UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE CIÊNCIAS
DEPARTAMENTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

RAINARA MAIA CARVALHO

CORRELATE & LEAD: PROCESS AND CATALOG OF NON-FUNCTIONAL REQUIREMENTS CORRELATIONS IN UBICOMP AND IOT SYSTEMS

FORTALEZA
2019

RAINARA MAIA CARVALHO

CORRELATE & LEAD: PROCESS AND CATALOG OF NON-FUNCTIONAL REQUIREMENTS CORRELATIONS IN UBICOMP AND IOT SYSTEMS

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de doutor em Ciência da Computação. Área de Concentração: Ciência da Computação

Orientadora: Prof. Dra. Rossana Maria de Castro Andrade
Coorientadora: Prof. Dra. Káthia Marçal de Oliveira

FORTALEZA

2019

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

C327c Carvalho, Rainara Maia.
Correlate & lead: process and catalog of non-functional requirements correlations in ubicomp and iot systems / Rainara Maia Carvalho. – 2019.
230 f. : il. color.

Tese (doutorado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Ciência da Computação , Fortaleza, 2019.
Orientação: Profa. Dra. Rossana Maria de Castro Andrade.
Coorientação: Profa. Dra. Káthia Marçal de Oliveira.

1. Ubiquitous Computing. 2. Internet of Things. 3. Human-Computer Interaction. 4. Non-Functional Requirements. I. Título.

CDD 005

RAINARA MAIA CARVALHO

CORRELATE & LEAD: PROCESS AND CATALOG OF NON-FUNCTIONAL REQUIREMENTS CORRELATIONS IN UBICOMP AND IOT SYSTEMS

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de doutor em Ciência da Computação. Área de Concentração: Ciência da Computação

Aprovada em: 04 de Setembro de 2019

BANCA EXAMINADORA

Prof. Dra. Rossana Maria de Castro Andrade (Orientadora)
Universidade Federal do Ceará (UFC)

Prof. Dra. Káthia Marçal de Oliveira (Coorientadora)
Université Polytechnique Hauts-de-France (UPHF)

Prof. Dr. Windson Viana de Carvalho
Universidade Federal do Ceará (UFC)

Prof. Dra. Andréia Libório Sampaio
Universidade Federal do Ceará (UFC)

Profa. Dra. Claudia Cappelli
Universidade Federal do Rio de Janeiro (UFRJ)

Prof. Dr. Luiz Cysneiros
York University (YU)

This thesis is dedicated to my beloved grandmother, Maria Venilce Maia, my model of both kindness and strength.

ACKNOWLEDGEMENTS

The journey of a Ph.D. student requires not only technical skills but also emotional ones such as perseverance, resilience, and most important, the willingness to never give up. So, it is essential to have good people supporting you to meet all these qualities, and, luckily, I have many of them. In this long journey, I could not acknowledge other people first, if not my advisors Profa. Rossana Andrade and Profa. Káthia Oliveira.

First and foremost, I would like to express my gratitude to Profa. Rossana, with whom I have worked over the last nine years. Her experience, talent, hard work, and strong confidence has inspired me since I was an undergraduate student. Her tireless pursuit of making her students the best ones gave me a priceless example for my own career. She believed me since day one and gave me unique and wonderful opportunities during not only my Ph.D. but my entire academic life. Thank you, Rossana, for suggesting the topic of my thesis, and for your valuable reviews and guidance.

My sincere gratitude goes to Profa. Káthia who always challenged me to strive towards the best work I could do. I thank her for all those hours she has dedicated to review my work and to discuss with me. Always with joy and enthusiasm. Prof. Káthia has a unique talent to make every student feel capable and motivated. Thank you, Káthia, for your guidance throughout all these years, your critical reviews and all the great experiences.

I also would like to express my genuine gratitude to my committee, who accepted to review this thesis, dedicating their precious time. Prof. Luiz Cysneiros, Profa. Claudia Cappelli, Profa. Andreia Libório and Prof. Windson Viana. Special thanks to Prof. Luiz Cysneiros, who inspired me immensely with his work regarding NFRs catalogs, and Profa. Andreia, who has been part of my committee since my master's and was always supportive and caring.

A huge thanks to my GREat family. My fellow colleagues Adyson, Amanda Sousa, Amanda Pires, Armando, Atslands, Belmondo, Breno, Bruno, Candré, Carla, Cayk, Christiano, Danilo, Deborah, Evilasio, Gabriel, Ítalo, Ismayle, Jefferson Barbosa, Josy, Juliana, Lana, Lincoln, Mariana, Paulo Artur, Pedro, Rayner, Rodrigo, Rute, Tales, Thalisson, Ticianne, Valéria, Valdenir. Their continuous encouragement was essential to get through every little challenge I faced. Special thanks to Ítalo for his precious help with the experiment design. Paulo Artur for literally everything, from serious discussions to coach sessions. Juliana for supporting me in the hardest courses I had to take in the first years. Rodrigo for all the valuable technical and non-technical advices. Josy, a true friend I have found in this last and most crucial year of my

Ph.D. Well, the truth is that I went to the GREat lab with the purpose to learn and to get a degree, but I ended up creating a second family, which I will always look up.

In this family, I also include all GREat staff, especially Darilú, Janaína and Chris. They are always working quietly behind the scenes so that everything works perfectly well. Also, I would like to thank MDCC staff, especially Orley and Marquinhos, dear people who worked at MDCC during the first years of my Ph.D.

Many thanks to my colleagues at Quixadá Campus, Davi Romero, Andreia Liborio, Wagner, Marcio Maia, Paulo Armando, Ingrid Monteiro, Regis Pires, Bruno Gois, João Marcelo and Aragão. Special thanks to Prof. Davi Romero who first encouraged me to pursue an academic career when I was his student; and Prof. Aragão for his availability and priceless help with the statistical analysis.

Also, I would like to express my gratitude to my students from Quixadá Campus. Erika, Yuri, Maryzangela, Gabriel, Lucas, Deborah, Millena and André. They were very important to me in this last year, helping me whenever I needed their support.

My gratitude to the people I met in Valenciennes. First, I thank Prof. Christopher Kolski, who allowed me to join LAMIH lab, and who accepted to work with me, sharing his enormous knowledge about user interaction. I also thank Taísa Guidini, who assisted me with many issues during my stay there. Among them, she introduced me to Jocelene and Sandro, a sweet couple I also owe my gratitude.

Thank you to my family. First, my dear parents, Roberto and Vanessa, thank you for always trusting on my decisions. Dear father, I'll forever carry with me your lessons about how important is to study and work hard. Dear mother, thank you for always let me know that whenever I go and do, I can always come back home. My brother Matheus, who gave me the most precious gift in my life, my niece, Vitória. My dear husband Jefferson de Carvalho, in whom I comforted myself during the bad times and celebrated the good ones. Thank you for always being by my side, whatever decision I would make, and for trying to make "other Ph.D. problems" invisible for me.

I also owe my gratitude to my parents-in-law, Jacqueline and Jales, for their availability to help me whenever I need. My dear uncles and aunties, who gave me support throughout my life to complete my education. Special thanks to my uncle Joãozito and my aunties Mariá, Paulinha, Ana, Galgani, Mônica and Goret, for having received me in their homes and hearts when I most needed. My dear cousins, Misa, Virna, Mariana, Ramon, and Diego, for making me laugh in many difficult situations. Special thanks to Misa, my most beloved cousin and sister by heart. My best friends from high school times, Isabella and Vanelly. Thank you, girls, for always being there for me, for the endless talks about life and, most important, for your honest and precious encouragement in this endeavor.

I want to thank every person who participated in this work someway and somehow — being experts, developers, users, reviewers, or researchers. The essence of this thesis is collecting and organizing information. Then, I did this work with the help of many people. People who shared their knowledge, the most precious resource someone can have. Nobody can steal it from you; it only can be spread — my eternal gratitude to them.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. Therefore, my gratitude to CAPES for supporting me with a scholarship during the first two years of my Ph.D, which was essential to start this journey.

“The most beautiful thing we can experience is the mysterious. It is the source of all true art and science.” (Albert Einstein)

ABSTRACT

Ubiquitous Computing (UbiComp) and Internet of Things (IoT) have environments full of smart and interconnected things, which can be accessed and controlled by several systems running on different devices. These systems bring a new set of Non-Functional Requirements (NFRs), especially those that are quality characteristics related to Human-Computer Interaction (HCI), such as Context-Awareness, Mobility and Invisibility. Such NFRs may interact with traditional ones (*e.g.*, Usability, Security), revealing positive correlations, when one NFR helps another, and negative correlations, when a procedure favors an NFR but creates difficulty for another one. As software engineers gain knowledge about these correlations, they can avoid conflicting NFRs and select strategies to better satisfy different NFRs. A common solution in the literature that can help software engineers in this scenario is to use correlation catalogs, which is a body of knowledge about NFRs produced from previous experiences. The literature has several catalogs that generally focus on requirements, strategies and correlations that are generic to any system, but it lacks catalogs with the previously mentioned NFRs for the domain of UbiComp and IoT systems. Moreover, the literature does not present a systematic and reusable process that organizes how to build these catalogs with well-defined inputs, outputs, and approaches. Therefore, the present work proposes first a process called CORRELATE to capture, analyze, and catalog the correlations between NFRs of UbiComp and IoT systems and then build a catalog named LEAD for the Invisibility characteristic, providing a proof of concept of the process. In the CORRELATE process, NFRs must first be specified in the Softgoal Interdependency Graph (SIG) notation; then, correlations can be identified and evaluated. For example, LEAD, the first catalog based on CORRELATE, contains 2 subcharacteristics of Invisibility, 12 sub subcharacteristics, 66 development strategies and 110 correlations with 9 NFRs. Also, a controlled experiment was conducted in this work to evaluate whether the proposed catalog improves software engineers' decisions regarding NFRs in the UbiComp and IoT systems. The results provide evidence that negative interactions between the considered NFRs are minimized and positive interactions are maximized, when LEAD is used.

Keywords: Ubiquitous Computing. Internet of Things. Human-Computer Interaction. Non-functional Requirement. Quality Characteristic. Correlation. Catalog.

RESUMO

A Computação Ubíqua (UbiComp) e a Internet das Coisas (IoT) possuem ambientes repletos de coisas inteligentes e interconectadas, que podem ser acessadas e controladas por vários sistemas em diferentes dispositivos. Esses sistemas trazem um novo conjunto de Requisitos Não-Funcionais (RNFs), principalmente aqueles que são características de qualidade relacionadas à Interação Humano-Computador (IHC), como Sensibilidade ao Contexto, Mobilidade e Invisibilidade. Tais RNFs podem interagir com RNFs tradicionais (*e.g.*, Usabilidade, Segurança), apresentando correlações positivas, quando um RNF ajuda outro, e negativas, quando um procedimento favorece um RNF, mas cria dificuldade para o outro. À medida que os engenheiros de software obtêm conhecimento sobre essas correlações, eles podem evitar RNFs conflitantes e selecionar estratégias que melhor satisfaçam diferentes RNFs. Uma solução comum na literatura para ajudar engenheiros de software neste cenário é o uso de catálogos de correlações, que é um corpo de conhecimento sobre RNFs gerado a partir de experiências anteriores. A literatura tem vários catálogos que geralmente se concentram em requisitos, estratégias e correlações que são genéricos para qualquer sistema, mas não apresenta catálogos com os RNFs mencionados anteriormente. Além disso, não foi encontrado um processo sistemático e reutilizável que organize como construir catálogos com entradas, saídas e abordagens bem definidas. Portanto, o presente trabalho propõe um processo chamado CORRELATE para capturar, analisar e catalogar as correlações entre os RNFs de sistemas UbiComp e IoT e, em seguida, como uma prova de conceito desse processo, um catálogo, chamado LEAD, para a característica Invisibilidade é construído. No processo CORRELATE, os RNFs precisam primeiro ser especificados na notação Softgoal Interdependency Graph (SIG); depois, as correlações devem ser identificadas e avaliadas. Por exemplo, LEAD, o primeiro catálogo baseado no CORRELATE, contém 2 subcaracterísticas de Invisibilidade, 12 sub subcaracterísticas, 66 estratégias de desenvolvimento e 110 correlações com 9 RNFs. Além disso, um experimento controlado foi realizado para avaliar se o catálogo proposto melhora as decisões dos engenheiros de software em relação aos RNFs dos sistemas UbiComp e IoT. Os resultados fornecem evidências de que, quando o LEAD é usado, as interações negativas entre os RNFs considerados são minimizadas e as positivas maximizadas.

Palavras-chave: Computação Ubíqua. Internet das Coisas. Interação Humano-Computador. Requisito Não-Funcional. Características de Qualidade. Correlações. Catálogo.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

AMICCaS	Attention, Mobility, Invisibility, Context-Awareness, Calmness and Synchronicity
ARRANGE	Approach based on the Grounded Theory Method to Refine a Quality Characteristic
CA	Content Analysis
CE-RQ	Controlled Experiment Research Question
EC	Exclusion Criteria
GREat	Group of Computer Network, Software Engineer and Systems
GT	Grounded Theory
HCI	Human-Computer Interaction
IC	Inclusion Criteria
IoT	Internet of Things
ISO	International Standardization Organization
ITU	International Telecommunication Union
LEAD	CataLog of Invisibility SubcharactEristics, StrAtegies anD Correlations
MIT	Massachusetts Institute of Technology
NFC	Near Field Communication
NFRs	Non-Functional Requirements
PoC	Proof of Concept
RFID	Radio Frequency Identification
SIG	Softgoal Interdependency Graph
SM	Systematic Mapping
SM-RQ	Systematic Mapping Research Question
TRACE	Approach to define CorrelaTions to Quality ChaRActeristics by using the Interview and the ContEnt Analysis Methods
UbiComp	Ubiquitous Computing
WER	Workshop on Requirements Engineering
WSN	Wireless Sensor Network

CONTENTS

1 INTRODUCTION

This thesis work deals with the investigation of correlations between quality characteristics for Ubiquitous Computing and Internet of Things systems. A process to organize the step-by-step and supporting instruments and approaches to establish correlations are the solutions presented in this thesis. Additionally, a catalog for Invisibility, a specific quality characteristic of this kind of systems, is proposed.

This Chapter introduces the main areas related to this thesis work in Section 1.1, followed by the research problem in Section 1.2. Then, Section 1.3 introduces the research hypothesis and the goal of this thesis. Next, Section 1.4 presents the research questions that guide this entire work and Section 1.5 presents the research methodology that conducts this thesis. work. Finally, Section 1.6 presents a road map of the thesis and a summary of this chapter.

1.1 Contextualization

Ubiquitous Computing (UbiComp) was coined by Mark Weiser in 1991 and became a well-established area with a significant research community (WHITMORE *et al.*, 2015). The first papers about UbiComp paradigm suggested a future where computing technologies were no longer present only in desktop computers. Instead, computing would move into the user's everyday environment. Therefore, the primary motivation behind UbiComp is to support users in their daily activities with minimal user distraction (CARVALHO *et al.*, 2017). On that occasion, several applications emerged to help people. Nowadays, it is common to see tour guides assisting users to locate themselves in big malls or museums and to learn about what is around them. Moreover, software assistants are available to support drivers by proposing routes or suggesting where to park. Smartphones also have applications designed to mute the device in situations, such as while attending classes, meetings or in a movie theater.

This pervasive integration with everyday objects and the evolution in technology have given rise to what is known today as Internet of Things (IoT). IoT is a paradigm which defines a collection of smart things from our daily lives connected to the Internet to provide much more relevant services for users (MASHAL *et al.*, 2015). For example, air-conditioners can be controlled remotely or can act alone according to the environmental context or doors can be automatically unlocked to authorized users (HO *et al.*, 2016).

This work considers that IoT extends the ideas and goals of UbiComp and then creates an environment full of smart objects which are interconnected and aware of the surrounding events (BODEI *et al.*, 2012). These objects can be accessed and controlled by several applications running on different devices. These applications bring a new set of Non-Functional Requirements (NFRs), primarily related to Human-Computer Interaction (HCI) (CARVALHO *et al.*, 2018).

According to (WIEGERS; BEATTY, 2013), one kind of NFR is the quality characteristic, which represents the expectations beyond the system's functionalities, such as Usability, Security, Reliability, and Performance. Some quality characteristics can be particularly important for certain types of software, for example, Availability is essential for web systems as well as Reliability is for banking systems. These characteristics can also be important from a specific point of view. The International Standardization Organization (ISO/IEC 25010, 2011) defines characteristics related to static properties of software, characteristics related to dynamic properties and characteristics related to the outcome of interaction when a product is used in a particular context of use. In this way, it is natural to have quality characteristics particularly important for UbiComp and also specific from a certain point of view, HCI for example, which is the case of this work.

According to (SILVA *et al.*, 2016), when quality characteristics are not satisfied, the entire system can be disabled. Therefore, regardless of which focus has a set of quality characteristics, they are equally important as functional requirements. It is common to evaluate the system regarding these characteristics, using approaches such as Usability Testing (BEZERRA *et al.*, 2014) and Performance Evaluation (MAIA *et al.*, 2014). However, it is necessary and essential to consider quality characteristics since the beginning of system's development cycle because they can impact the choice of hardware and network resources, architecture design, and other issues (SILVA *et al.*, 2016). Once the system is developed, it is expensive or even impracticable to change it to consider quality characteristics (WIEGERS; BEATTY, 2013).

Moreover, dealing with quality characteristics during software development is challenging. One of the reasons is that there are many types of them, and each requires specialized knowledge on how to support it. There can be several alternative solutions to help a single quality characteristic in a specific application. This knowledge is not always effortlessly available for software engineers, specially the ones who has no experience in the domain of the system being developed (CHUNG *et al.*, 2000).

Another reason is that quality characteristics can interact with each other (WIEGERS; BEATTY, 2013), which means that achieving one characteristic can impact the achievement of another. This impact can be either positive or negative. When negative, this impact is called "conflict" or "negative correlation" by (CHUNG *et al.*, 2000). For example, a well-known conflict is between Security (authorization) and Usability (efficiency) (MAIRIZA *et al.*, 2013). Indeed, Usability characteristic is negatively impacted by several other traditional quality characteristics, such as Efficiency, Integrity, Performance, Portability, and Safety, being as such one of the most impacted NFR (WIEGERS; BEATTY, 2013). On the other hand, there are positive correlations, called "harmony". For example, Reusability helps to achieve Interoperability (WIEGERS; BEATTY, 2013).

Knowledge about how to satisfy quality characteristics regarding solutions and impacts, especially the negative ones, is useful for avoiding bad or wrong decisions towards the target characteristics (CHUNG *et al.*, 2000). Therefore, software engineers should search for information about solutions and correlations to help them to make decisions and to avoid making agreements that are impossible to achieve in the development.

A common solution in the literature for helping software engineers to fulfill quality characteristics is the use of NFRs catalogs (CHUNG *et al.*, 2000). A catalog is a body of knowledge that has been accumulated from previous experience. They are known for bringing to better requirements specifications since it allows the reuse of requirements (CYSNEIROS, 2007) (CARDOSO; GUIZZARDI, 2011) (GRAMATICA *et al.*, 2015). Additionally, the usage of catalogs prevents engineers from spending time searching at many diverse sources or relying on experts in the field to make decisions about how to achieve quality characteristics. Consequently, they can decrease the time for searching know-how and the cost for consulting experts (CHUNG *et al.*, 2000).

Although there are catalogs in a matrix format (MAIRIZA; ZOWGHI, 2011), the focus of this work is on catalogs represented by Softgoal Interdependency Graph (SIG). In this notation, quality characteristic is treated as softgoals[1], which is a goal with no clear-cut criteria of satisfaction. In this notation, softgoals should be refined into sub softgoals (*i.e.*, subcharacteristics), which should be refined into operationalizing softgoals (*i.e.*, implementation or design solutions for supporting softgoals), and, finally, these softgoals can present correlations regarding another softgoals. This work focuses on this kind of catalog due to its extensive use in literature for cataloging NFRs (LEITE; CAPPELLI, 2010) (MAIA *et al.*, 2009) (TORRES; MARTINS, 2014) (ZINOVATNA; CYSNEIROS, 2015) (PORTUGAL *et al.*, 2018). Moreover,

[1] The terms Non-Functional Requirements, Quality Characteristics and Softgoals are used interchangeably.

this kind of catalog allows to analyze in depth how different quality characteristics could be achieved with implementation strategies.

1.2 Motivation

The literature has several NFRs catalogs. For example, there is a catalog for Security, Performance and Accuracy in (CHUNG *et al.*, 2000), Usability in (CYSNEIROS *et al.*, 2005), Transparency in (LEITE; CAPPELLI, 2010), Privacy in (ZINOVATNA; CYSNEIROS, 2015), Fault Tolerance in (TORRES; MARTINS, 2014). These existing catalogs focus on requirements and strategies that are generic to any system.

However, as previously mentioned, software complexity has changed in the past years with the advent of applications for UbiComp and IoT environments (ANDRADE *et al.*, 2017). Naturally, quality characteristics particularly important for this kind of system have arisen, but specifically related to the quality of user interaction.

This topic was one of the issues investigated by the MAxIMUm project (A Measurement-based Approach for the Quality Evaluation of Human-Computer Interaction in Ubiquitous Systems)[2]. During this project, an initial set of quality characteristics was identified as essential for the quality of interaction in ubiquitous systems (CARVALHO, 2017). This set became a key point that has motivated this thesis work. To better understand how these characteristics were defined, a short chronological timeline is illustrated in Table 1.

The MAxIMUm project started in 2012 and aimed to define a measurement approach to evaluate ubiquitous systems. To achieve this goal, first, it was necessary to define a set of quality characteristics specific for this kind of system. Through a systematic mapping method, five characteristics were identified: Attention, Mobility, Invisibility, Context-Awareness, Calmness and Synchronicity (AMICCaS). Then, software measures were defined to evaluate these characteristics. Finally, during the thesis work, this set was updated to include one more characteristic because of IoT area: Synchronicity (ANDRADE *et al.*, 2017).

This work refers to this set as **AMICCaS** - **A**ttention, **M**obility, **I**nvisibility, **C**ontext-Awareness, **C**almness, and **S**ynchronicity. Despite their importance for a better interaction, these specific characteristics (*e.g.*, Context-Awareness, Invisibility, Mobility, Attention, Calmness and Synchronicity) can impact traditional ones such as Usability, Performance, Security, bringing

[2] Project supported by FUNCAP and CNRS under grant number INC-0064-00012.01.00/12 - http://great.ufc.br/maximum/

Table 1 – Timeline of the Definition of Specific HCI Quality Characteristics for UbiComp and IoT Systems

Year	Event
2012	The Maximum's project was accepted
2013	Conduction of a Systematic Mapping to define a set of quality characteristics for UbiComp
2014	Master defense and entrance of Rainara Maia Carvalho in the doctorate program
2016	Online journal publication about the quality characteristics for UbiComp (CARVALHO *et al.*, 2017)
2017	Update of the set of characteristics for IoT systems (ANDRADE *et al.*, 2017)
2018	Online journal publication about measures for quality characteristics in UbiComp (CARVALHO *et al.*, 2018)

Source: Author.

new negative or positive correlations.

Mobility, for example, refers to the continuous use of the systems as the user moves across several devices and through various networks (YU *et al.*, 2013). This characteristic can be implemented through handoff, which is a procedure that allows a device to connect to another network when the previous one is out of range or loses signal quality. However, according to (MAIA *et al.*, 2009), the new network may not have the appropriate level of security, which in turn can initiate a transmission of confidential data, impacting negatively on Security characteristic of the application. Therefore, while handoff strategies help Mobility, they may bring negative correlations to Security.

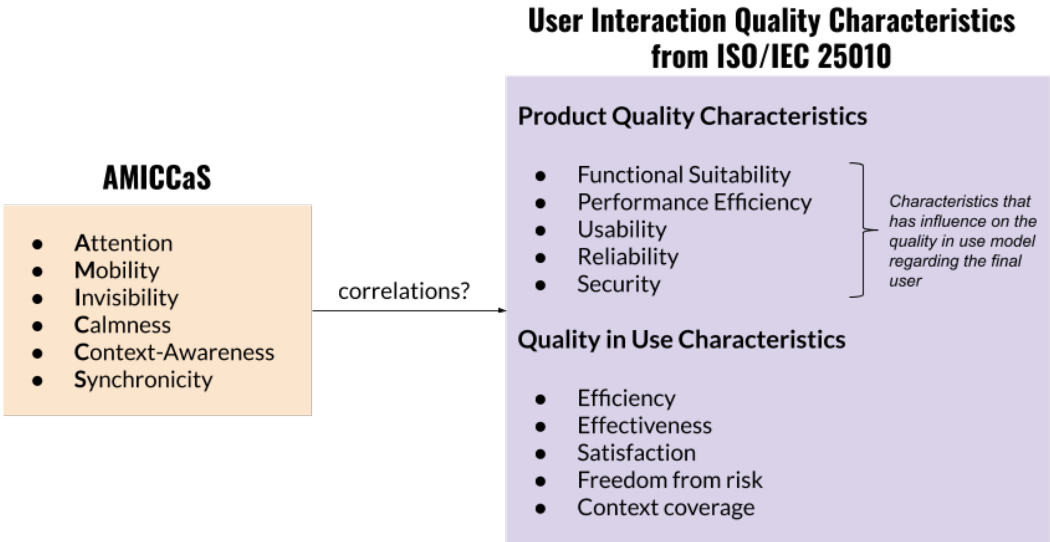
The same situation can happen with other characteristics, specially the ones related to user interaction quality. In this sense, the International Standardization Organization (ISO) defines two models of quality characteristics (ISO/IEC 25010, 2011). One of them is the "Quality in Use" Model, which defines five characteristics concerned with the impact that the product has on stakeholders and users: Effectiveness, Efficiency, Satisfaction, Freedom from Risk and Context Coverage. The other one is the "Product Quality" Model, which defines eight characteristics concerned with the software system in operation: Functional Suitability, Performance Efficiency, Compatibility, Usability, Reliability, Security, Maintainability and Portability. Although they are separated models, the standard states that there is an influence of the Product Quality Model into the Quality in Use Model, stating that the following five characteristics have an influence on the quality in use of the final user: Functional Suitability, Performance Efficiency, Usability, Reliability and Security. These characteristics, the five ones in the “Quality in Use” Model and the five ones that has an influence in the quality in use, are particularly important for UbiComp and IoT, because their systems are designed to be anywhere and to work anytime for users in their everyday lives. Then, these systems need to have the final user as their main concern. Therefore, this work defines these characteristics as a set of user interaction quality characteristics as shown in Figure 1.

In this scenario, it is possible that development strategies of specific HCI quality characteristics (such as AMICCaS) that have arisen from UbiComp and IoT impact the set of user interaction quality characteristics negatively. For that reason, this work addresses the following research problem, also illustrated in Figure 1:

Research Problem

Are there correlations between specific HCI quality characteristics (*e.g.*, AMICCaS) and user interaction quality characteristics (*e.g.*, Usability, Performance, Security) that can impact on the development of UbiComp and IoT systems?

Figure 1 – Overview of the Research Problem



Source: Author.

Knowing about these correlations helps avoiding commitments to conflicting NFRs (WIEGERS; BEATTY, 2013). Besides, knowledge about them helps dealing with trade-offs among NFRs, which means assisting software engineers to choose solutions with most benefit and least sacrifice for the chosen quality characteristics. Trade-offs analysis is even more critical when dealing with the quality in use of UbiComp and IoT applications since they are likely to make users feeling annoyed and overwhelmed due to its nature of being available every time and everywhere (CARVALHO *et al.*, 2017).

Additionally, during the literature search for the identification of the research problem, no generic approach that provides a suite of techniques, instruments and tools to help researchers to create their own NFR catalog was found out. This lack of such approach makes the definition of catalogs harder.

1.3 Research Hypothesis and Goal

Correlations between quality characteristics happen when a solution favors the first characteristic but create difficulties for the second (BERANDER *et al.*, 2005). For example, a module of the system may require Security mechanisms, which can increase the complexity of a module and, consequently, the difficulty of user interaction. Therefore, we can say that there is a conflict between Security and Usability requirements.

In case of UbiComp and IoT systems, it is likely that their specific quality characteristics (*e.g.*, context-awareness, mobility, invisibility) may bring new design solutions that are not used in traditional systems or that were not evaluated in UbiComp and IoT environments. Therefore, the following hypothesis is established and investigated in this thesis:

Research Hypothesis (RH)

Specific HCI quality characteristics such as AMICCaS impact on user interaction quality characteristics negative and positively, due to their design and implementation solutions.

Knowledge about NFRs (*i.e.*, subcharacteristics, solutions and correlations) should be cataloged to be reused for other software projects. SIG notation contains all elements to store information about NFRs. In this way, software engineers can consult these catalogs to help them to make decisions towards NFRs in a specific system.

However, during the problem investigation, this work found out that there is not a generic approach that provides a suite of techniques, instruments and tools to help a researcher to create his own NFR catalog. This lack of such approach makes the definition of catalogs hard. Therefore, the goal of this thesis work is twofold:

Research Goal (RG)

To define an approach to capture, analyze and catalog the impact of AMICCaS on user interaction quality of UbiComp and IoT systems and to use this approach to investigate the presence of negative and positive correlations.

1.4 Research Questions

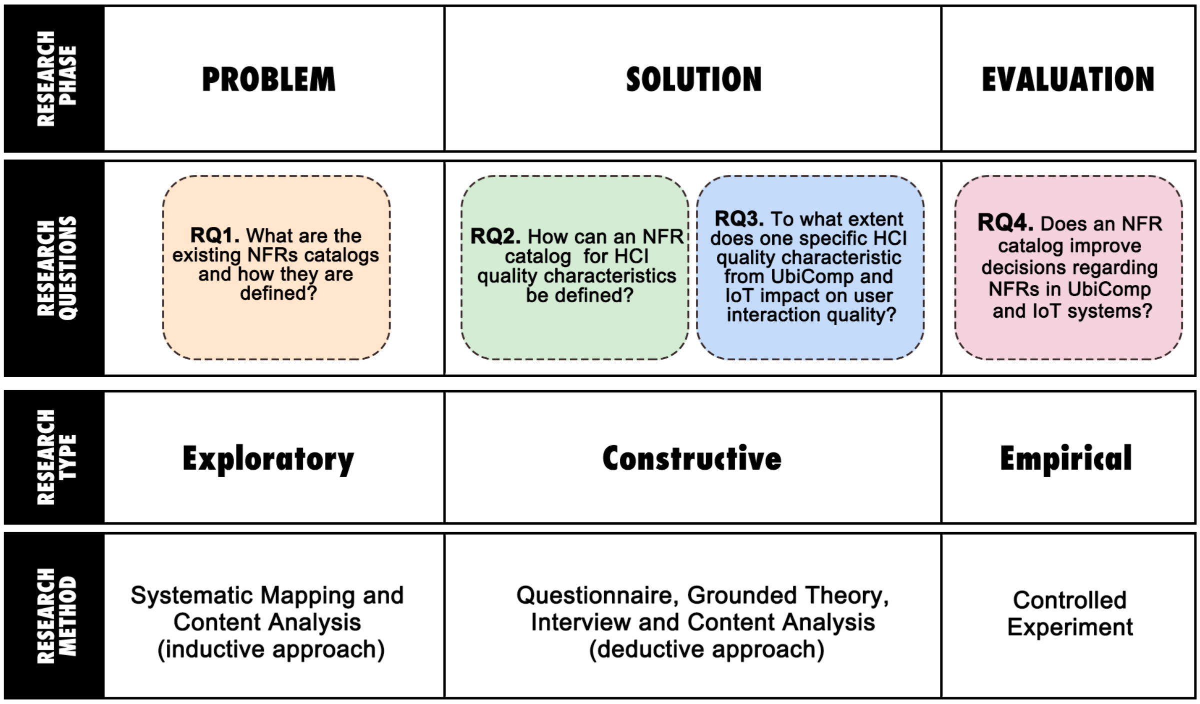
The following research questions guided this thesis work:

- ***RQ1 - Which are the existing NFRs catalogs and how they are defined?*** An in-depth investigation regarding existing NFRs catalog is necessary to understand the challenges and opportunities regarding them. It is important to understand how the existing catalogs are defined before starting the definition of the proposed catalog in this work.
- ***RQ2 - How can an NFR catalog for HCI quality characteristics in UbiComp and IoT systems be defined?*** This question is concerned about how to find out possible correlations and store them in a catalog.
- ***RQ3 - To what extent does a specific HCI quality characteristic from UbiComp and IoT impact on user interaction quality characteristics?*** This question aims to find out correlations of at least one specific HCI quality characteristic on user interaction quality characteristics.
- ***RQ4 - Does an NFR catalog improve decisions regarding NFRs in UbiComp and IoT systems?*** This question is concerned about the evaluation of a proposed catalog. It is expected that this proposed catalog help software engineers of UbiComp and IoT systems make decisions with most benefit and least sacrifice for chosen quality characteristics.

1.5 Research Methodology

The research methodology followed in this thesis work is inspired by the research steps in (GHAZI, 2018), which used the framework proposed by (WIERINGA *et al.*, 2006). In summary, this research is organized by three phases (See Figure 2): *Problem*, *Solution* and *Evaluation*. Problem phase is where the problem, hypothesis and goal of the thesis should be defined. Solution phase represents the development of the solutions proposed in this thesis that take into account the problem and issues investigated in Problem Phase. Finally, Evaluation phase is where the proposed solution should be evaluated.

Figure 2 – Research Methodology



Source: Author. Based on (GHAZI, 2018).

The Problem phase and its respective question (RQ1) are classified as an Exploratory research. This type of research is related to when the investigation has the purpose of providing more information about the topic that will be studied, allowing the problem refinement and goals definition (PRODANOV; FREITAS, 2013). In the case of this thesis work, the purpose in this phase is to investigate the existing catalogs and how they were defined before starting the construction of a catalog for UbiComp and IoT systems.

A Systematic Mapping (SM) study was used to perform this phase and reach the answers for the first research question. SM is a method to build a classification scheme and to structure a field of interest (PETERSEN *et al.*, 2008). It is defined as a rigorous, unbiased and auditable procedure for searching research literature. Systematic mapping studies use the same basic methodology as systematic review (KITCHENHAM; CHARTERS, 2007) guided by research questions. Nevertheless, the research questions for a mapping study are more general, related to research trend, and quite high level, including issues such as: which sub-topics have been addressed, what empirical methods have been used, and what sub-topics have sufficient studies for a more detailed system review.

The second phase (Solution) and its respective research questions (RQ2 and RQ3) are classified as Constructive Research. This type of research is related to develop solutions that will deal the research problem identified in the first phase. In the study performed to answer RQ1, several techniques that could potentially help identify correlations in UbiComp and IoT systems were found. However, it was possible to see that there is not a systematic and generic approach that provides a suite of techniques, instruments and guidelines to assist researchers to create their own NFR catalog, especially for quality characteristics that are new and no taxonomy is available, which is the case of AMICCaS. This lack of such approach makes the definition of catalogs harder. Thus, part of this thesis work is first dedicated to define an approach capable to define NFRs catalog (answering RQ2). Then, following this proposed approach, a catalog for specific quality characteristics in UbiComp and IoT could be constructed (answering RQ3).

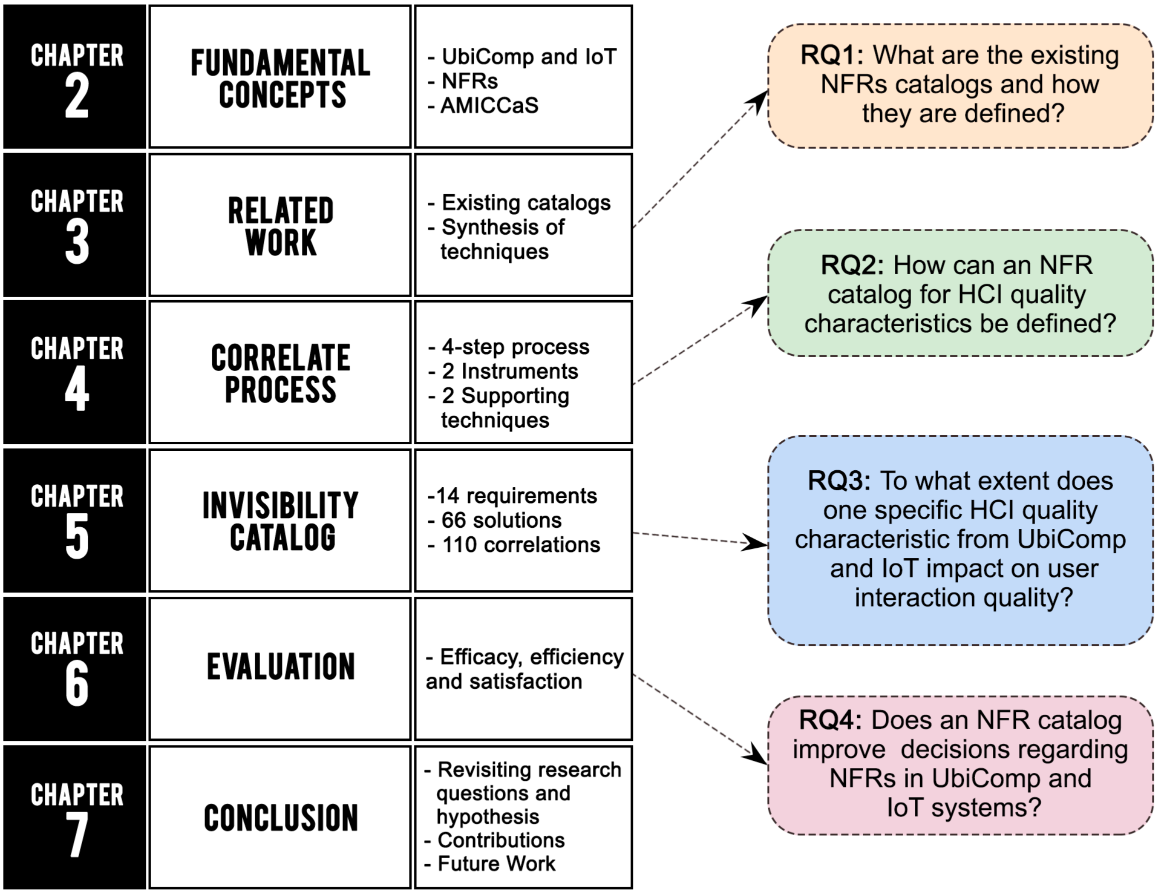
The research methods used in this phase were Questionnaire, Grounded Theory, Interview and, again, Content Analysis. A questionnaire is a pre-defined set of questions, organized in a pre-determined order. Respondents are asked to answer the questions, thus providing the researcher with data (OATES, 2005). Grounded Theory (GT) is a method developed for the purpose of building theory from data (GORBIN; STRAUSS, 2008). This method is based on grouping concepts derived from data until reaching the core category. According to (GORBIN; STRAUSS, 2008), it is like putting together a series of interlinking blocks to build a pyramid of knowledge. Interview is a particular kind of conversation between people. Usually, one person has a purpose for under taking the interview: they want to gain information from the other(s) (OATES, 2005). Content Analysis (CA) is a research method used to make replicable and valid inferences by interpreting and coding textual material. It is seen as a method to classify written or oral materials into identified categories of similar meanings (BARDLN, 1977). Each one of these methods will be better explained in the chapters they are used.

1.6 Road Map and Chapter Summary

This thesis is organized into six more chapters. Figure 3 shows an overview of the chapters, a brief summary and the research question each chapter is related to.

Chapter 2 provides the theoretical basis on the areas that compose this work. First, the concepts about UbiComp and IoT is presented. A brief overview about UbiComp is discussed and then the evolution towards IoT is presented. This Chapter also presents what this work considers as UbiComp and IoT systems. After that, concepts about NFRs are presented, specially regarding Quality Characteristics, which is one kind of NFRs. Furthermore, the sets of NFRs that are used in this thesis are introduced, both the set called AMICCaS and the set of user interaction

Figure 3 – Road Map of the Thesis



Source: Author.

quality characteristics.

Chapter 3 presents the related work to this thesis, which provides a basis for understanding the research gaps pointed out in this work. The collection and analysis of the related studies were performed through a SM study. Then, all the SM process is explained as well as its results.

Chapter 4 presents the proposed Process to Capture, Analysis and Catalog CoRRElations (CORRELATE). This process comprises 4 steps and each step includes a supporting instrument or approach to help a researcher or practitioner use the process and then build an NFR catalog.

Chapter 5 presents the CataLog of Invisibility SubcharactEristics, StrAtegies anD Correlations (LEAD), developed through the first execution of the CORRELATE process. First, the general results from each step of the process is presented, describing how each information in the catalog was achieved. Then, the information presented in the catalog is showed and explained.

Chapter 6 presents the evaluation executed to analyse the correlation catalog developed through the CORRELATE process. This evaluation was performed to evaluate the efficacy, efficiency and satisfaction of the correlation catalog. The results has showed that the catalog can substantially improve decisions regarding NFRs.

Chapter 7 presents the conclusions of this thesis by revisiting the research questions and main hypothesis, presenting the contributions and future work.

2 FUNDAMENTAL CONCEPTS

This Chapter presents fundamental concepts about the areas related to this research, which is useful for understanding the challenges and solutions pointed out in this work. Section 2.1 presents concepts related to UbiComp and IoT. Then, Section 2.2 gives an explanation about Quality Characteristics. Section 2.3 presents the set of specific HCI Quality Characteristics for UbiComp and IoT systems. Finally, Section 2.4 summarizes this chapter.

2.1 Ubiquitous Computing and Internet of Things

"The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it". With this statement made in the year 1991, Weiser idealized ubiquitous computing (UbiComp) as the main technology of the 21st-century (WEISER, 1991). Over the last two decades, UbiComp applications have increasingly been used to support hundreds of everyday activities. This idea has given rise to what is today called the Internet of Things (IoT), which is a collection of smart objects from our daily lives connected to the Internet to provide more relevant applications. The next subsections introduce an overview of UbiComp definitions and characteristics, the evolution from UbiComp towards IoT and main concepts of UbiComp and IoT systems.

2.1.1 Brief Overview about Ubiquitous Computing

Mark Weiser, the founder of ubiquitous computing paradigm, criticized the idea of making computers exciting objects per si. He argued that instead of making them the central object, computers should merge into people's daily environment (VASSEUR; DUNKELS, 2010). In this computing paradigm, the user can interact with computers that are presented in several shapes, such as tablets, TVs and any everyday objects like washing machines, fridges, coffee makers and lamps. Besides, the technology would help people achieve not only work needs but also daily tasks. In a nutshell, there are four differences between traditional systems and ubiquitous systems (POPPE *et al.*, 2007):

- *New possibilities of sensing*. In traditional systems the inputs of the users are provided often by hardware devices, such as keyboards. In ubiquitous systems, inputs can be captured by sensors (*e.g.*, GPS), or captured by the voice, gesture and touch.
- *Shift in initiative*. In traditional systems, the interaction corresponds to an explicit dialogue

between the user and the computer and it is the user who begins the interaction. In ubiquitous systems, dialogues can be initiated by the system itself, given its ability to sense the user, his/her environment and his/her needs.

- *Heterogeneity of physical interfaces*. Ubiquitous systems can be present in several everyday objects. Thus, there is a movement to make ubiquitous systems for both large interfaces, like interactive displays, and small ones, like smartphones and wearable devices.
- *Shift in application purpose*. Ubiquitous systems focus on the user and on everyday life, whereas traditional systems are, in general, task-based, helping users in their work.

Several applications emerged to embrace the ideals of UbiComp. For example, an application called GREatMute, developed by GREat Lab, puts a phone in silent mode when it detects the presence of the user in an event like a meeting or in cinema. This system does not use traditional input devices such as a keyboard or a mouse, but inputs (*e.g.*, activity and location) captured by physical and logical sensors (*e.g.*, GPS for location and Calendar for activity).

This idea of connectivity, pervasiveness and availability brought by UbiComp paradigm has given an ascension to what is called today the Internet of Things (IoT) (BODEI *et al.*, 2012). The next subsection takes a closer look at this evolution.

2.1.2 From Ubiquitous Computing to the Internet of Things

Looking to the brief timeline in Table 2, after UbiComp paradigm was coined in 1991, the term *context-aware* was created in 1994 by (SCHILIT; THEIMER, 1994), that would later become an essential area for the achievement of UbiComp goals.

Although coined by Schilit and Theimer (1994), the terms *context-aware* and *context* were improved a few years later by (DEY; ABOWD, 1999). According to them, *“a system is context-aware if it uses context to provide relevant information and services to the user, where relevancy depends on the user’s task”*. They also redefined context as *“any information that can be used to characterize the situation of an entity”*. Therefore, the UbiComp and Context-Aware communities proposed frameworks that should support the acquisition, representation and reaction, such as the Context ToolKit proposed by (DEY *et al.*, 2001).

Around the year of 1996, the ubiquitous community started to work with ambient environment. For example, the ambientROOM project at Massachusetts Institute of Technology (MIT) was developed as an example of enriching an entire environment with ubiquitous technologies, such as sensors. This scenario is interesting because it showed that technology started

Table 2 – Brief timeline for UbiComp to IoT movement

Year	Event
1991	The term *Ubiquitous Computing* was coined by Mark Weiser in his paper called *The Computer for the 21st Century* (WEISER, 1991)
1994	The term *context-aware* is first used by Schilit and Theimer in "Disseminating active map information to mobile hosts" (SCHILIT; THEIMER, 1994), but it was redefined years later by (DEY; ABOWD, 1999)
1996	The ubiquitous community started to work with ambient environment. The ambientROOM project is an example of that (ISHII *et al.*, 1998)
1999	The *Internet of Things* term is coined by Kevin Ashton, executive director of the Auto-ID Center (ASHTON, 2009), while they were working on turning RFID into a networking technology by linking objects to the Internet through the RFID tag.
2001	The MIT Auto-ID center presented their vision about Internet of Things (BROCK, 2001)
2005	IoT was introduced by the International Telecommunication Union (ITU) (Telecommunication Union, 2005)

Source: Author.

its presence not only in individual devices but also inside an environment.

A few years later, in 1999, Kevin Ashton coined the term *Internet of Things* (ASHTON, 2009), while proposing supply chain management improvements by using Radio Frequency Identification (RFID) to identify and track products automatically. After that, the term has gained considerable attention in academia and industry. The MIT Auto-ID center presented their IoT perception in 2001 (BROCK, 2001). Later, IoT was introduced by the International Telecommunication Union (ITU) in 2005 (Telecommunication Union, 2005). Then, many proposals of definitions and views started to arise, such as:

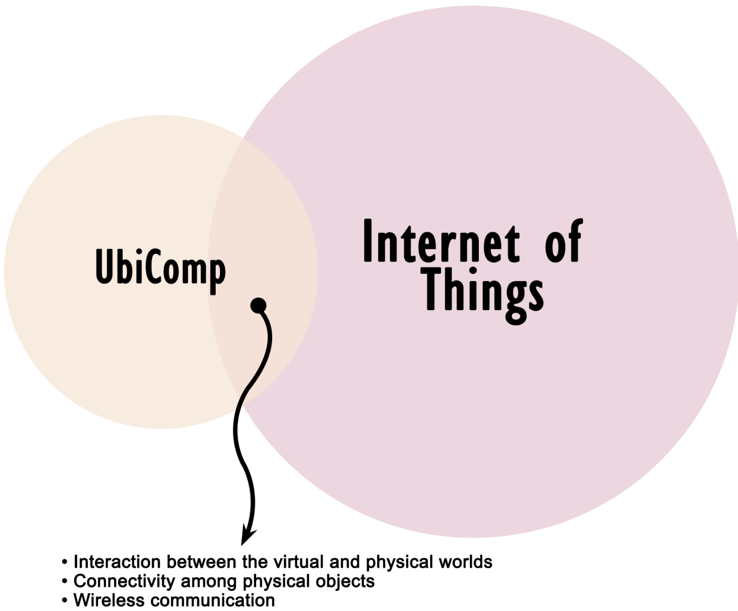
- *"The Internet of Things is the general idea of things, especially* ***everyday objects****, that are readable, recognizable, locatable, addressable, and controllable via the Internet - whether via RFID, wireless LAN, wide-area network, or other means"* (SWAN, 2012).
- *"An IoT is a network that connects uniquely identifiable "things" to the Internet. The "things" have* ***sensing/actuation*** *and potential programmability capabilities. Through the exploitation of unique identification and sensing, information about the "thing" can be collected and the state of the 'thing' can be changed from anywhere, anytime, by anything* (MINERVA *et al.*, 2015).
- *"The Internet of Things is nothing more than an extension of the current Internet, which allows* ***everyday things*** *connecting to the Internet. The connection to the global network*

will enable systems to control objects remotely. Also, this connection will provide things having communication skills with each other" (SANTOS *et al.*, 2016).

Considering the brief timeline, which shows the short interval between events and the definitions of IoT, it is possible to see that the vision of IoT inherits many goals of Ubiquitous Computing (*e.g.*, presence in everyday objects and environments and sensing/acting in the real world). Thus, this work considers that IoT is an extension of UbiComp, agreeing with (VASSEUR; DUNKELS, 2010), who state that many of the technologies and visions from UbiComp directly apply to IoT.

However, it is also true that IoT encompasses a much broader vision than UbiComp. The concepts of "connection" and "Internet" appear in almost all definitions, suggesting a system composed of many objects. For this reason, IoT does not just mean computing present in everyday objects that are capable of sensing and actuating, but a growing range of objects working together through network connections. Figure 4 illustrates this idea.

Figure 4 – Intersection between UbiComp and IoT



Source: Author.

This thesis work focuses on systems that are in the intersection of UbiComp and IoT. This kind of system has the following features in common: *(i)* interaction between the virtual and the physical world, which means there are sensing and actuating capabilities; *(ii)* connectivity among objects, which means systems are composed of more than one physical objects; and *(iii)* connection to the Internet, which means the system communicates through the Internet someway somehow. In this work, the term "UbiComp and IoT systems" is used to describe these kinds of systems.

2.1.3 UbiComp and IoT Systems

UbiComp and IoT systems are systems composed of objects connected with each other and with the Internet and have sensing and actuating capabilities. This section explains different kinds of objects and their connections to a better understanding of UbiComp and IoT systems. Also, examples of architectures are provided.

2.1.3.1 Types of Objects

In general, there are five types of objects (ROWLAND *et al.*, 2015): *(i)* Multipurpose computers; *(ii)* Specialized embedded devices; *(iii)* Connected Sensors; *(iv)* Passively Trackable Objects; and *(v)* Gateways. They are better explained as follows.

Multipurpose computers are devices created for performing a variety of computing tasks. PCs, smartphones, tablets, and TV are a few examples of these kinds of devices. They have rich interaction capabilities and allows connection to the Internet. Therefore, they are usually used to handle user interactions with Ubiquitous and IoT systems.

Specialized embedded devices are devices specialized in performing specific tasks. They can have several shapes to suit a task and may have other mechanical parts (*e.g.*, washing machine, car). Nest's thermostat[1] is an example of a specialized embedded device. These objects can have user interaction capabilities onboard and also can use multipurpose computers to interact with users. Also, they may connect directly to the Internet or connect indirectly via a multipurpose computer, such as the wearables or via a gateway.

Connected Sensors are small embedded devices used to capture data from the physical world. They convert readings about the environment into digital information. Each sensor is a computational element with a power supply, a processor, a memory, a wireless communication interface and a sensing unit (GARCIA, 2014). They are usually deployed as a Wireless Sensor Network (WSN) and also are increasingly found in mobile devices.

Passively Trackable objects are objects that can have a simple presence on the Internet without having an Internet connection, such as credit cards, bands. Passively trackable objects have a unique identity that is associated with online information about them, but at the

[1] https://nest.com/thermostats/

same time are not connected to the Internet. RFID, Near Field Communication (NFC), QR Code and Beacons are examples of these technologies.

Gateway is a combination of hardware and software whose goal is to deal with the heterogeneity between several sensors and mobile communication networks on the Internet (ZHU *et al.*, 2010). It is common to see gateways being used in home systems. Its role is interconnecting multiple things together to form an in-home network and share resources and information among various home appliances. On the other side, the gateway also plays another role to connect the external networks to the in-home network and provide access interface to the external networks (ZHU *et al.*, 2010).

2.1.3.2 Types of Connections

The objects mentioned in the last subsection can use different network technologies to connect with each other and with the Internet (SANTOS *et al.*, 2016) (ROWLAND *et al.*, 2015). In general, some objects can be directly connected to the Internet using, for example, Ethernet, Wi-Fi or Cellular Data. Other objects use local networks that do not support direct connections to the Internet - Bluetooth, Radio, ZigBee, RFID, NFC. However, objects with this kind of network can use gateways to connect to the Internet. The definitions of these two types of connections are explained as follows.

Internet Network Connections support Internet connections, which means that an object can directly connect to the Internet. The following options are some examples:

- Ethernet allows a wired connection between a device and the Internet. This type of connection is fast and reliable; however, mobility is compromised.
- Wi-Fi allows a direct connection between two devices and Internet connection via a router. This type of connection allows mobility; however, it works only in the range of the router.
- Cellular Data uses the same data networks provided by the mobile phone: GPRS or 3G/4G, allowing mobility in a major range.

Local Network Connections does not support Internet connections, which means the object should connect to a gateway through some local network, such as:

- Bluetooth is based on a wireless system designed for short-range and cheap devices to replace cables for computer peripherals, such as mice, keyboards and printers (LEE *et al.*, 2007).
- ZigBee is a low-powered radio network, suitable for battery-powered devices, found in

home automation, healthcare and remote control systems (CHALLOO *et al.*, 2012).

- RFID uses electromagnetic fields to send data from tags to a reader.
- NFC is a set of communications standards built on top of RFID technology to allow two-way communications. It is primarily designed for mobile devices.

2.1.3.3 Architectures

Figure 5 presents the architectures about how one object connects to another and to the Internet. In general, according to (ROWLAND *et al.*, 2015), there are five types of architectures: *(i)* dedicated gateway; *(ii)* smartphone as gateway; *(iii)* direct Internet connection; *(iv)* device-to-device connections; and *(v)* service-to-service connections. They are better explained as follows.

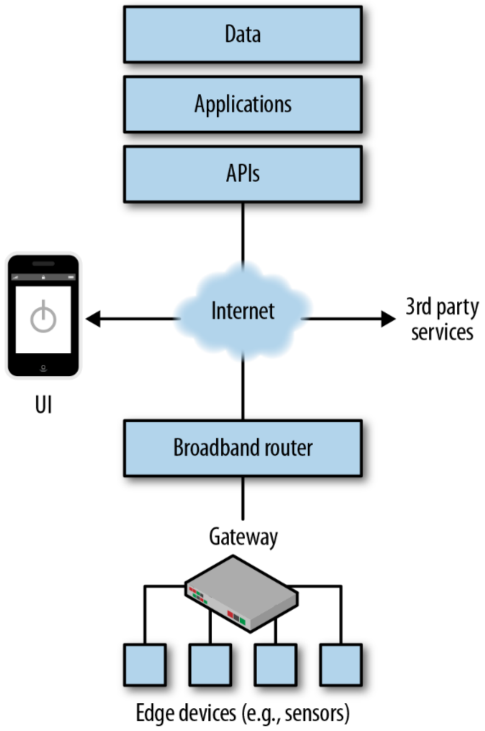
Dedicated Gateway is an architecture where objects such as sensors and embedded devices connect to a gateway object, which in turn connects to the Internet (See Figure 5 (a)). The endpoints of the system called edge devices (*e.g.*, light switches and sensors) usually use a local network, like ZigBee or Bluetooth, to connect to the gateway, while multipurpose computers, such as a mobile device, connect through an Internet connection. Home automation IoT systems are usually organized using this architecture.

Smartphone as a Gateway is an example of an architecture where the objects such as wearables connect to a multipurpose computer, usually a mobile device, as a gateway object, to connect to the Internet (See Figure 5 (b)). Typically, the connection between the wearable and the mobile device is established through Bluetooth. Besides acting as a gateway, the mobile device can have an application to interact with the user about the data collected by the wearable.

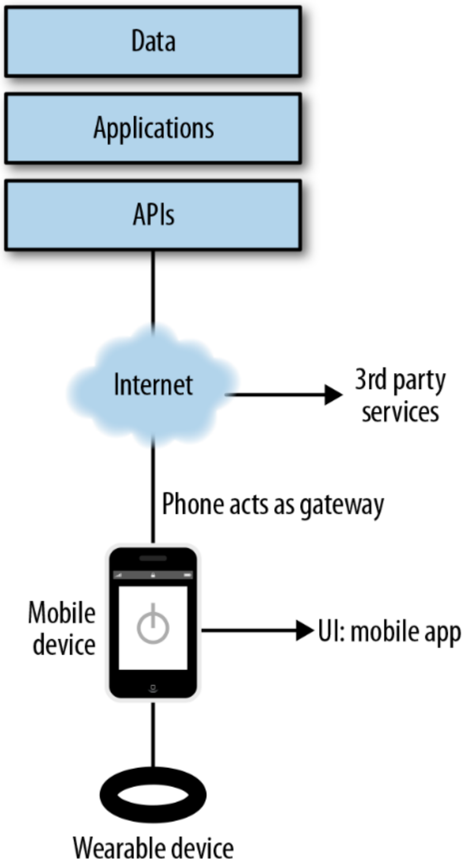
Direct Internet Connection is where the objects (*e.g.*, sensors, specialized embedded devices) connect directly to the Internet (See Figure 5 (c)). This connection is possible because they can have Internet interfaces and, therefore, do not need the gateway object. In **Device-to-device Connections** architecture, the objects can sometimes connect directly to each other without the usage of Internet or a gateway (See Figure 5 (d)).

Finally, the architecture **Service-to-Service Connections** is where the objects from different manufacturer services can be integrated without the need for a shared gateway (see Figure 6), which is possible due to application programming interfaces (APIs).

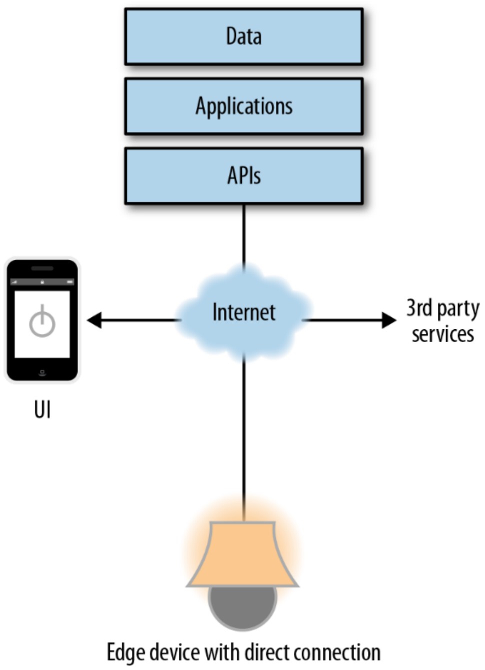
Figure 5 – Architectures



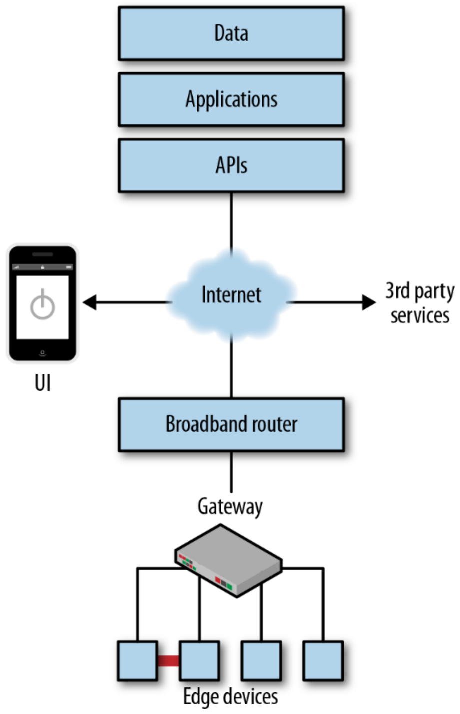
(a) Dedicated Gateway Architecture



(b) Smartphone as a Gateway Architecture



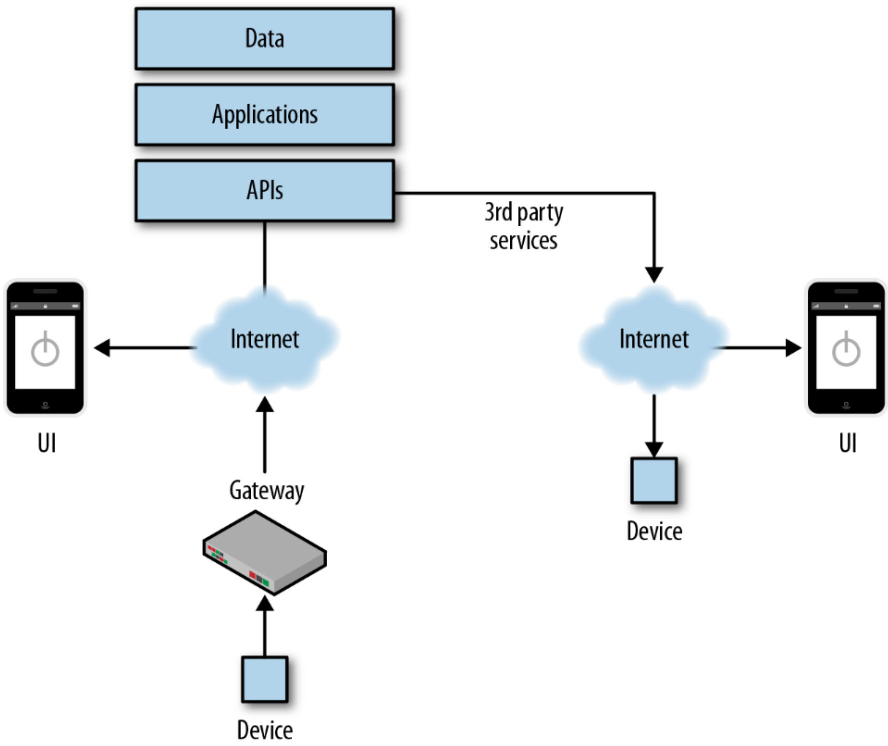
(c) Direct Internet Connection Architecture



(d) Device-to-device Connections Architecture

Source: (ROWLAND *et al.*, 2015)

Figure 6 – Service-to-Service Connections Architecture



Source: (ROWLAND *et al.*, 2015)

2.1.3.4 Interactions

The research about human-computer interaction (HCI) started in a world of desktop computers where users own just one computer and evolve to mobile and ubiquitous computing. Thus, a user can own several devices: a mobile phone, a tablet, a notebook, a smart watch, a watchband and many others. With that the world that is making a huge transition towards the Internet of Things, which means that humans will interact with connected things in almost all aspects of everyday life. All these connected things are also capable of interacting with each other (MASHAL *et al.*, 2015) (MINERVA *et al.*, 2015) (HOLLER *et al.*, 2014). Therefore, the interaction between users and IoT systems can be considered from two perspectives: Human-Thing and Thing-Thing interaction.

The first one is an interaction between users and things. The thing can be any object that the user will interact with (*e.g.*, watches, heaters, phones and others.) and that has capabilities of devices such as sensors, actuators, processing, identification and communication.

The area of HCI mentions that there are two types of interaction with computers: explicit and implicit (POSLAD, 2009) (SCHMIDT, 2005). Similarly, we can consider the interaction that takes place in UbiComp and the IoT environment (ZEZSCHWITZ *et al.*, 2015).

The explicit interaction between user and computer is direct communication in which the user explicitly enters data through a graphical interface and I/O devices, such as mice, keyboards, among others. The system, in turn, uses output devices such as displays or speakers, among others, to deliver information. This interaction can happen in many ways in UbiComp and IoT environments because the systems appear in a wide variety of objects with different capabilities of input and output data. Some may use gestures. Others may use interaction based on voice.

The implicit interaction or communication can be initiated by the system itself, given its ability to understand the context (POPPE *et al.*, 2007). The explicit interaction may have no context information, as the implicit interaction uses contextual information about the user, his/her environment and his/her needs, to provide relevant services. Thus, context-awareness is a key feature to make implicit interaction possible between users and systems. An example of implicit Human-Thing interaction is when the door is automatically unlocked by the simple presence of a person's smartphone.

The other perspective, the Thing-Thing interaction, is the communication between things themselves (TAN; WANG, 2010). Actually, the real power of the IoT comes from the things' ability to interact. For example, a smart thing that would switch on the door light can be capable of transmitting information that the door was opened to every other nearby smart object. Therefore, the Thing-Thing interaction refers to the communication and data exchange between things though the Internet (HOLLER *et al.*, 2014).

Most of UbiComp and IoT systems offer a limited Thing-Thing interaction, as seen in the architectures presented in the subsection 2.1.3.3. The interaction usually occurs between two things, for example, between a smartphone and an air-conditioned (ALAN *et al.*, 2016) (YANG; NEWMAN, 2013), or a smartphone and a wearable device.

Based on all these particularities of UbiComp and IoT systems, it is possible to realize that the development of this type of system should consider new techniques, strategies and methodologies, especially regarding the quality of the developed systems.

2.2 Quality Characteristics: A Type of Non-Functional Requirement

Ensuring that systems work in the best way they can is as important as delivering the right functionalities. Software engineers have a crucial role where they need to think about the systems' quality characteristics even from the beginning of the development. In this way, the systems are likely to work in their best way possible. This section discusses several notions related to quality characteristics. These include definitions and classifications, central concepts about how to achieve them in a system, correlations that may occur between these characteristics and catalogs that help software engineers during software design and analysis.

2.2.1 Definitions and Classifications

Quality characteristics are highly related to Software Requirements area. Therefore, firstly, it is necessary to explain the concepts about the mentioned Software Requirements. According to (WIEGERS; BEATTY, 2013) apud (SOMMERVILLE; SAWYER, 1997), software requirements are: *"A specification of what should be implemented. They are descriptions of how the system should behave, or of a system property or attribute. They may be a constraint on the development process of the system."*

This definition is broad and brings the possibility to classify requirements into more than one type, such as functionalities, properties and constraints. However, a traditional classification used regarding software requirements states that they are split into functional or non-functional requirements (respectively, FRs and NFRs) (ECKHARDT *et al.*, 2016).

FRs are the ones that are associated with specific functions, tasks or behaviors that must be provided by the system (NGUYEN *et al.*, 2016) (WIEGERS; BEATTY, 2013), while NFRs are related to how well systems behave (NGUYEN *et al.*, 2016).

Several researchers consider the term "quality characteristics" as synonyms of NFRs (CHUNG *et al.*, 2000) (MAIRIZA *et al.*, 2010) (SILVA *et al.*, 2016). However, the work in (WIEGERS; BEATTY, 2013) states that there are there are three types of NFRs: characteristics the system must present (*e.g.*, Performance), constraints and external interface requirements [2].

This work focuses on NFRs that are quality characteristics[3], because they represent a significant portion of NFRs (WIEGERS; BEATTY, 2013). These characteristics describe the expectations beyond the correct functioning of the system. For example, expectations such as how fast the system is (Performance), how secure it is (Security), and how easy it is to use (Usability) are examples of quality characteristics.

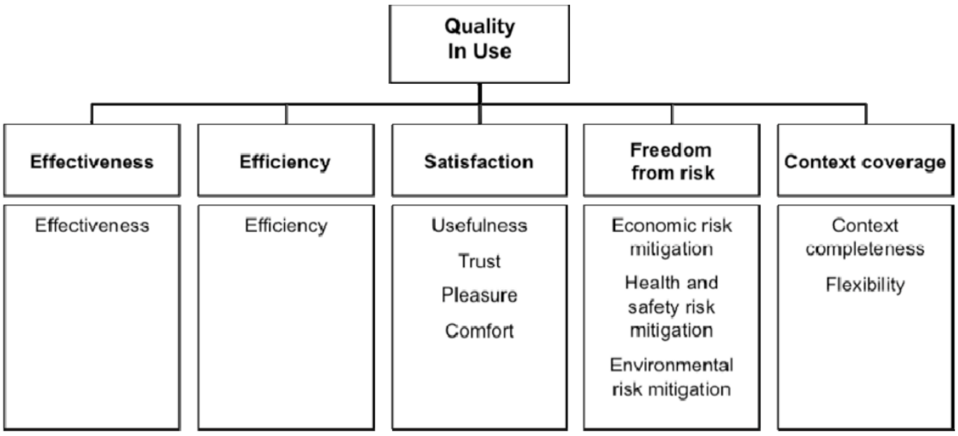
The International Standardization Organization (ISO/IEC 25010, 2011) classifies

[2] There is a huge discussion in the requirements community about the term "non-functional requirement". Several researchers disagree with this concept since many NFRs describe behavioral properties and may be treated as FRs functional requirements (ECKHARDT *et al.*, 2016)

[3] There are several synonyms for this term: quality attribute, quality requirement, quality factors, among others.

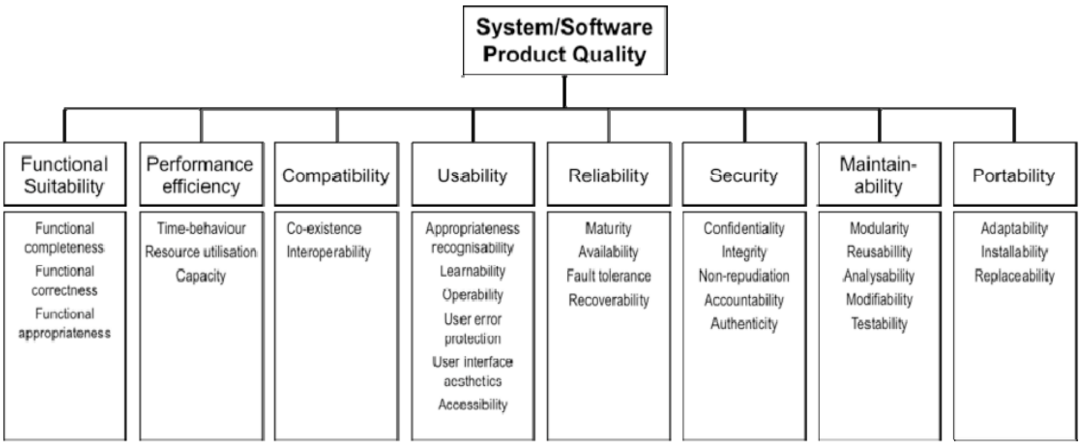
quality characteristics into two models: the Quality in Use model (See Figure 7) and the Product Quality model (See Figure 8). The quality in use model defines five characteristics related to outcomes of interaction with a system: Effectiveness, Efficiency, Satisfaction, Freedom from Risk, and Context Coverage. Then, this model is more important from an end user's point of view. The product quality model defines eight characteristics: Functional Suitability, Performance Efficiency, Compatibility, Usability, Reliability, Security, Maintainability, and Portability. This model is related to an internal and external point of view of software quality.

Figure 7 – Quality in Use Model



Source: (ISO/IEC 25010, 2011)

Figure 8 – Software Product Quality Model



Source: (ISO/IEC 25010, 2011)

Although ISO/IEC 25010 defines these two different models, the standard discusses the influence of the product quality model in the quality in use model depending on the kind of stakeholder. Table 3 presents the influence of quality characteristics. When the column has a star symbol (*), it means the characteristic in the row influences the user in the column.

Table 3 – Influence of the Quality Characteristics

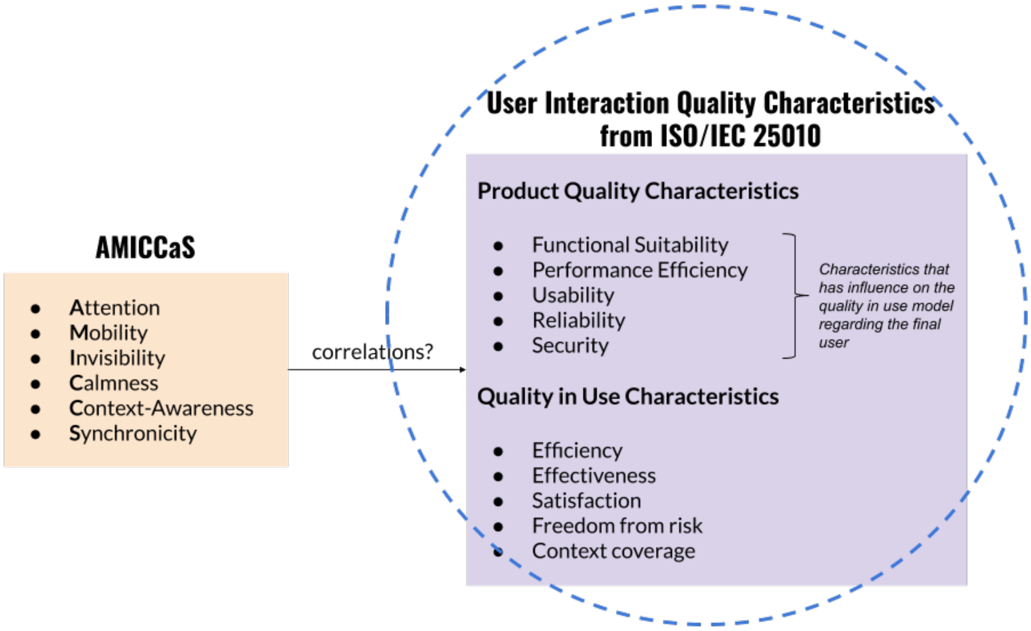
Product Quality Characteristics	Influence on quality in use for primary users	Influence on quality in use for maintenance tasks	Information system quality concerns of other stakeholders
Functional Suitability	*		
Performance Efficiency	*		*
Compatibility		*	
Usability	*		
Reliability	*		*
Security	*		*
Maintainability		*	
Portability		*	

Source: (ISO/IEC 25010, 2011)

Functional Suitability, Performance Efficiency, Usability, Reliability and Security will have a significant influence on the quality in use for primary users. Performance Efficiency, Reliability and Security can also be specific concerns of other stakeholders who specialize in these areas. Compatibility, Maintainability and Portability will have a significant influence on quality in use for secondary users who maintain the system.

The main goal of this thesis is defining an approach to capture, analyze and catalog the impact of AMICCaS on user interaction quality characteristics. Therefore, the information in Table 3 helped this work to define a set of quality characteristics that are particularly important for user interaction quality and then it better defined the scope of this thesis (See Figure 9).

Figure 9 – User Interaction Quality Characteristics



Source: Author.

The focus of the investigation from AMICCaS to traditional user interaction quality characteristics will be the five characteristics from quality in use model (Effectiveness, Efficiency, Satisfaction, Freedom from Risk and Context Coverage, presented in Table 4) and the five characteristics that has influence on quality in use for primary users (Functional Suitability, Performance Efficiency, Usability, Reliability and Security, presented in Table 5).

Table 4 – User Interaction Quality Characteristics from ISO/IEC 25010 Quality in Use Model

	Characteristic	Definition
1	**Efficiency**	Resources expended in relation to the accuracy and completeness with which users achieve goals
2	**Effectiveness**	Accuracy and completeness with which users achieve specified goals
3	**Satisfaction**	Degree to which user needs are satisfied when a product or system is used in a specified context of use
3.1	Usefulness	Degree to which a user is satisfied with their perceived achievement of pragmatic goals, including the results of use and the consequences of use
3.2	Trust	Degree to which a user or other stakeholder has confidence that a product or system will behave as intended
3.3	Pleasure	Degree to which a user obtains pleasure from fulfilling their personal needs
3.4	Comfort	Degree to which the user is satisfied with physical comfort
4	**Freedom from Risk**	Degree to which a product or system mitigates the potential risk to economic status, human life, health, or the environment
4.1	Economic Risk Mitigation	Degree to which a product or system mitigates the potential risk to financial status, efficient operation, commercial property, reputation or other resources in the intended contexts of use
4.2	Health and Safety Risk Mitigation	Degree to which a product or system mitigates the potential risk to people in the intended contexts of use
4.3	Environmental Risk Mitigation	Degree to which a product or system mitigates the potential risk to property or the environment in the intended contexts of use
5	**Context Coverage**	Degree to which a product or system can be used with effectiveness, efficiency, freedom from risk and satisfaction in both specified contexts of use and in contexts beyond those initially explicitly identified
5.1	Context Completeness	Degree to which a product or system can be used with effectiveness, efficiency, freedom from risk and satisfaction in all the specified contexts of use
5.2	Flexibility	Degree to which a product or system can be used with effectiveness, efficiency, freedom from risk and satisfaction in contexts beyond those initially specified in the requirements

Source: (ISO/IEC 25010, 2011)

2.2.2 *Quality Characteristics in the Software Development*

Many studies regarding quality characteristics take an evaluation perspective in the sense that they propose software measures to evaluate the degree to which a system/software product achieves the characteristic (CARVALHO *et al.*, 2018) (KARAISKOS *et al.*, 2009) (KOUROUTHANASSIS *et al.*, 2008). It is essential to have this kind of approach in the software development, however, supporting quality characteristics from the initial stages of requirement

Table 5 – User Interaction Quality Characteristics from ISO/IEC 25010 Product Quality Model

	Characteristic	Definition
1	**Functional Suitability**	Degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions
1.1	Functional Completeness	Degree to which the set of functions covers all the specified tasks and user objectives
1.2	Functional Correctness	Degree to which a product or system provides the correct results with the needed degree of precision
1.3	Functional Appropriateness	Degree to which the functions facilitate the accomplishment of specified tasks and objectives
2	**Performance Efficiency**	Performance relative to the amount of resources used under stated conditions
2.1	Time Behaviour	Degree to which the response and processing times and throughput rates of a product or system, when performing its functions, meet requirements
2.2	Resource Utilization	Degree to which the amounts and types of resources used by a product or system, when performing its functions, meet requirements
2.3	Capacity	Degree to which the maximum limits of a product or system parameter meet requirements
3	**Usability**	Degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use
3.1	Appropriateness Recognizability	Degree to which users can recognize whether a product or system is appropriate for their needs
3.2	Learnability	Degree to which a product or system can be used by specified users to achieve specified goals of learning to use the product or system with effectiveness, efficiency, freedom from risk and satisfaction in a specified context of use
3.3	Operability	Degree to which a product or system has attributes that make it easy to operate and control
3.4	User Error Protection	Degree to which a system protects users against making errors
3.5	User Interface Aesthetics	Degree to which a user interface enables pleasing and satisfying interaction for the user
3.6	Accessibility	Degree to which a product or system can be used by people with the widest range of characteristics and capabilities to achieve a specified goal in a specified context of use
4	**Reliability**	Degree to which a system, product or component performs specified functions under specified conditions for a specified period of time
4.1	Maturity	Degree to which a system, product or component meets needs for reliability under normal operation
4.2	Availability	Degree to which a system, product or component is operational and accessible when required for use
4.3	Fault Tolerance	Degree to which a system, product or component operates as intended despite the presence of hardware or software faults
4.4	Recoverability	Degree to which, in the event of an interruption or a failure, a product or system can recover the data directly affected and re-establish the desired state of the system
5	**Security**	Degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization
5.1	Confidentiality	Degree to which a product or system ensures that data are accessible only to those authorized to have access
5.2	Integrity	Degree to which a system, product or component prevents unauthorized access to, or modification of, computer programs or data
5.3	Non-Repudiation	Degree to which actions or events can be proven to have taken place, so that the events or actions cannot be repudiated later
5.4	Accountability	Degree to which the actions of an entity can be traced uniquely to the entity
5.5	Authenticity	Degree to which the identity of a subject or resource can be proved to be the one claimed

Source: (ISO/IEC 25010, 2011)

analysis and design is equally essential (CHONG *et al.*, 2014).

The translation from characteristics into corresponding technical solutions (or strategy) is part of the requirements analysis and high-level design (WIEGERS; BEATTY, 2013). The solutions can be classified into several categories, depending on which quality characteristic they will help. Table 6 presents the likely categories for them according to the quality characteristics. Development strategies to support a quality characteristic in a system can be of any type, such as a function or some component in architecture, a design constraint, a design guideline.

Table 6 – Quality Characteristics and its Technical Solutions

Quality Characteristic	Likely Technical Solution
Installability, integrity, interoperability, reliability, robustness, safety, security, usability, verifiability	Functional requirement
Availability, efficiency, modifiability, performance, reliability, scalability	System architecture
Interoperability, security, usability	Design constraint
Efficiency, modifiability, portability, reliability, reusability, scalability, verifiability, usability	Design guideline
Portability	Implementation constraint

Source: (WIEGERS; BEATTY, 2013)

Although all characteristics are essential, it is hard or even impractical to achieve all of them simultaneously. One of the reasons is that they have negative correlations with each other (WIEGERS; BEATTY, 2013) (MAIRIZA; ZOWGHI, 2011) (EGYED; GRUNBACHER, 2004) (AFREEN *et al.*, 2016) (BOEHM; IN, 1996) (LIU, 2016). The next subsection will explain in more details the meaning of correlations between quality characteristics and how can software software engineers deal with them during software development.

2.2.3 Correlations between Quality Characteristics

Quality characteristics may conflict or cooperate with each other. Conflicts between quality characteristics mean that achieving one characteristic can negatively impact the achievement of another (BERANDER *et al.*, 2005). Cooperation between quality characteristics means that one can characteristic can help another one (CHUNG *et al.*, 2000). Negative correlations symbolize conflicts and positive correlations represent cooperations.

Figure 10 illustrates a matrix of positive ("plus" signal) and negative ("minus" signal) correlations among pairs of traditional NFRs (WIEGERS; BEATTY, 2013). These are examples of high-level correlations (between quality characteristics). A plus signal means that increasing the characteristic in a row can affect the characteristic in the column positively. A minus symbol means that one characteristic negatively affects the other one.

Figure 10 – Correlations between traditional quality characteristics

	Availability	Efficiency	Installability	Integrity	Interoperability	Modifiability	Performance	Portability	Reliability	Reusability	Robustness	Safety	Scalability	Security	Usability	Verifiability
Availability									+		+					
Efficiency	+				−	−	+	−			−		+		−	
Installability	+								+					+		
Integrity			−		−		−			−		+		+	−	−
Interoperability	+		−	−			−	+	+		+	−		−		
Modifiability	+		−				−		+	+			+			+
Performance		+			−	−		−			−		−		−	
Portability		−			+	−	−			+				−	−	+
Reliability	+	−		+		+	−				+	+		+	+	+
Reusability		−		−	+	+	−	+						−		+
Robustness	+	−	+	+	+		−		+			+	+	+	+	
Safety		−		+	+		−				+			+	−	−
Scalability	+	+		+			+	+	+		+					
Security	+			+	+		−	−	+		+	+			−	−
Usability		−	+				−	−	+		+	+				−
Verifiability	+		+	+		+			+	+	+	+		+	+	

Source: (WIEGERS; BEATTY, 2013)

Another representation of correlations is from (MAIRIZA; ZOWGHI, 2011) and (MAIRIZA; ZOWGHI, 2010). This study splits a negative correlation into three types: absolute conflict (labeled as “x”), relative conflict (labeled as “*”) and non-existent conflict (labeled as “0”). Absolute conflict is related to NFRs that are always in conflict, no matter how developers implement them. Relative conflict is concerning NFRs that sometimes are in conflict and, finally, non-existent conflict means that NFRs are never in conflict with each other.

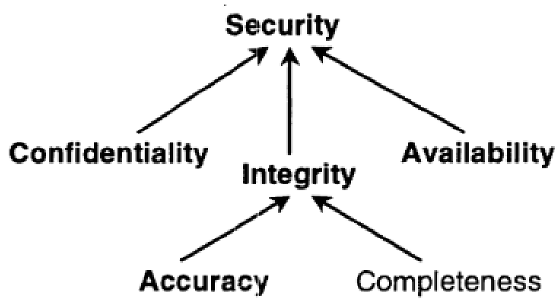
According to (MAIRIZA *et al.*, 2009), most of the research on conflict among NFRs provide documentation, catalogs, or list of potential conflicts among the types of NFRs. These potential conflict catalogs are used to identify and analyze the conflict among NFRs since the beginning of the development. Therefore, this thesis uses the catalog as a solution to deal with correlations between quality characteristics.

2.2.4 *Catalogs for supporting Quality Characteristics*

A catalog is a body of knowledge that engineers accumulate from previous experience (CHUNG *et al.*, 2000). There can be three types of information to catalog regarding quality characteristics: *(i)* subcharacteristics; *(ii)* solutions (*i.e.*, strategies to support the quality characteristic in a system) and *(iii)* correlations.

Subcharacteristics represents knowledge about an NFR (*e.g.*, Security, Performance), which are concepts and terminology, usually documented as subcharacteristics. For example, Figure 11 presents a partial catalog for Security that presents three subcharacteristics (Confidentiality, Integrity and Availability). Subcharacteristic Integrity is composed of two sub subcharacteristics (Accuracy and Completeness).

Figure 11 – Example of Subcharacteristics Catalog



Source: (CHUNG *et al.*, 2000)

A catalog for supporting NFRs can also store development strategies which are intended to satisfy subcharacteristics and are available to developers. Figure 12 shows a catalog of strategies for addressing Confidentiality, subcharacteristic of Security. The catalog is hierarchically organized: more specific strategies are placed under general ones. For example, the strategy "authentication" is composed of five specific strategies: password, cardKey, requireAdditionalID, biometrics and PIN.

Finally, a catalog can store correlations between quality characteristics. Figure 13 presents a partial correlation catalog. It shows that a method called "validation" contributes to Confidentiality (plus sign) but affects negatively (minus sign) Response Time. Additionally, correlations can be documented as a rule (CHUNG *et al.*, 2000), which is expressed in the following format:

Correlation Rule Format

<softgoal x> <kind of impact> <characteristic or subcharacteristic> <condition>.

Where conditions can be used when the correlation is applied to a specific situation that constrains the rule.

Correlation catalogs can be defined with different axes showing correlations between distinct levels of quality characteristics (CHUNG *et al.*, 2000).

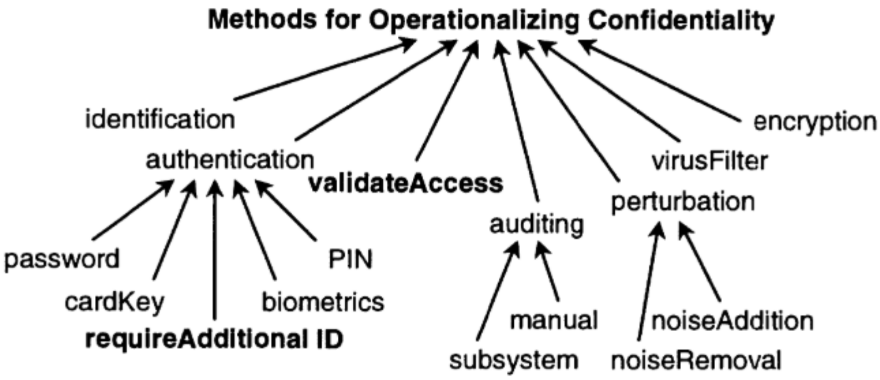
A more formal approach to represent catalogs proposed by (CHUNG *et al.*, 2000) is called Softgoal Interdependency Graph (SIG). SIGs have been used to represent all knowledge about NFRs (quality characteristic, subcharacteristic, strategies, and correlations). Next subsection explains in more details this notation.

2.2.4.1 Softgoal Interdependency Graph (SIG)

SIG is described and proposed in the NFR Framework (CHUNG *et al.*, 2000). Software engineers can use this framework for representing and analyzing non-functional requirements. The key element is the concept of softgoal, which represents a goal that has no clear-cut definition or criteria about its satisfaction. Therefore, softgoals are used to represent NFRs (*i.e.*, quality characteristics) because it is very hard or even impossible that a quality characteristic is completely satisfied.

A SIG is composed of softgoals and links between them. Softgoals can be either NFR softgoals or operationalizing softgoals. NFR softgoals act as overall constraints on the system (*i.e.*, quality characteristics) and are represented as a cloud in the SIG. They can be refined into more specific NFR softgoals. For example, Figure 14 presents a SIG in which the main

Figure 12 – Example of Strategies' Catalog



Source: (CHUNG *et al.*, 2000)

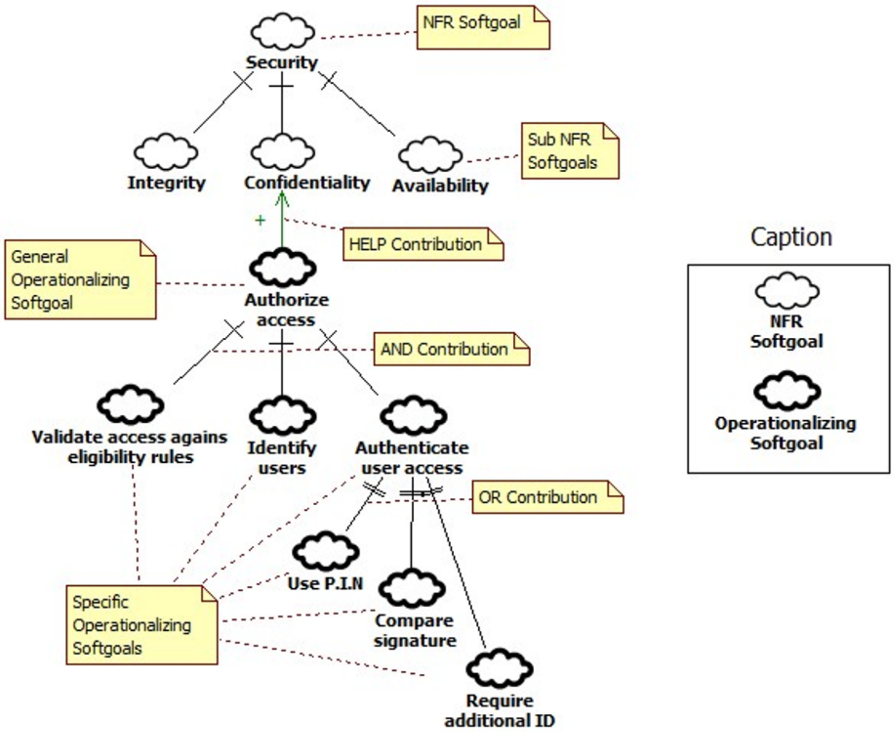
Figure 13 – Example of Correlations Catalog

Informal Correlation Catalogue

Impact of offspring *Operationalizing Softgoal*	upon parent *NFR Softgoal*				
	Accuracy	Confidentiality	Response Time	Space	User-friendliness
Validation	+	+	–		
Compression			–	+	
Indexing			+		
Authorization		+			
Additional ID		+			–

Source: (CHUNG *et al.*, 2000)

Figure 14 – Example of SIG



Source: Adapted from (CHUNG *et al.*, 2000)

NFR softgoal is Security, composed of three specific NFR softgoals: Integrity, Confidentiality and Availability.

The NFR softgoals are satisfied by the operationalizing softgoals, drawn as dark clouds in the graph. They represent design or implementation strategies. For example, in Figure 14, there is an operationalizing softgoal called “Authorize access”, which is used to help Confidentiality. The operationalizing softgoals can even be refined into more specific ones (specific operationalizing softgoals), such as “Validate access against eligibility rules”, “identify users" and "authenticate user access" in Figure 14.

The links between softgoals can be made by using the following relations:

- AND/OR contribution: "AND" means that all the sub softgoals need to be satisfied to make their parent satisfied. This contribution is shown by an arc connecting the lines from the sub softgoals to the parent. Figure 14 presents an example of an AND relation: the "Authorize access" softgoals is refined into "Validate access against eligibility rules", "identify users" AND "authenticate user access". "OR" means that at least one sub softgoals should be satisfied to make their parent satisfied. For "OR" contributions, a double arc connects the lines from the group of sub softgoals to the parent. Figure 14 presents an example of OR contribution. The "authenticate user access" softgoals is refined into "use PIN", "compare signature" OR "required additional ID".
- BREAK/HURT/UNKNOWN/HELP/MAKE contribution: The BREAK correlations (labeled as "–") mean that a softgoal certainly denies the achievement of another softgoal. The HURT correlations (labeled as "-") means that there is a negative partial contribution of a softgoal towards another softgoal. The UNKNOWN correlations (labeled as "?") mean that there is no knowledge about the relation between two softgoals. The HELP correlations (labeled as "+") mean that there is a positive partial contribution. Finally, the MAKE correlations (labeled as "++") mean a certain positive contribution. As an example, Figure 14 presents a HELP contribution: "authorize access" HELPS Confidentiality.

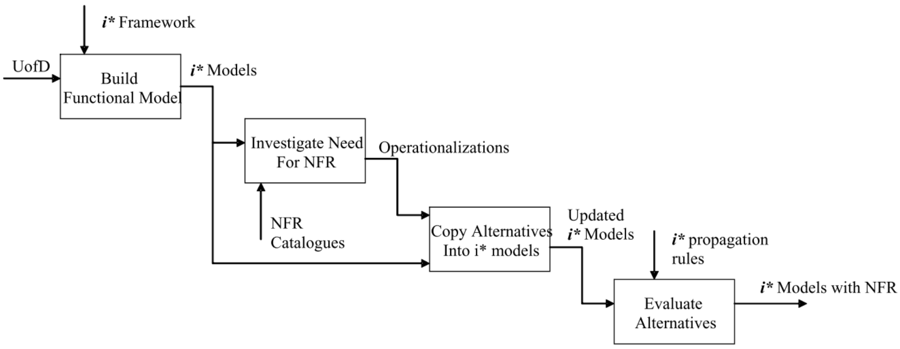
This work focuses on this kind of catalog due to its extensive use in literature for cataloging NFRs (CAPPELLI *et al.*, 2010) (SERRANO; LEITE, 2011) (TORRES; MARTINS, 2014) (ZINOVATNA; CYSNEIROS, 2015) (PORTUGAL *et al.*, 2018).

2.2.4.2 Usage of NFRs Catalogs

NFR catalogs are used in the requirements analysis and specification; and high-level design (WIEGERS; BEATTY, 2013). Two of the goals of these phases are avoiding commitments to conflicting NFRs and reaching a balance among them. Then, it is most used by requirements engineers when they need to check if requirements are in conflict; and also by developers and designers, when they need to determine the best way to satisfy each quality characteristic.

An approach proposed by (CYSNEIROS, 2007) systematize the usage of NFR catalogs into four steps (See Figure 15). The first step is about building functional models, which refers to perform elicitation of functional requirements. This approach uses Framework i*. The second step is to enrich the functional model with NFRs, which is performed with the help of catalogs. Third step aims to select development strategies to the NFRs. This step also needs NFRs catalogs. The final and fourth step is to evaluate the decisions made towards the NFRs through propagation rules.

Figure 15 – Example of Approach to use NFRs Catalogs



Source: Adapted from (CHUNG *et al.*, 2000).

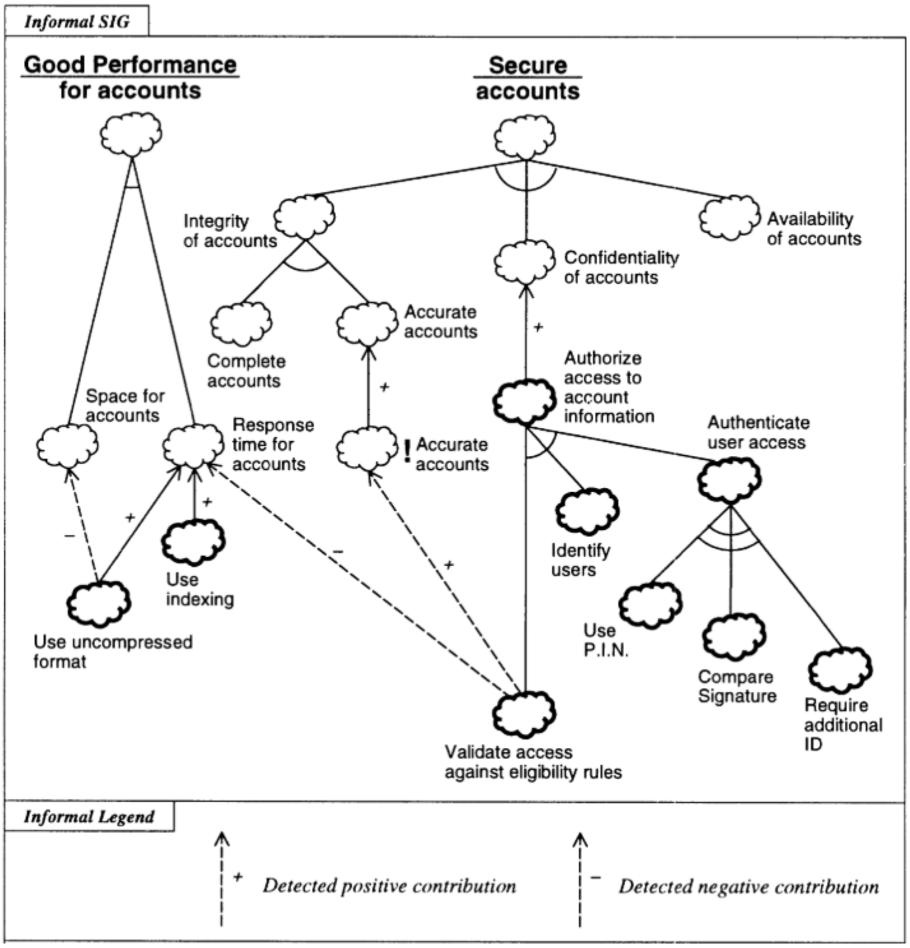
Another approach by which software engineers use catalogs, especially correlation catalogs, is in helping the detection of conflicts between decisions made by the developer. The NFR Framework proposes a mechanism to detect not known relations between different SIGs. For a given set of softgoals, there can be more relations than the amount that has already been explicitly stated when the developer created the model.

Such identification can be made with the help of correlation rules stored in a catalog. Therefore, this detection is done by comparing portions of a SIG to a catalog of correlations. It is then up to the developer to decide whether to incorporate the detected interdependencies into the softgoal interdependency graph. For example, Figure 16 presents two SIGs, one for Performance and another one for Security. A softgoal used to achieve Security called “Validate access against eligibility rules” generates a negative correlation to “Response time for accounts”, softgoal of Performance. Therefore, identifying negative correlations can be viewed as revealing the (hidden) trade-off of a development technique - it can be good for one softgoal but at the same time bad for another (CHUNG *et al.*, 2000).

2.3 HCI Quality Characteristics for UbiComp and IoT Systems

A set of quality characteristics can be particularly important for certain types of software or perspectives of quality (WIEGERS; BEATTY, 2013). For example, Availability is essential for web systems; Reliability is essential for banking systems, and Trust is essential for

Figure 16 – Example of SIG with Negative and Positive Correlations detected



Source: (CHUNG *et al.*, 2000)

the quality of use.

UbiComp and IoT systems moved away from the desktop environment to immerse in the user's everyday objects. They have different sensing possibilities as well as the ability to start an interaction with the user or be present in various objects connecting to the Internet. Therefore, this kind of system changed the way users interact with technology and bring us a set of new quality characteristics important from the user's point of view.

This thesis work uses a set of six HCI quality characteristics of UbiComp and IoT systems. These characteristics were selected based on two studies: a previous work for ubiquitous computing (CARVALHO, 2017), performed through the systematic mapping (SM) method, and a literature survey performed considering only IoT (ANDRADE *et al.*, 2017). Next subsections present each set.

2.3.1 *HCI Quality Characteristics for UbiComp Systems*

A previous Systematic Mapping (SM) study identified a set of HCI quality characteristics and software measures for ubiquitous systems (CARVALHO, 2017). A total of 27 quality characteristics had been identified, as presented in Table 7.

These quality characteristics from UbiComp are similarly essentials for IoT applications since IoT inherits many of UbiComp ideas. Proof of that is the studies performed to evaluate IoT applications used a subset of these characteristics and software measures from UbiComp (ARAGAO *et al.*, 2019) (CARVALHO *et al.*, 2017). Therefore, this work considers that IoT systems can have the same NFRs from UbiComp. However, in order to complement the set of quality characteristics for both UbiComp and IoT applications, an exploratory literature review was performed by only focusing on IoT applications (ANDRADE *et al.*, 2017). Next subsection presents these characteristics.

2.3.2 *HCI Quality Characteristics for IoT Systems*

Rowland (2015) describes a set of quality attributes for IoT systems. This work classifies these characteristics into two perspectives: connection and objects (ROWLAND *et al.*, 2015).

Connection-level Quality Characteristics are related to intrinsic characteristics of the network between the objects. An object in the UbiComp and IoT environment sends data through the Internet to another thing. Therefore, latency, intermittency, and reliability are significant issues to the quality of the interaction.

- Synchrony: many objects in the IoT environment do not keep connected because they want to conserve power. This issue can result in objects being out of sync with other objects. Imagine a heating system where the heating is 19^o C. Then, a user uses the mobile application to turn it up to 21^o C. It can take a few minutes for the heating controller to check for new instructions. During this time, the application shows 21^o C, and the controller shows 19^o C. The time it takes to synchronize the objects cannot be prolonged.
- Responsiveness: there are no certainties about how fast it will be for two objects to communicate with each other through the Internet. Therefore, latency (the time it takes for a message to pass through the network) can be unpredictable. This inability to control latency is a big issue because it affects the system responsiveness, which is a perceivable

Table 7 – HCI Quality Characteristics for Ubiquitous Systems

Characteristic	Definition
Acceptability	Represents the desire to use an application and its utilization rates
Attention	The ability to keep the user's attention to her/his main activity and not on the system and the technology involved
Availability	The service is always available, regardless of hardware, software or user fault, and it is often taken for granted until downtime occurs
Calmness	The ability to prevent users from feeling overwhelmed by information system
Context-Awareness	The ability to perceive contextual information system and proactively adapt its functionality
Device Capability	Properties of the device where the application will run (*e.g.*, screen size, color depth, battery life)
Ease of Use	The system should be easy to use by a target user group
Effectiveness	It refers to completeness in performing tasks proactively adapt its functionality
Efficiency	It refers to the amount of effort and resources required to reach a certain goal in the system
Familiarity	User interactions with the system should improve the quality of her/his work. The user should be treated with respect. The design should be aesthetically pleasing
Interconnectivity	An interconnected network between devices
Invisibility	The ability to hide the system, so users may not be aware of it. Moreover, the interaction is performed through natural interfaces
Mobility	The ability to provide users with continuous access to resources and information system, regardless of its location within the limits of the systems
Network Capability	Represents the collection of network information (*e.g.*, signal strength, delay, jitter)
Predictability	The ability, from past experiences, to predict the result of the system
Privacy	The ability to maintain information and data protected
Reliability	The ability to maintain a particular level of performance when used under specific software conditions
Reversibility	The user's activities should be reversible to be able to restore to pre-existing states of the system
Robustness	Degree to which a system or component can execute correctly in the presence of invalid inputs or stressful environmental conditions
Safety	The level of risk of harm to people, business, software, hardware, property or the environment in a specified context of use
Scalability	The ability to provide services to a few or a large number of users
Security	The protection to transport and to store information and also security controls who can access, use and modify context information
Simplicity	The user interface and the instructions should be simple
Trust	It is the belief of the user that the system uses data properly and not cause any harm. Implies awareness, privacy and control
Usability	The ability of the software to be understood, learned, used and attractive to the user, when used under specified conditions
User Satisfaction	The degree of user satisfaction and how the system is attractive for the user
Utility	The ability to provide value to user. The system provides a contribution to user that was not available before its development

Source: Author.

characteristic for final users.

- Reliability: another aspect of the network that may impact the interaction quality is

Reliability. Reliability is the degree to which a system, product, or component performs specified functions under specified conditions for a specified time (ISO/IEC 25010, 2011). In UbiComp and IoT systems, it is crucial if a message gets through or not. In the real world, if a person turns a light switch on, something is expected to happen.

Object-level Quality Characteristics are related to components/characteristics that the objects may have, and that can be very challenging for the interaction quality. The reason is that IoT is allowing an explosion in the diversity of objects connected to the Internet (CARY; JUMPELT, 2016). It is possible to see familiar objects with connectivity and increased computational power. Sensors and actuators create new possibilities for human-things interaction. However, many of these things have to make careful use of energy and computing power (ROWLAND *et al.*, 2015). Also, they have to lead with the generation of several millions of data items over its lifetime (VASSEUR; DUNKELS, 2010). Context-awareness plays a critical role in deciding which data needs to be processed (PERERA *et al.*, 2014). Therefore, battery consumption, context-awareness, and interoperability are essential characteristics of interaction quality. These quality characteristics are better described above.

- Battery Consumption: many connected things run on batteries. The usage of batteries can be because they are typically being moved around and are not using a power plug. Some things need to be connected all the time, and this requires much energy. Managing the trade-off between network and battery is a big issue. If a thing goes offline because of an empty battery, all other things will not be synchronized. Batteries and power packs are problematic in several scenarios because of their size, weight, and maintenance requirements (FRIEDEMANN; FLOERKEMEIR, 2011).
- Context-Awareness: smart object decide and react automatically. It does not necessarily require the initiation by the user. Example: Sun sensors control the shutters of the building. If the sun is shining at a specific angle, the shutters are moved down (ZEZSCHWITZ *et al.*, 2015). This example of decision implies that things must have the ability to sense the environment and the user correctly. If a single thing fails about the context, a chain of failures can happen between things that are connected with it.
- Interoperability: many devices are locked in proprietary ecosystems. In a real IoT, anything can connect mutually, but for now, things do not interoperate in this form. This lack of Interoperability represents a severe problem for the quality of interaction. Imagine a house with 20 connected things from different producers. This way, a user may have 20 separate

applications to access and control things. A simple task may become a terrible waste of time.Furthermore, these things may not work together. For example, a user may want the light turned on when the alarm rings. If they are from different producers, it may not be easy to make them work together (ROWLAND *et al.*, 2015).

- Difficult of installation: the installation of the devices from a UbiComp and IoT system revolves around the process of connecting an object with other already existent objects. This process can be challenging for a user, since they may use different network technologies (JEWELL *et al.*, 2015) (CHONG *et al.*, 2014) (SUOMALAINEN, 2014) (BROWN *et al.*, 2013).

2.3.3 AMICCaS: Specific HCI Quality Characteristics for UbiComp and IoT Systems

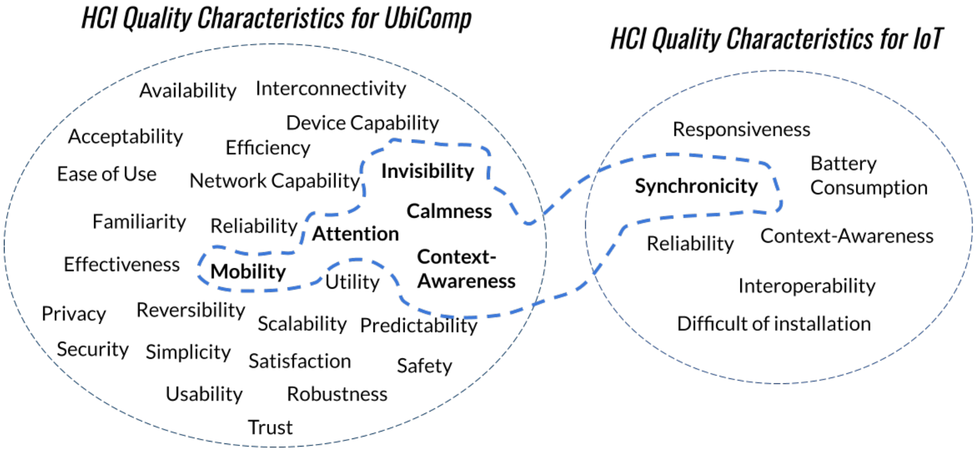
The two sets of characteristics presented in the previous sections are presented in Figure 17. Analyzing the quality characteristics for UbiComp (See left side of Figure 17), it was realized that:

- Several characteristics are already defined with the same name in the System/Software Product Quality Model, as follows: Usability, Reliability, Availability and Security;
- Several characteristics can be subcharacteristics of existing characteristics. For example, Simplicity and Familiarity characteristics are components of Usability. Privacy can be encapsulated in Trust. Robustness in Reliability. Network capability, Device capability, Scalability, and Interconnectivity should be taken into account while evaluating Performance Efficiency as defined in SQuaRE (ISO/IEC 25010, 2011); and
- Other characteristics are presented in (NIELSEN, 1994) for user interface evaluation (Acceptability, Utility, Usability, and Ease of Use), usually used to evaluate the quality in use model characteristics.
- There are specific quality characteristics for UbiComp systems: Context-Awareness, Mobility, Invisibility, Calmness, and Attention.

Regarding the HCI quality characteristics for IoT (See the right side of Figure 17). Two of them are already represented on the first set (Context-Awareness and Reliability). From the rest of them, only one was considered specific for IoT: Synchronicity, which is the ability of the things to be synchronized with each other.

Taking into account that IoT inherits UbiComp solutions, this thesis work also considers that the quality characteristics from UbiComp can be used in IoT. Therefore, this

Figure 17 – HCI Quality Characteristics for UbiComp and IoT



Source: Author.

work has selected a set of six quality characteristics as essential for the quality of interaction in UbiComp and IoT systems (CARVALHO, 2017) (ANDRADE *et al.*, 2017). This set is called **AMICCaS** - **A**ttention, **M**obility, **I**nvisibility, **C**ontext-Awareness, **C**almness, and **S**ynchronicity, explained as follows:

- Context-Awareness is the system's capability of perceiving contextual information related to the user, the system and the environment;
- Invisibility is the ability to hide the system, so users may not be aware of it. Moreover, the interaction is performed through natural interfaces;
- Attention refers to verify if the user's focus is on daily activities such as walking and driving rather than in technology;
- Calmness is the ability to prevent users from feeling overwhelmed by the systems;
- Mobility refers to the continuous use of systems as the user moves across several devices and through various networks; and
- Synchronicity is the ability of things to be synchronized with each other.

2.4 Chapter Summary

This Chapter described the main concepts related to the topics involved in the thesis work: Ubiquitous Computing and Internet of Things, Quality Characteristics and HCI Quality Characteristics for UbiComp and IoT systems.

Regarding Ubiquitous Computing and Internet of Things, this chapter first presented a brief overview of ubiquitous computing, emphasizing its main goals. After that, the evolution to what is now called the Internet of Things has been explained. It is concluded that IoT is an extension of UbiComp, but this work considers systems that are in the scope of these two environments, which is called UbiComp and IoT systems.

Afterwards, concepts about these systems were presented, such as their main components (objects and connections), examples of architectures and the interaction with user. Thus, it was possible to conclude that such kind of systems require that the engineering activities must be revisited.

Second, this chapter has introduced concepts of quality characteristics, which, as explained, represent a large part of non-functional requirements. In this part, this document explains the definition of the user interaction quality characteristics set, which was defined based on the quality models (Quality in Use and Product Quality) proposed by the International Organization for Standardization (ISO/IEC 25010, 2011) and its discussion about the influence of the Product Quality Model in the Quality in Use Model. The characteristics that comprise the set are: Functional Suitability, Performance Efficiency, Usability, Reliability and Security.

Furthermore, concepts about correlations that may exist between quality characteristics and catalogs to support the software engineers were also presented.

Finally, the set of specific quality characteristics for UbiComp and IoT systems were presented. At the end, this work considers the following ones: Context-Awareness, Mobility, Invisibility, Attention, Calmness and Synchronicity, named AMICCaS.

The next chapter presents the related work to this thesis through an exploratory study that collected several catalogs of quality characteristics.

3 RELATED WORK

This Chapter presents the related work to this thesis, which serves as a basis for understanding the research gaps pointed out in this work. Therefore, this chapter answer the RQ1 - *What are the existing NFRs catalogs and how they are defined?* Firstly, Section 3.1 presents the research method used to collect and analyze related studies. Then, Section 3.2 presents the results. In Section 3.3, the results are discussed in terms of key findings, research opportunities and threats to validity. Finally, Section 3.4 summarizes this chapter.

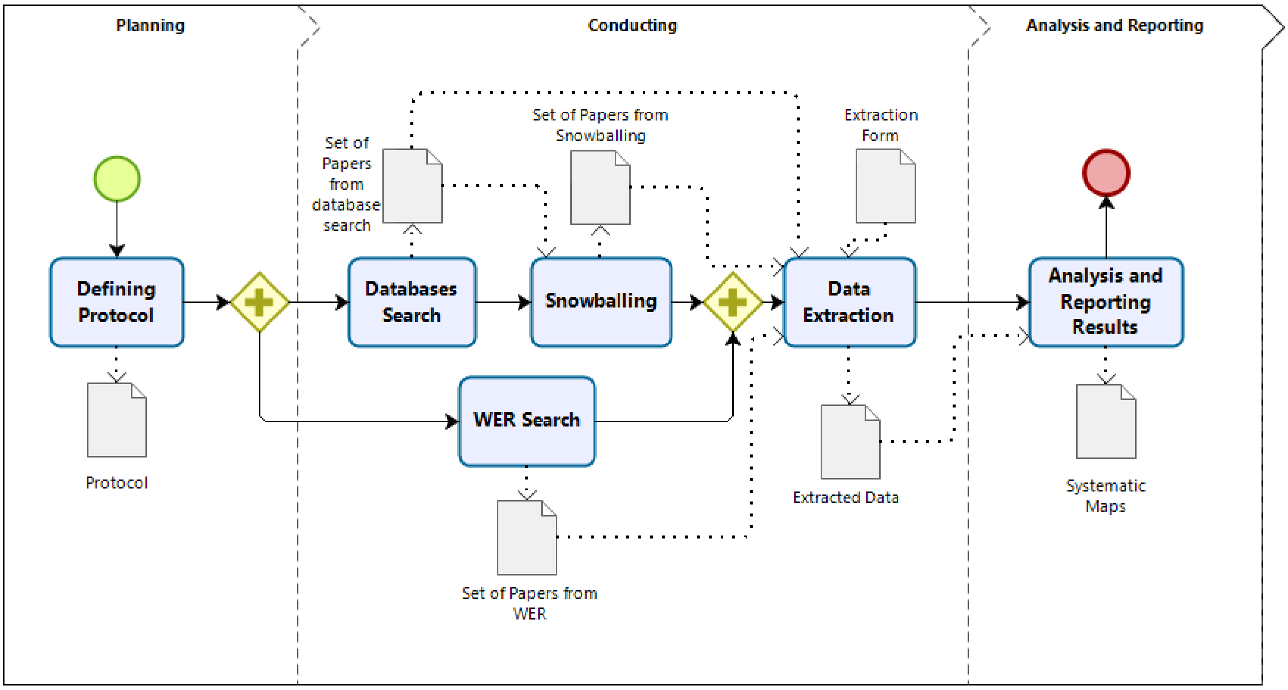
3.1 Research Method

There are several catalogs available in the literature. Many are scattered in several online libraries of scientific studies. Despite their importance, the requirements community lacks a synthesized study about NFRs catalogs in order to understand how existing catalogs are constructed, represented and evaluated. Therefore, to collect an overview of the state of the art and then understand better the related studies, this work performed a systematic mapping (SM) study. The aim is to identify what NFRs catalogs have been proposed in the literature, how they have been represented, defined and evaluated. SM studies are a research method that provides an overview of an area and allows to discover research gaps and trends.

To perform this SM study, a process with three main phases proposed by (KITCHENHAM; CHARTERS, 2007) for systematic studies: *(i)* Planning; *(ii)* Conducting, and *(iii)* Analysis and Reporting (see Figure 18). These phases are usually performed in systematic studies.

In Planning phase, the research protocol should be defined (KITCHENHAM; CHARTERS, 2007). This document combines all information necessary to perform the study. In conducting phase, the researcher should execute what was planned in the protocol. In this work, the conduction was performed through a databases' search followed by a snowballing and, additionally, a search in a specific research event (Workshop on Requirements Engineering - WER) was performed. Then, data extraction took place, where papers from databases, snowballing and WER were used as inputs together with the extraction form. By having all the data, the analysis and report was performed. The next subsections better explain how and why each activity was done.

Figure 18 – Systematic Mapping Process



Source: Author.

3.1.1 Planning

A mapping protocol is usually composed of the following information (KITCHENHAM; CHARTERS, 2007): research goal, questions, databases, selection criteria, screening process, and extraction strategy. All this information is explained as follows.

3.1.1.1 Research Goal and Questions

The goal of this SM is to provide an overview of the literature regarding NFRs catalogs. To achieve our goal, this work investigates the following questions (Systematic Mapping Research Question (SM-RQ)):

- **SM-RQ1 - What NFR catalogs have been proposed in the literature?** This question aims to understand what the existing catalogs are and more specifically what NFRs are covered by these catalogs and how they are classified.
- **SM-RQ2 - How is the information represented in the NFR catalogs?** This question seeks to identify the forms of representation of the extracted catalogs (such as Softgoals Interdependency Graph, Matrices, and others).
- **SM-RQ3 - How are the NFR catalogs defined?** This question aims to check what the approaches are used to define all information presented in these catalogs. In this way,

researchers who want to build their own catalogs can use the answers to this question to guide themselves.

- **SM-RQ4 - How are the NFR catalogs evaluated?** This question aims to know how the proposed catalogs have been evaluated regarding their usefulness and effectiveness.

3.1.1.2 Search Terms, String and Databases

The strategy PICO (Population, Intervention, Comparison, and Outcomes) suggested by (PETERSEN *et al.*, 2015) was considered to identify keywords and formulate the search string. However, only P and I were used. According to (PETERSEN *et al.*, 2015), the other dimensions (C and O) could restrict the search too much and remove important papers. After testing several combinations of terms, the final set is presented in Table 8.

Table 8 – Search Terms

Population	non-functional requirements
	NFR OR “quality characteristic” OR “quality attribute” OR “non-functional property” OR “extra-functional requirement” OR “non-behavioural requirement” OR “quality requirement” OR “quality factor”
Intervention	catalog
	catalogue OR SIG OR “softgoal interdependency graph”

Source: Author.

The population refers to specific roles of software engineering, an application area or an industrial group. In the context of this work, the population is non-functional requirement. There are several synonyms for this term (*e.g.*, quality attribute, quality requirement), which were used in the set of search terms. Intervention refers to a methodology, tool, software procedure or something to be investigated regarding the population. In this work, the intervention is the catalogs. As synonyms, “catalogues” and “softgoal interdependency graphs” were used since this notation has been widely used as NFRs catalogs (CHUNG *et al.*, 2000).

The final search string is presented bellow. To evaluate the quality of it, three control papers that were knew before the execution of the search were selected (MAIRIZA; ZOWGHI, 2011) (NIXON, 2000) (CYSNEIROS, 2007). If this string would find the control papers, then the search string could be considered good.

Search String

(*“quality characteristic”* OR *“non-functional requirement”* OR NFR OR *“quality attribute”* OR *“non-functional property”* OR *“extra-functional requirement"* OR *“non-behavioural requirement”* OR *“quality requirement”* OR *“quality factor”*) **AND** (*catalog* OR *catalogue* OR SIG OR *“softgoal interdependency graph"*)

Once the string has been defined, it was applied in two general databases: Web of Science and Scopus. They were chosen, because, according to (SANTOS *et al.*, 2017), they have good coverage and stability, as well as Scopus cover other bases, such as IEEE.

In addition to the database searches, a backward snowballing was performed. This procedure was done, because four papers regarding NFRs’ Catalogs (MAIA *et al.*, 2009) (FEITOSA *et al.*, 2015) (EGYED; GRUNBACHER, 2004) (GARCIA-MIRELES *et al.*, 2015), known before the execution of this study, were not found out by the database search, since they did not call what they were presenting as “catalog” or even “Softgoal Interdependency Graph”, even though one of them used SIG to catalog requirements (MAIA *et al.*, 2009).

Furthermore, one specific database from the requirements area was also added: Workshop on Requirements Engineering (WER). This workshop was added for three reasons: (i) it represents an important event in the area where researchers usually publish NFRs catalogs; (ii) publications from there are not all indexed in the databases this study previously chose; and (iii) it provides a search engine of its own, making the search for catalogs easier.

Snowballing was not performed in the selected papers from WER, because the four previously known papers (MAIA *et al.*, 2009) (FEITOSA *et al.*, 2015) (EGYED; GRUNBACHER, 2004) (GARCIA-MIRELES *et al.*, 2015) were found out during the snowballing from the database search (i.e., Web of Science and Scopus).

3.1.1.3 Selection Criteria

The following Inclusion Criteria (IC) and Exclusion Criteria (EC) to select the most suitable studies were defined: IC1 - the study presents a NFR catalog; EC1 - the study is not written in English; EC2 - the study is not from Computer Science or Engineering related areas; and EC3 - the paper does not present a NFR catalog. It is important to notice that short papers or books were not excluded. In this systematic mapping, all kind of study was accepted if they meet the criteria.

3.1.1.4 Data Extraction

An extraction form was elaborated to be used for each selected paper in this phase. This form is organized into four parts. The first one is about the papers that were accepted during the conduction phase: title, authors, year, source (Scopus, Web of Science, Snowballing or WER), location venue, type of publication (*e.g.*, conference, journal) and main contribution of the paper.

The other parts are related to data to answer the research questions. For SM-RQ1, the following data was defined: catalog ID, type of the catalog, NFRs considered in the catalog, subcharacteristics, development strategies, quantity of negative and positive correlations, level of correlation and domain area by which the catalog was developed. Also, a data item called "Original Authorship" was defined because many studies did not by themselves proposed a catalog, but it used a catalog from another study. Thus, it was possible to obtain the information about the origin of the catalog to get more information and even more catalogs.

Regarding SM-RQ2, the following data were defined to be extracted: general form of representation for each information in the catalog extracted by SM-RQ1 and form of representation for correlations. For answering SM-RQ3, data about how each component in the catalog (subcharacteristics, strategies, correlations) was defined. Finally, regarding SM-RQ4, data about how the catalog was evaluated should be extracted.

3.1.1.5 Data Analysis

In this work, the results should be analyzed according to the type of data extracted. Table 9 summarizes the analysis strategy and type planned to be used. For SM-RQ1 and SM-RQ2, a quantitative analysis should be performed due to the nature of the data collected, which would be numerical in a nominal scale. Measures like central tendency (mode) and dispersion (frequency) could be used to present the results. Regarding SM-RQ3 and SM-RQ4, to better explore the data, it was decided to use a qualitative method and, consequently, answer more appropriately the question. This work used Content Analysis (CA) (BARDLN, 1977).

CA is seen as a research method to classify any kind of communication material into identified categories of similar meanings (CHO; LEE, 2014). It is suitable for subjective interpretation of the content of text data through the systematic classification process of coding and identifying themes or patterns (HSIEH; SHANNON, 2005). Therefore, this method comes as

Table 9 – Systematic Mapping Data Analysis

Question	Types of Analysis	Strategy of Analysis
SM-RQ1, SM-RQ2	Quantitative	Descriptive Statistic: Mode and Frequency
SM-RQ3, SM-RQ4	Quantitative and Qualitative	Descriptive Statistic: Mode and Frequency, and Content Analysis Method

Source: Author.

a strategy to better analyze the extracted data in order to create categories of different approaches regarding how the catalogs have been defined and how they have been evaluated.

The same procedures from (CHO; LEE, 2014) was planned to be followed to perform the inductive approach for the qualitative content analysis. The unit of analysis is all extracted data for answering both questions SM-RQ3 and SM-RQ4.

3.1.2 Conducting

As presented in Figure 18, this phase was performed in four steps. The first one is the conduction of a search in two digital databases. The second step is the snowballing based on the papers selected in databases search, and, then, the third step consisted of searching papers at all proceedings of the Workshop on Requirements Engineering (WER). Finally, the last step is the Data Extraction. All these steps are described in the next subsections.

3.1.2.1 Database Search

This step is about selecting studies in Scopus and Web of Science libraries. This selection started by applying the search string into the search machines of those libraries. To select the most suitable set of papers after applying the search string, a screening step was performed using the following five filters: (Filter 1) Applying EC1 and EC2 in the found studies; (Filter 2) Excluding duplicate studies; (Filter 3) Applying the exclusion criteria in the abstract and title reading; (Filter 4) Applying the exclusion criteria in the introduction and conclusion reading, also apply the exclusion criteria by checking if there is an image of a catalog in the study (*e.g.*, graph, table); and (Filter 5) Applying the exclusion criteria in the entire paper reading.

Filter 1 was performed in the own search machines (Web of Science and Scopus) whereas the other filters (2, 3, 4 and 5) were performed with support of Parsif.al tool[1]. Filters 3, 4 and 5 were executed by peers - an undergraduate student and the author of this thesis. First, the

[1] https://parsif.al/ - Free online collaborative tool for systematic studies

student performed the filter alone. Then, the author reviewed the papers excluded by the student. For example, if the student had excluded a paper by reading the abstract and title, the author would review them to check whether she agrees with the exclusion. If not, the paper would be included in the study again.

3.1.2.2 Snowballing

This work also performed the backward snowballing procedures defined by (WOHLIN *et al.*, 2013) to complement the set of papers found by the database search. The backward procedure consists of checking the references list of a set of papers. In the case of this work, the set of papers were the ones selected in the databases searches.

The procedure from (WOHLIN *et al.*, 2013) was adapted and used to conduct this search, consisting of four filters. These filters are not totally similar to the filters used earlier since the selection through backward snowballing is slightly different. First, all references of each accepted paper from databases searches were manually reviewed. Then, for each reference the following filters were applied: (Filter 1) Applying all exclusion criteria in the reference. In this case, it was necessary to be more specific since only titles, authors, venue and year were being read. Thus, papers who presented any keyword from the search string were accepted; (Filter 2) Applying all exclusion criteria in the abstract reading; (Filter 3) Applying the exclusion criteria in the most relevant part of the papers (introduction, conclusion, and images of catalogs); and (Filter 4) Applying the exclusion criteria in the entire paper reading.

Then, the Data Extraction step with the same extraction strategy was followed to collect data from the papers selected by the snowballing procedure.

3.1.2.3 WER Search

The Workshop on Requirements Engineering (WER) exists since 1998 and has been an advance for the Ibero-American community of researchers. This workshop provides a Google search engine that explores papers in all WER editions. Therefore, the same search string was applied on it and the same filters were used in the databases searches. Also, just like the other papers, studies selected from WER search also used the same extraction strategy.

3.1.2.4 Data Extraction

After performed all searches, the data extraction took place. In this way, first, the extraction of data was done in the papers found in the electronic databases. Then the extraction was done in the papers of snowballing and, finally, the extraction was carried out in the papers from the workshop.

Regarding to the papers obtained by the databases, the same student who performed the filters during the filtering phase also performed the data extraction. Each extraction was reviewed by the author of this thesis. The extractions of the papers obtained by the snowballing and the workshop were made by the author of this thesis and reviewed by the student.

During extraction, the extraction form was updated to include a data not considered during planning: type of correlation. Through one of the obtained papers, it was possible to note that the correlations can appear between requirements (*e.g.*, Usability and Performance), which is called INTER-NFR, or within a same requirement (*e.g.*, Performance subcharacteristics conflict with one another), called INTRA-NFR. Thus, all papers that had already been extracted were revised again to extract this specific data.

All information extracted was managed in a online collaborative spreadsheet editor. At the end of this phase, the data to draw conclusions was analyzed and then it was possible to give an overview about this topic.

3.1.3 Analysis and Reporting Results

As presented in Figure 18, this phase was about analysis and reporting results. SM-RQ1 and SM-RQ2 were indeed analyzed with the measures Mode and Frequency, generating bar and line graphs, as well as a word cloud and a frequency table. SM-RQ3 and SM-RQ4 were also analyzed using quantitative measures, but it was also analyzed through a qualitative method, Context Analysis.

According to (CHO; LEE, 2014), there are two approaches of conducting qualitative content analysis: inductive approach and deductive approach. The inductive approach is suitable when prior knowledge regarding the topic under investigation is limited or fragmented. Therefore, codes, categories, or themes are directly drawn from the data. The deductive approach starts with preconceived codes or categories derived from prior relevant theory, research, or literature. In this work, an inductive approach was used to analyze data from both SM-RQ3 and SM-RQ4

(See Figure 19).

Figure 19 – Procedures used in Content Analysis - Inductive Approach

Source: Adapted from (CHO; LEE, 2014).

The main procedure for qualitative content analysis is the coding. According to (CHARMAZ, 2006), coding means that labels are attached to segments of data that depict what each segment is about. In open coding, each extracted data form was read. Then, the preliminary codes that emerged from the text were formulated. After that, all texts were codified with those codes. Every time a new information was found out but it did not fit into an existing code, a new one was created. The next steps were to review these codes in order to group the similar ones into categories, always with the care that the categories are mutually exclusive (a rule in CA method). The MAXQDA12 tool (GODAU *et al.*, 2004) was used to support all these procedures.

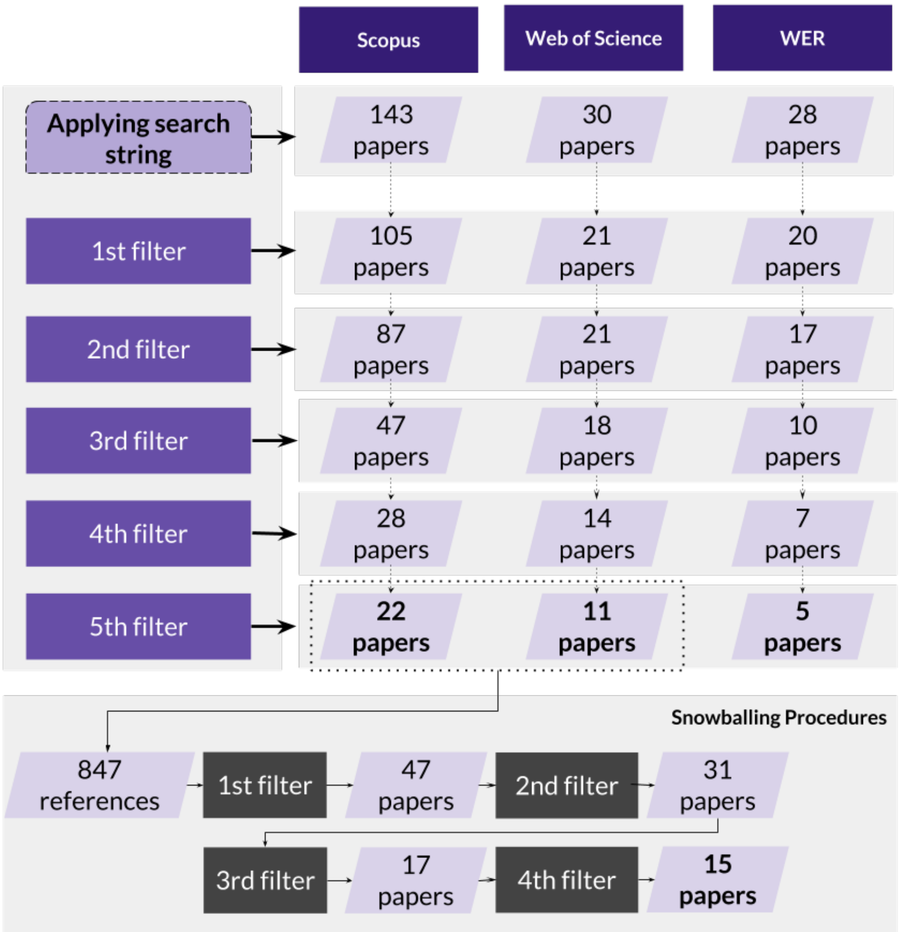
In the case of data from SM-RQ3, existing techniques to define knowledge were already known, however, the knowledge about this topic was still minimal and very fragmented. In the case of data from SM-RQ4, which were data about evaluation, approaches, such as experiment and case study, were knew, but it was clear that the extracted data could bring more information on the evaluation of this specific artifact: NFRs catalogs.

3.2 Results

Figure 20 presents the results from all searches during the conducting phase. In summary, the databases search resulted in 173 papers (143 from Scopus and 30 from Web of Science). After filtering them according to the selection criteria, 33 papers were accepted. This set of papers was defined as the start set for the backward snowballing, which after screening them, the result was a set of 15 additional papers.

Besides, the WER search, which initially brought 28 papers after applying the search string in their google engine, resulted in 5 papers. Therefore, this systematic mapping study resulted in 53 papers (33 from databases, 15 from snowballing and 5 from WER). They are listed in Appendix A. Next subsections present the results for each research question.

Figure 20 – Results from Filtering



Source: Author.

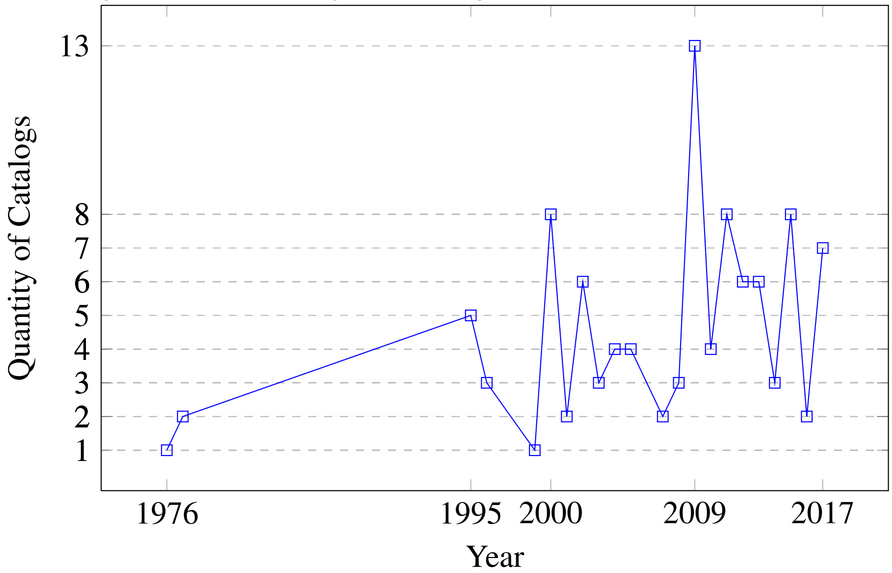
3.2.1 SM-RQ1 - What catalogs have been proposed in the literature?

102 catalogs were found out, ranging from 1976 to 2017 by year of publication, being 2009 the year with most published catalogs (13), as illustrated in Figure 21.

During extraction, it was realized that many catalogs cannot be classified exclusively in one of the three kinds of catalogs: catalog of subcharacteristics, catalog of implementation or design strategies and catalog of correlations. The found catalogs present a mix of stored knowledge. Many of the found catalogs (30) present three types of information: Subcharacteristics, Strategies, and Correlations (See Figure 22). Others present only subcharacteristics (28) or present subcharacteristics and strategies (22). Therefore, this work came up with a different classification from the previous one presented in Background Section for types of catalogs:

- T1. Subcharacteristics - this kind of catalog stores only decomposition of NFRs into more specific NFRs, which this work calls as subcharacteristics;
- T2. Subcharacteristics and Strategies - this catalog stores not only decompositions of

Figure 21 – Quantity of Catalogs vs. Year of Publication



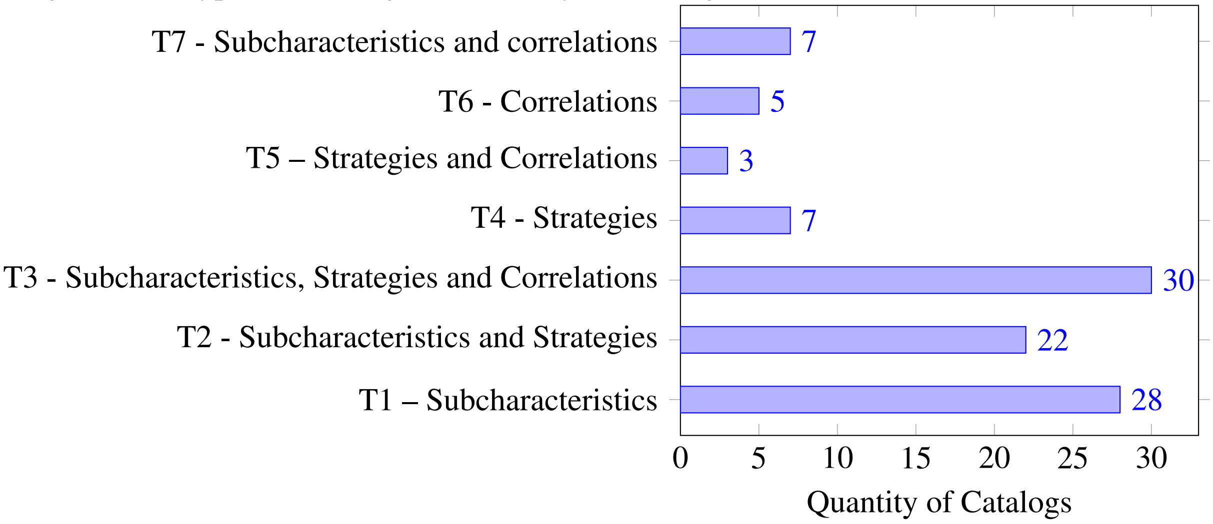
Source: Author.

NFRs, but also development strategies that help achieve subcharacteristics;

- T3. Subcharacteristics, Strategies and Correlations - this kind of catalog contains all the three knowledge;
- T4. Strategies - this catalog stores only strategies to operationalize NFRs. Therefore, this kind of catalog does not present subcharacteristics;
- T5. Strategies and Correlations - this kind of catalogs store correlations between strategies of NFR;
- T6. Correlations - these catalogs store correlations between NFRs, not considering subcharacteristics. That means the correlations are defined in the level of characteristics;
- T7. Subcharacteristics and correlations - this catalog stores correlations between subcharacteristics of NFR. Therefore, the catalog classified in this type should present subcharacteristics and correlations;

The NFRs supported by these catalogs were also analyzed. In total, 348 NFRs were extracted. However, this number includes duplicated NFRs, which means they had the same name or were synonyms. Therefore, they were analyzed in order to establish a set with different NFRs. The analysis was performed in two steps. The first was the removal of NFRs that had exactly the same name. The second step consisted of grouping NFRs with similar names, such as traceability and traceable, functional suitability and functionality, among others. At the end of this analysis, it was concluded that these catalogs support 86 different NFRs. Figure 23 presents all NFRs and Table 10 presents the frequency of the five most cited NFRs. Performance was

Figure 22 – Types of Catalogs vs. Quantity of Catalogs



Source: Author.

the most cited (34), followed by Security (29), Usability (23) and Reliability (14). Regarding subcharacteristics and strategies, 1269 and 1113 were found out, respectively.

Figure 23 – Word Cloud of NFRs identified in the SM Study

Source: Author.

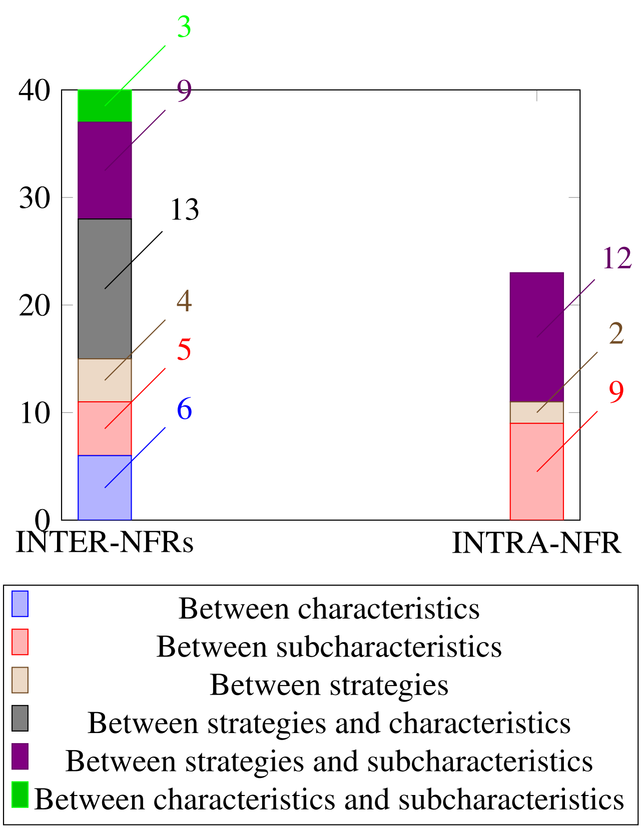
Table 10 – Frequency of NFRs in the SM Study

NFR	Frequency	Relative Frequency
Performance	34	9,7%
Security	30	8,6%
Usability	23	6,6%
Reliability	14	4%
Reusability	14	4%

Source: Author.

With respect to the correlation catalogs, they present 473 positive and 395 negative correlations in total. They are mostly classified as INTER-NFRs (24 out of 45 catalogs that present correlations). Furthermore, there are also catalogs presenting INTRA-NFR correlations (11) and catalogs presenting both (10). Figure 24 shows the type of correlation regarding the level of correlation. Catalogs that are INTER-NFRs type have most correlations in the level "between strategies and characteristics", 13 catalogs in total. While for INTRA-NFR catalogs, the level of correlation is "between strategies and subcharacteristics". This information is interesting because it gave the idea that no matter what type of correlation is, the catalogs show them in a more specific level, which is considered good for developers who wants to make a quick decision.

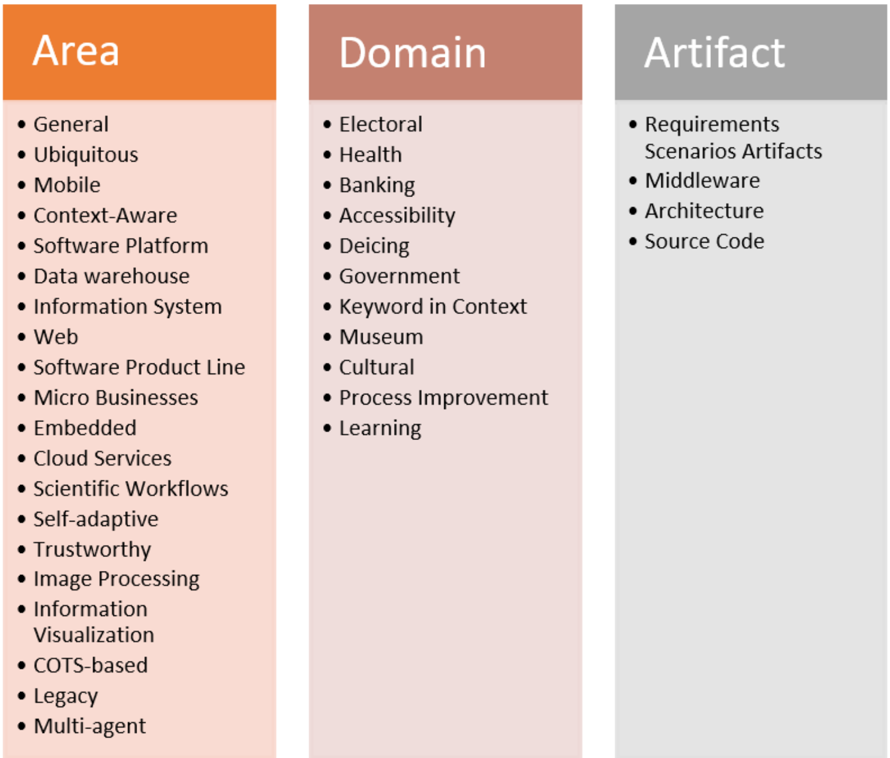
Figure 24 – Type of Correlation vs Level of Correlation



Source: Author.

Furthermore, while analyzing the domain by which the catalog was proposed, it was found out that there are catalogs specific not only for a domain, but also for areas and artifacts. Figure 25 presents them, ordered by highest frequency. Area represents the kind of system the catalog is dealing with. For example, there are catalogs specific for mobile applications, web application or embedded applications. Domain represents the kind of problem the catalog is proposed to. For example, health, learning, banking are domains. Artifact represents the part of

Figure 25 – Domain, Area and Artifacts of Catalogs identified by the SM Study



Source: Author.

the system for which the catalog is being proposed. For example, there are catalogs specific for source code or specific for requirements scenarios.

Looking to the NFR catalogs proposed for UbiComp, Context-Aware, Self-Adaptive, Mobile areas, eight catalogs that contain one or more characteristics from AMICCaS were found out. Table 11 presents them. However, it was possible to see that there are few correlations and they are very specific for one kind of system: health domain.

Seven catalogs are related to Context-Awareness characteristics, six are related to Mobility and four are related to Invisibility. Most of the catalogs (5) do not present correlations. The ones that have correlations use to present them in a specific level: between strategies and subcharacteristics.

3.2.2 SM-RQ2 - How is information represented in the catalogs?

In the set of identified catalogs, eight types of representing the knowledge about NFRs were identified: SIG, matrices, i* notation, tables, hierarchical structures, SIG adaptations, list, template, and pattern. To better describe these representations, the types of catalogs that are most stored in each of them were analyzed. Figure 26 presents a bubble plot crossing data

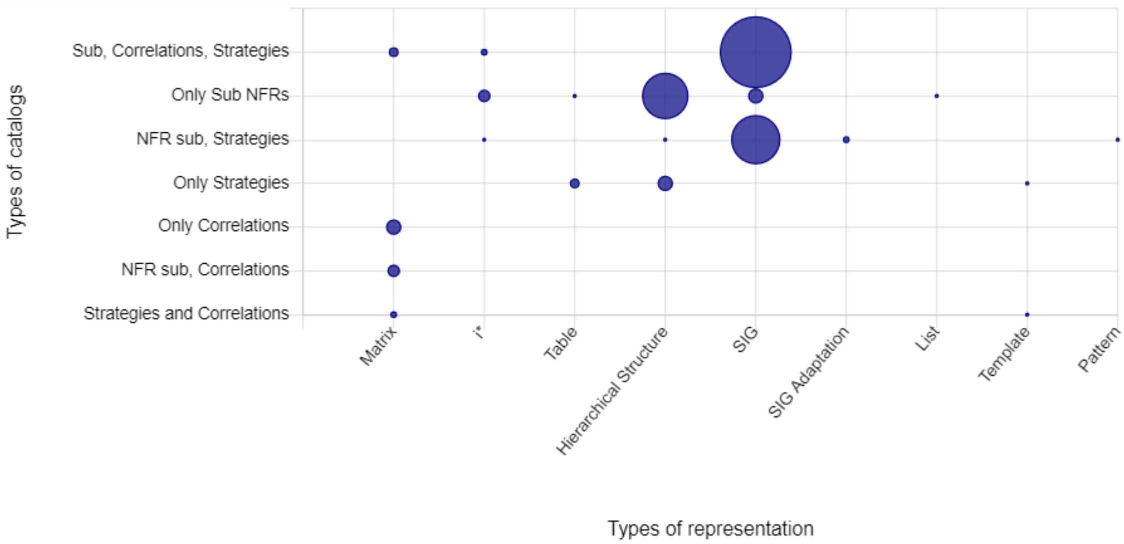
Table 11 – NFRs Catalogs related to AMICCaS identified by the SM Study

CAT ID	*Presence of AMICCaS*	*# Positive Correlation*	*# Negative Correlation*	*Level of Correlation*
18	Context-Awareness	-	-	-
22	Mobility, Context-Awareness	-	-	-
26	Invisibility, Context-Awareness, Mobility	-	-	-
27	Context-Awareness	-	-	-
28	Invisibility, Mobility, Context-Awareness	2	1	Between strategies and subcharacteristics / Between strategies
29	Invisibility, Mobility, Context-Awareness	1	5	Between strategies and subcharacteristics
30	Invisibility, Mobility, Context-Awareness	0	1	Between strategies and subcharacteristics / Between strategies
67	Mobility	-	-	-

Source: Author.

between the type of catalog and the type of representation. The size of a bubble is proportional to the number of catalogs that are in a pair of categories corresponding to the bubble coordinates. Therefore, it is possible to see that SIGs and hierarchical structures are the most used.

Figure 26 – Types of Catalogs vs Type of Representation

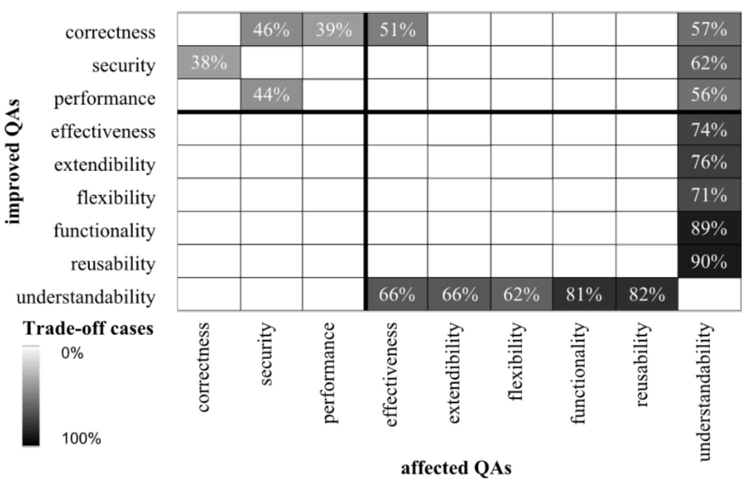


Source: Author.

The first representation is Matrix, which is a rectangular array organized in columns and rows to store information such as numbers and symbols. In case of NFRs catalogs, matrices are usually used to store correlations among pairs of NFRs. Figure 26 shows that regarding

"Matrix", there are only catalogs that contain correlations (T7 - Subcharacteristics and Correlations, T6 - Correlations, T5 - Strategies and Correlations, T3 - Subcharacteristics, Strategies and Correlations). Furthermore, Matrix is most used to store correlations directly between NFRs, *e.g.*, Usability hurts Security. Figure 27 presents an example of a matrix (FEITOSA *et al.*, 2015), where the correlations are correlations between NFRs (INTER-NFRs), each cell represents the effect of improving one NFR (vertical axis) over another (horizontal axis).

Figure 27 – Example of NFR Catalog represented as a Matrix



Source: (FEITOSA *et al.*, 2015)

Framework i* is a goal-oriented modeling language which consists of two main modeling components (YU, 1997). In this notation, it is possible to describe relationships among actors in an organizational context, stakeholder interests and concerns, and how several configurations of systems and environments can address them. Figure 26 shows that it is possible to represent all three knowledge (subcharacteristics, strategies, and correlations).

Hierarchical Structures are typically used to display a hierarchy of NFRs (ISO/IEC 25010, 2011). Figure 26 shows that this representation is more used in catalogs that present only subcharacteristics of NFRs. However, it also has been adopted to represent a hierarchy of strategies.

Softgoal Interdependency Graph (SIG) is a notation proposed by the NFR Framework to analyze and rationale about NFRs (CHUNG *et al.*, 2000). SIGs use NFRs catalogs to support the analysis and help developers. However, this notation has been largely used to not only analyze but also store information that can be reused in other opportunities. This notation was one of the most cited, unsurprisingly given the defined search string in this work. Furthermore, with this notation, it is possible to represent all kinds of knowledge: subcharacteristics, strategies and correlations.

There are also some Adaptations to SIGs, such as the Softgoal Interdependency Ruleset Graphs (SIRG) (BURGESS *et al.*, 2009), which includes an automated propagation of labels by introducing a new node type: Interdependency Rulesets (IR's). It creates the possibility of analysis without developer input. Another adaptation is from (CYSNEIROS; LEITE, 1999), which is the integration of NFRs into data models, aiming to be more helpful in identifying conflicts. Both adaptations were used to catalog subcharacteristics and strategies.

The other types of representation are List, similar to hierarchical structures but only used in one catalog to store subcharacteristics (FREITAS *et al.*, 2013), Template (BOEHM; IN, 1996) and Pattern (CARVALLO, 2015). The last two are a more formal structure to organize a piece of information.

3.2.3 SM-RQ3 - How the catalogs are defined?

As previously mentioned in Section 3, the Content Analysis (CA) methodology (BARDLN, 1977) was used to analyze the data to answer this question. Therefore, all extracted data of SM-RQ3 was used to find out how catalogs are defined. Thus, the coding activity was performed in the material exploration step. This activity means extracting and relating codes from raw data through inspection. Codes are conceptual names that represent the understanding of the researcher about a text. A set of codes can be grouped to form a category, which is a higher-level concept. Table 12 presents examples of text segments, their codes and category.

Each information that was being read was compared with other information along the set of data for similarities and differences. Every time a similar understanding was found, the existing code could be used. Then, codes with the same characteristics or purpose could be grouped in a category. This was the case for the codes presented in Table 12: Expert, Literature and Author are all sources for extracting knowledge to define a NFR catalog. Therefore, a category called "Source of Knowledge Extraction" was created.

At the end of CA steps, 173 texts segments were obtained and they were represented by 20 codes. These codes were analyzed and organized into 3 identified categories: *(i)* Sources of Knowledge Extraction; *(ii)* Extraction Techniques; and *(iii)* Analysis Techniques. These categories were created because during the analysis it was realized that there is a variation among where to collect knowledge, how to collect knowledge and how to analyze knowledge (*i.e.*,

Table 12 – Examples of Texts, Codes and Category for SM-RQ3

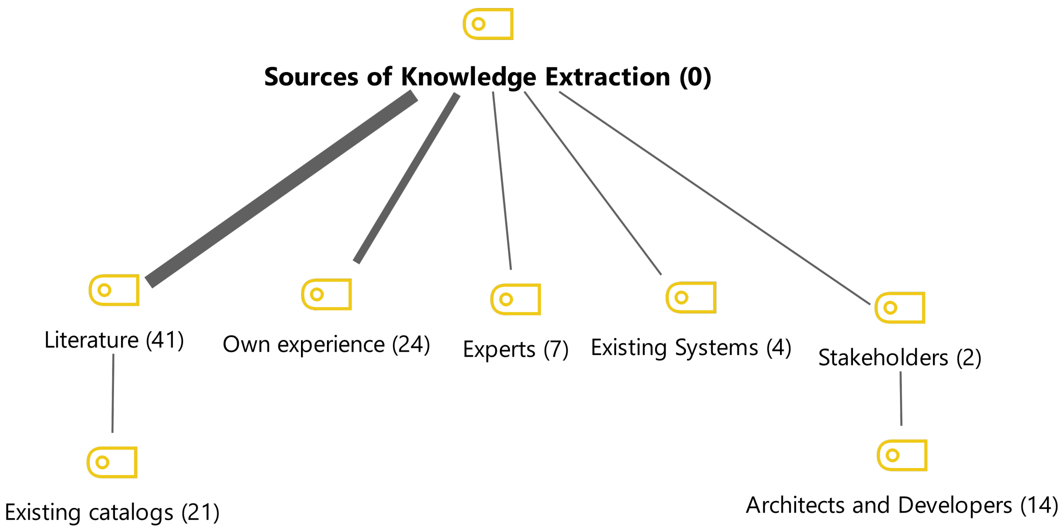
Text Segment	Code	Category
"The positive or negative impact of the alternative on fulfilling the related alternatives is assessed. The assessment is done based on expert knowledge from the design alternatives..." (SADI; YU, 2017)	Expert	Source of Knowledge Extraction
"Based on our bibliographical research, together with our Software Engineering experience" (SILVA et al., 2003)	Literature, Own Experience	
The content of this group of catalogues is extracted from a set of Software Engineering and Business literature discussing problems, concerns, and requirements in opening up software platforms." (SADI; YU, 2017)	Literature	

Source: Author.

subcharacteristics, strategies and correlations).

Figures 28, 29 and 30 present them organized in trees, where the categories are at the highest level and codes are below them. Also, each code presents a number that represents the frequency that this code appeared in data. For example, Figure 28 presents a code "Experts", where the number next to it is 7, meaning this code appeared seven times in data.

Figure 28 – "Sources of Knowledge" Category



Source: Author.

The first category illustrated in Figure 28 refers to the source of extraction, *i.e.*, from where the knowledge of the catalogs was extracted. Six sources were identified: stakeholders; architects and developers; experts; literature; existing catalogs and own experience of the authors.

The source with most citations is Literature, which means that knowledge may have come from books, papers, technical reports, among others. Below literature, there is the source "existing catalogs", these are also part of the literature, but are a particular class because they represent sources of literature with more organized content. Some studies adapt the knowledge of existing catalogs to structure a new catalog (SILVA *et al.*, 2003) (TOTIYA; SENIVONGSE, 2017) (SADI; YU, 2017).

The second source most cited is the own experience of the authors. Many studies are based on the authors' own knowledge and experience (SILVA *et al.*, 2003) (CHUNG *et al.*, 1995). Therefore they do not use an external source, but catalog knowledge based on their experience. Experts also appear as a source of extraction (GARCIA-MIRELES *et al.*, 2015), *i.e.*, experienced people that the authors of the studies consulted to extract knowledge.

The last source identified is the "Stakeholders", which relates to Architects and Developers. They appeared because many catalogs were built for specific systems and thus were made from the experience of the stakeholders who were working on the development of that system.

The relative frequency of each source regarding the type of NFR knowledge that the source was used to collect was analysed. Table 13 presents these frequencies regarding subcharacteristics (S), development strategies (DS) and correlations (C) and the catalogs where these sources were used.

Table 13 – Frequency of Extraction Sources

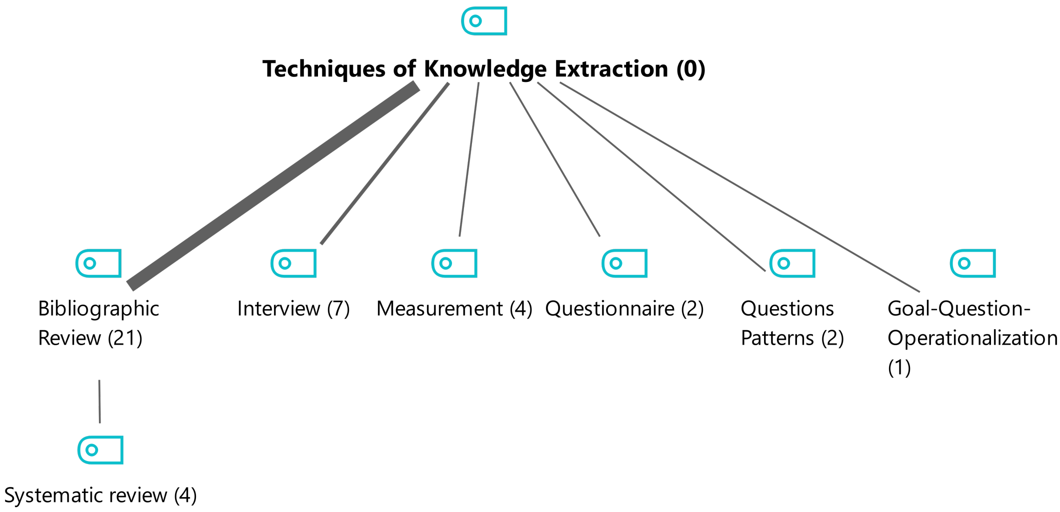
	NFR Knowledge			
Source	**S**	**DS**	**C**	**Catalogs**
Literature	56%	27%	17%	C2, C3, C5, C6, C9, C10, C11, C14, C15, C18, C21, C22, C25, C26, C27, C28, C29, C30, C31, C44, C58, C61, C63, C64, C86, C88, C89, C93, C101
Existing Catalogs	71%	14%	14%	C5, C6, C7, C8, C13, C15, C16, C19, C22, C43, C54, C61, C74, C97, C90, C79
Own Experience	33%	29%	38%	C14, C17, C19, C28, C29, C30, C42, C43, C44, C85, C86, C92, C101
Experts	43%	0%	57%	C4, C25, C74, C75, C76, C88, C89
Stakeholders	100%	0%	0%	C20, C21
Architects and Developers	43%	36%	21%	C12, C40, C41, C87, C88, C89
Existing Systems	0%	0%	100%	C72, C73, C98, C99

Source: Author.

In this way, it is possible to highlight that Literature is a source more used to extract subcharacteristics as well as Existing Catalogs; Own Experience is a common source for correlations; Experts are sources for only subcharacteristics and correlations; Stakeholders were cited only for subcharacteristics; and Architects and Developers are almost equally balanced among the three types of knowledge.

The second category (see Figure 29) refers to the technique of extraction, which means how the knowledge of the catalogs was extracted. Seven techniques were identified. The most used technique is Bibliographic Review, which can be done through a Systematic Review of the literature. It makes sense that this technique is the most cited because the source most used to extract is the literature itself. Interview is another found technique, which can be used together with two sources: stakeholders and experts. Questionnaires were also used to extract knowledge. Another one is Measurement (FEITOSA *et al.*, 2015) (ANDREOPOULOS, 2004), which is about using existing systems to find possible correlations between NFRs. Unlike other techniques, this is specific for extracting correlations.

Figure 29 – "Techniques of Extraction" Category



Source: Author.

Different techniques to extract knowledge are: Goal-Question-Operationalization (SERRANO; LEITE, 2011) and Question Patterns (LEAL *et al.*, 2015). Both help to obtain "strategy" knowledge because they refine subcharacteristics in questions that help to identify developing strategies. These techniques may support interviews or questionnaires for experts.

The relative frequency of each technique regarding the type of knowledge and the catalogs where these techniques were used are presented in Table VII.

It is possible to observe that **Bibliographic Review** and **Systematic Reviews** are

Table 14 – Frequency of Extraction Techniques

Source	NFR Knowledge S	DS	C	Catalogs
Bibliographic Review	62%	24%	14%	C15, C18, C22, C25, C28, C29, C30, C31, C44, C58, C63, C64, C93, C101
Systematic Review	50%	25%	25%	C14, C78
Interview	43%	14%	43%	C20, C21, C74, C88, C89, C91
Measurement	0%	0%	100%	C72, C73, C98, C99
Questionnaire	50%	0%	50%	C15
Question Patterns	0%	100%	0%	C13, C18
Goal-Question-Operationalization	0%	100%	0%	C79

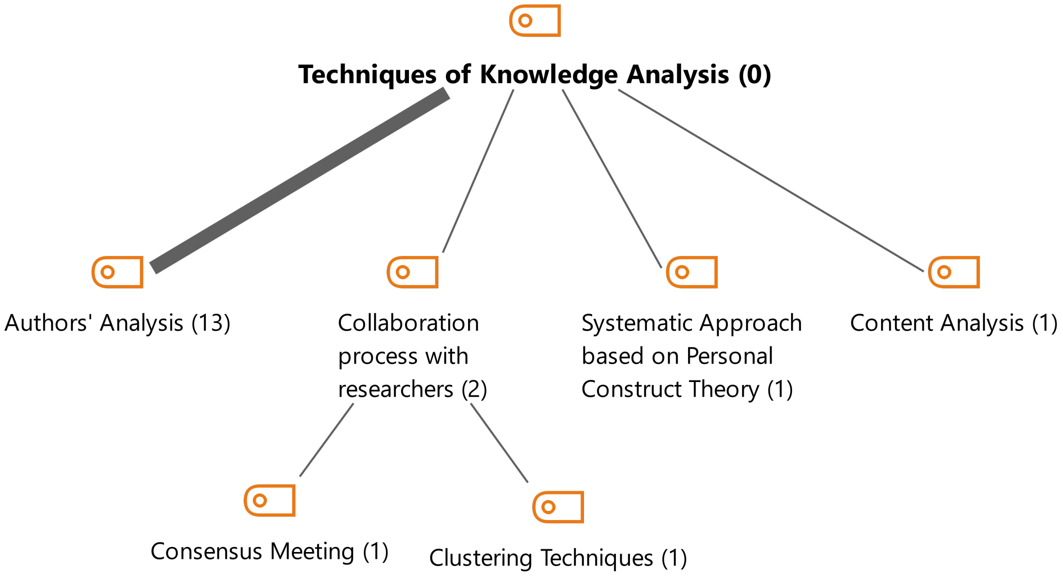
Source: Author.

techniques used to extract all kinds of knowledge: subcharacteristics, strategies and correlations; **Interviews** are used to extract all types of knowledge, being most cited for subcharacteristics and correlations; **Questionnaires** are cited for subcharacteristics and correlations, but it could be used for strategies as well; **Goal - Question - Operationalization** and **Question Patterns** are specific techniques for extracting strategies; and **Measurement** is specific for extracting correlations.

The third category (Techniques of Knowledge Analysis) refers to approaches to analyze data to arrive at an accurate result of knowledge (see Figure 30). In general, the studies present analyzes by the **authors** themselves, not specifying a technique. However, there are also approaches that involve authors in a **collaboration process with experts' researchers**, where they discuss data through **consensus meetings** and **grouping techniques** (LEITE; CAPPELLI, 2010). Also there is one catalog in which correlations were defined through **Content Analysis** (MAIRIZA; ZOWGHI, 2011). In another catalog, the correlations were defined through on an approach based on **Personal Construct Theory** (CAPPELLI *et al.*, 2010).

The relative frequency of each analysis technique regarding the type of knowledge is presented in Table 15 as well as the catalogs where these techniques were used. From this data, it is possible to conclude that many catalogs do not present a technique for the analysis of the extracted data, which is done by the authors themselves; **Consensus Meeting** and **Clustering Techniques** appeared as alternatives to analyze data for defining subcharacteristics; and **Content Analysis** and a proposed technique based on **Personal Construct Theory** are specifically used to identify correlations.

Figure 30 – "Techniques of Analysis" Category



Source: Author.

Table 15 – Frequency of Analysis Techniques

Source	NFR Knowledge S	DS	C	Catalogs
Author's Analysis	38%	23%	38%	C15, C17, C28, C29, C30, C42, C43, C44, C63, C64
Collaboration Process with researchers	50%	50%	0%	C15, C79
Consensus Meeting	100%	0%	0%	C15
Clustering Techniques	100%	0%	0%	C15
Content Analysis	0%	0%	100%	C31
Technique based on Personal Construct Theory	0%	0%	100%	C6

Source: Author.

3.2.4 SM-RQ4 - How the catalogs are evaluated?

The data extracted to answer this question provided a great variety of information to be analyzed, different ideas and concepts related to evaluation appeared in the texts. Thus, the same methodology used to analyze the SM-RQ3 data was used to analyze the SM-RQ4 data. In summary, 88 segments were codified with 18 codes. These codes were analyzed and organized into three categories (see Figures 31, 32 and 33). Table 16 presents examples of text segments, their codes and categories related to this question.

During the analysis, it was observed that some catalogs were not directly evaluated, but rather used to support a proposal. This situation occurred when the focus of the paper was on

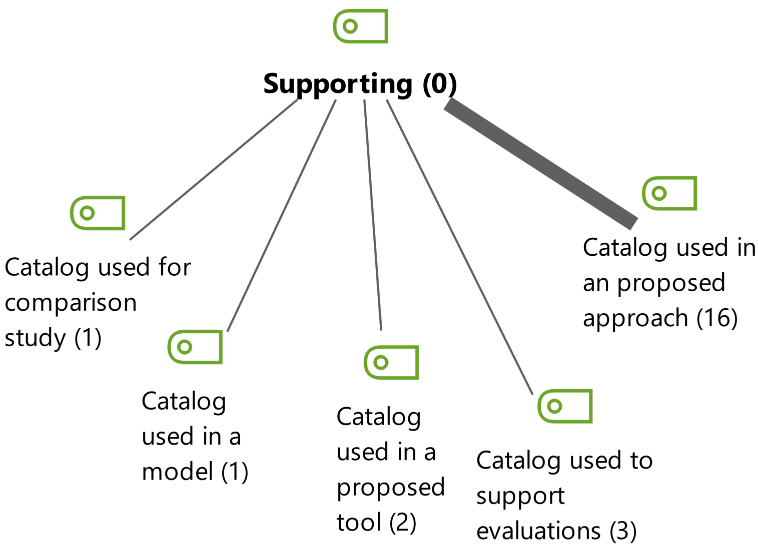
Table 16 – Examples of Texts, Codes and Category for Quest4

Text Segment	**Code**	**Category**
"we have carried out a survey with several stakeholders..." (SUBRAMANIAN *et al.*, 2014)	Survey	
"...continued refining the SIG, using questionnaire responses submitted to 16 international modeling experts" (LEITE; CAPPELLI, 2010)	Questionnaire	Evaluation Approaches
"In order to exemplify the use of Provenance SIG, this section describes an usage scenario where the scientific software SWf is modelled" (LEAL *et al.*, 2015)	Design	
"We use the proposed approach to revisit the high-level architectural design of data pro vision service in two real-world open software platform" (SADI; YU, 2017)	Redesign	Evaluation Purpose
"this model can calculate all NFRs contribution values by which developers could make tradeoff decisions among NFRs competing alternatives" (ZHU *et al.*, 2012)	Catalog used in a model	Supporting

Source: Author.

some solution proposal that uses catalogs instead of the catalog itself. In this way, a category "Supporting" was created to represent this concept (See Figure 31).

Figure 31 – "Supporting" Category



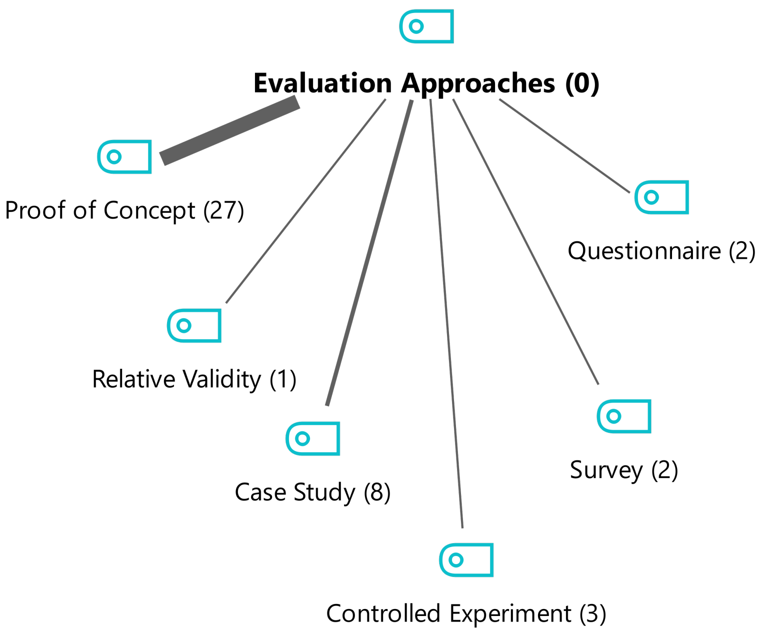
Source: Author.

Five codes were identified in this category: *(i)* Catalog used to support evaluations –

catalogs used as a base to create verification checklists or software measures (ISO/IEC 25010, 2011) (SOAD; BARBOSA, 2016); *(ii)* Catalog used in a model – catalog to support a model that can calculate all NFRs contribution values by which developers could make decisions (ZHU *et al.*, 2012); *(iii)* Catalog used for comparison study – catalog used to compare methodologies (SILVA *et al.*, 2003); *(iv)* Catalog used in a tool – catalogs used inside tools, for example, to help modeling properties of NFRs (UCHÔA *et al.*, 2017); and *(v)* Catalog used in proposed approaches – an example of approach is one from (EGYED; GRUNBACHER, 2004) which aims to identify more precise conflicts through requirements traceability. This last category was the one most cited, 16 texts segments were codified in it.

Besides, a category called "Evaluation Approaches" was created to represent all approaches found out in the studies (see Figure 32). Although, at the beginning, some of the well known approaches (such as case studies and experiments) were expected, this information was scattered in the papers and not explicitly defined, so it was essential to perform the content analysis. Figure 32 shows that Proof of Concept (PoC) is the approach that most appears.

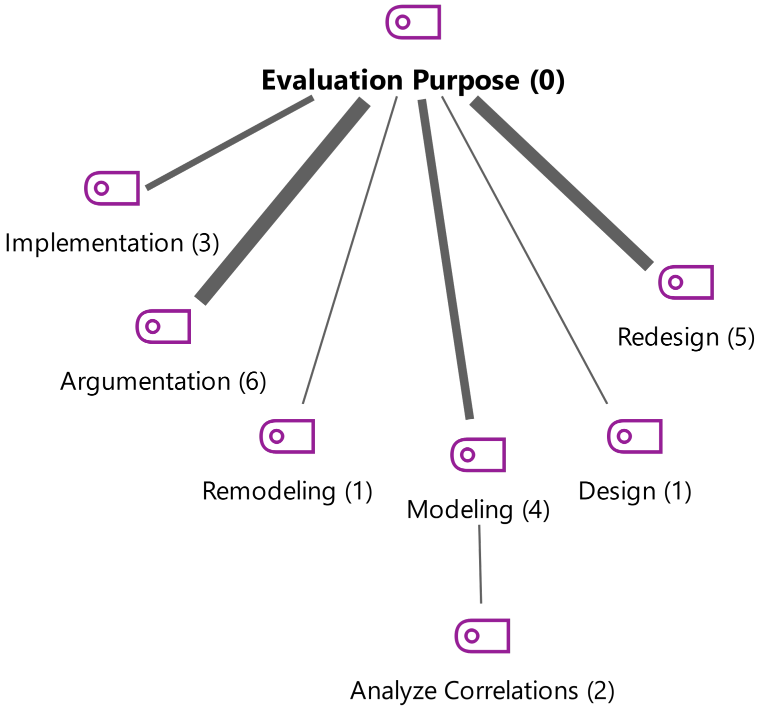
Figure 32 – "Evaluation Approaches" Category



Source: Author.

The third category created is called "Evaluation Purpose". This category was created because, besides the approach, there are still different purposes. Also, many papers did not clarify exactly which approach (Case Study, Experiment or other) they used but have explained about the purpose of the evaluation. As can be seen in Figure 33, some evaluations had the following purpose: *(i)* designing a specific system considering a NFR; *(ii)* redesigning an existing system to show improvement with the use of the catalog; *(iii)* building a new model, often reusing knowledge from the existing catalog; *(iv)* remodeling an existing catalog; *(v)* arguing about the effectiveness of the catalog; and *(vi)* Support a system's implementation.

Figure 33 – "Evaluation Purposes" Category



Source: Author.

3.3 Discussion

Interesting findings about NFRs catalogs were obtained during this study and they were presented through answers for each research question (SM-RQs). A synthesis of these findings is presented and discussed in Subsection 3.3.1. Then, research opportunities are presented in Subsection 3.3.2 and, finally, the threats to validity in Subsection 3.3.3.

3.3.1 Synthesis of the Results

This work consolidates the key findings into a single comprehensive view presented in Figure 34 and each one of these findings is discussed in this section.

The primary focus with this mapping study was first to collect as many catalogs as possible to understand them more deeply. Through SM-RQ1, 102 catalogs were obtained and then it was possible to better understand how they are characterized. One of the main findings is

Figure 34 – Key Findings of the SM Study

RQ1: What NFRs catalogs have been proposed in the literature?

- 102 NFRs Catalogs
- 7 - Types of NFRs Catalog Classification
- 3 - Types of Focus Classification
- 2 - Types of Correlations Classification
- 6 - Levels of Correlations Classification

RQ2: How information is represented in the NFR catalogs?

- 8 - Types of Representation

RQ3: How the NFRs catalogs are defined?

- 7 Sources of Knowledge
- 7 Techniques of Knowledge Extraction
- 6 Techniques of Knowledge Analysis

RQ4: How the NFRs catalogs are evaluated?

- 6 Evaluation Approaches
- 7 Evaluation Purposes
- 5 Supporting Purposes

Source: Author.

that catalogs cannot be classified as mutually exclusive in three types of catalogs, as previously proposed by (CHUNG *et al.*, 2000). In fact, a catalog can be in more than one classification. In this way, the initial classification of (CHUNG *et al.*, 2000) was extended to include seven types: T1 – Subcharacteristics, T2 - Subcharacteristics and strategies, T3 - Subcharacteristics, Strategies and Correlations, T4 – Strategies, T5 - Strategies and Correlations, T6 – Correlations and T7 - Subcharacteristics and correlations.

Another interesting finding is that initially it was expected that the catalogs could be for a specific area, but in fact, they can be proposed to particular areas, domains or artifacts. These different views are named as the "focus" of the catalog. Catalogs specific to artifacts of the system (*e.g.*, middleware) were found out, but also to a domain (*e.g.*, health) and to a area (*e.g.*, mobile). It was also possible to find combinations of these foci, such as catalogs for mobile applications focused on health. Or even catalogs for ubiquitous applications with a focus on middleware. The more specific the more the catalog can help developers in decision making.

As seen previously, many catalogs have correlations. This kind of knowledge is more complex than subcharacteristics and strategies and deserves a separate analysis. This SM study found out that correlations can occur not only between NFRs but also within the same NFR because their subcharacteristics or strategies may conflict with one another. In this way, a classification of correlation types was found out: INTER-NFRs and INTRA-NFR. This is interesting because it shows that even a single NFR cannot be wholly achieved in a system, thus demonstrating the high complexity that a NFR can present. Besides, this SM study found out that there are six levels of correlation. This level varies from the most generic, which are correlations directly between NFRs, to more specific levels, which are correlations between strategies and strategies and characteristics. Correlations with strategies are also more useful to developers because they can help more accurate decision making. It is difficult for the developer to understand or decide only with a more general level correlation, *e.g.*, Performance Hurts Security. Even because this relationship may be relative, in one type of system, this may be true, in another, it does not. Also, it depends on the strategy used.

The catalogs found are represented in eight different ways (SM-RQ2). Some representations are specific notations that the developer can use to analyze the satisfaction of NFRs, as is the case of SIGs, their adaptations and i *. Other representations are more informal, such as hierarchical structures, matrices, tables, and lists. Some are not notations, but they better organize knowledge, such as patterns and templates. SIG is the representation that was knew before starting this work, which is why it was part of the search string and was obviously the representation with more catalogs. Despite this, other representations have been found, and the interesting fact is to see that some stand out for a specific type of knowledge. For example, hierarchical structures are widely used to store a hierarchy of NFRs and their subcharacteristics. Matrices are widely used to represent correlations. Understanding these representations is important because a researcher or practitioner who wants to catalog knowledge about NFRs can use one of these representations.

Additionally, a mapping of approaches to define a catalog was investigated(SM-RQ3). Although there are catalogs defined by the authors themselves based on their experience, this study realized that the definition of a catalog can be done in two steps. First, it is necessary to collect the information, and then it is necessary to analyze this information to arrive at a more organized knowledge. Concerning the collection, there are six external sources by which the catalog creator can search for information: literature, existing catalogs, existing systems, experts, stakeholders and architects/developers. Also, there are seven techniques to extract the information from these sources: bibliographic review, systematic review, interview, questionnaire, questions patterns, goal-question-operationalization and measurement.

Table 17 presents a mapping among these sources, techniques and the type of knowledge that can be extracted from them. For sources like “literature” and “existing catalogs”, the technique is the “bibliographic review”, which can be done through a “systematic review”. Although this study did not identified it, this literature review could also be done through systematic mapping or snowballing. For sources “experts” and “stakeholders”, “questionnaire” and “interview” can be used to extract all kinds of knowledge (subcharacteristics, strategies and correlations). Additionally, the techniques “question patterns” and “goal-question-operationalization” can be used to extract strategies. From the source “existing systems”, “measurements” can be performed to find out correlations.

Table 17 – Mapping among Source, Technique and Kind of Knowledge

Source of Knowledge	***Technique of Knowledge Extraction***	***Kind of Knowledge***
Literature, Existing Catalogs	Bibliographic Review, Systematic Review	Subcharacteristics, Strategies, Correlations
Experts, Stakeholders	Questionnaire, Interview	Subcharacteristics, Strategies, Correlations
	Question Patterns Goal-Question-Operationalization	Strategies
Existing Systems	Measurement	Correlations

Source: Author.

In addition to the sources and extraction techniques, few researchers use techniques to better understand the data that has been extracted and thus define the knowledge more reliably. The techniques found were: Collaboration Process with researchers, Consensus Meeting, Clustering Techniques, Content Analysis and a Technique based on Personal Construct Theory. Collaboration Process with researchers together with consensus Meeting and clustering techniques were used to define subcharacteristics. Content analysis was used to analyze correlations in the level “between characteristics”, which means correlations between NFRs *e.g.*, Security hurts Usability. The technique based on Personal Construct Theory was used to define correlations in the level “between subcharacteristics”, but using strategies to analyze conflicts and cooperations among NFRs.

Thus, different studies use different combinations of approaches to arrive at a proposed catalog. Some papers cite that they used well-known research strategies, others use their own knowledge and experience. However, a systematic and reusable process that organizes a step by step with inputs, outputs, and approaches on how to create a complete NFR catalog, including steps for refinement of NFRs and strategies, especially for innovative NFRs, was not found in this systematic mapping.

The last findings were regarding the evaluation of a catalog. Six evaluation approaches were found out: proof of concept, case study, survey, questionnaire, relative validity and controlled experiment. Seven evaluation purposes: designing a specific system considering a NFR; redesigning an existing system to show improvement with the use of the catalog; building a new model, often reusing knowledge from the existing catalog; remodeling an existing catalog; arguing about the effectiveness of the catalog; and support a system's implementation. Also, many catalogs are not evaluate themselves but used to support another proposal. Five supporting purposes were found out: to support evaluations, to be used in a model, to be used for comparison study, to be used in a tool, to be used in proposed approaches.

3.3.2 Research Opportunities

For each research question, it is possible to identify a research opportunity. Figure 35 presents an overview of these four opportunities.

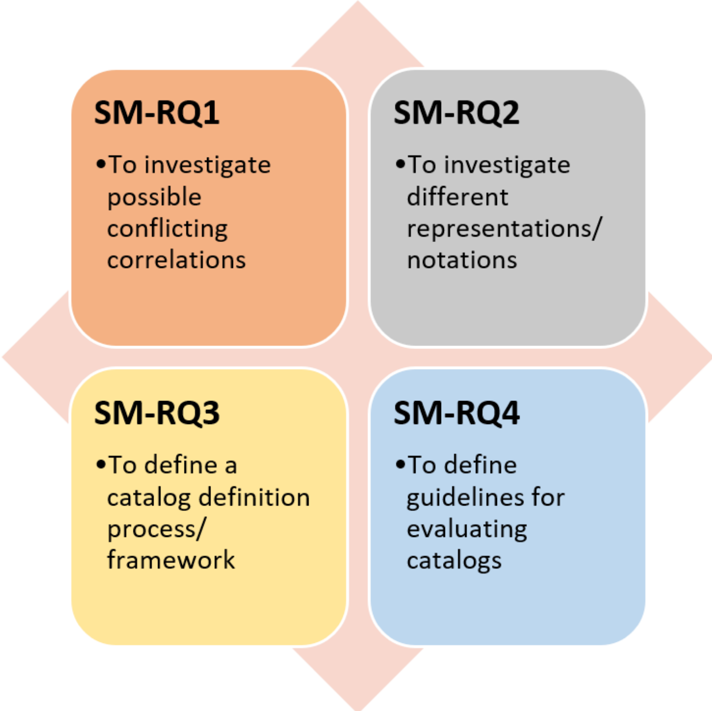
The first opportunity is related to the investigation of possible conflicting (positive or negative) correlations. 473 positive correlations and 395 negative correlations were found. However, they have not been analyzed so that there may be opposing correlations of different catalogs. This research is interesting to show that correlations between NFRs are relative, not always it is true that a pair of NFRs will be in conflict or harmony. Then, a general catalog of these catalogs can be built to support developers.

The second opportunity is related to the catalog's representation. Future research could investigate these representations in a way that could indicate which would be the most appropriate to deal with each knowledge of a catalog of NFRs: subcharacteristics, strategies and correlations. Or even if there is one that prevails in all these senses.

The third opportunity is related to the definition of a complete catalog. Although some papers explain their ways of constructing the proposed catalogs, a generic framework or process that provides a detailed guideline to create NFRs catalog was not found.

The fourth and last opportunity is related to the evaluation of a proposed catalog.

Figure 35 – Research Opportunities



Source: Author.

Few catalogs presented detailed evaluation procedures. Thus, there is a need to create a guideline to guide NFRs catalog evaluations.

This thesis deals with the third research opportunity, which is the definition of NFR catalog. The idea is to group a source of knowledge; and techniques of extraction and analysis into a single comprehensive process to help researchers in defining NFRs catalogs.

3.3.3 Threats to Validity

Following the suggestion of (PETERSEN *et al.*, 2015), which is also a systematic study, the following types of validity were considered: descriptive validity, theoretical validity, generalizability, interpretive validity.

Descriptive validity means the extent to which observations are described accurately. This kind of threat usually has more risk in qualitative studies. In the case of this study, there are two questions by which the extracted data was expressed as textual paragraphs. Therefore, to reduce this threat, a qualitative methodology was used to avoid an informal analysis and bias during the analysis. Hence, this threat is considered as being under control.

Theoretical validity is determined by the ability to being able to capture what the researchers intend to capture. Therefore, study identification, study selection and data extraction are two important threats to be considered. In this study, two databases (Scopus and Web of Science) were initially selected, but to reduce the threat regarding study identification, a snowballing for the selected papers from these databases were performed. Also, a requirement engineering workshop (WER) database was added, because it represents an important event in the requirements area where researchers usually publish their NFRs catalogs, but publications are not all indexed in the databases. However, it is worth to say that only a small number (5) of new studies was obtained from WER, indicating that the overall conclusions of this mapping would not change.

Regarding the study selection, valid papers could have been rejected. To avoid this threat, during the studies selection from Web of Science and Scopus, the review was performed by peers, the author reviewed the selection every time a paper was rejected by the undergraduate student. Therefore, a paper who was rejected could be considered again after the review. Also, whenever an excluded paper returned to the set of selected papers, an explanation was given to the student who was making the selection to achieve a common understanding of the studies.

The suggestions from the literature say that it is also appropriate to have one researcher extracting data and another reviewing the extraction (PETERSEN *et al.*, 2015). Therefore, extractions were also performed by peers in studies from Web of Science and Scopus.

Generalizability is determined by the degree that researchers can generalize results and it can be classified between external and internal. A possible threat is that data interpretation could be different for different researchers. To mitigate this threat, most of the extraction was performed in peers. Thus, this work considers that internal generalizability is not a major threat. Regarding external generalizability, this threat was minimized by performing this work with researchers from three organizations and two countries, increasing the possibilities of generalization. However, this study cannot generalize completely so this risk is accepted.

Interpretive validity is concerned about if the conclusions were based on the data, whether objective or subjective. In this work, most of the extractions were reviewed by the author. Also, the data analysis, both objective and subjective, following a methodology, not being conducted informally. The quantitative analysis followed Wholin's suggestions for presenting the data in graphs. Qualitative analysis was performed through Content Analysis.

Repeatability requires detailed reporting of the research process. To achieve this kind of validity, guidelines from literature were followed to perform several steps of this research. For performing the SM study in general, instructions proposed by (PETERSEN *et al.*, 2015) were used, and also this work considered other SM processes, such as (SANTOS *et al.*, 2017), (MONTAGUD *et al.*, 2012) (NETO *et al.*, 2011) (CARVALHO *et al.*, 2017). Besides, guidelines for performing snowballing proposed by (JALALI; WOHLIN, 2012), (BADAMPUDI *et al.*, 2015) were followed. Finally, the whole methodology from (HSIEH; SHANNON, 2005) to conduct content analysis was followed as well.

3.4 Chapter Summary

This Chapter has presented an exploratory study about NFRs' catalogs through a systematic mapping[2]. The purpose of this study is to investigate the existing catalogs, specially if there are existing catalogs dealing with AMICCaS, and also understand how they are defined and evaluated before starting the construction of a catalog for UbiComp and IoT systems.

102 NFRs catalogs were found out in this study. Most of them present the three kinds of knowledge: subcharacteristics, strategies and correlations. There are catalogs with a general focus and others with a specific focus. Regarding the area of this work, Internet of Things and UbiComp, few catalogs present characteristics related to AMICCaS (8), and only three of them present few correlations, which is still very scarce.

Furthermore, several techniques that could potentially help identify correlations in UbiComp and IoT systems were found. However, it was possible to see that there is not a systematic process that groups techniques to help researchers and developers to define correlations by looking to the development strategies, especially for quality characteristics that are new and no taxonomy is available, which is the case of AMICCaS.

This lack of such approach makes the definition of catalogs arduous. Thus, part of this thesis work is first dedicated to define a process capable to define NFRs catalog composed of subcharacteristics, development strategies and correlations (answering RQ2).

[2] All materials and results from this SM study are available at https://github.com/great-ufc/SM-NFRsCatalogs

4 CORRELATE PROCESS

This Chapter introduces a process to help researchers and developers to establish correlations between quality characteristics. This process is called CORRELATE - Process to Capture, Analysis and CatalOg CoRRElations between QuaLity ChAracTEristics and can be used by both researchers and developers[1]. Therefore, CORRELATE answers RQ2 - *How can an NFR catalog for HCI quality characteristics in UbiComp and IoT systems be defined?*

Additionally, this Chapter presents the proposed instruments and approaches that support the execution of the CORRELATE process.

So, in short, Section 4.1 presents an overview of the process. Sections 4.2, 4.3, 4.4 and 4.5 provides a more detailed explanation about each step and supporting instruments and approaches. Finally, Section 4.6 concludes this chapter.

4.1 Overview of the CORRELATE Process

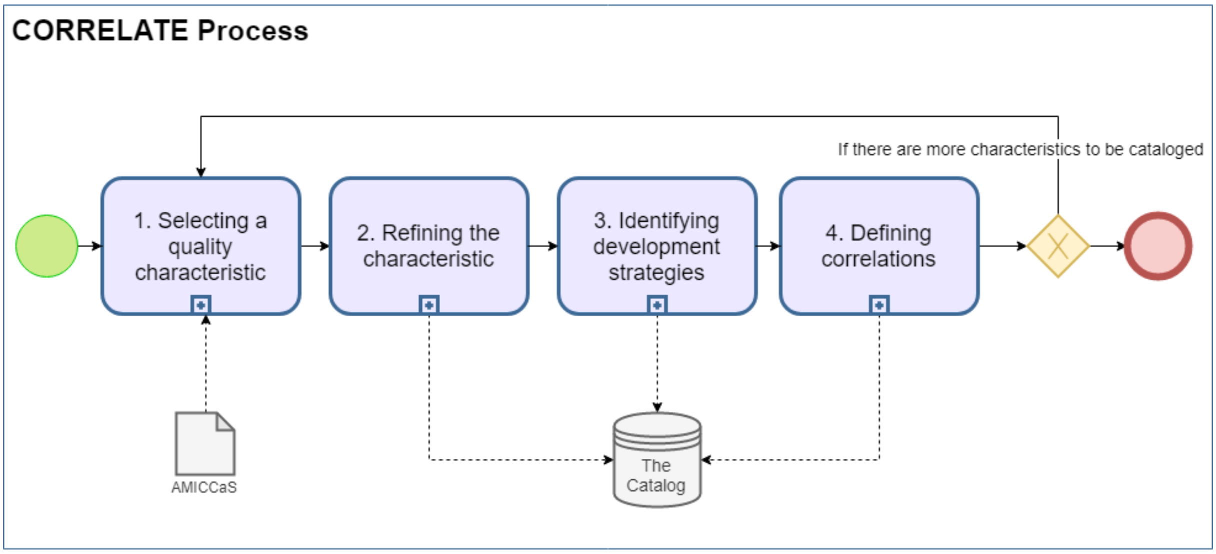
Chapter 3 showed a need for a systematic and reusable process on how to create a catalog that contains specific correlations that take into account development strategies of an NFR regarding another NFRs. This thesis assumes that specific correlations can better support developers when they need to make decisions. Therefore, a process should have steps related to the refinement of an NFR until reaching the software development strategies. Only after these steps, correlations can be defined at a specific level of the strategies.

Figure 36 presents an overview of the proposed process, which is called CORRELATE - Process to Capture, Analysis and CatalOg CoRRElations between QuaLity ChAracTEristics. This process is composed of four general steps that are briefly explained as follows and, in the next subsections, they are described in details.

The first step is to select the quality characteristic to start the definition of the catalog. Existing approaches usually analyze pairs of NFRs (CYSNEIROS; LEITE, 2004). For example, in this work, there is a list of specific HCI quality characteristics (*e.g.*, AMICCaS) that can be analyzed against a list of user interaction quality characteristics from (ISO/IEC 25010, 2011). Then, in this case, since several possibilities of pairs are possible, this process recommends choosing what characteristic the process user will consider first. This choice can be made based on the interest of the process user or the process user can perform a prioritization.

[1] For the sake of clarity throughout this document; this thesis refers to “process user” for both researchers and developers.

Figure 36 – Overview of the Proposed Process to Define a Correlation Catalog



Source: Author.

Then, the second step is to refine the selected quality characteristic into subcharacteristics. As briefly explained at the beginning of this section, specific correlations require a refinement of the NFR being investigated. This refinement can be performed by reusing available taxonomies in literature, from International Standards such as (ISO/IEC 25010, 2011), for example. If this is not the case, this thesis work proposes an approach based on the Grounded Theory method (GORBIN; STRAUSS, 2008), from which the process users can perform the refinement.

After refining the characteristic, development strategies should be identified to support the implementation of the subcharacteristics. This identification corresponds to the third step and interviews can be performed with experts or developers in the area (*e.g.*, Ubicomp and IoT, in this case) or even by reviewing the literature in the area.

Then, after identifying these strategies, the process user can analyze the impacts that each strategy can have in the other quality characteristic, which is fourth step.

Although the main goal of this process is to catalog correlations, which is done in Step 4, other types of knowledge are also generated and cataloged by this process. Steps 2 and 3 create subcharacteristics and development strategies. These data are useful to help the elicitation of system requirements and the high-level design.

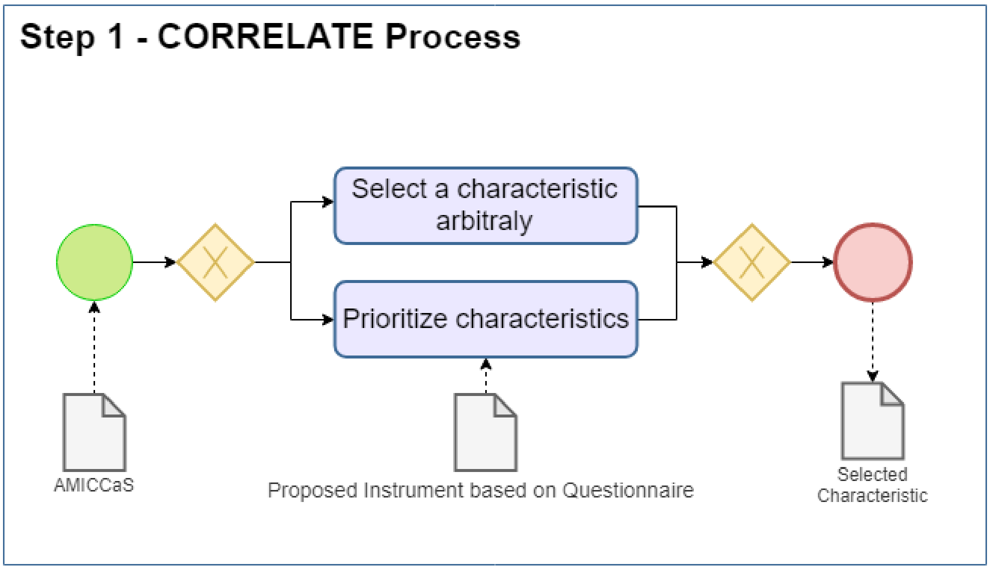
Next subsection explains better each one of these steps, its inputs/outputs, and supporting instruments and approaches.

4.2 Step 1: Selecting a quality characteristic

In this step, the process user should select the characteristic that will be cataloged. In the scope of this thesis, AMICCaS is the input of this step. Therefore, one of the characteristics of AMICCaS must be chosen to start the process of catalog definition. Then, in its turn, each one can be selected later.

There are two possibilities to make a choice, as presented in Figure 37: *(i)* a prioritization based on some criteria, or *(ii)* the process user can select the characteristic arbitrarily that he/she wants to catalog. The prioritization can be made by applying techniques such as Literature Review or Questionnaires and Interviews with Experts to make a choice based on some criteria, for example, an NFR with less information in literature, or more critical in the experts' opinion.

Figure 37 – Step 1 Activities of CORRELATE Process



Source: Author.

This thesis work aims to investigate specific characteristics of UbiComp and IoT applications (Mobility, Context-Awareness, Invisibility, Attention, Calmness, and Synchronicity (AMICCaS)), mostly because they are not complete cataloged in literature. Therefore, it is necessary to select one of them to start the cataloging process.

Following the step where a prioritization can be made, the criterion of selection was defined as follows: the one that is more likely to have negative correlations with user interaction quality characteristics according to experts in the area. This criterion was defined because negative correlations are the most damaging correlations between NFRs. Therefore, they need to be discovered as soon as possible.

To make the prioritization simple and quick, only one characteristic from the set of user interaction quality characteristics (See Figure 9) was used in this selection: Usability.

Knowing that Usability represents a vital characteristic to achieve quality during user interaction, this characteristic was used in this thesis to prioritize a characteristic of AMICCaS.

This work proposes an instrument based on the questionnaire data collection technique to be used to choose a quality characteristic of AMICCaS by checking which one is more likely to have negative correlations with Usability. The Questionnaire technique was chosen because it is possible to collect large amounts of data in a short time, unlike interviews, for example (WOHLIN *et al.*, 2012).

In this instrument, experts can give their opinion. The idea is to put in a column all characteristics the process user is interested in investigating and putting Usability in a row followed by the scale from the NFR Framework: BREAK, HURT, UNKNOWN, HELP and MAKE (CHUNG *et al.*, 2000) (see Table 18).

Table 18 – Questionnaire-based Instrument to prioritize Quality Characteristics

	Likely Impact on Usability				
AMICCaS	Break	Hurt	Unknown	Help	Make
Invisibility: the ability to hide the system, so users may not be aware of it					
Context-awareness: it is the system's capability of perceiving contextual information and dynamically and proactively adapts its functionalities.					
Attention: it refers to verifying if the user's focus is on various mental and physical activities such as walking, driving or other real-world interactions rather than on technology.					
Calmness means "free from distraction, excitement or disturbance"					
Mobility: it refers to the system's capability to provide users with continuous access to information resources irrespective of their location within the system's boundaries					
Synchronicity: the ability to keep things in sync with other things, which prevents different information from being presented in different things that are part of the same system.					

Source: Author.

In this way, HCI experts should indicate the likely impact of each specific characteristic of UbiComp and IoT on Usability using the scale. Thus, the characteristic with more indications on “break”, “hurt” and “unknown” would be the selected one. The two first values (Break and Hurt) represent negative correlations. The unknown value represents no knowledge from the expert about the correlation. Therefore a negative one still can exist.

Although this questionnaire is built specifically for AMICCaS, the process users can adapt it for other contexts.

4.3 Step 2: Refining the characteristic

This step is concerned to narrow down more specific concepts for the characteristic selected in the previous step. Figure 38 summarizes how to perform this step. The characteristic chosen in the last step is an input to this step. Furthermore, it is necessary to search for information about that selected characteristic, knowledge such as existing taxonomies or models. If there is an existing well-defined taxonomy from literature or standards from industry such as ISO standards (ISO/IEC 25010, 2011), then the process user can refine the characteristic by using this existing body of knowledge. Another option is refine the characteristic using other sources of data regarding that characteristic, especially in the case there is no taxonomy of subcharacteristics and the knowledge of that characteristic is fragmented or spread out among several sources. Additionally, the process user can use an existing body of knowledge together with other sources, then following the two paths in the sub-process.

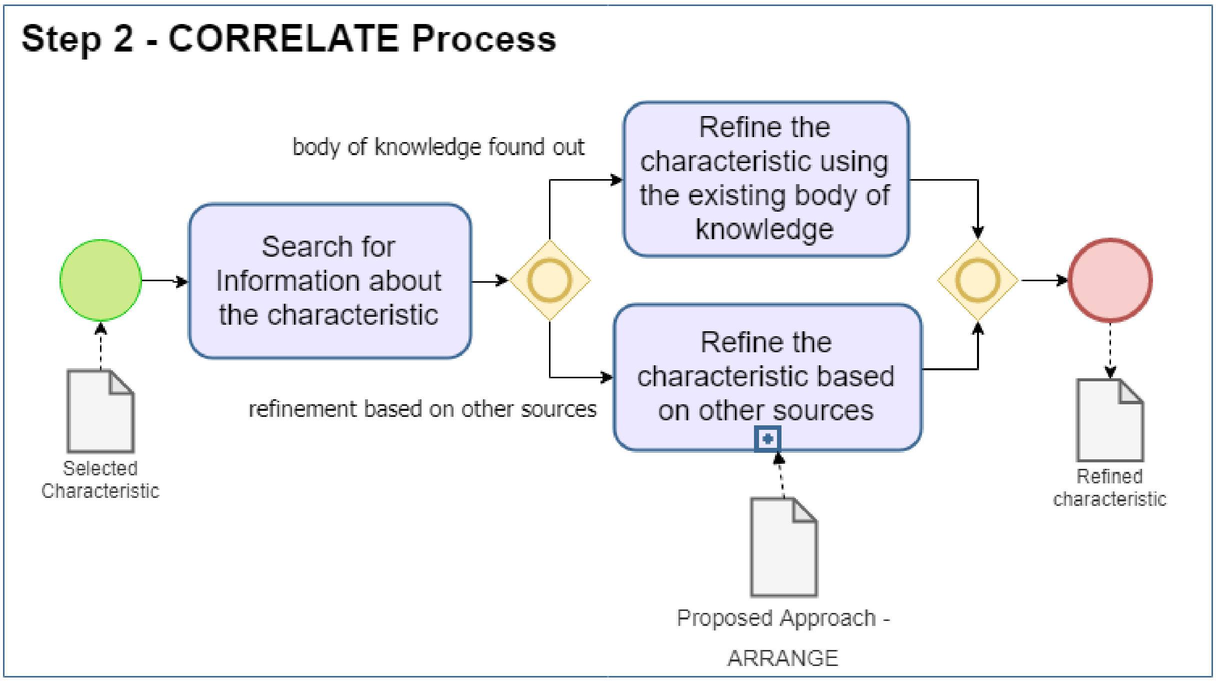
This thesis presents an approach called ARRANGE to support the activity in which the process user should refine the characteristic based on other sources. Before explaining it, this section presents the rationale for the activities in this step and the proposed approach.

4.3.1 Rationale

The evidence found out in the SM study presented in Chapter 3 supports the reasons behind each decision in this process. Regarding refinement of one characteristic into one or more subcharacteristics, the results of the SM study showed that:

- It is essential to collect and analyze data when refining a quality characteristic;
- Although the literature is the most widely used source of information to obtain knowledge about subcharacteristics, the own experience of the researchers appear to be a frequent source of knowledge;

Figure 38 – Step 2 Activities of CORRELATE Process



Source: Author.

- A bibliographic review is the extraction technique most used to obtain knowledge of the literature, followed by a systematic review. Furthermore, researchers can also use Interview and Questionnaire; and
- Most of the found catalogs do not analyze data using some technique of qualitative analysis. Since the data in this kind of study is textual and must be related to an NFR, a qualitative technique could bring more reliable outcomes. Two catalogs found in the literature use a collaborative process with researchers, and one of these uses consensus meetings and grouping techniques, which shows the importance of analyzing the data more reliably.

Therefore, the process users could refine the characteristic by their own experience. However, it is hard to refine a new NFR, for example, Calmness and Invisibility, by themselves. Additionally, such systematic refinement process is not the main focus of the previous related studies. Thus, this motivates the author of this thesis to the definition of a reusable approach that can support the process users in refining an NFR, especially a new one. Such approach should include not only part of the information gathered but also the analysis of this information so that the process user can reach a set of interrelated subcharacteristics.

Then, this work proposes to use the literature as a source of knowledge since it is the most widely used source of information to obtain knowledge about subcharacteristics.

Additionally, this work proposes to use a qualitative data analysis technique to analyze data from the literature.

One example of a qualitative data analysis method is the Grounded Theory (GT), which is a method developed for building theory from data (GORBIN; STRAUSS, 2008). This method groups together concepts derived from data until reaching the core category. According to (GORBIN; STRAUSS, 2008), it is like putting together a series of interlinking blocks to build a pyramid of knowledge.

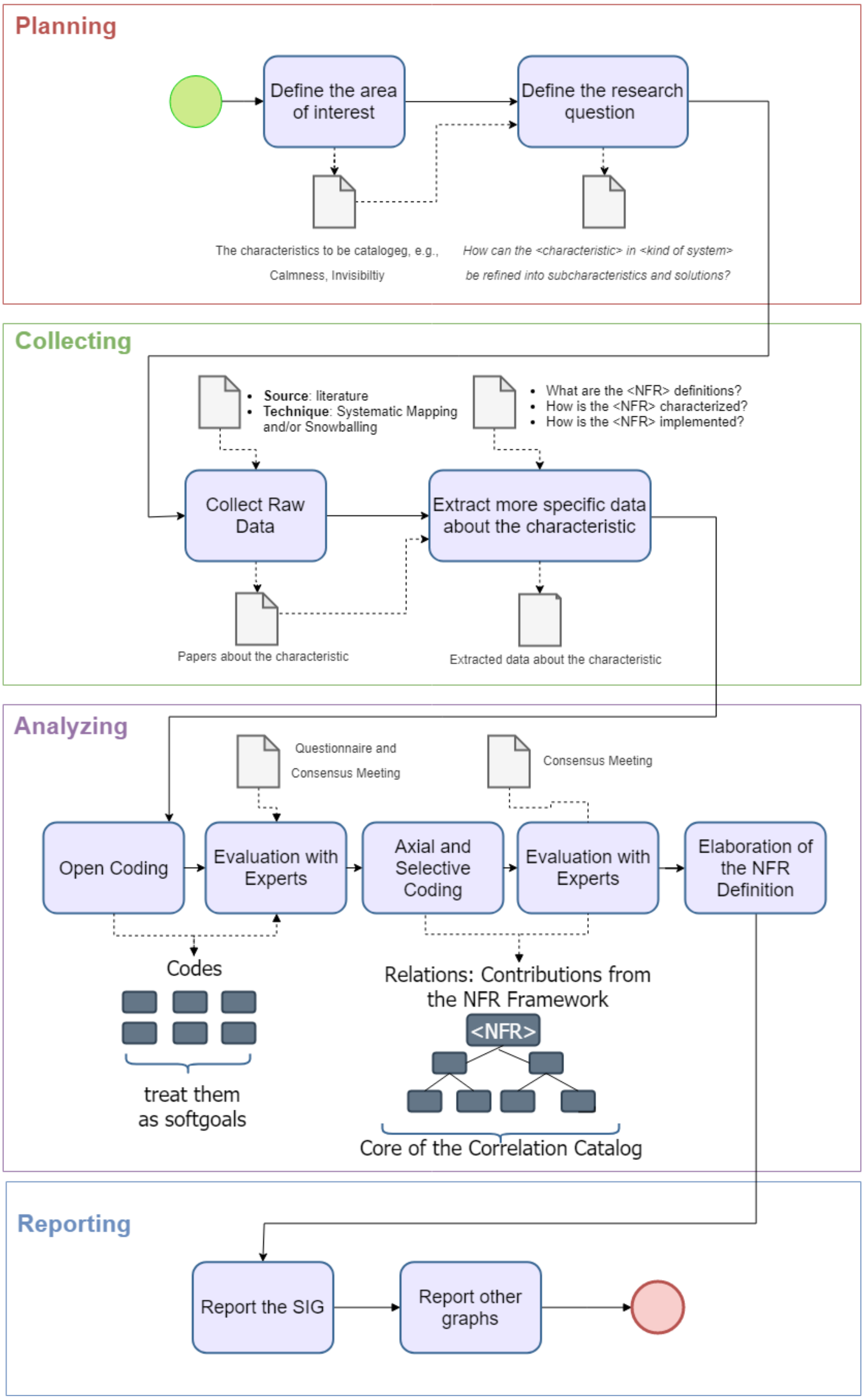
Researchers have already used GT in Software Engineering studies for many purposes: analyze data from a survey (VALE *et al.*, 2011), analyze data from an observational study (CONTE *et al.*, 2009), analyze data about success factors to software quality improvement (MONTONI, 2010) and analyze data from a systematic review (MOTTA, 2016).

Although GT has not been used to build an NFR catalog, its usage as an approach to extract and relate concepts from the literature data fits into the structure of an NFR catalog in the SIG format (NFR, subcharacteristics, development strategies), which is a pyramid of knowledge. Therefore, this thesis used this method in the context of refining an NFR into sub NFRs (*i.e.*, subcharacteristics) as a foundation for proposing an approach called Approach based on the Grounded Theory Method to Refine a Quality Characteristic (ARRANGE).

4.3.2 ARRANGE: Approach based on the Grounded Theory Method to Refine a Quality Characteristic

ARRANGE is composed of four phases: *(i)* planning; *(ii)* collecting; *(iii)* analyzing; and *(iv)* reporting results. This approach follows the findings of the SM study, which states that the refinement of a characteristic should be done through data collecting and analyzing, and uses the GT method, which is composed of the four steps: *(i)* planning; *(ii)* data collection; *(iii)* coding; and *(iv)* reporting results. Although all recommendations from the systematic literature review method and the grounded theory method should be followed in ARRANGE, each phase was instantiated specifically to build an NFR catalog, always with the concern to make this approach reusable for any NFR to be cataloged. Figure 39 gives an overview of ARRANGE, which steps are explained in the next subsections.

Figure 39 – ARRANGE Overview



Source: Author.

4.3.2.1 Planning and Collecting

The planning step aims to identify the research area of interest and the research question that will drive the work. In the context of this work, which is refining an NFR, the area of interest must be the NFR itself (*e.g.*, Invisibility, Calmness, Mobility). The process user should write the question as follows: *"How can the <NFR> in <kind of system> be defined and refined into subcharacteristics and solutions?"*.

The data collection step aims to collect the necessary data to answer the general research question. ARRANGE suggests the use of a systematic literature review or snowballing procedure (KITCHENHAM; CHARTERS, 2007) (PETERSEN *et al.*, 2015) to systematize this task. Snowballing refers to the use of a reference list of a paper (backward snowballing) or citations to that paper (forward snowballing) to identify additional papers.

Additionally, to build a data set of an NFR, it is necessary to extract information from the papers. Therefore, the following questions are proposed to guide the data extraction:

- *What are the <NFR> definitions?* This question aims to collect all definitions available on literature about the NFR that will be cataloged;
- *How is the <NFR> characterized?* This question aims to collect existing subcharacteristics; and
- *How is the <NFR> implemented?* This question aims to identify any kind of solution used to implement the interested NFR.

With all data extracted from the papers through these questions, it will be possible to execute the next activity of ARRANGE, which is *Analyzing*.

4.3.2.2 Analyzing

The analysis in ARRANGE is concerned to obtain a well-defined set of interrelated subcharacteristics from all data extracted in the previous step.

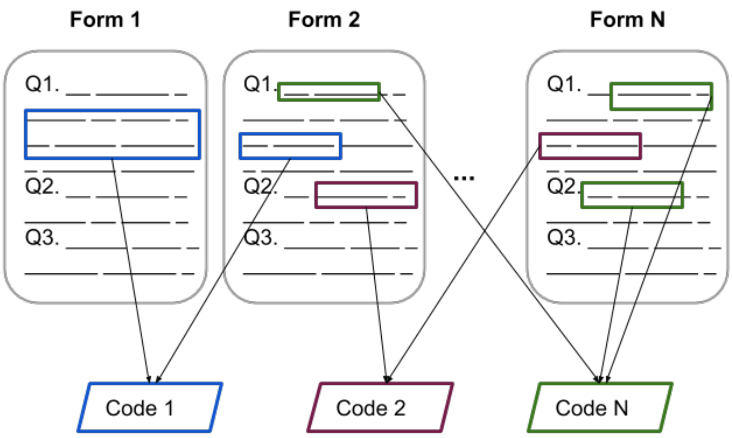
The GT method contains an activity that can be used for this purpose, which is the coding activity. Coding means extracting concepts from raw data and relating them to each other until a core concept is reached, which is the theory (GORBIN; STRAUSS, 2008).

In the context of this work, coding will be executed on the extracted data from the papers. Then concepts extracted from this activity will become subcharacteristics. Then, the process users can relate them to each other until reach the NFR being refined.

Some tools can be used to help this coding step. For example, the MAXQDA12 tool [2], used in the SM study, is a recommendation from (GORBIN; STRAUSS, 2008). Another example is the QDA Miner Lite tool [3], which is free. These tools can help in the entire coding process, which the GT method performs in three tasks: open, axial, and selective coding.

In Open Coding, the process user should inspect the textual data to understand the essence of what is being expressed (GORBIN; STRAUSS, 2008). In the context of this work, the process user should examine the data extracted from the papers, which are the answers to the extraction form (See Figure 40).

Figure 40 – Overview of Open Coding Activity



Source: Author.

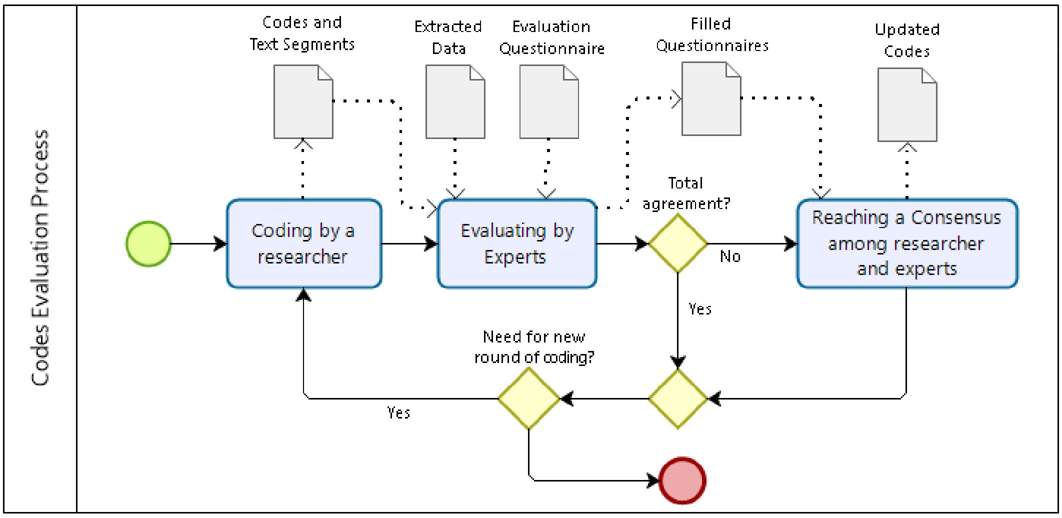
Therefore, following the fields of the extraction form, the coding process should start by reading all the texts related to the NFR definition and then for every data from the extraction form. Then, the process user should create a conceptual name (code) to represent his/her understanding of the data. Codes can represent a single word, a phrase, or a whole paragraph. The process user should compare each information that is being read with other information along with the set of data for similarities and differences. Every time he/she finds similar information, the existing code should be reused. The name of this strategy is "constant comparison", usually used for qualitative analysis (GORBIN; STRAUSS, 2008).

At the end of open coding, the process user will have created a set of codes that represents his/her understanding of the subcharacteristics of the NFR being investigated. To minimize possible misinterpretation and to help with specialized concepts related to the NFR, it is important to evaluate theses codifications. The idea of this work is that experts in the area of

2 https://www.maxqda.com/
3 http://provalisresearch.com/products/qualitative-data-analysis-software/

the NFR evaluate all the matchings between text segments and codes. Figure 41 presents the evaluation process.

Figure 41 – Evaluation Process of the Open Coding



Source: Author.

After coding by a process user (first step), the suggested codes and text segments should be sent to experts for evaluating them (second step). The evaluation should be performed through a questionnaire where every match between the text segment and the suggested code should be evaluated and classified as: *(i)* agree; *(ii)* partially agree; or *(iii)* disagree. Then, an analysis of the results of the evaluation should be done. If a total agreement was not obtained, a meeting should be performed for discussion and consensus (third step). Finally, if the process user noticed a need for more sessions of coding, the process can restart.

In case that there is no feasibility to evaluate every match to reach a consensus, the process user can evaluate part of the matches, and can use the Kappa method to assess the agreement (CARLETTA, 1996).

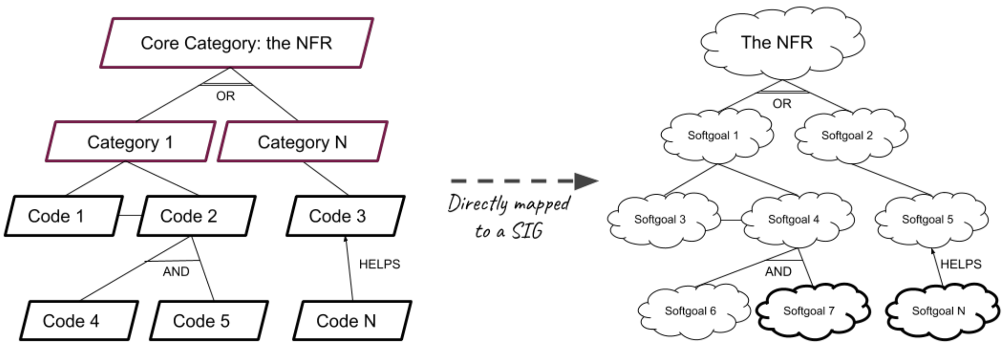
All the resulting codes will be directly represented as softgoals of the SIG, either as an NFR softgoal (*i.e.*, characteristics or subcharacteristics) or an operationalizing softgoal (*i.e.*, development strategy). Furthermore, they need to be related to each other, which is the next step of GT - the axial coding.

Axial Coding is the process of relating concepts or grouping them by creating categories (a high-level concept that represents a group of codes) (GORBIN; STRAUSS, 2008). These relations between concepts can be defined by the process user, although there are already existing relations that can be reused, such as “is a”, where a concept is a kind of another concept. This work proposes to create relations by analyzing implicit meaning between the codes and by using the types of contributions from the SIG notation (AND, OR, MAKE, HELP, BREAK and HURT). Furthermore, the process user can group categories into new categories, creating a chain of categories. That is why axial coding is like putting together a series of interlinking blocks to build a knowledge pyramid (GORBIN; STRAUSS, 2008), fitting in the structure of SIGs catalogs.

When all codes and categories can be related to a core category, it means the process user is doing Selective Coding. According to (GORBIN; STRAUSS, 2008), the process of linking concepts around a core category, refining and trimming the resulting theoretical construction is called Selective Coding. This work proposes that the core category should be the investigated NFR.

After the axial and selective coding, this work proposes that the concepts (*i.e.*, codes and categories) and its relations should be directly mapped to a Softgoal Interdependency Graph (See Figure 42). Each concept will either be an NFR softgoal or an operationalizing softgoal. Therefore, the process user should classify each code according to the definition of the NFR and operationalizing softgoals.

Figure 42 – Overview of Axial and Selective Coding



Source: Author.

The axial and selective coding can be performed together by the process user, the one who will lead the open coding. The process user should analyze the codes to group them into categories, to make the necessary relations and to relate them to the core category, always with the effort to describe how the concepts are grounded in data. Then, meetings and discussions can be held with the same experts who participated in the open coding to reach an agreement about the organization of the concepts. During these meetings, the experts can help not only with the organization but also by complementing the NFR SIG by indicating additional softgoals.

Finally, after the selective coding, the process user will be able to elaborate a definition for the NFR being cataloged. For that, the definitions of the subcharacteristics should be used to compose the description of the NFR.

4.3.2.3 Reporting

At the end of GT, it is possible to create a SIG for the NFR being investigated. The code structure generated by the Grounded Theory method will be exactly represented in a SIG. In fact, this knowledge becomes the core of the SIG in which, with the proper arrangement, it can be expanded to include more softgoals. Therefore, in this step, the SIG can be reported, as well as some statistics about the codes. Two measures are the groundedness of each concept, meaning how many text segments are related to it and the density of each concept, meaning how many cases (in this case, papers) support it.

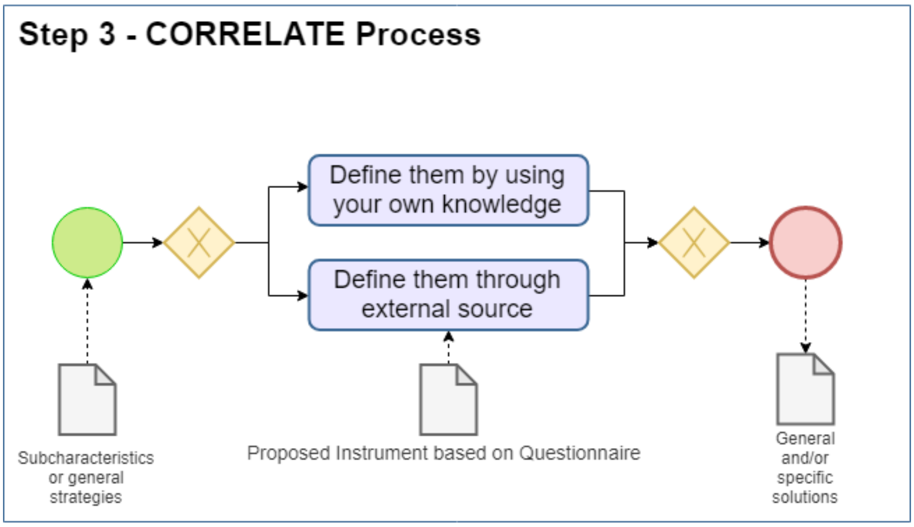
4.4 Step 3: Identifying development strategies

As seen in the previous section, it is possible to build a SIG through the Grounded Theory. The SIG resulting from the GT may already come with strategies that help the developer in the implementation of the quality characteristic since one of the issues of extraction is *"How is the <NFR> implemented?"*. However, these strategies may not yet be at a more specific level. As explained in Section 2.2, an operationalizing softgoal can not only be generic (broad), but it can also be refined into specific operationalizing softgoals (CHUNG *et al.*, 2000), helping even more software developers and designers. Then, the SIG generated from GT could be complemented with specific operationalizing softgoals. Figure 43 summarizes what can be done to perform this step.

Subcharacteristics or general solutions are inputs to this step. As seen in Chapter 3, the literature shows that development strategies can be captured through external sources such as literature, existing catalogs, and developers, or through the knowledge of the catalog creator himself. In this way, the creator can define on his own or use an external source for this definition.

If the process user chooses to identify strategies through an external source, CORRELATE proposes a questionnaire-based instrument to collect design and development strategies from UbiComp and IoT developers. It is expected that they accumulate the solutions most used in

Figure 43 – Step 3 Activities of CORRELATE Process



Source: Author.

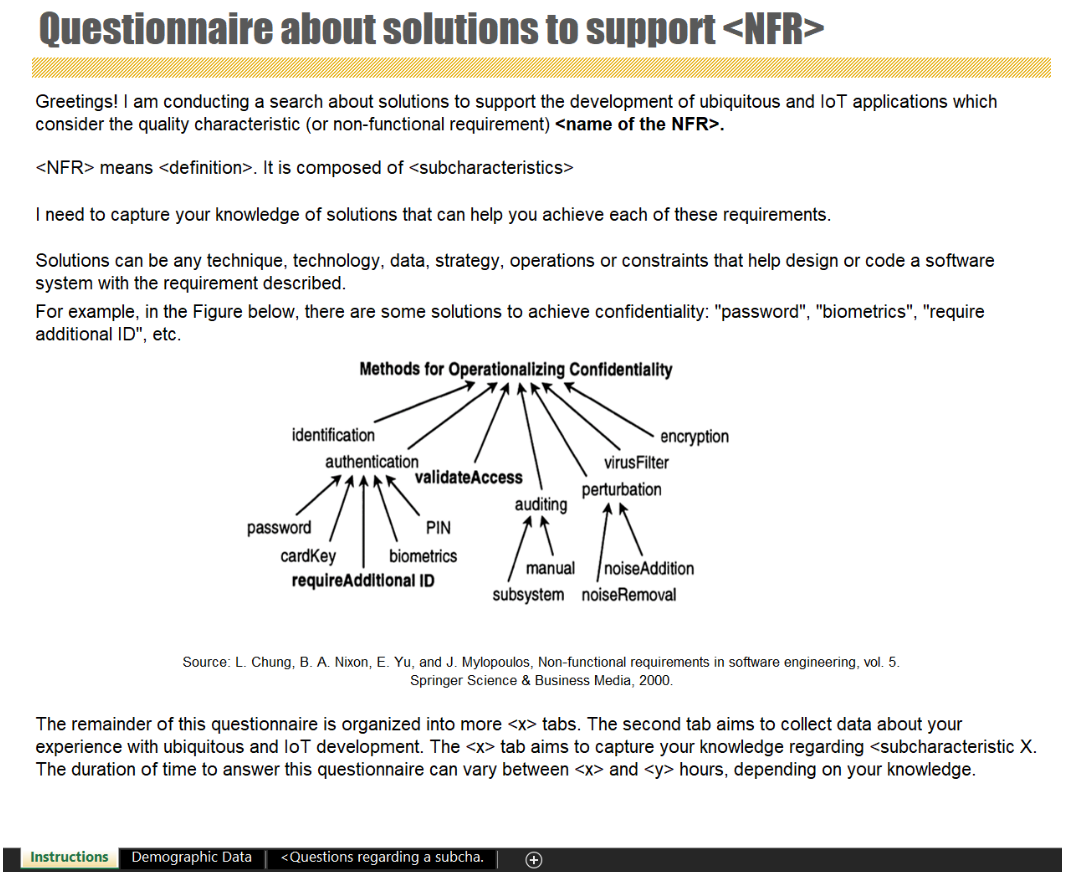
the industry. Furthermore, it is important to emphasize that collecting strategies from developers is just one strategy, other sources could be used, such as literature. Additionally, questionnaire was chosen because it is a technique in which it is possible to collect large amounts of data in a short time (WOHLIN *et al.*, 2012).

The proposed questionnaire is split into three parts. The first part is the presentation of the research (See Figure 44). In this part, the investigated NFR should be presented, its definition and subcharacteristics, as it was defined in the previous step of the process (Step 2). After that, it should be left clear for the developers that solutions can be any technique, technology, data, strategy, operations or constraints that help design or code a software system with the requirement described (CHUNG *et al.*, 2000). An example should be given to make it clear to developers how they can respond. Finally, the rest of the questionnaire should be explained.

The second part of the questionnaire is related to demographic data. It is essential to understand that the developers work with the domain studied. Thus, four questions should be asked for them: *(i)* Have you worked with applications for environments such as Ubiquitous Computing or Internet of Things? *(ii)* Describe what activities you have worked on for this type of application (example: programming, testing) *(iii)* Describe which domains you've previously worked with (*e.g.*, healthcare, smart building) and *(iv)* How long have you worked with this type of application? (*e.g.*, for 5 years).

The third part of the questionnaire is composed of questions about solutions (See Figure 45). For the questionnaire to not get too big and exhaustive or discouraging developers

Figure 44 – Questionnaire for the Developers – First Part



Questionnaire about solutions to support <NFR>

Greetings! I am conducting a search about solutions to support the development of ubiquitous and IoT applications which consider the quality characteristic (or non-functional requirement) **<name of the NFR>.**

<NFR> means <definition>. It is composed of <subcharacteristics>

I need to capture your knowledge of solutions that can help you achieve each of these requirements.

Solutions can be any technique, technology, data, strategy, operations or constraints that help design or code a software system with the requirement described.

For example, in the Figure below, there are some solutions to achieve confidentiality: "password", "biometrics", "require additional ID", etc.

Source: L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos, Non-functional requirements in software engineering, vol. 5. Springer Science & Business Media, 2000.

The remainder of this questionnaire is organized into more <x> tabs. The second tab aims to collect data about your experience with ubiquitous and IoT development. The <x> tab aims to capture your knowledge regarding <subcharacteristic X. The duration of time to answer this questionnaire can vary between <x> and <y> hours, depending on your knowledge.

Instructions | Demographic Data | <Questions regarding a subcha. | ⊕

Source: Author.

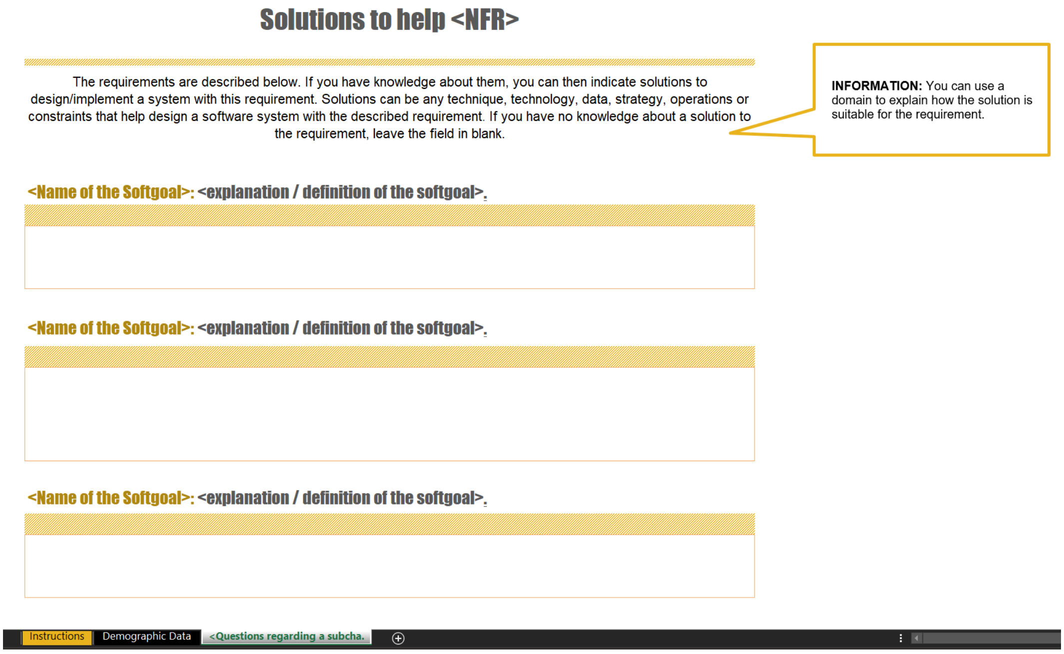
to respond, this process suggests that the questions for the developers should be designed to solutions for the softgoals collected at the lowest level of the SIG generated by the GT.

4.5 Step 4: Defining correlations

As mentioned in Section 2.2, the reason why two characteristics may present a correlation (positive or negative) is the use of a specific strategy. When solutions are used to improve a system concerning one characteristic, it can lead to harming another. In this way, the strategy used to support a specific quality characteristic for UbiComp and IoT systems should be analyzed.

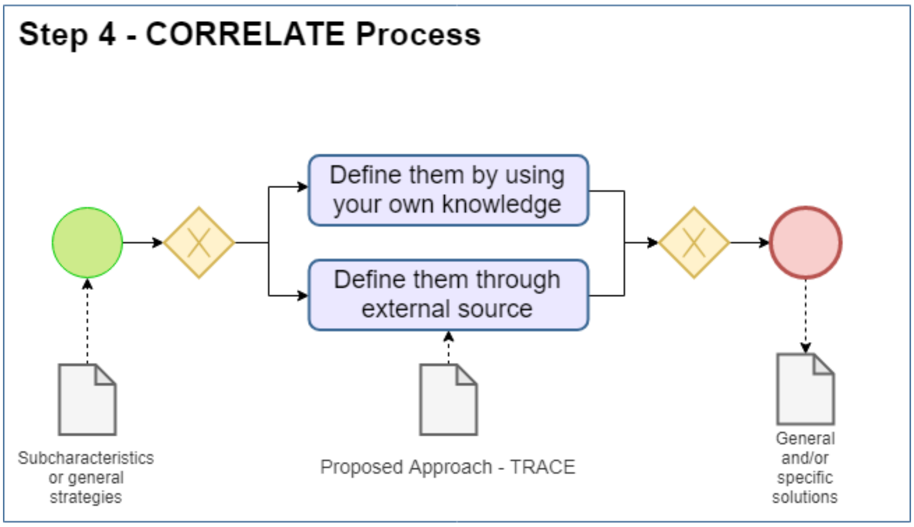
This step is then concerned to define such correlations and to store them in a catalog. Figure 46 presents an overview of this step. Softgoals from the lowest level of the SIG obtained in the previous step (*i.e.*, mostly specific or general solutions) are used as input to this step. Then, the process users can define correlations by their own experience or using an external source of knowledge.

Figure 45 – Questionnaire for the Developers – Third Part



Source: Author.

Figure 46 – Step 4 Activities of CORRELATE Process



Source: Author.

This work proposes an approach, which is called Approach to define CorrelaTions to Quality ChaRActeristics by using the Interview and the ContEnt Analysis Methods (TRACE). TRACE uses the Interview and the deductive Content Analysis methods to collect and analyze data in order to establish correlations. The next subsection explains the rationale to understand the activities in this step and the proposed approach.

4.5.1 Rationale

Regarding the definition of correlations, the results from the SM study show that:

- The own author of many catalogs defined correlations, therefore, he/she can use their own knowledge and rationale to define correlations;
- External sources were also used, such as literature and experts;
- In case of using an external source, it is essential to extract the information and then analyze the data to define the correlation;
- Concerning extraction, researchers can use systematic reviews, interviews, questionnaires, and measurements; and
- Regarding analysis, few catalogs use some technique. One of these catalogs uses a technique based on Personal Construct Theory, where an indirect elicitation process offers a systematic way of analyzing interactions, and it uses a filter to determine which characteristics will be analyzed. The other one uses Content Analysis, but it is used to define correlations in the level of characteristics, and it does not detail how the approach is used. The last ones use measurement to define correlations, then the researchers need several existing systems, and the correlation will be defined in the level of characteristics.

This work proposes an approach called TRACE. This approach is based on these existing studies regarding the source of knowledge and extraction technique, but it differs from them regarding the analysis since it considers that any development strategy may have correlations with any other NFR. Thus, none filter of possible correlations is made and no existing systems are necessary. Instead, the process user investigates each development strategy regarding if there is any impact on another NFR or a group of NFRs.

TRACE uses developers' opinion as a source of knowledge to define correlations. In the case of this thesis, the suggestion is asking developers of mobile, ubiquitous or IoT applications as a knowledge source of correlations since they have experience with development and can give specific information.

There are different ways of extracting information from people. This work used the interview approach. According to (OATES, 2005), the interview is considered a suitable data generation approach when a researcher wants to:

- obtain detailed information;
- ask questions that are complex, or open-ended, or whose order and logic might need to be different for different people;

- explore emotions, experiences or feelings that cannot easily be observed or described via pre-defined questionnaire responses; and
- investigate sensitive issues or privileged information that respondents might not be willing to write about on paper for a researcher that they have not met.

Therefore, considering that correlations of specific development strategies regarding quality characteristics may not be easily collected through closed questionnaires; that these strategies may have sensitive issues; and more information about the strategy can be obtained; this work decided to use the interview as an extraction approach.

TRACE proposes that the data collected through interviews be analyzed with the Deductive Content Analysis method, which is suitable to demonstrate a hypothesis.

All recommendations on how to plan, conduct, and report an interview and content analysis should be followed. Therefore, this work does not replace these techniques but bring them together to a specific purpose: defining correlations. Furthermore, this work highlights and better specify some activities concerning the definition of correlations. Therefore, each phase of TRACE is explained regarding these specific activities.

4.5.2 *TRACE: Approach to define CorrelaTions to Quality ChaRActeristics by using the Interview and the ContEnt Analysis Methods*

TRACE is composed of four phases: *(i)* planning; *(ii)* collecting; *(iii)* analyzing; *validating*; and *(iv) reporting*.

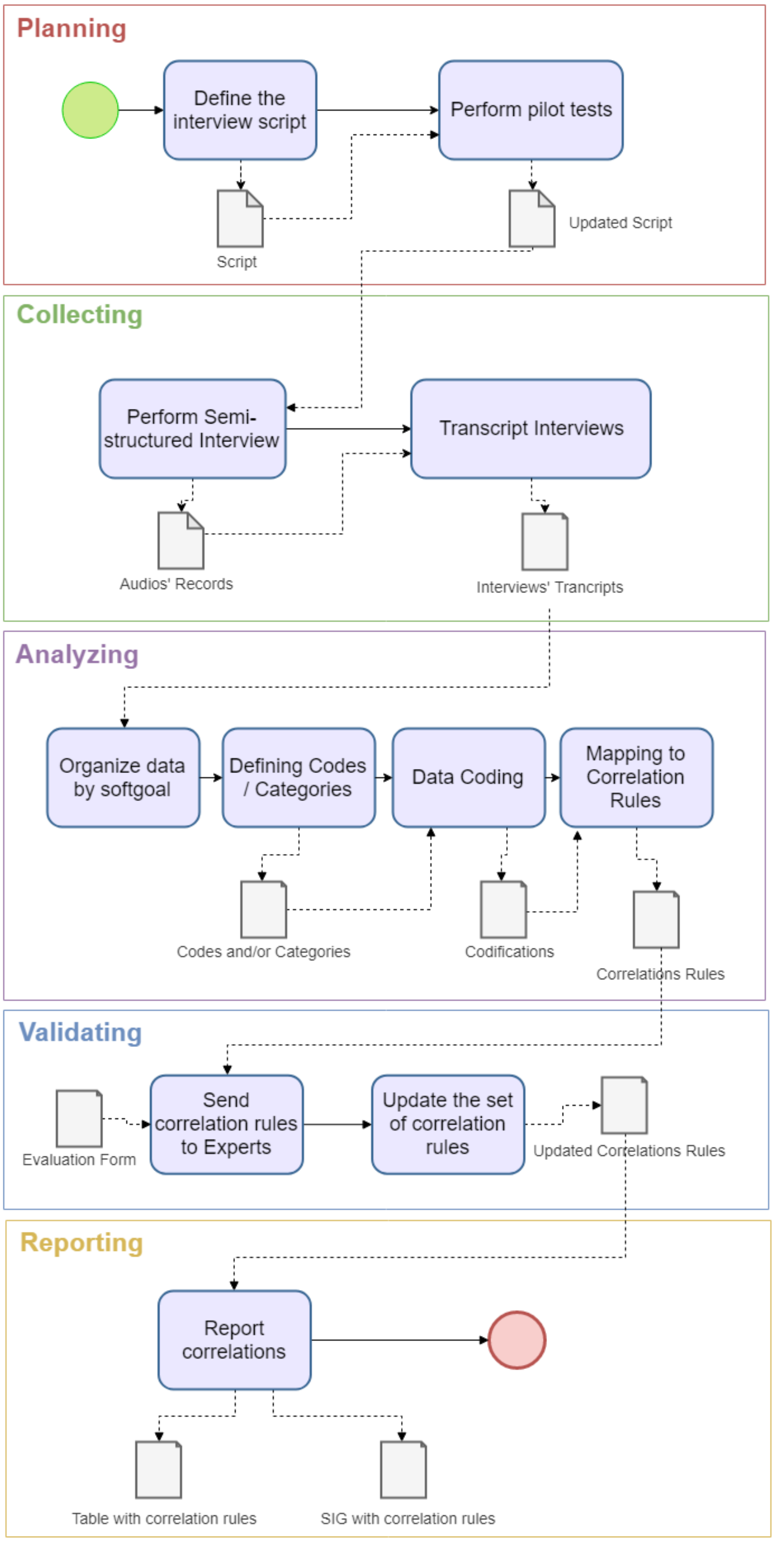
The phases *(i)* planning and *(ii)* reporting exist in TRACE because they are part of any research method. The phases *(ii)* collecting and *(iii)* analyzing follow the findings obtained in the SM study, regarding the definition of correlations, which states that there is a need not only to *collect* information but also to *analyze* them. Additionally, the phase *validating* can guarantee a more reliable correlation.

Figure 47 gives an overview of TRACE, which is better explained in the next subsections.

4.5.2.1 *Planning and Collecting*

One of the interview planning activities is to prepare the interview's script. Besides information such as the goal of the interview, instructions, demography data, such script should contain the questions itself. Furthermore, the script's questions depend on the type of interview

Figure 47 – TRACE Overview



Source: Author.

that will be performed: structured, unstructured, and semi-structured. In the case of this work, it is recommended to use a semi-structured interview because both closed and open-ended questions can be asked. Therefore, more correlations can be found and discussed by the interviewer.

Table 19 – Interview Script to be followed

Part	***Description***
Introduction	Explain the general goal of the research
Instructions	Explain the quality characteristics investigated Explain the questions Explain the impact scale
Demography Data	Ask profile and expertise data
Interview	For each strategy, ask *(1) "What is your experience in this strategy?"* ( ) Not Know, ( ) Know, ( ) Know and Worked *(2) "What is the impact of this strategy on user interaction quality characteristics"?* ( ) Break, ( ) Hurt, ( ) Unknown, ( ) Help, ( ) Make

Source: Author.

Table 19 presents the script proposed in this work. First, the process user should introduce the overall goal of the work. Then, instructions should be given. The first one is about the quality characteristics to be investigated; definitions of such characteristics should be made clear since people can have different perceptions about the quality characteristics. Then the questions to be answered should be explained. Finally, the impact scale used to guide the discussion about correlations should be explained. In the case of this work, the impact scale is that one from the NFR Framework: BREAK, HURT, UNKNOWN, HELP and MAKE (CHUNG *et al.*, 2000). After instructions, demography data should be collected to make sure the developer has minimal experience in UbiComp or IoT.

With the script and all materials prepared, a pilot test should be performed to verify their correctness. It is essential to highlight here that this pilot test should be executed with a developer, which is the target of the interview. Pilot tests with developers can demonstrate how they behave and then reveal presumptions that the interviewer may have.

After that, the interview can take place. For each strategy, the interviewer should ask to developers their experience with that strategy and then ask about the impact that each strategy can have on the NFR chosen or in a set of characteristics, which is the case of this work (user interaction quality characteristics). By indicating what is his/her opinion on that characteristic or set of characteristics, a quantitative overview about what the developers think can be collected.

Furthermore, the indication of such impact by using the scale is not the only data that should be collected. A mandatory step in this phase is that the interviewer should always make an effort and ask the developer if he/she has any comment on that impact, if he/she thinks it may have positive or negative effect on some other aspect of the NFR. For example, if the developer replies that there is a positive correlation with a particular characteristic, the interviewer can ask the reason and if he/she thinks there may be some negative relation to some other quality feature or even if there exist a negative impact on some aspect of that NFR, which can indicate a correlation to a subcharacteristic. In this way, more data can be collected and, thus, correlations can be defined. After all, the interviews should be transcribed, then the process user will have a set of data to be analyzed.

4.5.2.2 Analyzing

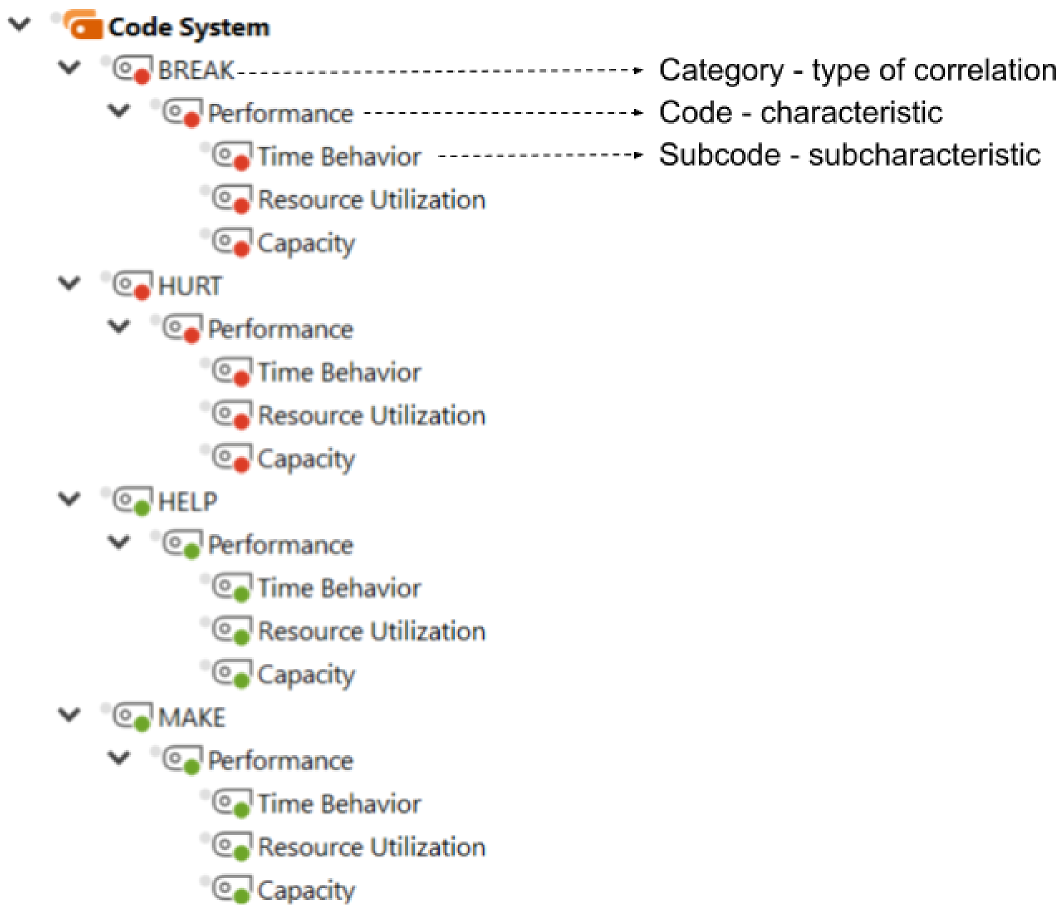
This work proposes to instantiate the Content Analysis (CA) method to analyze data (BARDLN, 1977). As previously explained, CA is a research method to classify any kind of communication material into identified categories of similar meanings (CHO; LEE, 2014). It is suitable for subjective interpretation of the content of text data through the systematic classification process of coding and identifying themes or patterns (HSIEH; SHANNON, 2005). Therefore, this method comes as a strategy to better analyze the collected data through the interview to define correlations.

As explained in Chapter 3, there are two manners of conducting qualitative content analysis: inductive approach and deductive approach (CHO; LEE, 2014). The inductive approach is suitable when prior knowledge regarding the topic under investigation is limited or fragmented. Therefore, codes, categories, or themes are directly drawn from the data. The deductive approach starts with preconceived codes or categories derived from prior relevant theory, research, or literature. In this part of the work, the deductive approach is proposed to be used because at this point the process user already knows what he/she wants to analyze. In the case of this work, the set of user interaction quality characteristics is the target of the analysis.

Furthermore, first, it is necessary to organize all collected data by the strategy, since the correlations will be defined according to them. For example, all feedbacks regarding a specific strategy should be grouped together and, then, the process user will have a general picture of what was said about that strategy.

After that, the preconceived codes or categories should be defined so that the data coding can start. In the case of this work, the quality characteristics will be the codes as well as the types of correlations. Therefore, the process user can create codes for each characteristic he/she wants to investigate: Performance, Usability, Security, among others. Subcharacteristics can also be defined as subcodes. Categories representing the types of correlations should also be created to group the characteristics. Figure 48 presents an example of predefined codes, in which Performance and its subcharacteristics were used and grouped into the four types of Correlations (break, hurt, help and make).

Figure 48 – Example of Predefined Codes and Categories to perform Data Coding

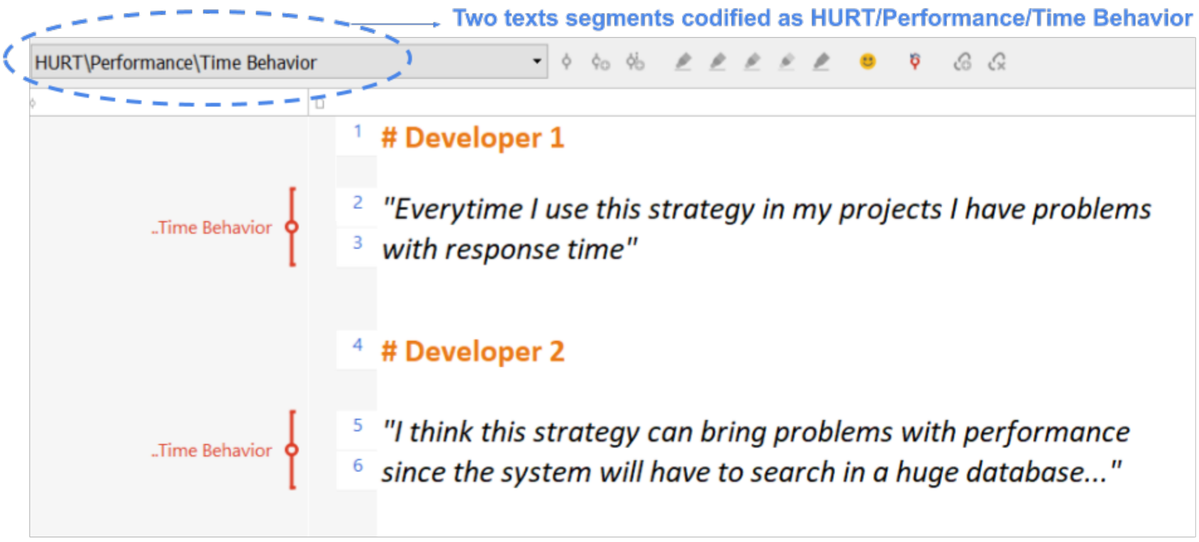


Source: Author

In this example, the types of correlations are categories because they group codes. The codes are the quality characteristics and the subcodes are the subcharacteristics. Furthermore, Performance and its subcharacteristics are defined to each correlation type. This repetition happens, because codifications can be made regarding this characteristic in any of these types of correlation. For example, a a strategy may hurt Time Behavior but it can help Resource Utilization.

Then, the process user can start the data coding, which means he/she inspects the data to understand the essence of what is being expressed by the developer. In the context of this work, the process user should examine the data from the interviews, which are the answers to questions about possible impacts. Therefore, the coding process should start by reading all the texts related to each strategy, and then the process user can codify texts segments that indicate an impact on the quality characteristics or subcharacteristics defined in the previous step. Figure 49 demonstrates an example of codification, where the process user coded two opinions from two different developers as a "hurt" impact in Time Behavior, subcharacteristic of Performance.

Figure 49 – Example of Codifications in TRACE



Source: Author.

This work proposes that the codifications become correlation rules in the following format: <*Strategy*> <kind of impact> <*Characteristic or Subcharacteristic*>. Following the example of Figure 49, one correlation rule would be created: *Softgoal X* HURTS *Time Behavior*. If conflicts are found, *e.g.*, there are codifications to both help and hurt, the process user should decide which one will be selected. A criteria to make the decision can be choose the codifications that have more citations. Finally, once a set of correlation rules is defined, they can be validated.

4.5.2.3 Validating and Reporting

As well as in the refinement of a characteristic using ARRANGE (See Section 4.3.2), the process user should perform validation to obtain more trustworthy correlation rules. The same validation from ARRANGE could be performed since the general activity is the same: texts' codification. However, depending on the number of coded textual segments, which depends on the number of interviewed developers, this task can be very time-consuming. Then, another approach is to ask experts in the area to evaluate the correlation rules directly instead of each codification. Knowing it is difficult to find experts who understand all characteristics, the process user can split the correlations rules among different experts, sending correlation rules to the ones that are more experienced to that strategy or characteristic.

This validation can be performed using a questionnaire in which every correlation rule and the corresponding rationale should be evaluated and classified as: *(i)* agree; *(ii)* partially agree; or *(iii)* disagree. Then, an analysis of the results of the evaluation should be done. If the expert disagree with a correlation rule, then, this rule should be excluded.

At the end of validation, it is possible to update SIG with the correlations, or a table can be created to store all correlations.

4.6 Chapter Summary

This chapter aimed to answer the second research question: RQ2 - *How can a correlation catalog for HCI quality characteristics in UbiComp and IoT systems be defined?* This research question was determined due to the exploratory study regarding NFRs Catalogs, which had shown the need for an approach to establish correlations between NFRs that takes into account specific development strategies.

Then, this chapter detailed a process called CORRELATE - Process to Capture, Analysis and CatalOg CoRRElations between QuaLity ChAracTEristics. In this process, NFRs are specified in the notation Softgoal Interdependency Graph (SIG).

CORRELATE is composed of four steps and supporting instruments and approaches[4]. The first step is to select the quality characteristics to start the definition of the catalog. A questionnaire-based instrument is proposed to make a prioritization.

Then, the next step is to refine the specific quality characteristic into subcharacteristics. In order to help the process users making refinements, this work instantiates the Grounded Theory method and proposed to use literature as a source of knowledge. ARRANGE is the name of this proposed approach. Through ARRANGE, it is possible to define a SIG that will be the core of a correlation catalog. Furthermore, ARRANGE is also useful when used alone (out of the process) for setting requirements and help in the high-level analysis, because it

[4] The documents of CORRELATE are available at https://github.com/great-ufc/CORRELATE

generates a SIG containing subcharacteristics and general solutions. This approach can be used by researchers and developers interested in refining characteristics and then creating a catalog of subcharacteristics and/or development strategies.

After refining the characteristic, specific development strategies should be identified to support the implementation of the subcharacteristics, and then completing the SIG generated by ARRANGE. This thesis proposed an instrument based on questionnaire to gather development strategies from developers.

At this point, every knowledge captured and analyzed are represented as softgoals. Then, it is necessary to analyze the impacts that each softgoal of the lowest level of the constructed SIG can have in the other quality characteristic. This analysis is more complex and can be done using TRACE, which was proposed in this thesis to identify correlations based on Interviews and Content Analysis.

The next chapter presents a complete NFR Catalog for the Invisibility characteristic, which is a result of the first execution of CORRELATE and, thus, a proof of concept of the process.

5 LEAD CATALOG

This Chapter presents the results achieved with the first execution of the CORRELATE process and its supporting instruments and approaches that generated LEAD - CataLog of Invisibility SubcharactEristics, StrAtegies anD Correlations. Then, it addresses RQ3 - *To what extent does one specific HCI quality characteristic from UbiComp and IoT impact on user interaction quality characteristics?*

The first step of this process is concerned with selecting a quality characteristic from AMICCaS to start the cataloging process. The result, which will be explained in this chapter, is the Invisibility characteristic. Therefore, this thesis investigates the impact of Invisibility on the user interaction quality characteristics as a strategy to answer RQ3 and to deal with the research problem defined in this thesis. All the knowledge captured and analyzed from Invisibility is cataloged in LEAD, which can be used by developers and requirements analysts to design a UbiComp and IoT system.

So, in short, Section 5.1 explains how each activity is executed and provides results from each step of the CORRELATE first execution. Then, all knowledge captured and analyzed from this process is cataloged and presented in Section 5.2.

5.1 General Results from the CORRELATE Process Execution

Results of each step of CORRELATE are presented in the following subsections, together with an explanation regarding how the results are obtained using the CORRELATE process with its instruments and approaches (ARRANGE and TRACE). Also, threats to validity to these results are presented.

5.1.1 Results of Step 1: Selecting Invisibility

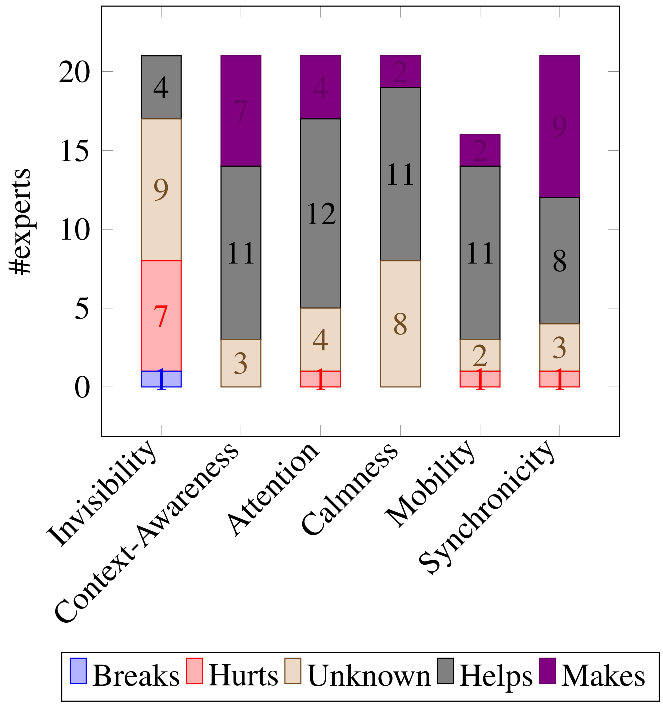
The first step of the proposed process is to select a quality characteristic to start the definition of the catalog. Then, the questionnaire presented in Table 18 was applied in an offline form to twenty-one HCI experts. They were all from Europe, selected by convenience in three situations: *(i)* during an international symposium about HCI; *(ii)* during a PhD thesis defense where the subject was HCI; and *(iii)* during a visit to the Laboratory of Industrial and Human Automation control, Mechanical engineering and Computer Science (LAMIH). The average of their experience working on HCI area was 16,5 years. Most of them (67%) had more than

10 years of experience (Between 10 and 35 years), while 33% had between 3 and 9 years of experience.

In this questionnaire, HCI experts should indicate the likely impact of each specific characteristic of UbiComp and IoT on Usability using the scale from the NFR Framework: BREAK, HURT, UNKNOWN, HELP and MAKE (CHUNG *et al.*, 2000). It is worth to remind here that although this thesis aims to investigate correlations with the set of user interaction quality characteristics, only Usability was used to make the prioritization. The purpose of this step is to be simple and quick. Furthermore, Usability represents an important characteristic to achieve quality in a system.

The results of the questionnaire are presented in Figure 50. In total, 125 answers were obtained. Most of them (57) were for the option "Helps". From them, 12 answers were for Attention, followed by Context-Awareness (11 answers), Calmness (11 answers), and Mobility (11 answers).

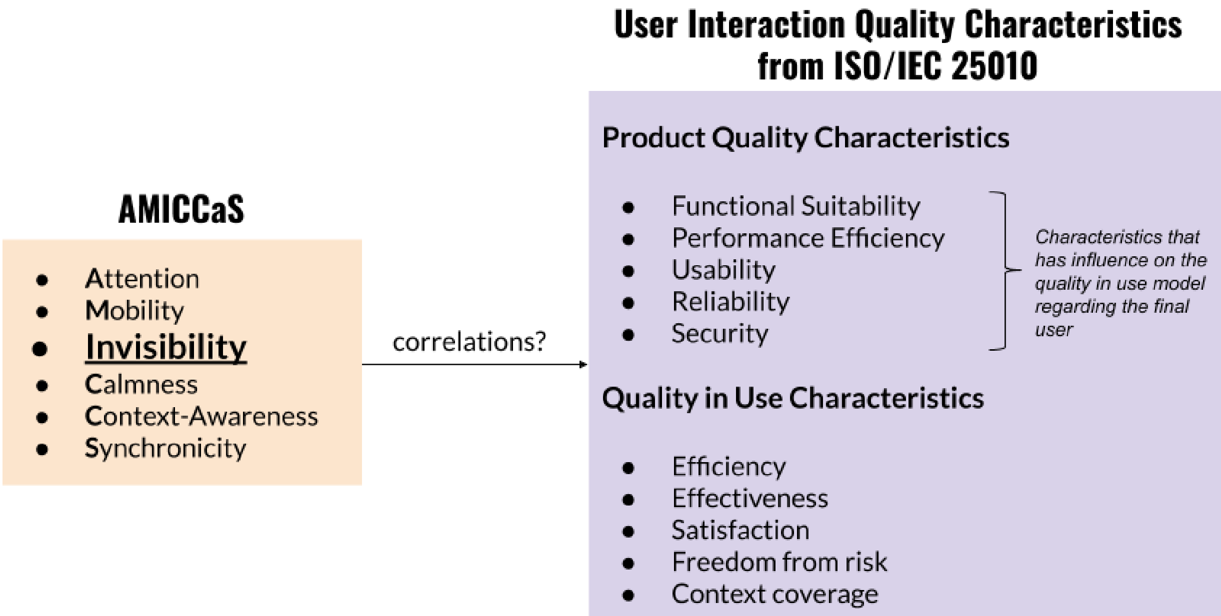
Figure 50 – Results of the Questionnaire-based Instrument to Prioritize Characteristics



Source: Author.

The option with least answers was "Breaks", with only one answer (Invisibility). The option "Hurts" had 10 answers, most of them (7) for Invisibility. The option "Unknown" had 29 answers and most of them (9) were for Invisibility, followed by Calmness (8). Therefore, according to this study, Invisibility is the characteristic most likely to have negative correlations with Usability, being the selected one to start the investigation that answers RQ3 (See Figure 51).

Figure 51 – Invisibility selected to be investigated regarding correlations



Source: Author.

Furthermore, it is interesting to note that there is no complete agreement between the experts, which shows that negative correlations between quality characteristics are relative. It depends of the subcharacteristic and the development strategies used to implement the subcharacteristic.

Then, based on this result, Invisibility was chosen as the first quality characteristic for study. This step is explained in the next subsection.

5.1.2 Results of Step 2: Refining Invisibility using ARRANGE

This step is concerned to narrow down more specific concepts for the Invisibility characteristic, selected in the previous step. Although Invisibility has been pointed out as a primary and essential characteristic for UbiComp (SATYANARAYANAN, 2001) (KARVONEN; KUJALA, 2014) (SAHA; MUKHERJEE, 2003) (COSTA *et al.*, 2008) (SCHOLTZ; CONSOLVO, 2004) and, consequently, IoT applications (CARVALHO *et al.*, 2017) (ANDRADE *et al.*, 2017), the literature lacks well-defined taxonomies or a body of organized knowledge about how to achieve it during software development. Therefore, it is necessary to refine it, which was performed through ARRANGE approach presented in Subsection 4.3.2. The next sections explain in more details results of each phase from the ARRANGE approach (planning, collecting, analyzing and reporting).

5.1.2.1 Planning and Collecting

As previously explained in the ARRANGE approach (See Subsection 4.3.2), the planning phase aims to identify the area of interest and the research question that will drive the work. Following the suggestion of ARRANGE, in which the area of interest must be the NFR itself, *Invisibility* becomes the area of interest. Thus, the research question is: *"How can the Invisibility characteristic in UbiComp and IoT applications be defined and refined into subcharacteristics and solutions?"*.

The collecting phase aims to collect the necessary data to answer the general research question. ARRANGE suggests the use of a systematic literature review or snowballing procedure (KITCHENHAM; CHARTERS, 2007) (PETERSEN *et al.*, 2015) because ARRANGE has the literature as a source of knowledge. To collect the data about Invisibility, only the snowballing strategy was performed in one iteration. Snowballing was chosen because there was already a SM study related to quality characteristics from UbiComp (CARVALHO *et al.*, 2017).

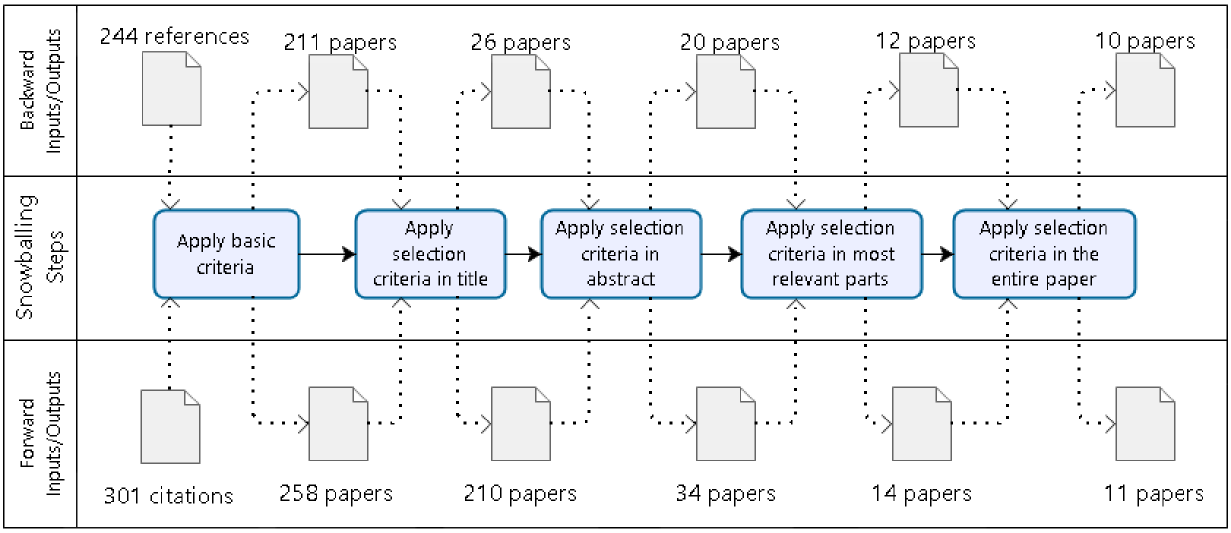
Both snowballing strategies (backward and forward) need a starting set of papers. Since the existing SM study had papers related to 27 characteristics, only the papers (9 in total) citing Invisibility were selected. Thus, a consistent set of papers was identified as a starting set for the snowballing method.

Figure 52 presents the snowballing steps. To perform backward snowballing, all references from the 9 papers, which contained 244 references in total, were identified. For the forward snowballing, all the citations of the 9 papers using Google Scholar, *i.e.*, 301 in total, were identified.

The first step of both snowballing strategies was applying the following basic criteria to filter references and citations: *(i)* Papers not written in English; *(ii)* Papers published before 1991 (year of the first paper about Ubicomp); and *(iii)* References and citations that are only web addresses of research groups, newspapers and/or companies. It was possible to reduce 244 references to 211 papers and 301 citations to 258 papers.

Then, the following criteria was applied in all steps of snowballing: *(i)* the study is not related to invisible interaction in UbiComp or IoT, which means we were looking for papers related to the interaction perspective, regardless of the research type (*i.e.*, solution proposals, experiments, case studies).

Figure 52 – Snowballing Steps and Results



Source: Author.

The second step consisted of applying criteria while reading the title, which resulted in 26 papers (backward) and 210 papers (forward). The third step consisted of applying it in abstract reading, which reduced the set to 20 papers (backward) and 34 papers (forward). The fourth step consisted of reading the most relevant parts of the paper, resulting in 12 papers (backward) and 14 papers (forward). The last step was performed by applying the criteria during the full paper reading. This step resulted in 10 and 11 final papers, for backward and forward, respectively.

Additionally, a complementary search in a database for the IoT domain was performed to get more pieces of evidence regarding IoT in literature. Scopus library was used with the following search string: ((“internet of things” OR iot) AND (interaction) AND (transparency OR invisibility OR disappearance OR diffusion OR implicit OR “minimal user distraction” OR unobtrusive)). This library was selected because of its broad coverage and stability (SANTOS *et al.*, 2017). It also covers major publishers in the requirements area (HORKOFF *et al.*, 2016).

This search resulted in 40 papers. Then, the same criteria from the snowballing procedure was applied in the title and abstract reading, which resulted in 5 papers. Finally, the most relevant parts of the paper were read and the criteria was applied, resulting in only 1 paper.

Therefore, this collecting step resulted in 9 papers from a previous SM, 10 papers from backward snowballing, 11 papers from forward snowballing and 1 paper from the database search, resulting in 31 papers.

Following the recommendation of ARRANGE for extracting data, the papers were extracted using a form with the following questions: *(a)* What are the Invisibility definitions? *(b)* How is Invisibility characterized? and *(c)* How is Invisibility implemented?. After this data extraction, a set of existing data about Invisibility was collected. For example, regarding the definitions, it was discovered that there are several definitions proposed by researchers. Some of them are:

- "A system that requires minimal human intervention offers a reasonable approximation of invisibility" (SAHA; MUKHERJEE, 2003);
- "Invisibility refers to the integration of a system into the user environment..." (SCHOLTZ; CONSOLVO, 2004);
- "The extent to which the system consists of hidden components in the physical space and interaction is performed through natural interfaces" (KOUROUTHANASSIS *et al.*, 2008).

It is possible to see that definitions differ from each other in some respects and have similarities in others. The first definition talks about minimal user intervention, while the last two address the concept by another respect: the user's physical environment. Still, in the latter, the concept has been linked with natural interfaces. The results from data extraction are even worse for the subcharacteristics, which leads us to conclude that there is no consensus about a characterization of Invisibility.

All data needs to be deeply analyzed to understand how Invisibility can be broken down into subcharacteristics. This activity was done through the analyzing phase, explained in detail in the following subsection.

5.1.2.2 Analyzing

As explained in Subsection 4.3.2, the analyzing phase is performed through data coding, which is performed in three tasks: open, axial and selective, as originally performed in the GT method. In the open coding, the researcher should inspect the data to understand the essence of what is being expressed (GORBIN; STRAUSS, 2008). MAXQDA12 tool[1] was used in this work to support all steps of coding.

First, all extracted data about Invisibility was imported into this tool. Then, this data was analyzed in depth, which enabled us to codify text segments by creating codes (open coding). Codes can represent a single word, a phrase or a whole paragraph. Two examples of them from the open coding to Invisibility are given in Table 20.

[1] https://www.maxqda.com/

Table 20 – Codes and GT

ID	Text segments from the extracted data	Code
1	“A system that requires minimal human intervention offers a reasonable approximation of invisibility.” (SAHA; MUKHERJEE, 2003)	Minimal User Involvement
2	“To achieve invisibility, the system must keep the focus of a user to the task while keeping computing invisible...” (ABDULRAZAK; MALIK, 2012)	
3	“According to the ‘invisibility’ ideal, the user should not be bothered with details of the system’s functioning...” (KARVONEN; KUJALA, 2014)	
4	“...ubicomp applications should allow users to utilize the skills they have obtained from daily lives to interact with computers.” (YUE *et al.*, 2007)	Usage of Natural Interfaces
5	“PICT technology must be transparent to users, and must have interfaces that are intuitive for humans.” (THOMPSON; AZVINE, 2004)	
6	“If ubicomp applications support more natural human forms of communication, they will create more natural and expressively powerful means of interaction...” (THOMPSON; AZVINE, 2004)	

Source: Author.

Since all of them dealt with the minimum intervention and because the user should not be focused on the system, but in their day to day tasks, the definitions with IDs 1, 2 and 3 were represented by a code called “Minimal User Involvement”. Regarding text segments with ID 4, 5 and 6, Invisibility is related to a natural way of interaction. Therefore, these segments were coded with a code named “Usage of Natural Interface”. For that reason, “Minimal User Involvement” and “Usage of Natural Interface” represent the understanding of those text segments.

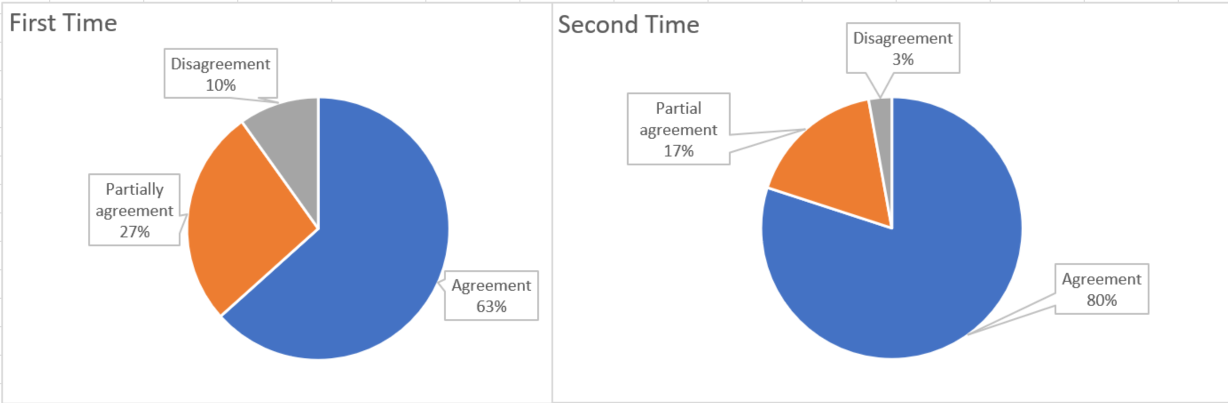
After finishing one round of open coding all the matching between text segments and codes were evaluated by experts, as proposed in ARRRANGE. In this work, two experts in HCI area evaluated the suggested codes. One expert had 32 years of experience in HCI area while the other had 8 years. The evaluation process showed in Figure 41 was followed for open coding.

This process was carried out twice until reaching a total consensus. The evaluation rate among researchers is shown in Figure 53. The agreement rate was 63% and 80% in the first and second time, respectively.

At the end of open coding, after total consensus, this activity resulted in 118 text segments represented by 18 codes. The codes and the number of segments coded by them (*i.e.*, groundedness) are presented in Figure 54. Therefore, codes with higher groundedness represent stronger concepts.

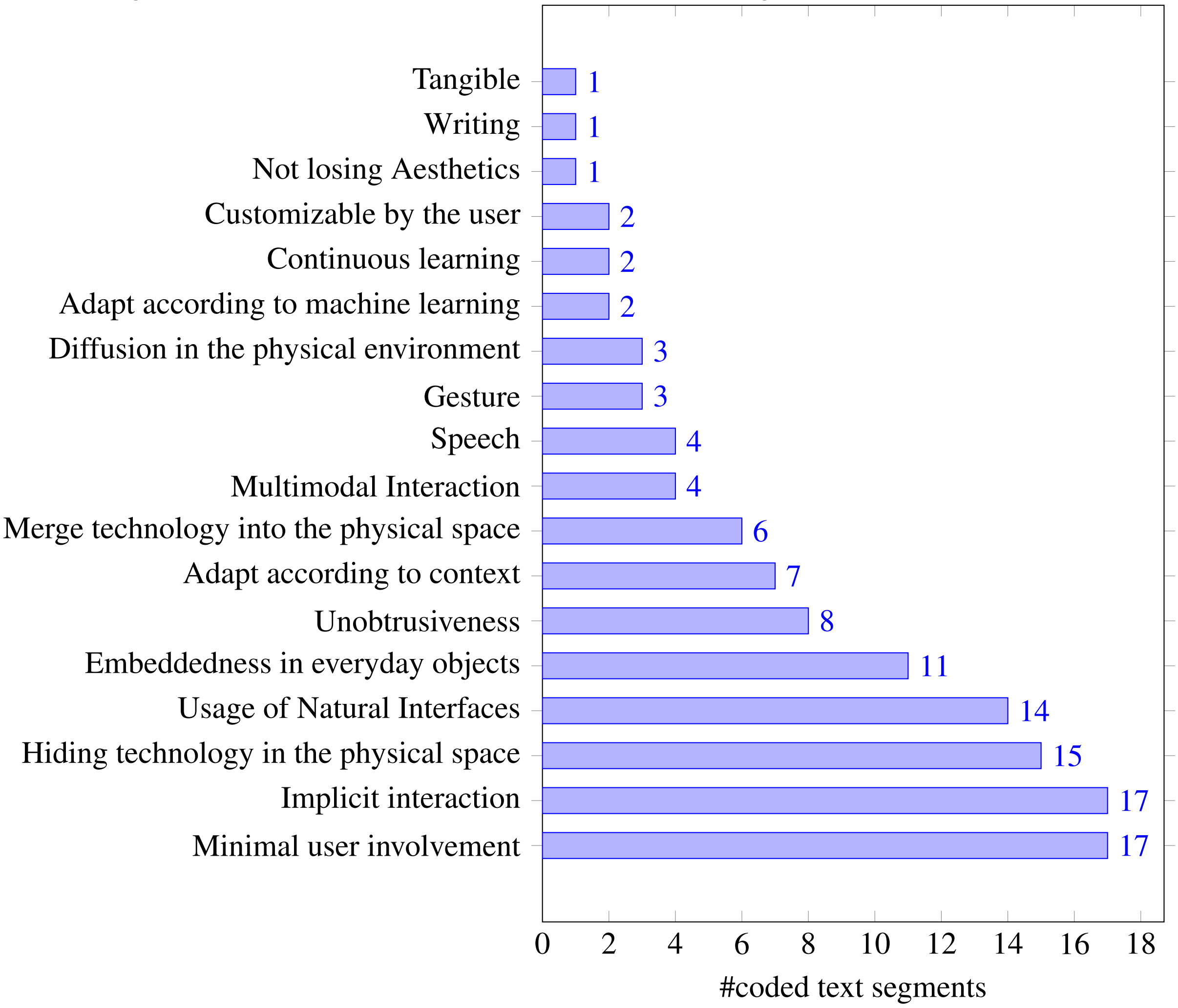
It is possible to realize that “Minimal user involvement” and “Implicit interaction” are the codes with higher groundedness, followed by “Hiding technology in the physical space” and “Usage of natural interfaces”. Three codes (“tangible”, “writing” and “not losing aesthetics”) have the lowest groundedness. However, these codes were kept because they represent a low-level

Figure 53 – Code Evaluation Rate among Researchers



Source: Author.

Figure 54 – Codes and its Amount of Coded Text Segments (Groundednesss)



Source: Author

concept, which is part of another high-level concept.

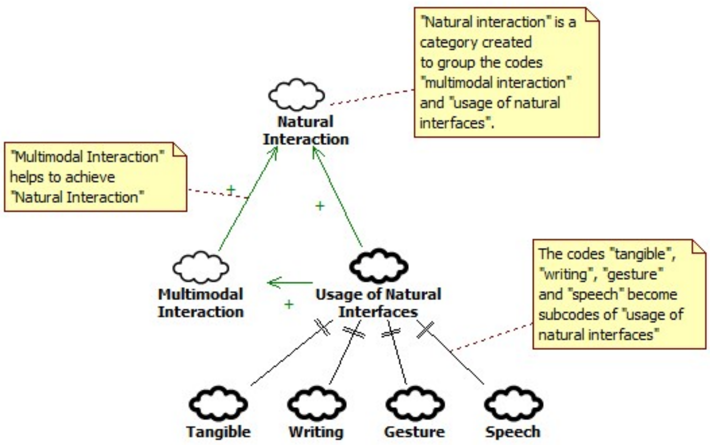
All these concepts were directly represented as softgoals of the Invisibility SIG, either as a NFR softgoal or operationalizing softgoal. Therefore, all codes were classifief according to the type of softgoal: NFR softgoal or operationalizing softgoal. For example, “gesture” is an operationalizing softgoal, since it operationalizes Invisibility. On the other hand, “implicit interaction” is still seen as a NFR softgoal, because it is an overall constraint in the system.

Furthermore, these codes need to be related with each other and to the main softgoal Invisibility, which are the next steps (axial and selective coding).

Axial coding is the process of relating concepts or grouping them by creating categories (a high-level concept that represents a group of codes) (GORBIN; STRAUSS, 2008). As suggested by the GT method, the relations should be created by analyzing implicit meaning between the codes and, as suggested by ARRANGE, these relations should be based on the types of contributions from the SIG notation (AND, OR, MAKE, HELP, BREAK and HURT).

Figure 55 presents examples of axial coding. The codes “tangible”, “writing”, “gesture” and “speech” are examples of “usage of natural interfaces”. They were related with OR contribution, which means the softgoal “usage of natural interfaces” can be achieved by using any of these softgoals. Additionally, Figure 55 shows the creation of a category called “Natural Interaction”, which embraces the codes “multimodal interaction” and “usage of natural interfaces”, since they help to achieve more natural interaction.

Figure 55 – Example of Axial Coding – Category of Codes



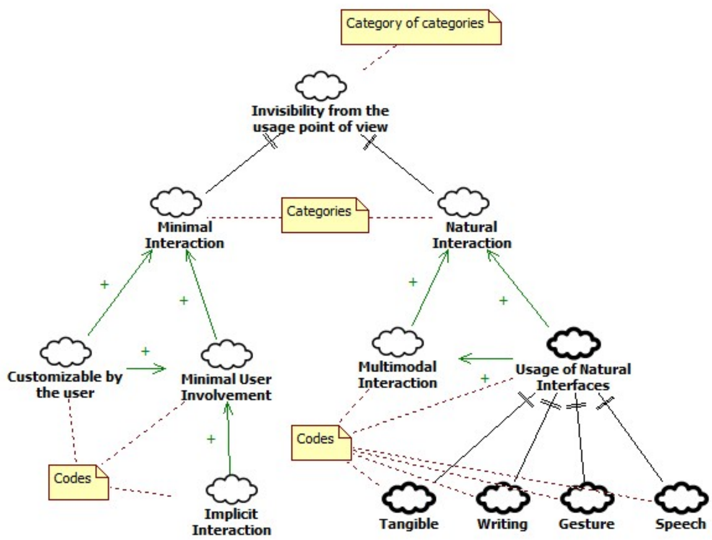
Source: Author.

At the end, only the following relations were used: “AND, OR and HELP”. AND contribution was only used when it was sure that all sub softgoals were needed to achieve the parent. OR contribution was used when the sub softgoals are examples of the same kind to achieve the parent softgoal. HELP was used when it was known from the data about the positive support and the AND/OR contribution was not suitable.

Furthermore, the researcher can group categories into new categories, creating a chain of categories. That is why axial coding is like putting together a series of interlinking blocks to build a knowledge pyramid (GORBIN; STRAUSS, 2008). Figure 56 presents an example of a category of categories. The codes "customizable by the user" and "minimal user involvement" were grouped in a category called "Minimal Interaction" because they have the goal of minimizing the interaction. Customizable by the User means that the system should, whenever possible, let the users make changes in the system according to their personal preferences, minimizing future interaction. Minimal User Involvement means the system should, whenever possible, require less user interaction by decreasing user inputs and actions. That is the reason why these softgoals were grouped into Minimal Interaction.

This category and the other category "Natural Interaction" were grouped into a new category called "Invisibility from the usage point of view" because they all are related to the system's usage.

Figure 56 – Example of Axial Coding - Category of Categories



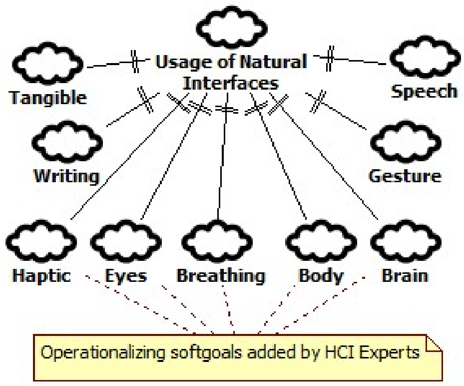
Source: Author.

Finally, when all codes and categories can be related to a core category, the selective coding is taking place. The core category of this work is Invisibility, which is the main NFR softgoal this study would like to achieve. Therefore, all codes and categories from open and axial coding were related in some way to this characteristic.

In this work, axial and selective coding were performed together by only one researcher who also led open coding (*i.e.*, the author of this thesis). The researcher analyzed the codes to group them into categories, to make the necessary relations and to relate them in the core category, always with an effort to describe how the concepts are grounded in data. Then, meetings and discussions were held with the same two HCI experts, who participated in open coding, to reach an agreement about the organization of the concepts.

During these meetings, the experts helped not only with the organization, but also with complementing the Invisibility SIG by indicating additional softgoals. For example, the softgoal "Usage of Natural Interfaces" is composed by "tangible", "writing", "gesture" and "speech". However, these are not the only existing natural interfaces in literature. Therefore, HCI experts suggested adding the following interfaces in the SIG: body, eyes and haptic. Also, the experts suggested to add "Minimize user's effort in tasks" under the "Minimal User Involvement" softgoal (See Figure 57).

Figure 57 – Example of softgoals added by HCI experts



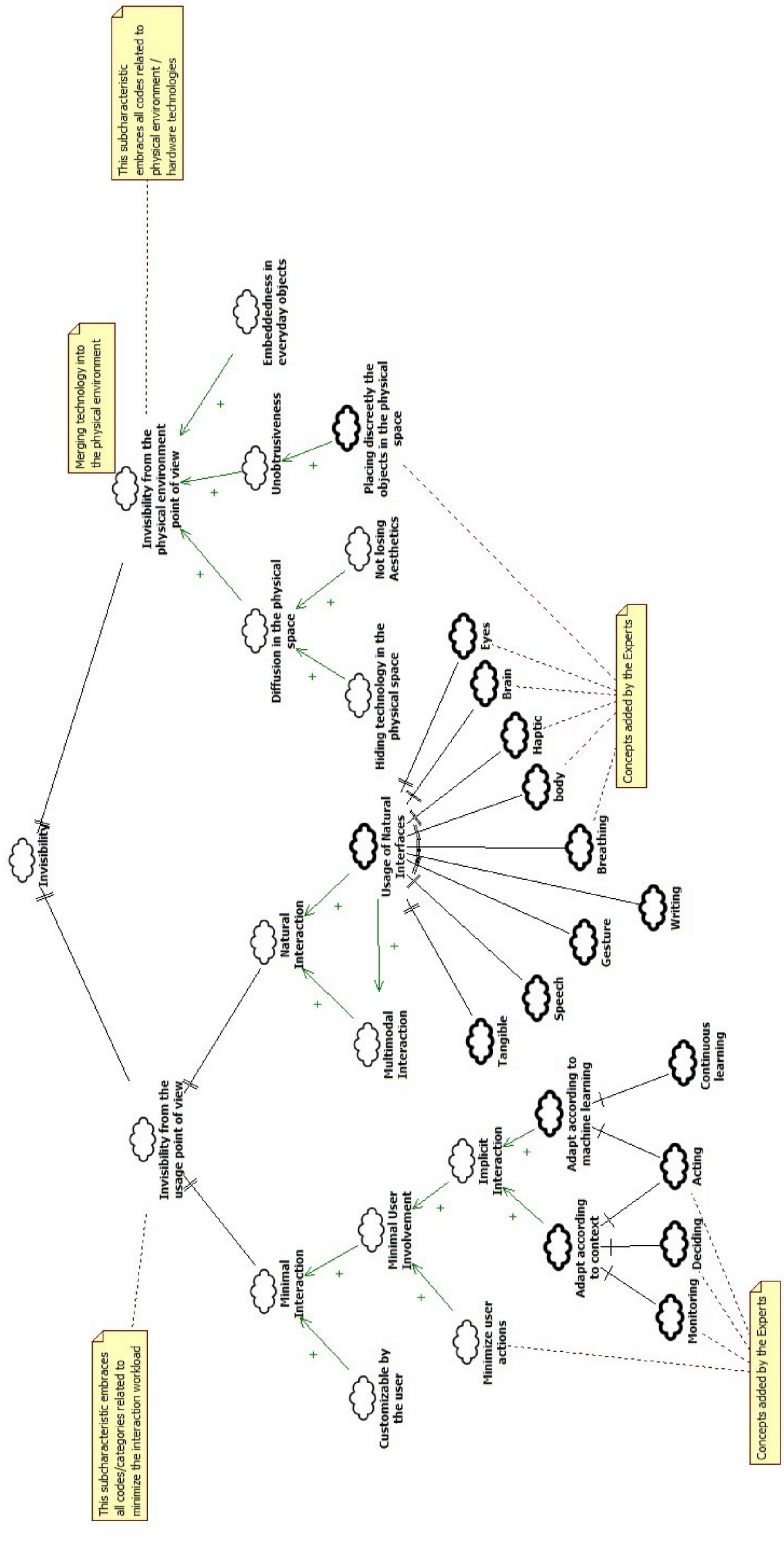
Source: Author.

At the end of GT, it was possible to create a SIG for Invisibility (See Figure 58) containing 2 sub characteristics, 12 subcharacteristics (*i.e.*, NFR softgoals), three general strategies and fourteen specific strategies (*i.e.*, operationalizing softgoals). In fact, this knowledge became the core of the Invisibility catalog, later expanded to include more softgoals with the results from the next step.

5.1.3 Results of Step 3: Identifying development strategies for Invisibility

As explained in Section 2.2.4.1, an operationalizing softgoal can not only be generic (broad), but it can also be refined into specific operationalizing softgoals (CHUNG *et al.*, 2000), helping software developers even more. Then, the Invisibility SIG generated from GT with HCI experts could be complemented with specific operationalizing softgoals accumulated from

Figure 58 – Core of the Invisibility SIG generated by ARRANGE



Source: Author.

developers of ubiquitous and IoT applications.

As suggested by Step 3 of the CORRELATE process, this identification can be done based on the own experience of researchers or through an external source. Considering that Invisibility brought specific subcharacteristics and general solutions, developers were used as external source since they accumulate knowledge regarding the strategies in real projects.

Therefore, the proposed instrument presented in Figures 44 and 45 was used to design a questionnaire specific for the Invisibility softgoals. This questionnaire comprised 11 questions that consider the softgoals at the lowest level of the SIG generated by the GT. For example, developers should answer how they implement “Speech” from the “Usage of Natural Interfaces” softgoal.

According to (CHUNG *et al.*, 2000), solutions can be any technique, technology, data, strategy, operations or constraints that help designing or coding a software system that has to satisfy a characteristic (CHUNG *et al.*, 2000). Therefore, the developers could answer any solution about how to design or code that softgoal.

Then, the questionnaire was sent to seven UbiComp and IoT developers to collect data from their experience. These developers were selected through the convenience sampling, which means the nearest and most convenient persons are selected as subjects (WOHLIN *et al.*, 2012). Additionally, they were selected because they have experience with UbiComp and IoT applications. Furthermore, the questionnaire was sent as an editable spreadsheet form. So, developers could feel free to respond in their time.

It is worth noting that an experienced researcher in HCI area evaluated the questionnaire before sending it out to the developers. This researcher read the questionnaire and simulated a developer answering it. Then, improvements were suggested and corrected.

All seven developers answered the questionnaire. They have been working mainly with smart building and healthcare applications for six years on average. Six of them have been working with programming and testing and one has worked with architecture project.

After receiving all answers, an analysis was performed to extract common and most cited solutions. When an uncertainty about any of the solutions arose, the developers were contacted to solve any doubt.

An example of a strategy indicated by the developers is the Speech API from Google, which helps implementing “Speech” in Android applications, allowing conversion from speech to text and vice versa. Another example of solution suggested by the developers is the “Integration with third-party services", which means using data from existing accounts to minimize the need to type personal data to sign in into an application. They suggested using the Google Sign-in API or Facebook API.

In total, seven general and forty-two specific development strategies were identified respectively and added to the catalog. Figure 59 presents the final SIG, resulting in 44 softgoals in the lowest level (See Table 21). These softgoals were then considered as inputs to the next step of the CORRELATE process.

Table 21 – Examples of Coded Segment Texts

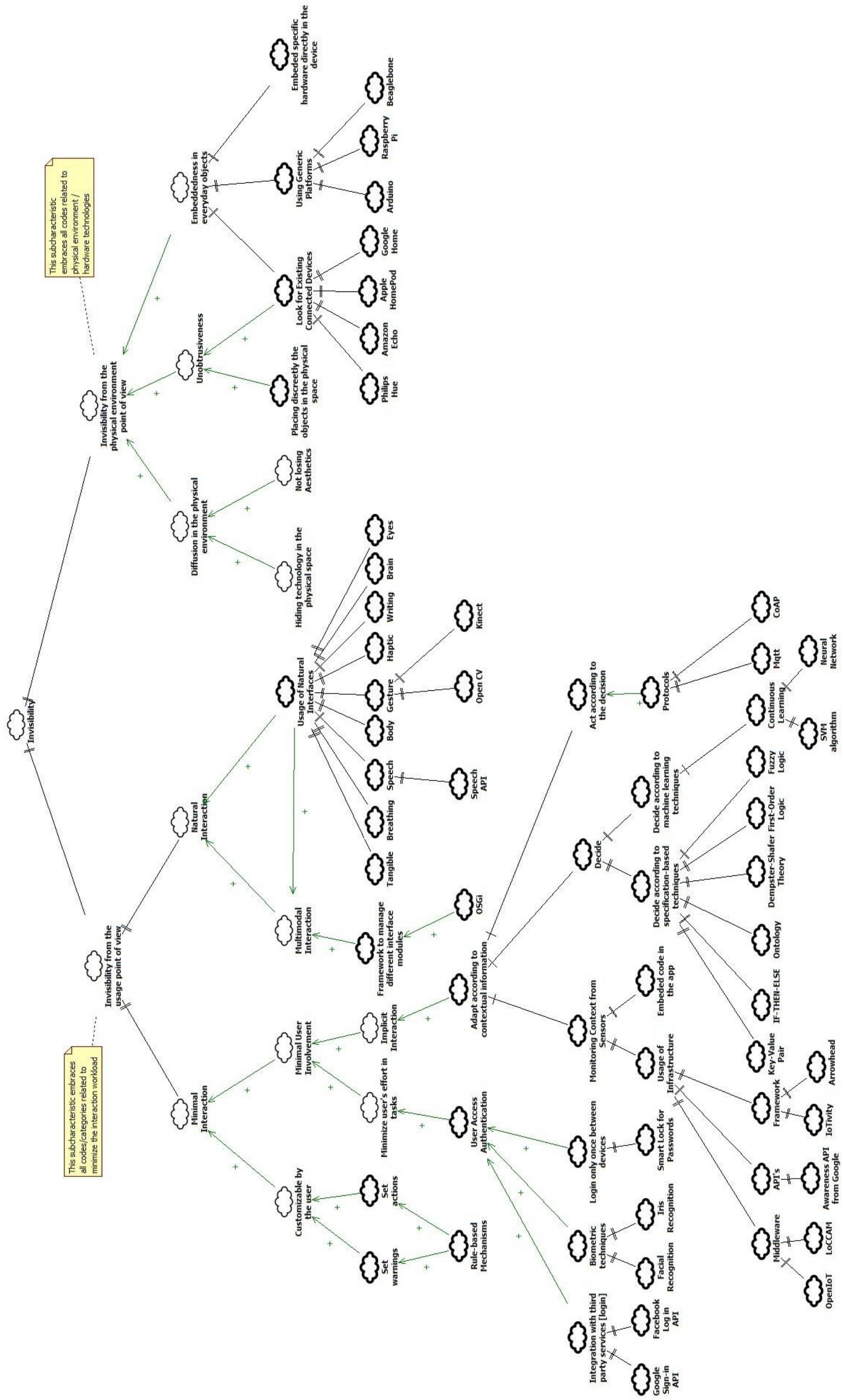
ID	***Softgoal***	***ID***	***Softgoal***
S1	IFTTT	S23	OSGi
S2	Google Sign in API	S24	Speech API
S3	Facebook Log in API	S25	OpenCV
S4	Facial Recognition	S26	Kinect
S5	Iris Recognition	S27	Tangible
S6	SmartLock	S28	Breath
S7	OpenIoT	S29	Body
S8	LoCCAM	S30	Haptic
S9	Awareness	S31	Writing
S10	IoTivity	S32	Brain
S11	Arrowhead	S33	Eyes
S12	Embedded Code	S34	Arduino
S13	Key value pair	S35	Raspberry
S14	If then else	S36	Beaglebone
S15	Ontology	S37	Philips Hue
S16	Dempster-Shafer Theory	S38	Amazon Eco
S17	First Order Logic	S39	Apple HomePod
S18	Fuzzy Logic	S40	Google home
S19	SVM Algorithm	S41	Embedded specific hardware
S20	Neural Networks	S42	Hide technology
S21	MQTT	S43	Not losing aesthetics
S22	CoAP	S44	Place objects discreetly

Source: Author.

5.1.4 Results of Step 4: Defining correlations from Invisibility

The TRACE approach (See Subsection 4.5.2) was applied to define correlations. As previously explained, this approach groups the Interview and Content Analysis research methods and it contains five phases: planning, collecting, analyzing, validating and reporting. Each phase is presented as follows.

Figure 59 – The final Invisibility SIG



Source: Author.

5.1.4.1 Planning and Collecting

The planning phase is concerned to prepare a script to guide the interview. The script prepared to this phase followed the structure presented in Section 4.5.2, containing four parts: introduction, instructions, demography data and questions of the interview itself.

In the interview part, each softgoal from the lowest level of the SIG generated in the last step (44 softgoals) was linked to the question: *What is the impact in the user interaction quality?*, resulting in forty-four questions. Developers answered first what was their experience regarding that softgoal: *a)* Not know; *b)* Know; or *c)* Know and Already Worked. Only developers who *b)* Know; or *c)* Know and Already Worked gave their feedback regarding the impact of the softgoal.

The script was improved in two rounds of evaluation and feedback: first, it was evaluated by two professors and one HCI researcher in order to discover possible ambiguities and shortcomings. Second, a pilot interview was conducted with an undergraduate student who works in the development of an IoT application at GREat lab.

Fifteen (15) developers were recruited by the convenience sampling technique (WOHLIN *et al.*, 2012). Table 22 presents the demographic data from them.

Table 22 – Demographic Data of the Respondents

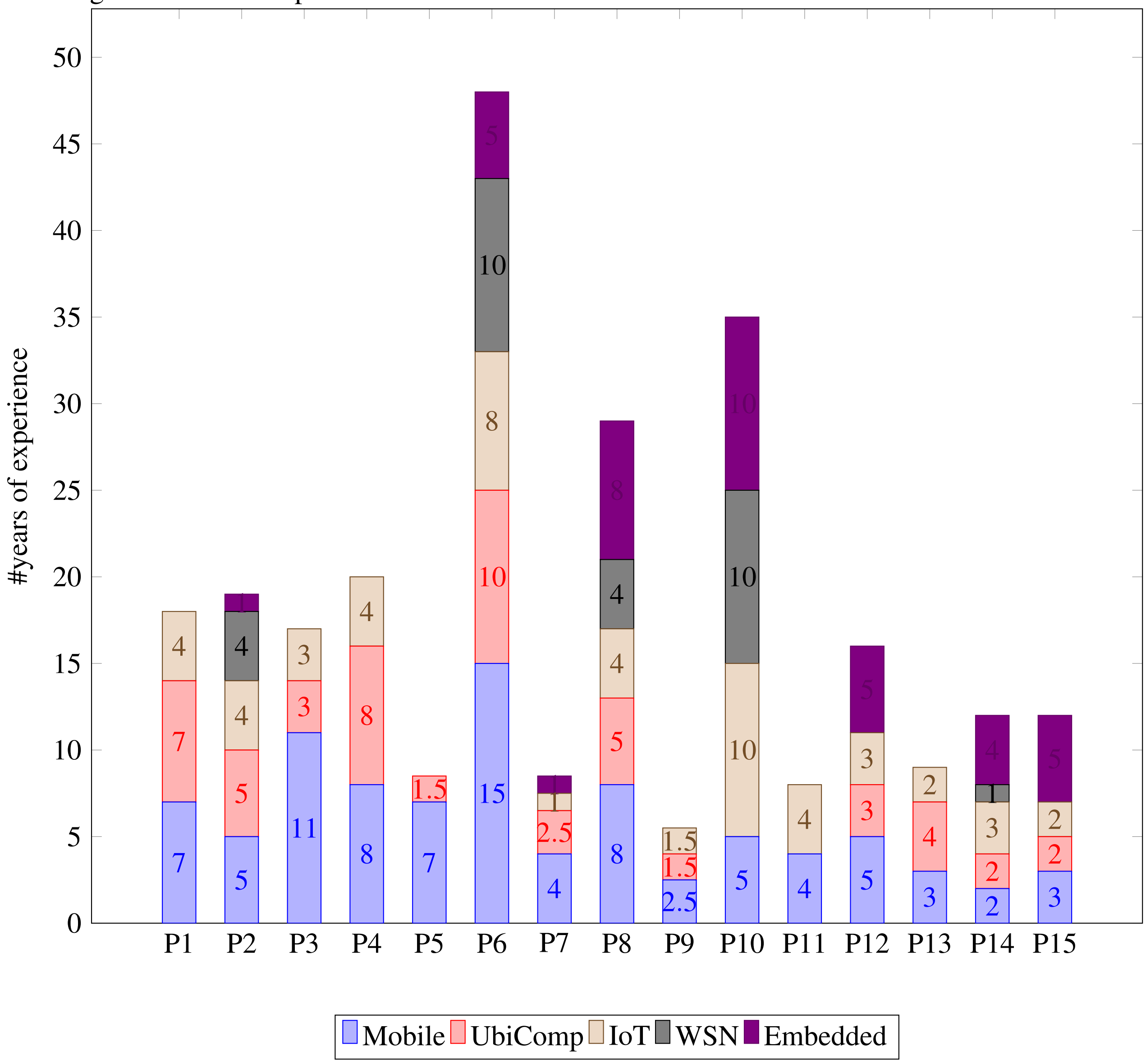
ID	Company / Institute	Country	Degree	Current Ocupation
1	GREat - UFC	Brazil	D.Sc Candidate	Requirements Analyst / Researcher
2	GREat - UFC	Brazil	D.Sc Candidate	Developer / Researcher
3	GREat - UFC	Brazil	D.Sc Candidate	Researcher
4	GREat - UFC	Brazil	D.Sc Candidate	Requirements Analyst / Researcher
5	Université Polytechnique Hauts-de-France	France	D.Sc Candidate	Professor / Researcher
6	Quixadá Campus - UFC	Brazil	D.Sc	Professor / Researcher
7	Quixadá Campus - UFC	Brazil	D.Sc Candidate	Professor / Researcher
8	Crateus Campus - UFC	Brazil	D.Sc Candidate	Professor / Researcher
9	GREat - UFC	Brazil	Master Candidate	Developer / Researcher
10	Quixadá Campus - UFC	Brazil	D.Sc Candidate	Professor / Researcher
11	GREat - UFC	Brazil	Master Candidate	Researcher
12	GREat - UFC	Brazil	Master Candidate	Developer / Researcher
13	GREat - UFC	Brazil	D.Sc	Researcher
14	OSF Global Services	Brazil	Master	Developer / Professor
15	Infovista	Sweden	Master	Developer

Source: Author.

One criteria was defined to recruit them, which was to have at least two years of experience of being a developer in any of these areas: Mobile Computing, Ubiquitous Computing, Internet of Things, Wireless Sensor Network and Embedded Systems.

Figure 60 presents the years of experience of each developer regarding each area. All of them had experience with Mobile Computing, varying between 2 and 15 years, with an average of 6 years. Regarding UbiComp, only two developers stated they did not have experience, the rest varied between 1,5 and 10 years, 4,2 years being the average. Most developers also had experience with IoT, varying between 1 and 10 years, with an average of 3,8 years. Wireless Sensor Network and Embedded Systems had less developers with experience, presenting an average of 5,8 and 4,9 years of experience, respectively.

Figure 60 – Developers' Profile



Source: Author.

Most interviews were conducted face-to-face, except for three of them that were performed through video conferences due to the location and availability of developers. The duration varied between 49 and 87 minutes, with an average of 60 minutes. All of them were recorded and transcribed.

5.1.4.2 Analyzing

A summary of the quantitative answers to the impact on the set of user interaction quality characteristics are presented in Table 23. These answers gave an overview about what developers think in general, but it was not possible to define correlations from them. An important observation made from these answers was that for every softgoal, regarding positive impact, the quantity of answers to HELP was bigger than answers to make. In the same way, regarding negative impact, the quantity of answers to HURT was bigger than answers to BREAK. This data shows that correlations should be defined with HELP and HURT, which became the categories of data coding activity.

Then, a qualitative analysis was performed following TRACE recommendations, described in Section 4.5.2. Then, first, all data was organized by softgoal. Therefore, answers and comments by all developers for a softgoal were grouped together to make the analysis easy and clear.

Then, codes were defined and added in the tool since analysis uses a deductive approach. Figure 61 illustrates the initial set of codes. They correspond to the type of correlations (HELP and HURT correlations), the type of quality model from (ISO/IEC 25010, 2011), the user interaction quality characteristics (See Figure 9) and their subcharacteristics.

The type of correlations (HELP and HURT) groups the type of quality model, which in turn groups their characteristics and subcharacteristics.

Then, the data coding was performed by reading all the feedbacks from the developers of each softgoal. This coding activity was executed with the help of the MAXQDA tool, which Figure 62 gives an overview about. Box 1 is where all strategies are listed, they are treated as documents by the tool. Box 2 keeps the categories, codes and subcodes while Box 3 is used to visualize data of the strategies and codify them. Every time a sentence seems to have a reference for a quality characteristic, a code representing that characteristic is used to match the sentence. Data that could not be coded into the predetermined characteristics, but coded instead with another known quality characteristic were also coded.

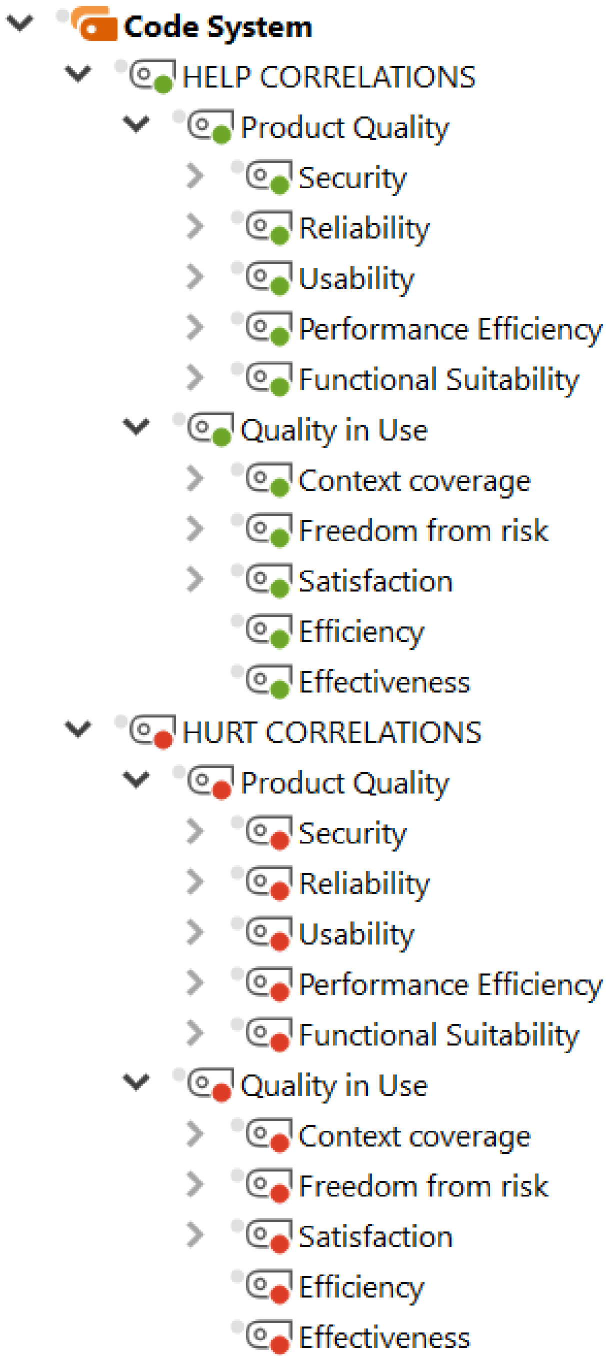
Several examples of coded text segments are presented in Table 24. For the "Ontology" strategy, the Table shows three examples of developers' comments, which state that this

Table 23 – Developers' Quantitative Answers

ID	*Softgoal*	*Impact on User Interaction Quality* *Break*	*Hurt*	*Unkown*	*Help*	*Make*
1	Mechanisms based on rules	0	1	0	12	2
2	Google Sign in	0	0	1	9	5
3	Facebook Sign in	0	0	1	9	5
4	Facial Recognition	0	4	0	10	1
5	Iris Recognition	0	3	3	7	1
6	Smart lock for passwords	0	0	2	11	1
7	OpenIoT	0	0	3	2	0
8	LoCCAM	0	0	4	9	0
9	Awareness API from Google	0	0	1	5	1
10	IoTivity	0	0	0	2	0
11	Arrowhead	0	0	0	1	0
12	Embed code without using infra	0	3	3	3	2
13	Key value pair	0	0	7	4	1
14	if then else	0	3	5	4	0
15	Ontology	0	3	7	3	0
16	Dempster-Shafer Theory	0	0	0	0	0
17	First Order Logic	0	1	7	0	0
18	Fuzzy Logic	0	1	7	3	0
19	SVM Algorithm	0	2	4	4	1
20	Neural Networks	0	1	5	8	1
21	MQTT	0	1	3	6	1
22	CoAP	0	0	5	4	0
23	OSGi	0	1	4	3	0
24	Speech API	0	0	0	12	2
25	OpenCV	0	1	3	4	2
26	Kinect	0	1	2	6	3
27	Tangible	0	0	1	7	3
28	Breath	0	0	0	2	1
29	Body	0	0	1	7	3
30	Haptic	0	1	0	7	1
31	Writing	1	2	2	7	0
32	Brain	0	0	6	2	0
33	Eyes	0	1	2	4	1
34	Arduino	0	3	3	9	0
35	Raspberry	0	1	4	10	0
36	Beaglebone	0	2	4	7	0
37	Philips Hue	0	0	3	10	1
38	Amazon Eco	0	1	0	8	1
39	Apple HomePod	0	0	0	6	1
40	Google home	0	0	1	9	1
41	Embed specific hardware	0	1	2	8	3
42	Hide technology	0	3	3	8	0
43	Not losing aesthetics	0	1	1	9	0
44	Place objects discretly	0	2	1	9	1

Source: Author.

Figure 61 – Predefined Codes used in TRACE Execution

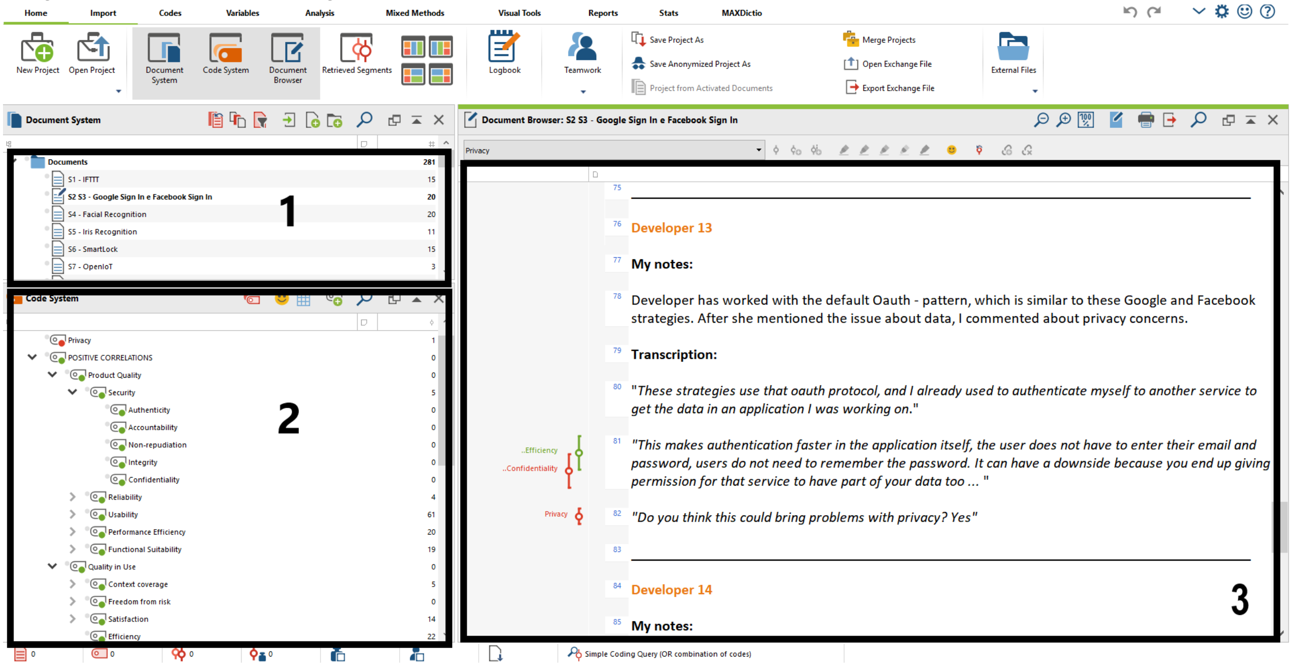


Source: Author.

strategy has a negative impact on performance. On the other hand, for Google's and Facebook login API's strategies, three examples point to a positive impact for Efficiency of the interaction.

In total, 329 codifications were performed, 161 of them had positive mentions (HELP correlations) of the quality characteristics and 168 had negative mentions (HURT correlations). Table 25 shows statistics between the kinds of characteristics and kinds of correlations. It is possible to see that there are more segments for product quality characteristics than for quality in use characteristics, in both positive and negative impacts. This number can show that developers

Figure 62 – Data Coding at MAXQDA tool



Source: Author

are more likely to point out information about quality of the product itself than quality of use issues, which is already foreseen by (ISO/IEC 25010, 2011). Furthermore, there are references to other quality characteristics only in respect to Negative Correlations, which were Maintainability, Cost and Privacy. This work did not use Maintainability and the Cost codification, because they are out of the scope of this thesis. Only Privacy codifications were considered. Even though it is not a characteristic defined at (ISO/IEC 25010, 2011), it is very relevant to user interaction quality.

Table 26 shows the amount of coded segments and the number of softgoals that each characteristic or subcharacteristic has correlations with. Also, Table 26 presents the kind of correlation in the first column. If it is positive, then the symbol (+) is used and colored green, if it is negative, then the symbol (-) is used and it is colored red.

In the first row, for example, the Security characteristic has five positive encoded segments in two softgoals, which means that five developers mentioning positive impacts of two softgoals in Security.

Furthermore, this table gives an overview of how characteristics and softgoals are mentioned by the developers. It is possible to see that regarding the help correlations, Efficiency has the most coded segments. Regarding the hurt correlations, Functional Correctness is pointed out as the most cited. Looking at the number of softgoals, it is possible to see that Accessibility's

Table 24 – Examples of Coded Segment Texts

Softgoal	Text Segments	Codification
Ontology	"...depending on the size of the system, the ontology grows very fast and can have an impact on performance, and this can hurt the user, they realize the delay" "impacts on the performance, ontology may slow the interaction" "...has a problem at runtime, when you need to infer some information on top of that knowledge, it becomes heavier than other approaches"	HURT / Product Quality / Performance Efficiency
Google Sign in and Facebook Log in APIs	"as you already have your account, you will not need to create another account, so it's one step less to do for the user and this becomes easier" "They make it easier to register, shorten the registration time" "...sometimes the application needs some information and when you log in with any of these API's, this information already comes to the application so the user does not need to put his/her name, age. Therefore, this task gets easier"	HELP / Quality in Use / Efficiency
LoCCAM	"LoCCAM asks permission for everything because of the CACs, the app can have access to any CAC even if the app can't have that permission." "I think it enters the same issue of data privacy because the user in LoCCAM does not have much control of what it is storing and where it will store ..."	HURT / Privacy
MQTT	"...it is a good performance implementation, whatever you build on it will make sure that it is not consuming all the way through." "...it is optimized... so for example if application one uses http and the other mqtt, the tendency is to use mqtt to be more efficient..."	HELP / Product Quality / Performance Efficiency

Source: Author.

Table 25 – Statistics between the Types of Characteristics and the Types of Correlations

	HELP CORRELATIONS		**HURT CORRELATIONS**	
	Segments	**Percentage**	**Segments**	**Percentage**
Product Quality Characteristics	108	67,1	109	64,9
Quality in Use Characteristics	53	32,9	27	16,1
Others	-	-	32	19,1
TOTAL	161	100,00	168	100,00

Source: Author.

is most cited among softgoals. There are twelve different softgoals impacting Accessibility positively. On the other hand, there are 8 softgoals impacting Privacy negatively.

Table 27 presents softgoals of each characteristic or subcharacteristic. For example, Security is impacted by Facial Recognition (S4) and Iris Recognition (S5) positively. The corresponding softgoals for each ID can be seen in Table 21.

Furthermore, each one of these impacts were directly mapped as correlation rules. However, while mapping these correlations, three softgoals (S4 - Facial Recognition, S12 -

Table 26 – Quantity of Codifications

Correlation	Characteristic	Subcharacteristic	# Coded Segments	#Softgoals
+	Security	-	5	2
+	Reliability	-	4	1
+	Usability	-	6	2
+	Usability	Accessibility	17	12
+	Usability	Operability	6	1
+	Usability	Appropriateness Recognizability	32	8
+	Performance Efficiency	-	14	5
+	Performance Efficiency	Resource Utilization	2	1
+	Performance Efficiency	Time Behavior	3	1
+	Functional Suitability	-	17	6
+	Functional Suitability	Functional Correctness	2	1
+	Context coverage	Flexibility	5	4
+	Satisfaction	-	12	5
+	Satisfaction	Trust	2	2
+	Efficiency	-	34	5
-	Security	-	9	8
-	Security	Authenticity	3	1
-	Security	Confidentiality	12	2
-	Reliability	Availability	1	1
-	Reliability	-	16	7
-	Usability	Accessibility	2	1
-	Usability	Operability	10	4
-	Usability	Learnability	7	4
-	Performance Efficiency	-	14	6
-	Performance Efficiency	Capacity	2	1
-	Performance Efficiency	Resource Utilization	1	1
-	Performance Efficiency	Time Behavior	2	2
-	Functional Suitability	Functional Correctness	30	8
-	Context coverage	Flexibility	9	3
-	Satisfaction	Comfort	4	1
-	Satisfaction	Trust	9	7
-	Efficiency	-	5	2
-	Privacy	-	18	9
-	Maintainability	-	7	3
-	Cost	-	7	1

Source: Author.

Embedded Code, S19 - SVM algorithm) presented four conflicts of correlations, which means there were negative and positive mentions to the same characteristic. The impact with most citations was selected to become a correlation rule. Then, the others were excluded. Besides, the positive impact was represented as “HELP”, while the negative one was represented as “HURT”. Other options in the scale of impact from (CHUNG *et al.*, 2000) were not used since most of developers thought they were very strong contributions (break and make), which can be seen in quantitative answers.

At the end of mapping, 120 correlation rules were defined. This number corresponds to the sum of numbers (128) in the last column of Table 26 minus four conflicting correlations and four correlations of Maintainability and Cost, since these last two were not the focus of this

Table 27 – Softgoals of each Characteristic or Subcharacteristic

Correlation	*NFR*	*Softgoals IDs*
+	Security	S4, S5
+	Reliability	S41
+	Usability	S30, S37
+	Accessibility	S4, S5, S24, S25, S26, S29, S30, S32, S33, S38, S39, S40
+	Operability	S1
+	Appropriateness Recognizability	S1, S24, S27, S31, S33, S38, S39, S40
+	Performance Efficiency	S21, S22, S13, S19, S20
+	Resource Utilization	S12
+	Time Behavior	S12
+	Functional Suitability	S7, S8, S9, S11, S12
+	Functional Correctness	S7, S8, S9, S11, S12
+	Flexibility	S18, S19, S20, S28
+	Satisfaction	S1, S26, S42, S43, S44
+	Trust	S4, S5
+	Efficiency	S2, S3, S6, S19, S20
-	Security	S6, S8, S24, S35, S36, S38, S39, S40
-	Authenticity	S4
-	Confidentiality	S2, S3
-	Availability	S8, S12, S14, S23, S34, S35, S36
-	Reliability	S12
-	Accessibility	S31
-	Operability	S6, S42, S43, S44
-	Learnability	S1, S27, S19, S20
-	Performance Efficiency	S8, S15, S17, S18, S19, S20
-	Capacity	S34
-	Resource Utilization	S12
-	Time Behavior	S4, S5
-	Functional Correctness	S5, S24, S25, S26, S27, S38, S39, S40
-	Flexibility	S1, S13, S14
-	Comfort	S30
-	Trust	S2, S3, S6, S8, S9, S38, S39, S40
-	Efficiency	S4, S5
-	Privacy	S2, S3, S8, S9, S38, S39, S40, S42
-	Maintainability	S12, S13, S14
-	Cost	S41

Source: Author.

thesis.

5.1.4.3 *Validating*

A validation of the Invisibility correlations has been made to obtain more reliable data. Unlike what is done in Step 2, when using the ARRANGE approach, in which each mapping between text and code was evaluated, the validation here occurred in the rules because the data analysis in this step generated 329 codifications, which can be quite costly.

Thus, the validation of the rules was done by experts. Each rule was evaluated using a scale: agree, partially agree, disagree. Even though the amount of data to be evaluated was smaller, the set had to be split between experts because the correlations rules refer to different topics and it was hard to find experts for all softgoals or characteristics. Thus, seven experts were consulted in total. Two criteria were defined to recruit them: they must have a D.Sc degree; and they should be from the area of the softgoal or the characteristic.

Table 28 presents the profile of the experts together with the quantity of correlation rules and the kind of softgoals they received to validate. For example, Expert 1 has worked with context-aware and mobile computing, therefore, rules regarding the softgoal "Adapt according to the context" were sent to Expert 1, 25 of them in total.

Table 28 – Profile of the Experts who evaluated the correlation rules

Expert	*Expertise Area*	*#Correlation Rules*	*Kind of Evaluated Softgoals*
1	Context-Aware Computing, Mobile Computing	25	Adapt according to the context
2	IoT, WSN,	2	Protocols
3	Security	24	User Access Authentication
4	Computer Vision, Digital Image Processing, Virtual Reality, Security	9	Usage of Natural Interfaces
5	HCI, End-user development	19	Minimal Interaction, Multimodal Interaction
6	IoT, Embedded systems	33	Invisibility from the usage point of view
7	Machine Learning, Data Science	8	Decide according to machine learning techniques

Source: Author.

Table 29 presents the results of their evaluations. Most correlations were agreed by the experts (94 in total - 78%), some of them were partially agreed (16 in total, 13%) and only ten correlation rules (8%) were disagreed and then excluded. Expert 3 was the one who had the highest rate regarding disagreement. Analyzing his/her evaluation about each disagreed rule, made it possible to see that Expert 3 took into account a different definition of Trust, characteristic present in 4 correlations. Expert 3 disagreed, even though the evaluation asked of him to take into account the definition from (ISO/IEC 25010, 2011). The correlation rules that were disagreed by the experts were excluded.

The rules with partially agree rates were analyzed to include some kind of condition. For example, the correlation rule: "SVM algorithm hurts Usability / Learnability" is stated because machine learning algorithms can impact users negatively when they are learning how to use a system since at the beginning of use (users may be confused as the system may not perform optimally). The expert who evaluated this rule agreed that this problem exists and it is

Table 29 – Agreement Rate for the Correlation Rules

Expert	*Total*	*Agree*	*Partially Agree*	*Disagree*
1	25	15	9	1
2	2	2	0	0
3	24	14	0	7
4	9	7	2	0
5	19	16	1	2
6	33	33	0	0
7	8	4	4	0

Source: Author.

called “cold start”, which means the system can take a while to infer correctly. However, the expert said that there are techniques to minimize the cold start problem. In this way, the rule was changed to include a condition: “SVM algorithm hurts Usability / Learnability when a technique for minimizing the cold start problem is not used”.

Finally, the correlations rules which were agreed by the experts were kept and did not change. At the end, 110 correlation rules were defined and then cataloged.

5.1.4.4 Reporting

These correlation rules can be viewed in the SIG itself or in a table. For clarity reasons, the rules in this thesis will be presented in a table. Next section presents all of them together with the description of each softgoal, thus comprising the proposed NFR Catalog of this thesis.

5.1.5 Threats to Validity

Many threats to validity appears in studies performed through interviews. This work considered the four categories of validity defined in (WOHLIN *et al.*, 2012), and used in (GHAZI; GLINZ, 2016). However, it is important to emphasize that in qualitative research, which is the case of this work, it is not possible to solve all threats (GHAZI; GLINZ, 2016).

Conclusion validity is concerned with issues that affect the ability to draw the correct conclusion. In the case of this work, that the correlations represent the conclusions from interviews, this threat was minimized by applying a systematic qualitative method, the Content Analysis. Also, while mapping the correlations, only 4 codifications out of 329 were considered conflicting to each other, indicating that there is a consistency among the opinion of the developers. Furthermore, all correlation rules generated from this study were validated by experts. When an expert disagreed with the correlations, then they were excluded. However, only 8% of the correlations were disagreed, indicating that the set of correlations was well defined.

Internal validity is related to how the participants (in this work, the interviewees) are selected, how they are treated and compensated during the study, if special events occur during the experiment, among other. In this work, the script remained the same for all participants during all the study. The interviews were performed in one month, which is a short time, then we could avoid advances in software or hardware. Furthermore, the participants (developers and experts) did not receive any compensation.

Construct validity ensures that interview actually ask what it is supposed to ask. All interviews were conducted by the author of this thesis. Moreover, the interview script was validated with an expert and a pilot study was performed to correct any issue. Furthermore, most of the participants were interviewed in the same environment, except the ones who were interviewed by video conference. Furthermore, it was made sure for them that the data would be used only for research purposes and that any material with their identity would not be disclosed.

External validity of a research means that the results are generalizable. For this validity, the criteria of this work was selecting participants as developers with experience. Their feedback was asked only when they knew or have worked with the softgoal in question. Furthermore, it is known that the number of interviewed developers may not be statistically significant. However, this number of interviewers (15) is enough to collect valuable information about correlations in this specific topic.

5.2 Invisibility Characteristic, Subcharacteristics, Strategies and Correlations

This section presents LEAD, the proposed NFR catalog for the Invisibility characteristic. Each softgoal is presented in a top-down way. First, the main NFR softgoal, Invisibility, is discussed in terms of a proposed definition based on and grounded in data. After that, each one of its sub softgoals (*i.e.*, subcharacteristics and strategies) and correlations are explained. Figure 59 presents the final SIG.

5.2.1 Invisibility Definition

The existing definitions of Invisibility are concerned mainly about the disappearance of technology to allow users to focus on everyday tasks. However, as previously discussed, there are still differences between them. Through the GT analysis, it was possible to identify two groups of concepts regarding Invisibility: physical environment and workload during the usage of systems. Therefore, Invisibility can be related not only to devices in the user physical environment, but also with the minimization of the interaction workload.

Then, based on and grounded in data, it was possible to propose the following definition: ***"Invisibility in UbiComp and IoT applications refers to either a merge of technology in user's physical environment or an interaction workload decrease, both aiming to provide a greater focus of the user on his everyday's tasks."***

Therefore, Invisibility is considered either from physical view or usage view, which are its two subcharacteristics: Invisibility from the usage point of view and Invisibility from the physical environment point of view. Each one of these subcharacteristics is further refined into more subcharacteristics and then into strategies that operationalize the subcharacteristic. Next subsection presents them.

5.2.2 Invisibility Subcharacteristics and strategies

Invisibility from the usage point of view refers to a decrease of workload of user's interaction with the system. The workload reduction can be achieved in two ways: reducing interactions or designing an interaction more natural for the user. Thus, it is further refined into two others: *(i)* Minimal Interaction or *(ii)* Natural Interaction.

Minimal Interaction refers to the system's ability to design tasks without them being entirely or constantly dependent on explicit user inputs. Minimal Interaction is supported by two other softgoals: Customizable by the User or Minimal User Involvement.

- **Customizable by the User** means that a system should, whenever possible, let the users make changes in it according to their personal preferences, minimizing future interactions (COSTA *et al.*, 2008) (SCHOLTZ; CONSOLVO, 2004). Table 30 presents two operationalizing softgoals (*i.e.*, strategies) that help this NFR softgoal: **set warnings** and **set actions**, both helped by rule-based mechanisms.
- **Minimal User Involvement** means a system should, whenever possible, require less user interaction by decreasing user inputs and actions. The following softgoals can help here: Minimize user's effort in tasks and Implicit Interaction.
  - *Minimize user's effort in tasks* is the ability to minimize user's effort in tasks that cannot be excluded from a system. User access authentication is a general opera-

tionalizing softgoal, suggested by the UbiComp/IoT developers to help "Minimize user's effort". Table 31 presents its specific softgoals.

– *Implicit Interaction* is the ability to interact with the user without his/her explicit command or awareness (GINER *et al.*, 2011). The general operationalizing softgoal used to help Implicit Interaction calls "Adapt according to context", which means an ability of a system to adapt in response to contextual information. Table 32 presents its specific operationalizing softgoals.

Table 30 – Operationalizing "Customizable by The User"

General Operationalizing Softgoals	Specific Operationalizing Softgoals
Set warnings: the system can allow user to set warnings according to their preferences. **Set actions**: the system can allow user to set actions either in the system itself or in the physical environment.	**Rule-based mechanisms**: mechanisms that can be implemented using rules such as If-This-Then-That (IFTTT).

Source: Author.

Natural Interaction: it refers to supporting more natural and expressively powerful means of interaction by using natural interfaces and letting the user switch between modes of interaction (YUE *et al.*, 2007). This support will significantly reduce the workload, since users are used to natural communication in their everyday lives. Two softgoals that can help natural interaction are:

- Multimodal Interaction: the system should support user alternating modes and switch modalities as needed during the changing conditions (YUE *et al.*, 2007).
- Usage of Natural Interfaces: This softgoal helps a system to achieve a more natural communication by using interfaces that are natural for a person (YUE *et al.*, 2007)

Table 31 – Operationalizing "Minimize user's effort in tasks"

General Op. Softgoals	*Specific Operationalizing Softgoals*
User Access Authentication - an ability to minimize user's effort to access a system	**Integration with third-party services**: using data from existing accounts minimize the need to type personal data to sign in, which can be implemented with **Google Sign-in API**[5] or **Facebook API**[6] **Biometric techniques**: automated techniques for identifying a human being based on physiological or behavioral characteristics. For example, **face** or **iris recognition**. **Login only once between devices**: technique used to save passwords, so the user does not have to remember them when he/she enters the same service on a different device. One solution is the **Smart Lock for Passwords API from Google**[7]

Source: Author.

Table 32 – Operationalizing “Adapt according to context”

Specific Op. Softgoals	*Specific Operationalizing Softgoals*
1. Monitoring: collecting data from sensors	**1.1 Usage of Infrastructure**: Sensor data can be collected through context management infrastructures, which encapsulate data access: **Middleware** (*e.g.*, **LoCCAM** (MAIA *et al.*, 2013) or **OpenIoT**[14]) **API** (*e.g.*, **Awareness from Google**[15]) **Framework** (*e.g.*, **IoTivity** [16] or **Arrowhead**[17]) **1.2 Embedded code directly in the app**: data collection can be performed inside the application code, without any supporting infrastructure.
2. Deciding: once the data is collected, there are two types of techniques to identify the current context situation: **specification-based** or **machine learning-based techniques** (YE *et al.*, 2012)	**2.1 Specification-based techniques**: they are based on a priori expert knowledge (YE *et al.*, 2012). Examples of these techniques are: **Key-value pair**, **If-then-else**, **Ontology**, **Dempster-Shafer Theory**, **First-Order Logic** or **Fuzzy Logic** **2.2 Machine learning-based techniques**: they allow learning complex associations between situations and sensor data (YE *et al.*, 2012). It is implemented by continuous learning with an **SVM algorithm** or a **Neural Network**, for example.
3. Acting: once a situation is detected, the action is automatically performed on the system.	**3.1 Protocols**: the interaction with things can be performed through communication protocols, such as **MQTT**[18] or **CoAP**[19].

Source: Author.

Table 33 – Operationalizing “Usage of Natural Interfaces” and ‘Multimodal Interaction”

General Op. Softgoals	*Specific Operationalizing Softgoals*
Usage of Natural Interfaces: the system should support more natural human forms of communication, which are referred to as natural interfaces	**Speech** is the ability to interact by voice. **Google Speech API**[24] is a solution to achieve this softgoal. This API converts speech into text and vice-versa. **Gestures/Body/Eyes** are other examples of natural interfaces. They can be implemented by **OpenCV**[25], a library that provides an infrastructure to computer vision applications, or **Kinect**[26], a programming toolkit that includes rich APIs for raw sensor streams and natural user interfaces. Other interfaces are: **Tangible**, **Breathing**, **Haptic**, **Writing**, **Brain**
Framework to manage interface modules: an application must be modular, allowing the management of each module (*e.g.*, speech, writing, gestures, etc.) at runtime.	**OSGi**[27]: framework to manage a life cycle of dynamic components

Source: Author.

(KAASINEN *et al.*, 2013). Besides helping the Natural Interaction softgoal, it also helps Multimodal Interaction by supporting different modalities.

Table 33 presents the operationalizing softgoals for Usage of Natural Interfaces and Multimodal Interaction.

Invisibility from the physical environment point of view refers to the merging of the technological infrastructure in the physical space to ubiquitously support their users (KARAISKOS *et al.*, 2009). That way, this softgoal is helped by three others: Embeddedness in everyday objects, Diffusion in the physical environment and Unobtrusiveness.

Embeddedness in everyday objects refers to augmenting everyday objects with sensors, processing and communication without compromising the primary functions (BEIGL *et al.*, 1998). There are three strategies (general operationalizing softgoals) to embed computational power into objects: (i) using generic platforms capable of embedding sensors and actuators, such as Arduino[28], Raspberry PI[29] or Beaglebone[30]; (ii) using devices already equipped with sensors and communication, such as Philip Hues[31], Amazon Echo[32], Apple HomePod[33] or Google Home[34]; or (iii) embedding specific hardware directly into objects, which requires specific knowledge about microelectronics.

Diffusion in the physical environment: The computational resources (*e.g.* data collection devices, communication technologies) should be diffused in the physical environment in order to promote effective use of services, without compromising physical space of the user (BEIGL *et al.*, 1998). Two NFR softgoals help in achieving it: *(a)* Hiding technology in the physical space and *(b)* Not losing Aesthetics. Hiding technology in the physical space is about hiding the hardware infrastructure (*e.g.*, data collection devices, communication devices) from the user in a way that does not catch his/her attention. Not losing Aesthetics is related to the physical appearance of the space, which should not be changed.

Unobtrusiveness: the hardware resources should not be conspicuous or attracting attention. Even with little weight, knowing that a device is present in an environment increases the risk of invading user's personal space, causing discomfort (MORAN; NAKATA, 2010). Therefore, an operationalizing softgoal that can help it is Placing the objects discreetly in the physical space. If it is not possible to entirely hide hardware devices, they should be placed discreetly in the physical area. Therefore, places where the user does not need to perform actions, such as wall corners and roofs, would be ideal. Another operationalizing softgoal that can help Unobtrusiveness is usage of existing connected devices from popular manufactures such as Philip Hues, since they already are specifically designed with sensors and actuators.

---

[28] https://www.arduino.cc
[29] https://www.raspberrypi.org/
[30] http://beagleboard.org/bone
[31] https://www.philips.co.in/c-m-li/hue
[32] https://www.amazon.com/
[33] https://www.apple.com/homepod/
[34] https://store.google.com/gb/product/google$_{h}ome$

5.2.3 *Invisibility Correlations*

This subsection presents all correlation rules[35] generated from TRACE in Tables 34 and 35. In total, there are 51 positive and 59 negative correlations. When the ID comes together with a *, it means that a correlation rule contains a condition. Therefore, 19 rules have conditions, presented as follows:

- Facial Recognition hurts Performance / Time Behavior *when the system needs to search through a large database*
- Iris Recognition hurts Performance / Time Behavior *when the system needs to search through a large database*
- OpenIoT helps Functional Suitability *when it does not require adaptations of requirements and features originally planned for the system*
- LoCCAM helps Functional Suitability *when it does not require adaptations of requirements and features originally planned for the system*
- Awareness helps Functional Suitability *when it does not require adaptations of requirements and features originally planned for the system*
- IoTivity helps Functional Suitability *when it does not require adaptations of requirements and features originally planned for the system*
- Arrowhead helps Functional Suitability *when it does not require adaptations of requirements and features originally planned for the system*
- Embed code helps Functional Suitability *when developers are experienced*
- Embed code helps Time Behavior *when data is not previously available*
- Embed code helps Resource Utilization *when an application runs in a device that is not likely to have many apps running*
- Ontology hurts Performance Efficiency *when data set is large*
- SVM algorithm hurts Usability / Learnability *when no technique to minimize the cold start problem is used*
- Neural Network hurts Usability / Learnability *when no technique to minimize the cold start problem is used*
- OSGi hurts Reliability *when an app is simple*
- OSGi hurts Performance Efficiency *when an app is simple*

[35] These correlations can also be seen in the Invisibility SIG at https://github.com/great-ufc/LEAD/blob/master/Correlations.jpg

- OpenCV hurts Functional Correctness *when developers are not experienced*
- Kinect helps Satisfaction *when interactions are short*
- Tangible helps Appropriateness Recognizability *when there are good materials and digital mapping*
- Writing helps Appropriateness Recognizability *when used in some professional contexts*

5.2.4 Discussion

The main goal of this chapter is to answer RQ3 -*To what extent does a specific HCI quality characteristic from UbiComp and IoT impact user interaction quality characteristics?*, which can be modified to *To what extent does Invisibility from UbiComp and IoT impact on user interaction quality characteristics?*, since Invisibility was the characteristic selected to be investigated.

Figure 63 presents an overview of the correlations from Invisibility to the User Quality Characteristics. In summary, Invisibility correlates with Security, Reliability, Usability, Performance Efficiency, Functional Suitability, Context Coverage, Satisfaction, Efficiency and Privacy. The HELP correlations are presented in the upper part of the graph and colored green. The HURT correlations are presented in the upside down part of the graph and colored red.

Looking at the upper part, it is possible to see that Invisibility has more positive impact on Usability, where 22 correlations are positively related to this characteristic, followed by Performance (7), Functional Suitability (7), Satisfaction (5), Efficiency (5) and Context Coverage (4). Invisibility has only one softgoal impacting Security and Reliability.

The positive correlations with Usability appeared mostly in its subcharacteristics such as Accessibility (11) and Appropriateness Recognizability (8). They are strongly related to softgoals that gives another possibility of interaction for a user, such as: facial recognition, iris recognition, speech API, OpenCV, Kinect, haptic, brain, eyes, Amazon Echo, Apple Homepod and Google Home. Therefore, when developers use natural interfaces and minimize the user's effort, they are helping more users to access the system and they recognize these attempts as suitable.

Positive correlations with Performance are more related to the strategies of deciding how to adapt to the context. When techniques of machine learning are used, they are likely to help Performance. Additionally, specific protocols for the Internet of Things, such as MQTT and CoAP, are more likely to help Performance.

Table 34 – Correlations Rules - Part 1/2

ID	Strategy	Type	Quality Characteristic
1	Rule-based mechanisms	HELPS	Usability/ Appropriateness Recognizability
2	Rule-based mechanisms	HELPS	Usability/Operability
3	Rule-based mechanisms	HELPS	Satisfaction
4	Rule-based mechanisms	HURTS	Usability / Learnability
5	Google Sign-in	HELPS	Efficiency
6	Google Sign-in	HURTS	Privacy
7	Google Sign-in	HURTS	Security / Confidentiality
8	Facebook Log-in	HELPS	Efficiency
9	Facebook Log-in	HURTS	Privacy
10	Facebook Log-in	HURTS	Security / Confidentiality
11	Facial Recognition	HELPS	Usability / Accessibility
12	Facial Recognition	HURTS	Functional Suitability / Functional Correctness
13	Facial Recognition	HURTS	Privacy
14*	Facial Recognition	HURTS	Performance Efficiency / Time Behavior
15	Facial Recognition	HURTS	Efficiency
16	Facial Recognition	HURTS	Security / Authenticity
17	Iris Recognition	HELPS	Security
18	Iris Recognition	HELPS	Usability/Accessibility
19*	Iris Recognition	HURTS	Performance Efficiency / Time Behavior
20	Iris Recognition	HURTS	Efficiency
21	SmartLock	HELPS	Efficiency
22*	OpenIoT	HELPS	Functional Suitability
23*	LoCCAM	HELPS	Functional Suitability
24	LoCCAM	HURTS	Privacy
25	LoCCAM	HURTS	Performance Efficiency
26	LoCCAM	HURTS	Security
27	LoCCAM	HURTS	Reliability
28*	Awareness	HELPS	Functional Suitability
29	Awareness	HURTS	Privacy
30*	IoTivity	HELPS	Functional Suitability
31*	Arrowhead	HELPS	Functional Suitability
32*	Embed code	HELPS	Functional Suitability
33*	Embed code	HELPS	Performance Efficiency / Time Behavior
34*	Embed code	HELPS	Performance Efficiency / Resource Utilization
35	Embed code	HURTS	Reliability
36	Embed code	HURTS	Reliability / Availability
37	Key-value pair	HELPS	Performance Efficiency
38	if then else	HURTS	Context Coverage / Flexibility
39	if then else	HURTS	Reliability
40*	Ontology	HURTS	Performance Efficiency
41	First order logic	HURTS	Performance Efficiency
42	Fuzzy logic	HELPS	Context Coverage / Flexibility
43	Fuzzy logic	HURTS	Performance Efficiency
44	SVM algorithm	HELPS	Efficiency
45	SVM algorithm	HELPS	Performance Efficiency
46	SVM algorithm	HELPS	Context Coverage / Flexibility
47*	SVM algorithm	HURTS	Usability / Learnability
48	Neural Network	HELPS	Efficiency
49	Neural Network	HELPS	Performance Efficiency
50	Neural Network	HELPS	Context Coverage / Flexibility
51*	Neural Network	HURTS	Usability / Learnability
52	MQTT	HELPS	Performance Efficiency
53	CoAP	HELPS	Performance Efficiency
54*	OSGi	HURTS	Reliability
55*	OSGi	HURTS	Performance Efficiency

Source: Author.

Table 35 – Correlations Rules - Part 2/2

ID	Strategy	Type	Quality Characteristic
56	SpeechAPI	HELPS	Usability/ Appropriateness Recognizability
57	SpeechAPI	HELPS	Usability / Accessibility
58	SpeechAPI	HURTS	Security
59	SpeechAPI	HURTS	Functional Suitability / Functional Correctness
60	OpenCV	HELPS	Usability / Accessibility
61*	OpenCV	HURTS	Functional Suitability / Functional Correctness
62*	Kinect	HELPS	Satisfaction
63	Kinect	HELPS	Usability / Accessibility
64	Kinect	HURTS	Functional Suitability / Functional Correctness
65*	Tangible	HELPS	Usability/ Appropriateness Recognizability
66	Tangible	HURTS	Functional Suitability / Functional Correctness
67	Tangible	HURTS	Usability / Learnability
68	Breath	HELPS	Context Coverage / Flexibility
69	Haptic	HELPS	Usability / Accessibility
70	Haptic	HELPS	Usability
71	Haptic	HURTS	Satisfaction / Comfort
72*	Writing	HELPS	Usability/ Appropriateness Recognizability
73	Writing	HURTS	Usability / Accessibility
74	Writing	HURTS	Efficiency
75	Brain	HELPS	Usability / Accessibility
76	Eyes	HELPS	Usability / Accessibility
77	Eyes	HELPS	Usability/ Appropriateness Recognizability
78	Arduino	HURTS	Reliability
79	Arduino	HURTS	Performance efficiency / capacity
80	Raspberry	HURTS	Reliability
81	Raspberry	HURTS	Security
82	Beaglebone	HURTS	Reliability
83	Beaglebone	HURTS	Security
84	Philips Hue	HELPS	Usability
85	Amazon Eco	HELPS	Usability / Accessibility
86	Amazon Eco	HELPS	Usability / Appropriateness Recognizibility
87	Amazon Eco	HURTS	Privacy
88	Amazon Eco	HURTS	Satisfaction / Trust
89	Amazon Eco	HURTS	Security
90	Amazon Eco	HURTS	Functional Suitability / Functional Correctness
91	Apple HomePod	HELPS	Usability / Accessibility
92	Apple HomePod	HELPS	Usability / Appropriateness Recognizibility
93	Apple HomePod	HURTS	Privacy
94	Apple HomePod	HURTS	Satisfaction / Trust
95	Apple HomePod	HURTS	Security
96	Apple HomePod	HURTS	Functional Suitability / Functional Correctness
97	GoogleHome	HELPS	Usability / Accessibility
98	GoogleHome	HELPS	Usability / Appropriateness Recognizibility
99	GoogleHome	HURTS	Privacy
100	GoogleHome	HURTS	Satisfaction / Trust
101	GoogleHome	HURTS	Security
102	GoogleHome	HURTS	Functional Suitability / Functional Correctness
103	Embed hardware	HELPS	Reliability
104	Hide Technology	HELPS	Satisfaction
105	Hide Technology	HURTS	Usability / Operability
106	Hide Technology	HURTS	Privacy
107	Not losing aestheics	HELPS	Satisfaction
108	Not losing aestheics	HURTS	Usability / Operability
109	Place objets discret	HELPS	Satisfaction
110	Place objets discret	HURTS	Usability / Operability

Source: Author.

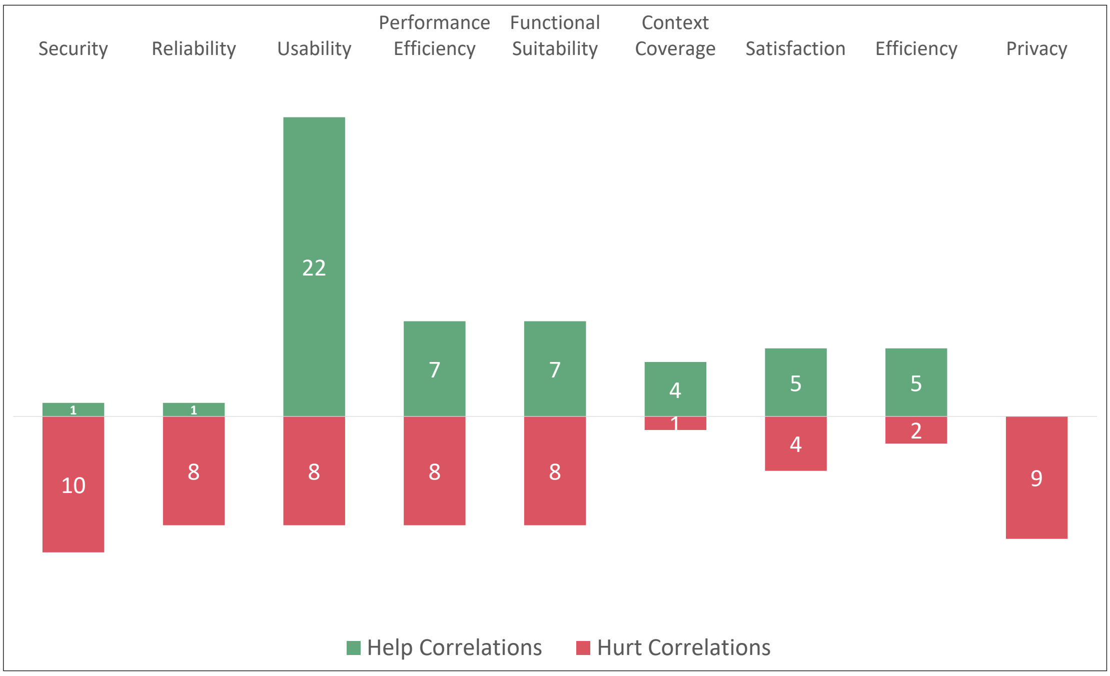
Regarding Functional Suitability, strategies with the purpose of monitoring context usually help in a degree to which a system provides functions that meet stated and implied needs when used under specified conditions (definition of Functional Suitability according to (ISO/IEC 25010, 2011)). Indeed, context monitoring allows an application to know user's possible needs, even if they have not even been explicit.

Satisfaction is positively impacted by strategies that mask technology from user's eyes (hiding technology, not losing aesthetics, place objects discreetly) . This fact is explained because users are concerned with the appearance of things, specially things that will change their house, which is what IoT systems can do.

Efficiency is positively impacted when strategies are used to minimize user's effort. They are related to strategies to user authentication, such as Google Sign in API, Facebook Login API and Smart Lock, and also to strategies that learn the behavior of the user, such as SVM algorithm and Neural Network.

The positive correlations with Context Coverage appear in Flexibility, its subcharac-

Figure 63 – Overview of the correlations from Invisibility to the User Interaction Quality Characteristics



Source: Author.

teristic, which is a degree to which a product or system can be used in contexts beyond those initially specified in the requirements. Therefore, strategies regarding continuous learning are likely to help this characteristic.

Finally, only one positive correlation appears with Security and Reliability. Iris Recognition helps Security since it is much more difficult to cheat, being an advantage for Security. Regarding Reliability, a positive correlation appears when developers use specific hardware sensors and actuators with a system being developed, avoiding failures from the general sensor platforms.

Looking to the upside down of the graph in Figure 63, it is possible to see that Invisibility has a more negative impact on Security, where 10 correlations are negatively related to this characteristic, followed by Privacy (8), Reliability (8), Performance (8), Usability(8), Functional Suitability (8). Satisfaction presents 4 negative correlations, while Efficiency and Context Coverage are the characteristics with less negative correlations, 2 and 1, respectively.

The negative correlations with Security are related to softgoals that gives another alternative to authentication, such as: Google Sign in API, Facebook Login API, Facial Recognition. Indeed, a recent study (WIJAYARATHNA; ARACHCHILAGE, 2019) showed that the security of applications using Google's authentication API would depend on how programmers are using it. They pointed out different ways for a programmer to develop an application that is not secure. This way, these APIs can adversely affect the confidentiality of personal data.

Moreover, regarding Facial Recognition, many algorithms are still subject to spoofing attack, in which a photo can be used in place of the user (OH *et al.*, 2019). This way, authenticity is impaired.

Privacy is not a characteristic defined in (ISO/IEC 25010, 2011), however, it appeared 18 times in interviews. Therefore, this work decided to also consider Privacy as a characteristic related to user interaction quality. Regarding correlations to it, all of them are negative and mostly related to softgoals which somehow take the user's control over their data:

- Google Sign in API, Facebook Login API - through these APIs it is possible to collect personal data of the users, which imposes privacy concerns;
- Facial Recognition - a study revealed that there is a growing concern about privacy due to possible sharing of images (OH *et al.*, 2019);
- LoCCAM - this middleware requests all permissions in order to work. As other applications call it as a service, then they no longer need these permissions to work, and this can lead

to a security and privacy problem;

- Awareness API - this type of API collects sensitive information from the user, which may harm their privacy.
- Amazon Echo, Google Home, Apple HomePod - such devices can collect, record and save user conversations on the server.
- Hide technology - it can bring harm to Privacy because by being hidden, the user may not know what is being collected or if something is being collected.

Reliability is mostly negatively impacted when generic platforms of sensors and actuators are used. Platforms such as Arduino, Raspberry and BeagleBone should not be used in the final product. The reliability is very low due to their fragility.

Performance mostly appears when strategies such as Ontology, First Order Logic and Fuzzy Logic are used. Reasoning on ontology models is resource-intensive and is not suitable to real-time knowledge representation when the number of entities is large. To infer some information over a knowledge modeled with first order or fuzzy logic is heavier than logic “if then else”. The possibilities grow too much and until the system can make a decision. It can be a long time for the final user.

Although it appears as the characteristic with most positive correlations, Usability also appeared to be negatively impacted by Invisibility. However, the negative correlations appears in subcharacteristics such as Operability and Learnability. Every time a system masks something from a users, the user may loose control, which is related to Operability. Therefore, softgoals such as Hide technology, Not losing aesthetics and Place discreetly impact negatively Operability. Regarding Learnability, softgoals such as IFTTT, SVM algorithm and Neural Network are the reasons that this characteristic is impacted. IFTTT, as a mechanism based on rules, can at first bring a difficulty to a user who learns how to use a system with this type of interaction. Also, machine learning techniques can hurt a user when he/she is learning to use an application. At the beginning of usage, users may be confused because a system may not exhibit optimal behavior.

Functional suitability is negatively impacted regarding its subcharacteristic Functional Correctness, which is a degree to which a product or system provides correct results with a needed degree of precision. Many strategies used in UbiComp and IoT systems are not 100% precise, still presenting errors for the users. However they are maturing with time and investments, some of which are Amazon Echo, Google Home, Apple HomePod, Tangible interfaces, Kinect, OpenCV, Speech API, Facial Recognition etc.

Satisfaction is mostly impacted in its Trust subcharacteristics. Strategies such as Amazon Echo, Google Home, Apple HomePod, in which personal conversations can be recorded, may not be trusted by users.

As minor correlations, Efficiency is negatively impacted by Facial and Iris Recognition. Because they require that a user authenticates himself in a costly way, *i.e.*, he can not be in movement. Instead he needs to bring the phone to the face, which can cost more time and cause greater annoyance, thus affecting efficiency.

Context coverage is impacted in Flexibility subcharacteristic by If-then-else strategy to adapt to the context. This technique does not support reasoning, thus new context information or situations will not be considered.

It is possible to see in Figure 63 that Invisibility have have both positive and negative correlations with the same characteristics. What can differentiate this interaction is the subcharacteristic or the development strategy. For Example, in Usability, while Invisibility helps Appropriateness Recognizability, it does not hurt this subcharacteristic. However, there are also sucharacteristics impacted positively and negatively. For instance, Invisibility has positive and negative correlations with the subcharacteristic Operability and what differentiates them is a development strategy.

Despite this, the positive relationship of Invisibility with Usability is greater than their negative relationship. Thus, in general, Invisibility converges positively with Usability. On the other hand, Security is on the opposite side. Invisibility has a greater negative relationship with Security, followed by Privacy.

Some characteristics appear with the same intensity in both relationships (positive and negative), which is a case of Performance Functional Suitability and Satisfaction. More investigations are necessary to see how they could differentiate to each other.

5.3 Chapter Summary

This Chapter presented the LEAD Catalog, which is the proof of concept (PoC) of the CORRELATE process and its supporting instruments and approaches[36], then addressing RQ3 -*To what extent does one specific HCI quality characteristic from UbiComp and IoT impact on user interaction quality?*.

[36] All materials and results of LEAD are available at https://github.com/great-ufc/LEAD

The first step of the process is to select a quality characteristic, then, in this PoC, the characteristic from AMICCaS was chosen to start the use of the cataloging process. Through a questionnaire-based instrument, where twenty-one HCI experts answered it, Invisibility was the characteristic most likely to have negative impacts with Usability. Therefore, this characteristic was selected to be investigated regarding other user interaction quality characteristic, giving evidence to answer RQ3.

The second step of the CORRELATE process is related to refining the chosen characteristic in the previous step, which was Invisibility. ARRANGE was used to perform this refinement, resulting in a SIG for Invisibility containing 2 subcharacteristics, 12 sub subcharacteristics, 3 general strategies and 14 specific strategies.

The third step aims to collect specific strategies to satisfy the subcharacteristics that were defined in the previous step. Through a questionnaire-based instrument, where seven UbiComp and IoT developers gave anwsers to, seven general and forty-two specific development strategies were obtained.

The fourth and last step is to define correlations by using the softgoals defined in the previous step. TRACE was used, consisting in semi structured interviews with fifteen developers. In total, 110 correlation rules were defined. They made it possible to understand the extent to which Invisibility impacts on user interaction quality. On the one hand, Usability was the most positively impacted characteristic. On the other hand, Security was most negatively impacted.

Furthermore, in Section 5.2, this chapter presented all the knowledge captured, analyzed and cataloged in LEAD, a well-defined NFR catalog. Also, a discussion about how Invisibility impacts on the user interaction quality characteristics was presented.

The next chapter presents an evaluation of LEAD regarding its correlations. The goal is to verify whether by using the proposed correlation catalog, novice requirements engineers can make better decisions regarding the NFRs of a UbiComp and IoT system.

6 EVALUATION OF THE LEAD CATALOG

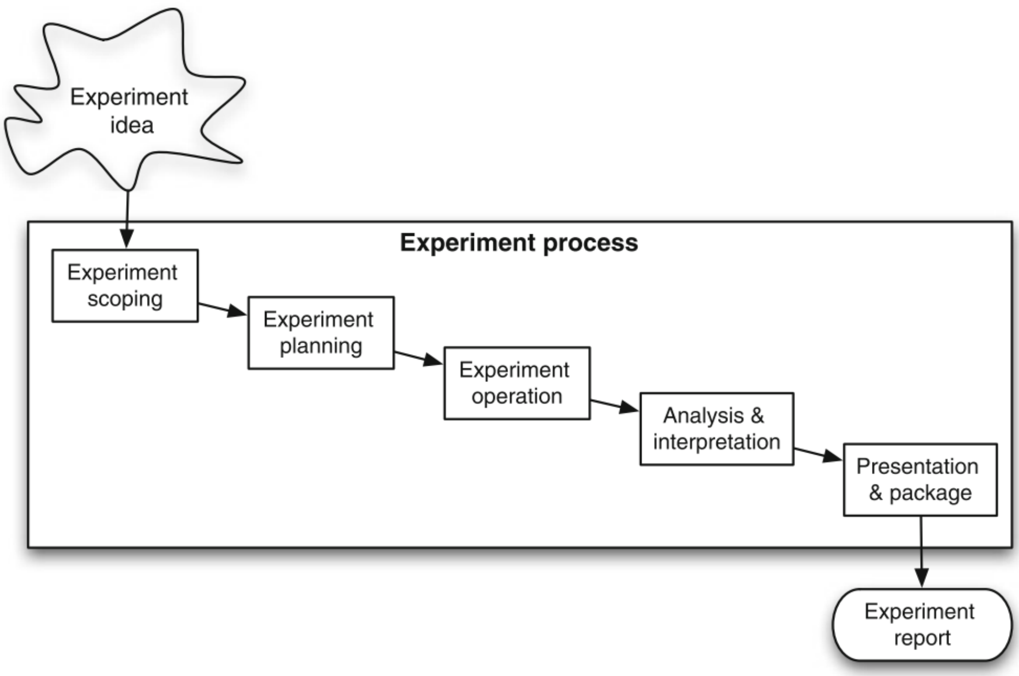
This Chapter describes the evaluation of LEAD, the proposed correlations catalog, which is performed using a controlled experiment and answers *RQ4 - Does an NFR catalog improve decisions regarding NFRs in UbiComp and IoT systems?*

In short, Section 6.1 gives an overview of the process used to perform the controlled experiment. Then, Sections 6.2 and 6.3 present the scope and planning of this experiment, respectively. Section 6.4 describes the experiment operation. Section 6.5 shows the analysis and interpretation of the results. Finally, Section 6.6 presents a discussion of these results.

6.1 Overview of the Controlled Experiment

The controlled experiment follows the process proposed by (WOHLIN *et al.*, 2012). Figure 64 shows an overview of the process.

Figure 64 – Overview of the Experiment Process



Source: (WOHLIN *et al.*, 2012)

The starting point of an experiment is that researchers have an idea of the cause and effect relationship. In the case of this work, the idea is that the usage of NFRs catalogs help in making better and faster design decisions compared with not using them.

In this thesis, better decisions mean choosing a set of strategies that has most benefit (maximize) and least sacrifice (minimize) to the required quality characteristics. Therefore, it is important to emphasize that decisions are always made towards to improve the NFRs, since it is known that it is hard or even impossible to satisfy NFRs completely (WIEGERS; BEATTY, 2013). Furthermore, faster decisions mean spending less time when using the correlation catalog, since the knowledge is already captured and cataloged[1]. This idea is especially valid for IoT and UbiComp systems, because they have relatively new development strategies and, as a consequence, new positive and negative impacts not known by novice requirements engineers.

6.2 Scoping

According to (WOHLIN *et al.*, 2012), the scope of the experiment is set by defining its goal, suggesting the template proposed by (SOLINGEN *et al.*, 2002) to describe it. In this template, the goal should have the following information: object of study, purpose, attribute of evaluation. Table 36 presents the goal of the experiment executed in this thesis.

Table 36 – Experiment Goal

Analyse	the usage of the correlations catalog
for the purpose of	characterizing
with respect to	efficacy, efficiency and satisfaction
from the point of view of the	researcher
in the context of	novice requirements engineers making decisions regarding what strategies of Invisibility should be used

Source: Author.

Therefore, three research questions (Controlled Experiment Research Question (CE-RQ)) are defined as follows.

- CE-RQ1: Is the set of selected strategies suitable to maximize the positive impact and minimize the negative impact of the required NFRs when the catalog is used? This question aims to evaluate the attribute “efficacy”, which means, in this work, checking if the proposed catalog helps in making better decisions than the participant’s own experience.
- CE-RQ2: Is the time spent to make decisions towards NFRs lower when the correlations catalog is used? The goal of this question is to assess the attribute “efficiency”, which means, in this work, checking if the participants spent more time using the catalog or not using the catalog.

[1] It is important to emphasize that this experiment differs from the one in (CYSNEIROS, 2007), where the author expected the time would be higher when using a catalog since the participants would have to model the entire SIG for a system.

- CE-RQ3: Will the participants in the role of requirements engineers feel more satisfied with using a catalog compared to when they are not using it? This question investigates the attribute "satisfaction", which means, in this work, checking their opinions regarding the usage of the catalog.

6.3 Planning

After the scope has been defined, the planning phase starts. In this phase, the following topics should be defined: context selection; variables, factor and treatments selection; hypothesis formulation; subjects selection, tasks and objects; design type definition; and instrumentation (WOHLIN *et al.*, 2012). They are described as follows.

6.3.1 Context Selection

In the context selection, the types of subjects and projects should be defined (WOHLIN *et al.*, 2012). Subjects are the people who participate in the experiment. There are two types of subjects: students or practitioners (WOHLIN *et al.*, 2012). In this work, students were chosen due to their availability to perform the experiment and also due to the fact that they were attending in the university a course for requirement engineers. The subjects perform the experiment tasks in projects, which can be real or toy (WOHLIN *et al.*, 2012). In this work, two real projects that have been developed at Group of Computer Network, Software Engineer and Systems (GREat) were selected to the experiment.

6.3.2 Variables, Factor and Treatment Selection

A controlled experiment is performed to study the effect of changing some of the input variables. Thus, there are two kinds of variables in an experiment: independent and dependent variables. All variables that are manipulated and controlled are called independent variables. The variables that show the effect of the changes are called dependent variables.

This experiment needs to study the effect of using a correlation catalog on the decisions of a requirements engineer. Therefore, one of the independent variables is the usage of the correlations catalog, because this study wants to manipulate if the participant uses or does not use catalogs to make decisions. Also, since the subjects are intentionally selected to have a similar background, the background experience is another independent variable, and then this will not affect the outcome.

The independent variable that researchers manipulate is called factor and treatment is one particular value of a factor. In this work, the factor is the catalog and the treatments are using the catalog and not using the catalog.

Then, the dependent variables need to be set to test the effect of changing the treatments (using or not using a correlation catalog). They are related to the factors defined in the goal: efficacy, efficiency and satisfaction. These factors are not directly measurable, which makes it necessary to measure it via an indirect measure instead.

For Efficacy, this study aims to see if the catalog supports better decisions than when the catalog is not used. As mentioned before, decisions in this work mean the participant selecting strategies that maximize positive effects and minimize negative effects in the required NFRs. These strategies are represented as operationalizing softgoals in a SIG.

Therefore, to measure Efficacy in this work, the confusion matrix from machine learning area was used as inspiration (SOKOLOVA; LAPALME, 2009) since it contains measures about how effective is a classification model, which can be mapped to the scenario of this work. Figure 65) presents these measures:

- True Positive (TP), which means the model predicted positive and it is true;
- True Negative (TN), which means the model predicted negative and it is true;
- False Positive (FP), which means the model predicted positive and it is false; and
- False Negative (FN), which means the model predicted negative and it is false.

Figure 65 – Confusion Matrix

Class \ Recognized	as Positive	as Negative
Positive	tp	fn
Negative	fp	tn

Source: (SOKOLOVA; LAPALME, 2009)

Therefore, these measures were then mapped to the scenario of this work, which is the selection of operationalizing softgoals that maximize the positive effect and minimize negative effects. Then, True Positive in this work can be read as “the participant selected an operationalizing softgoal that must be selected, because it has only positive impacts in one of the required NFRs” and True Negative in this work can be read as “the participant did not select an operationalizing softgoal that must not be selected, because it has only negative impacts in one of the required NFRs". Therefore, the following measures are defined to evaluate Efficacy.

- True Positive (TP), which refers to the percentage of operationalizations that they must choose, because it has a positive effect on some of the required NFRs. This is measured by the Equation 6.1.

$$\mathbf{TP} = \frac{\#ChosenOperationalizations}{\#PositiveOperationalizations} \tag{6.1}$$

- True Negative (TN), which refers to the percentage of operationalizations that they should not choose because it has a negative effect on some of the required NFR. This is measured by the Equation 6.2.

$$\mathbf{TN} = \frac{\#NotChosenOperationalizations}{\#NegativeOperationalizations} \tag{6.2}$$

For Efficiency, this study evaluates if the catalog supports better decisions in a faster way than when it is not used. Therefore, the following measure is defined in Equation 6.3.

$$\mathbf{TS} = Time\ Spent\ in\ minutes\ to\ complete\ the\ tasks \tag{6.3}$$

For the quality focus Satisfaction, this study evaluates if by using the catalog, the participants feel more satisfied. However, Satisfaction is hard to be measured, since it may be impacted by several other factors (GHAZI; GLINZ, 2018). In this work, Satisfaction was evaluated regarding the following statements: 1. I easily identified the impacts; 2. I quickly identified the impacts; 3. I easily made my decision; and 4. I quickly made my decision. Therefore, the participants were asked to rate through Likert Scale how much they agreed with a set of statements regarding these feelings. These statements and other statements are better explained in subsection 6.5.4.

6.3.3 Hypothesis Formulation

There are two types of hypothesis: null and alternative (WOHLIN *et al.*, 2012). The null hypothesis H_0 states that there are no statistical difference between the treatments. This is the hypothesis that the researcher wants to reject. The alternative hypothesis H_1 is used in case the null hypothesis is rejected.

In this thesis, there are three null and alternative hypothesis related to the three research questions for this experiment, defined as follows.

Null hypothesis related to **CE-RQ1**:
$H_{0-CE-RQ1}: TPwithCatalog = TPwithoutCatalog \wedge TNwithCatalog = TNwithoutCatalog$

This hypothesis means that using the correlation catalog results on the same values of True Positive (TP) and True Negative (TN) compared to not using the catalog.

Alternative hypothesis related to **CE-RQ1**:
$H_{1-CE-RQ1}: TPwithCatalog \neq TPwithoutCatalog \wedge TNwithCatalog \neq TNwithoutCatalog$

This hypothesis means that using the correlation catalog results on different values of True Positive (TP) and True Negative (TN) compared to not using the catalog.

Null hypothesis related to **CE-RQ2**:
$H_{0-CE-RQ2}: TSwithCatalog = TSwithoutCatalog$

This hypothesis means that using the correlation catalog results on the same value of Time Spent (TS) compared to not using the catalog.

Alternative hypothesis related to **CE-RQ2**:
$H_{1-CE-RQ2}: TSwithCatalog \neq TSwithoutCatalog$

This hypothesis means that using the correlation catalog results on different values of Time Spent (TS) compared to not using the catalog.

Null hypothesis related to **CE-RQ3**:
$H_{0-CE-RQ3}: SatisfactionWithCatalog = SatisfactionWithoutCatalog$

This hypothesis means that using the correlation catalog results on the same feeling of Satisfaction compared to not using the catalog.

Alternative hypothesis related to **CE-RQ3**:
$H_{1-CE-RQ3}: SatisfactionWithCatalog \neq SatisfactionWithoutCatalog$

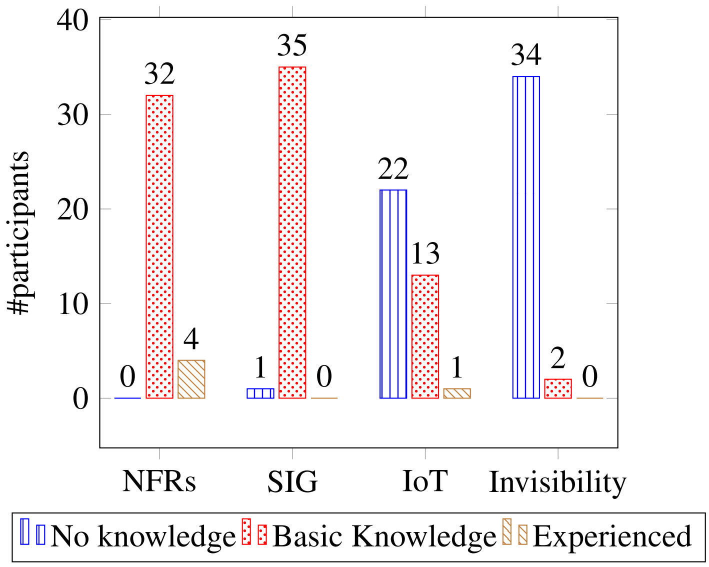
This hypothesis means that using the correlation catalog results on a different feeling of Satisfaction compared to not using the catalog.

6.3.4 Subjects Selection

In this work, the population is the requirements engineer. Then, to select subjects for this population, the convenience sampling (WOHLIN *et al.*, 2012), which is a non-probability sampling technique, was used. In this technique, the nearest and most convenient people are selected as subjects. In this way, 44 undergraduate students from the Requirements Engineering course at Federal University of Ceará, Campus Quixadá, were invited to participate. They were the nearest and most convenient people, since they had classes about NFRs; trade-offs between NFRs; and Softgoal Interdependency Graphs. Furthermore, they are considered to be novice in IoT and UbiComp systems. In total, 36 of them participated in the experiment.

Figure 66 presents the profile of the students that have participated in the experiment. Most of them (32) had basic knowledge about NFRs, obtained from the course they were enrolled. Regarding SIGs, 35 had basic knowledge, where only one stated that he/she had no knowledge. This student in particular missed one of the classes about SIGs, which was a practical class. However, he/she participated in the class about the theory of SIGs. Regarding IoT and UbiComp concepts, a lot of them (22) had no knowledge, which was previously expected. 13 of them had basic knowledge and only 1 was experienced. Finally, regarding the Invisibility characteristic, the majority had no knowledge about it and only 2 had basic knowledge.

Figure 66 – Profile of the Experiment Participants



Source: Author.

6.3.5 *Tasks and Objects Definition*

In experiments, the subjects need to perform tasks in objects, so then the experimenter can collect data about the variables. In this work, the tasks were based on the purpose of a correlation catalog: making decisions regarding operationalizations in a Softgoal Interdependency Graph for a specific system and its NFRs.

The selected objects for this experiment were two UbiComp and IoT systems called

AutomaGREat (Object 1) (ANDRADE *et al.*, 2017) and GREatBus[2] (Object 2). AutomaGREat (Object 1) is an application that proposes an intelligent environment for the Seminar Room of the GREat, a research lab. GREatBus (Object 2) is an application created to propose an intelligent system for passengers and bus drivers. In general, this system aims to facilitate the tasks related to the usage of buses.

Then, two SIGs were defined for these systems. These SIGs were an instance of the Invisibility catalog developed in this work. They are presented in Sections C.3 and C.5 of Appendix C. Also, a set of NFRs, including Invisibility for both applications, were defined.

For AutomaGREat, besides Invisibility, the NFRs are Security, Performance, Efficiency and Reliability. Security was defined because AutomaGREat is related to controlling objects of the building, therefore, only authorized persons should do this. Performance and Reliability were defined because users do not want to wait too much to control everyday objects or do not expect failures to happen when doing such controlling. Finally, Efficiency was defined for this application due to its goal, which is facilitating the controlling of objects as much as possible.

For GREatBus, besides Invisibility, two NFRs were defined: Accessibility and Privacy. The first was selected because the application should be used for a wide range of users. The second one was selected because this application deals with sensitive data of users, *e.g.*, location data.

In this way, the participant received a SIG model for these systems, with the description of the softgoals and the NFRs, and based on them they had to perform two tasks, described as follows.

- Task 1: Given a set of operationalizations in the last level of the SIG, analyze if they have positive and negative impact with the required NFRs for the system.
- Task 2: Choose the operationalizations that maximize the positive impact and minimize the negative impact to the required NFRs.

Task 1 was defined because it was important to guarantee that all participants, whether they used a catalog or not, would reason about the positive and negative impacts with the required NFRs.

[2] GREatBus is a system in development, part of the project called Smart Bus Stop, accepted by the Institutional Program of Innovation Scholarships at Federal University of Ceará

6.3.6 Design Type Definition

In this work, the design type is composed of one factor (the usage of the correlation catalog) with two treatments: (T1) With the correlation catalog and (T2) Without the catalog. Table 37 presents the experiment design used.

Table 37 – Experiment Design Type

Object	Control Group	Group 1	Group 2
1. AutomaGREat	T2	T2	T1
2. GREatBus	T2	T1	T2

Source: Author.

To perform the tasks of the experiments, the subjects were randomly divided in three groups: Control Group, Group 1 and Group 2. In the Control Group, the subjects performed the tasks in both objects without the correlation catalog (T2). In Group 1, the subjects also performed the tasks in both objects, but first they performed tasks in object AutomaGREat with treatment 2 - not using the catalog. Then they performed the tasks in the object GREatBus with treatment 1 - using the catalog to support them. In Group 2, the subjects started the tasks in AutomaGREat object with treatment 1 - using the catalog. Then they performed tasks in Object 2 with treatment 2 - not using the catalog.

This type of design uses the same objects for both treatments and assigns the subjects randomly to each treatment (WOHLIN *et al.*, 2012). The Control Group was added to increase the reliability of the hypothesis tests, since it is a group that receives only one treatment in both objects, which is T2.

Additionally, when the groups performed tasks in the second object, they received a portion of the SIG different from what they received in the first object. This strategy minimizes the possibility of memorizing the correlations, which is important to Group 2 since the participants started with the catalog and then they could not used it anymore.

6.3.7 Instrumentation

The instruments for an experiment are of three types, namely objects, guidelines and measurement instruments (WOHLIN *et al.*, 2012). The instruments in this study were:

- Consent Term (See Section C.1 in Appendix C);
- Background Form (See Section C.2 in Appendix C);

- Slides with training on UbiComp and IoT systems; and Invisibility concepts;
- Slides with training on the experiment tasks;
- Sheets for each object being used in the experiment, containing a description of the system, functional requirements, non-functional requirements, SIG of Invisibility to the system, a description of the operationalizations, a description of the tasks and the forms to get the starting and finishing time for tasks in each object and the selected operationalizations. Most of these documents can be seen at Appendix C; and
- Post-experiment questionnaire (See Section C.7 in Appendix C).

6.4 Operation

Once the planning is finished, the operation phase could take place, which means the data will be collected (WOHLIN, 2014). This phase comprises three steps: *(i)* Preparation, in which the material is prepared and the subjects are invited; and *(ii)* Execution, in which the subjects perform the experiment tasks and the data is collected.

6.4.1 Preparation

In this step, the participants were selected and informed. Therefore, all participants were informed in advance that the class would be replaced by the experiment. They were free to participate or not to participate.

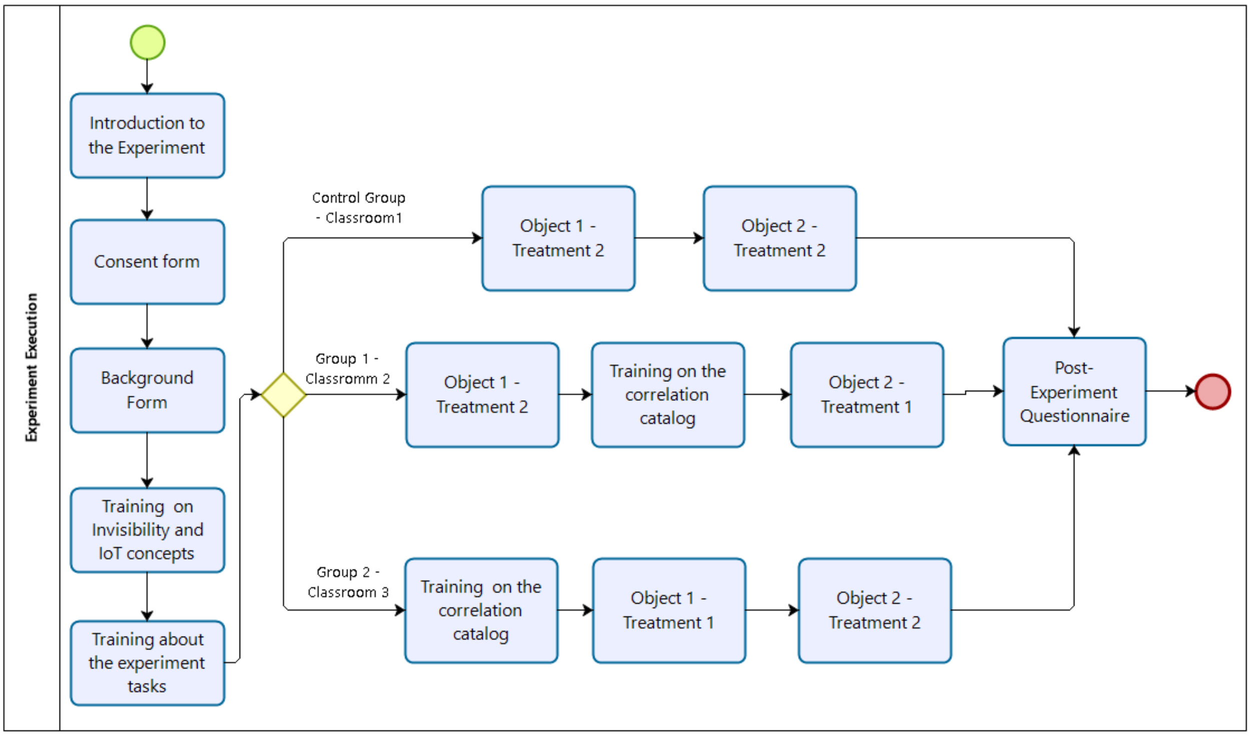
Also in this step, the material such as forms and tools should be prepared. All the instruments defined in subsection 6.3.7 were prepared. Also, three rooms were booked in order to execute the tasks in different rooms for all three groups. In this case, three undergraduate students acted as assistants to help the experimenter.

Additionally, in this step, two pilot studies were performed. The first one was executed by a D.Sc student, who performed the tasks in one object. Through this specific pilot test, several improvements in the training, forms and tasks were suggested. The second pilot test was performed with a D.Sc professor, who executed the tasks in both objects and made minor suggestions.

6.4.2 Execution

With all materials prepared, the experiment can take place, which happened at one day in this work. Figure 67 presents how was the execution, which was based on the design type established in this experiment and also based on the experiment performed in (SANTOS, 2018). The difference of the execution performed in this work to the work in (SANTOS, 2018) is the usage of an additional group, which is the Control Group.

Figure 67 – Overview of the Experiment Execution

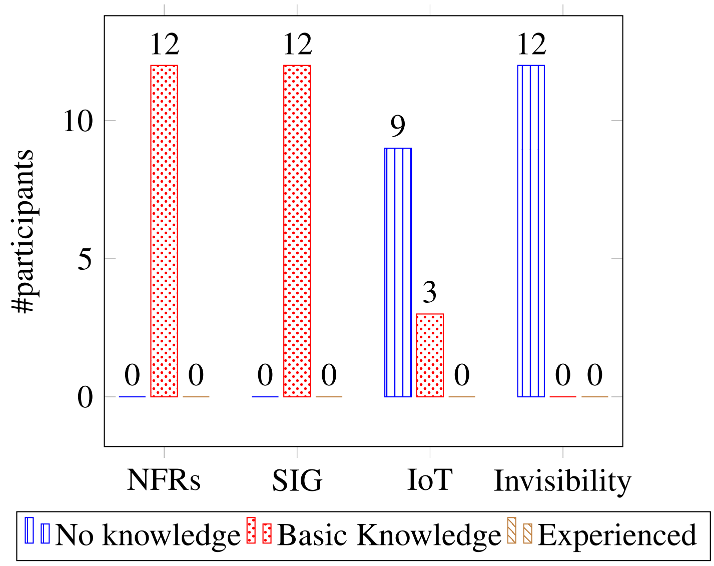


Source: Author. Based on (SANTOS, 2018).

First, an introduction about the experiment was done to the students and then their consent was asked and registered. After that, a background form was applied to get information about their experience. Then, the trainings took place. First, a training about IoT and UbiComp systems was done. After that, a training was performed to explain the tasks the students would be asked to do. In this training, the concepts about NFRs, SIGs and tradeoffs were revisited. These trainings were executed in the same classroom with all students to avoid different explanations for them.

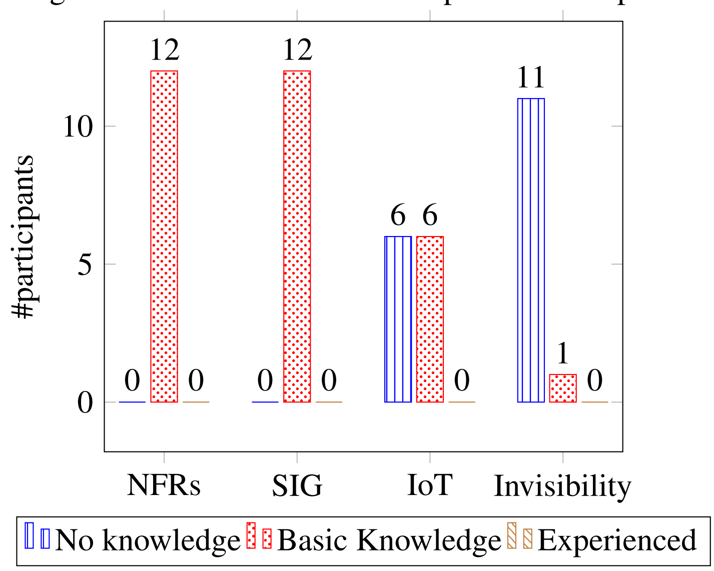
After the explanations, the subjects were randomly divided into three groups (Control Group, Group 1 and Group 2), keeping a balanced number of subjects (12) in each group, which is one the principles of experimentation according to (WOHLIN *et al.*, 2012). The distribution of participants by group also kept a similar background among groups, as it is possible to see in Figures 68, 69 and 70.

Figure 68 – Profile of the Participants in the Control Group



Source: Author.

Figure 69 – Profile of the Participants in Group 1

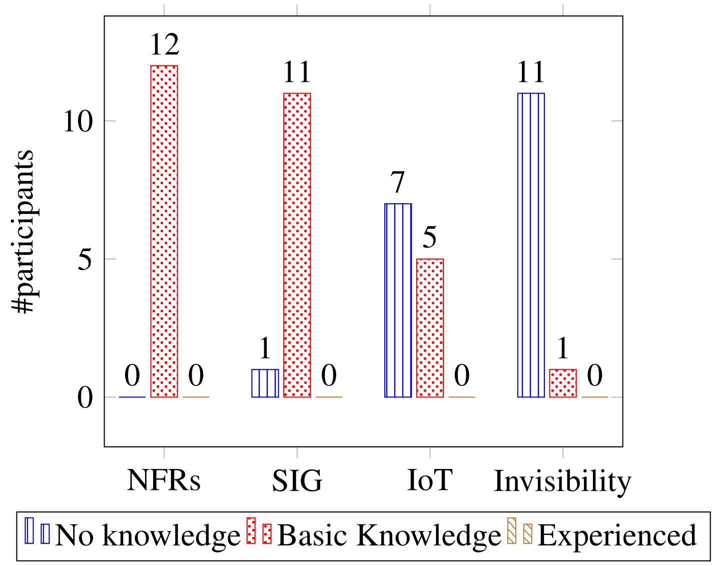


Source: Author.

Also, the groups went to three different classrooms, so that the experimenter could perform the training without interrupt the participants who do not need to watch the training.

The subjects of Group 1 performed the tasks in AutomaGREat (Object 1 - Section C.3) without the catalog. After all of the participants had finished the tasks in Object 1, they

Figure 70 – Profile of the Participants in Group 2



Source: Author.

received a training about the correlation catalog. Then, they started to do the same tasks, but in GREatBus (Object 2 - Section C.5) with the correlation catalog (See Table 55 in Appendix C).

The subjects from Group 2 first received the training about the correlation catalog and performed the tasks in Object 1 (AutomaGREat) with the support of the correlation catalog (See Table 54 in Appendix C). Then, they performed the tasks in Object 2 (GREatBus) without the catalog.

The subjects from Control Group performed tasks in both objects (AutomaGREat and GREatBus) without using the catalog. This group is important to give more reliability to the results through hypothesis tests. With it, more combinations to perform tests can be done.

After finishing the tasks in each object, all subjects filled out the Post-Task Questionnaire. This form consisted of questions to analyze their satisfaction regarding the tasks in that object. The set of questions is presented in Table 46, in which the participants should use the five-point Likert items to rate their feelings: Strongly Agree, Partially Agree, Neither Agree nor Disagree, Partially Disagree and Strongly Disagree.

Finally, after finishing the experiment, all subjects were asked to fill out the Post-Experiment Questionnaire (See Section C.7).

6.5 Analysis and Interpretation of the Results

After collecting experimental data in the operation phase, the conclusions can be drawn during the Analysis and Interpretation (WOHLIN *et al.*, 2012), as presented in Figure 64.

In this work, the analysis and interpretation are performed by discussing the measures and the hypothesis. Therefore, next subsection explains how the analysis was performed in this work. Then, the results for each attribute (Efficacy, Efficiency and Satisfaction) are presented.

6.5.1 Overview of the Analysis and Interpretation

Following the suggestion in (WOHLIN *et al.*, 2012), the analysis and interpretation of data is performed in two steps. First, the data is characterized by using descriptive statistics, which is the numerical processing of the data. In this work, the data from Efficacy and Efficiency was mostly characterized by the mean, median, mode and standard deviation, while data from Satisfaction was characterized by median and mode, since the data is represented in a ordinal scale (WOHLIN *et al.*, 2012).

Second, the data is analyzed by hypothesis testing, where the null hypothesis are evaluated statistically, in a given level of significance. In summary, the hypothesis testing calculates the p-value. This value represents the probability that the effect or difference between two or more samples is not because of the treatments. In this work, the level of significance was set to 0,05, which is a common value used in statistical analysis. Therefore, if the p-value is higher than 0,05, this work does not reject the null hypothesis.

There are several different statistical tests that can be used to evaluate the experiment data (WOHLIN *et al.*, 2012). They can be classified into parametric tests and non-parametric tests. Parametric tests are based on a model that involves a specific distribution, while non-parametric tests do not make the same type of assumptions concerning the distribution of parameters. Therefore, before testing the hypothesis, it is necessary to verify if the data is normally distributed through a normality testing.

To verify the data distribution, the Shapiro-Wilk test was used (RAZALI *et al.*, 2011) in all normality tests executed in this work. This test was used in this work because it is suitable for samples sizes in the range $3 \leq n \leq 5000$, which is the case of this experiment. Like the hypothesis testing, Shapiro-Wilk also calculates a p-value. If p-value > 0,05, the data is considered normal, therefore, a parametric test should be used, if not, a non-parametric test should be used.

In this work, when was necessary to use a non-parametric test, the Mann-Whitney is used because it is suitable to the type of design of this experiment (one factor, two treatments, completely randomized design). When was necessary to use a parametric test, the T-Test was used, because of the same reason.

Furthermore, normality tests were performed only for Efficacy and Efficiency data. The data of Satisfaction are not suitable to the normality tests because they are in ordinal scale. Therefore, a non-parametric test should be used. However, a suitable test to evaluate Likert answers is the Cronbach's alfa test. This test evaluates the reliability of the obtained answers (GLIEM; GLIEM, 2003)

To do all the normality testing, the hypothesis testing and Cronbach testing in this thesis, a software package for statistical analysis called SPSS IBM[3] was used in this work. This tool contains several kinds of tests and it is possible to perform all of these tests.

6.5.2 CE-RQ1: Efficacy

The research question related to efficacy is *"Is the set of operationalizations suitable to minimize the negative impact and maximize the positive impact of the required NFRs when the catalog is used?"* The measures used to answer this question was True Positive and True Negative, described in Subsection 6.3.2. To calculate the measures True Positive and True Negative, the answers of the participants were compared to a set of predefined answers that was established based on the correlation catalog. Next subsection presents the descriptive statistics, then, the normality and hypothesis testings is presented.

6.5.2.1 Descriptive Statistics

All raw data to draw conclusions about CE-RQ1 is presented in Table 38 and the descriptive statistics is presented in Table 39.

Regarding to the TP measure in Object 1 (AutomaGREat), it is possible to see that the mean of the group who was not using the catalog (Group 1 - 0,58) was smaller than the mean of the group who was using the catalog (Group 2 - 0,97). The mean of the Control Group was close to the mean of Group 1 (both did not use the catalog). Median and mode were also higher for Group 2 (1). Figure 71 presents an overview of the three groups regarding TP measure in Object 1, where it is possible to see that Group 2 was much better than the other groups, only participant 15 did not achieve the highest value.

Regarding the TN measure also in Object 1 (AutomaGREat), it is possible to see again that the group who did the best was Group 2, who was using the catalog. This group

[3] https://www.ibm.com/analytics/spss-statistics-software

Table 38 – Raw Data to answer CE-RQ1

		Object 1		Object 2	
Subject	**Group**	**True Positive**	**True Negative**	**True Positive**	**True Negative**
1	1	0,33	0,43	1	1
2	1	0,67	0,86	0	0,86
3	1	0,67	0,71	1	1
4	1	0,33	0,43	1	1
5	1	0,33	0,71	1	1
6	1	1,00	1,00	1	1
7	1	0,67	0,86	1	0,33
8	1	0,33	0,71	0	0,67
9	1	1	1	1	1
10	1	0,33	0,43	1	0,77
11	1	0,67	0,57	1	0,33
12	1	0,67	0,71	1	0,67
13	2	1	1	0	0,33
14	2	1	1	1	0,66
15	2	0,67	1	0	0,67
16	2	1	1	1	0,66
17	2	1	1	0	0,67
18	2	1	0,86	0	0,67
19	2	1	0,71	0	0,33
20	2	1	1	0	0,33
21	2	1	1	0	0,33
22	2	1	1	0	0
23	2	1	0,86	1	0,67
24	2	1	1	1	1
25	C	0,67	0,71	0	0,67
26	C	0,67	0,57	1	0
27	C	0,33	0,57	0	0,67
28	C	0,67	0,71	0	0,33
29	C	0,33	0,86	0	0,67
30	C	1	0,43	0	0,67
31	C	0,33	0,71	0	0,33
32	C	0,67	0,71	0	0
33	C	0,67	0,43	0	0,33
34	C	0,33	0,86	1	0,67
35	C	0	0,71	0	0,67
36	C	0,67	0,71	0	0,67

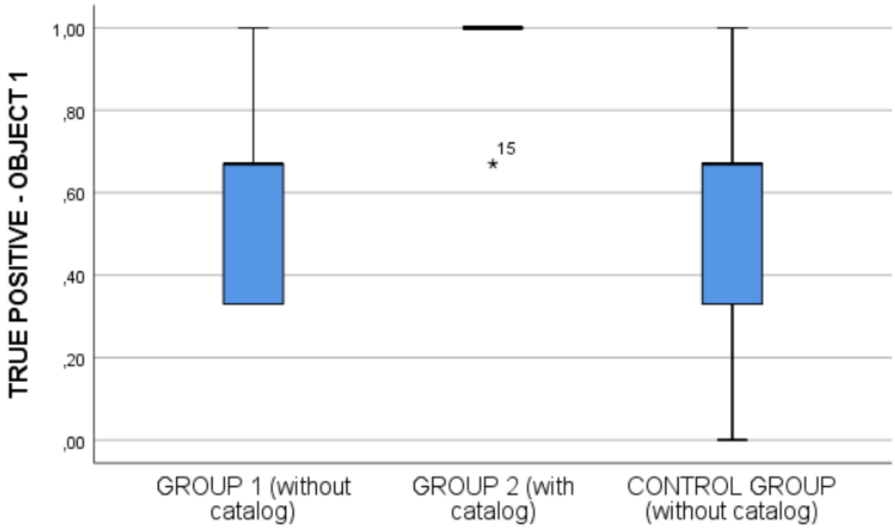
Source: Author.

Table 39 – Descriptive Statistics to answer CE-RQ1

	Group	**Mean**	**Median**	**Mode**	**St. Dev.**
TP_OBJECT1	1	0,58	0,67	0,33	0,25
	2	0,97	1	1	0,09
	C	0,52	0,67	0,67	0,26
TN_OBJECT1	1	0,7	0,71	0,71	0,20
	2	0,95	1	1	0,09
	C	0,66	0,71	0,71	0,14
TP_OBJECT2	1	0,83	1	1	0,38
	2	0,33	0	0	0,49
	C	0,16	0	0	0,38
TN_OBJECT2	1	0,8	0,93	1	0,25
	2	0,52	0,66	0,33	0,26
	C	0,47	0,67	0,67	0,26

Source: Author.

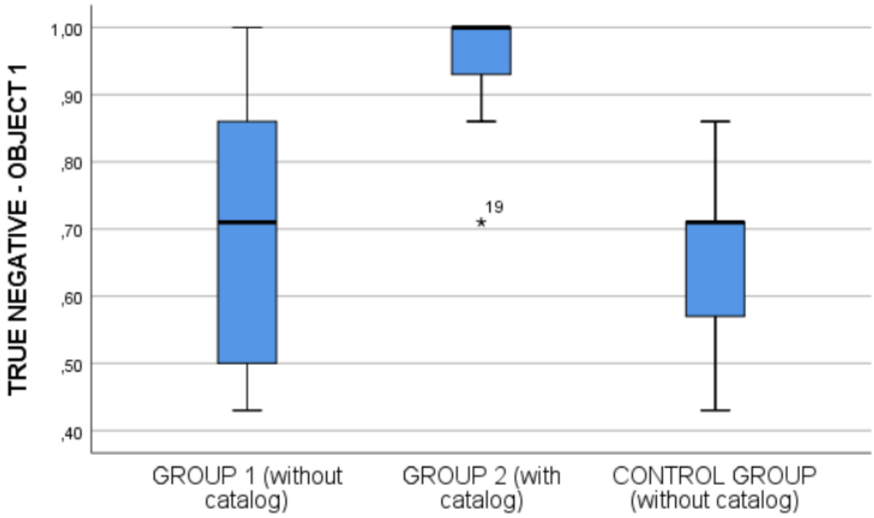
Figure 71 – Boxplot - True Positive in Object 1 (AutomaGREat)



Source: Author

obtained a mean of 0,95 while Group 1 and Control Group reached 0,7 and 0,66, respectively. Median and mode were also higher for Group 2 (1). Figure 72 presents an overview of the three groups regarding TN measure in Object 1, where it is possible to see that Group 2 was much better than the other groups.

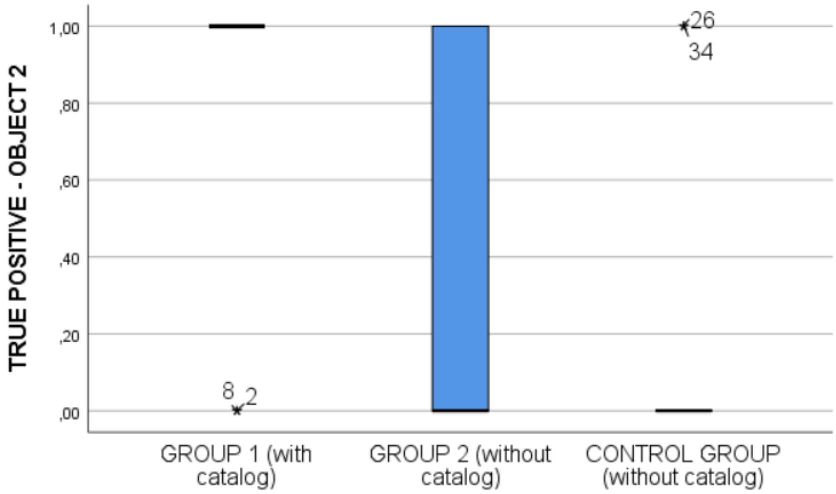
Figure 72 – Boxplot - True Negative in Object 1 (AutomaGREat)



Source: Author

Regarding the TP measure in Object 2 (GREatBus), the mean of the group who was not using the catalog (Group 2 - 0,33) was lower than the mean of the group who was using the catalog (Group 1 - 0,83). The mean of the Control Group was even lower than Group 1 (Control Group - 0,16). The median and mode of Group 1 (1 and 1 respectively) was greater than the median and mode of Group 2 (0 and 0) and Control Group (0 and 0). Figure 73 presents an overview of the three groups regarding TP measure in Object 2, where it is possible to see that Group 1 was much better than the other groups, except for participants 8 and 2.

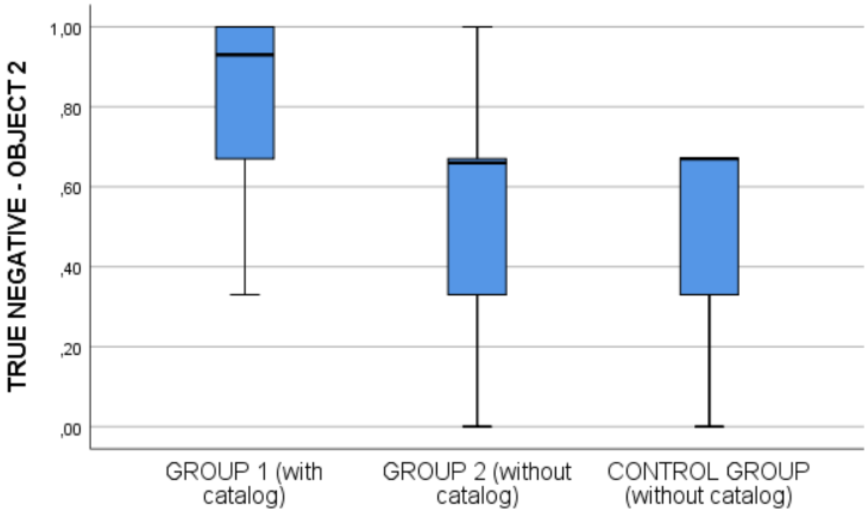
Figure 73 – Boxplot - True Positive in Object 2 (GREatBus)



Source: Author

Regarding the TN measure in Object 2 (GREatBus), the mean of the group who was using the catalog (Group 1 - 0,8) was higher than the average of the groups who did not use the catalog (Group 2 - 0,52, group of control - 0,47). The median and mode also are higher than Group 2 and Control Group. Figure 74 presents an overview of the three groups regarding TN measure in Object 1. Group 1 had better results than the other groups.

Figure 74 – Boxplot - True Negative in Object 2 (GREatBus)



Source: Author

Thus, it is notable that the use of the catalog resulted in better answers by the participants. Only three participants (Subject 5, 9 and 24) had the highest result for both situations (using or not using the catalog). Taking a closer look at these participants, it was possible to see that two of them had basic experience with IoT and UbiComp systems.

6.5.2.2 Normality and Hypothesis Testing

To test the null hypothesis of Efficacy, defined in Equation 6.4, both measures (True Positive - TP and True Negative - TN) should have significant differences (p<0,05) in both objects (AutomaGREat - OBJECT1 and GREatBus - OBJECT2).

$$H_{0-CE-RQ1} : TPwithCatalog = TPwithoutCatalog \wedge TNwithCatalog = TNwithoutCatalog \tag{6.4}$$

As the design of this experiment includes three groups using both objects, the tests should be executed for comparing three set of groups: *(i)* Group 1 and Group 2; *(ii)* Control Group and Group 1; and *(iii)* Control Group and Group 2. These three combinations of groups should be analyzed in each measure (TP and TN) and each object (AutomaGREat - OBJECT1 and GREatBus - OBJECT2). In this way, twelve hypothesis tests should be executed.

To select a suitable hypothesis tests, a normality test should be done, which was the Shapiro-Wilk, as mentioned before. The normality test also should be done for each combination of groups, measures and objects, due to the fact that each combination can suggest the need for different kinds of hypothesis tests.

Table 40 presents the results of the normality tests for CE-RQ1. The Shapiro-Wilk test indicated that the dataset for the three combinations of groups and for all measures (TP in Object 1, TN in Object 1, TP in Object 2 and TN in Object 2) do not follow a normal distribution. All results are near 0,000. They would follow a normal distribution if p > 0,05. Then, it is necessary to use a non-parametric test.

Table 40 – Shapiro-Wilk tests for CE-RQ1

Measures / Samples	Group 1 and Group 2	Control Group and Group 1	Control Group and Group 2
TP_OBJECT1	0,000	0,003	0,003
TN_OBJECT1	0,000	0,019	0,019
TP_OBJECT2	0,000	0,000	0,000
TN_OBJECT2	0,006	0,007	0,007

Source: Author.

The Mann-Whitney test is a non-parametric test used in this work because it is suggested by (WOHLIN *et al.*, 2012) for experiments with one factor, two treatments and randomized design, which is the case of this work. Table 41 presents the results.

Table 41 – Mann-Whitney tests for CE-RQ1

Measures / Samples	Group 1 and Group 2	Control Group and Group 1	Control Group and Group 2
TP_OBJECT1	0,000	0,755	0,000
TN_OBJECT1	0,002	0,671	0,000
TP_OBJECT2	0,039	0,005	0,514
TN_OBJECT2	0,012	0,005	0,977

Source: Author.

Regarding measure TP in Object 1 (TP_OBEJCT1) at groups 1 and 2, the p-value was 0,000, which is bellow 0,05. Therefore, the p-value indicates a statistically significant difference between the groups, while noting that Group 1 did not use the catalog and Group 2 used it.

Regarding measure TP in Object 1 (TP_OBEJCT1) at Control Group and Group 1, the p-value was 0,755. This result indicates that there is no statistically significant difference between the groups. Noting that both Control Group and Group 1 did not use the catalog in Object 1 . Therefore, nothing could indicate a difference between them. It is interesting to note that because it strengthens the conclusion that the only difference between groups is the usage of the catalog.

Regarding measure TP in Object 1 (TP_OBEJCT1) at Control Group and Group 2, the p-value was 0,000, which is bellow 0,05. Therefore, the p-value indicates a statistically significant difference between the groups. Noting that Group 2 used the catalog and Control Group did not use it.

Following the same pattern, the measure TN in Object 1 (TN_OBEJCT1) at Group 1 and Group 2, the p-value (0,002) was bellow 0,05, characterizing a difference when a group used the catalog and the other one did not use it. Also, the p-value was above 0,05 when both groups did not use the catalog (Control Group and Group 1 - 0,671). Furthermore, TN in Object 1 at Control Group and Group 2 obtained p-value bellow 0,05. Thus, the treatments (use the catalog or not use the catalog) make a difference for all measures, TP and TN, in Object 1.

Regarding measure TP in Object 2 (TP_OBEJCT2) at groups 1 and 2, the p-value was 0,039, being bellow 0,05. Therefore, the p-value indicates a statistically significant difference between the groups. Noting that Group 1 used the catalog and Group 2 did not use it in Object 2.

Regarding measure TP in Object 2 (TP_OBEJCT2) at Control Group and Group 1, the p-value was 0,005, which is bellow 0,05. This result indicates that there is statistically significant difference between the groups. Noting that Control Group did not use the catalog and Group 1 did used the catalog in Object 2.

Regarding measure TP in Object 2 (TP_OBEJCT2) at Control Group and Group 2, the p-value was 0,514. This result indicates that there is no statistically significant difference between the groups. Noting that both Control Group and Group 2 did not use the catalog in Object 2. Therefore, nothing could indicate a difference between them. Again, this result strengthens the conclusion that the only difference between groups is the usage of the catalog.

Following the same pattern, for the measure TN in Object 2 (TN_OBEJCT2) at Group 1 and Group 2, the p-value was 0,012, being bellow 0,05. Also, the p-value (0,005) in Control Group and Group 1 was bellow 0,05. Finally, the p-value was above 0,05 when both groups did not use the catalog (Control Group and Group 2). Thus, the treatments (use the catalog or not use the catalog) make a difference for all measures, TP and TN, in Object 2 , as well as in Object 1.

Therefore, the null hypothesis stated in Equation 6.4 is rejected, allowing the acceptance of the alternative hypothesis.

6.5.3 CE-RQ2: Efficiency

The research question related to Efficiency is *"Is the time spent to make decisions towards NFRs lower when the correlations catalog is used?"*. The measure used to answer this question was Time Spent (TS). Next subsection presents the descriptive statistics, then, the normality and hypothesis testing is presented.

6.5.3.1 Descriptive Statistics

All raw data used to draw conclusions in regards to Question 2 is presented in Table 42 and the descriptive statistics in Table 43.

Regarding to the time spent in Object 1 (TS_OBJECT1), it is possible to see that the mean of the group who was not using the catalog (Group 1 - 34,25 minutes) was bigger than the mean of the group who was using the catalog (Group 2 - 21,42). The median (33) and mode (33) were also higher for Group 1. The standard deviation for Group 1 was 15,89, higher than the other groups, indicating that there is considerable dispersion in the data. Figure 75 presents an

Table 42 – Raw Data to answer CE-RQ2

Subject	Group	Time Spent - OBJECT 1	Time Spent - OBJECT 2
1	1	0:27:00	00:11:00
2	1	0:46:00	00:12:00
3	1	0:33:00	00:12:00
4	1	0:26:00	00:12:00
5	1	0:48:00	00:14:00
6	1	0:43:00	00:11:00
7	1	0:38:00	00:12:00
8	1	1:08:00	00:18:00
9	1	0:33:00	00:08:00
10	1	0:24:00	00:08:00
11	1	0:09:00	00:05:00
12	1	0:16:00	00:07:00
13	2	00:33:00	00:08:00
14	2	00:26:00	00:16:00
15	2	00:24:00	00:15:00
16	2	00:22:00	00:19:00
17	2	00:20:00	00:12:00
18	2	00:13:00	00:16:00
19	2	00:19:00	00:12:00
20	2	00:20:00	00:10:00
21	2	00:15:00	00:09:00
22	2	00:12:00	00:19:00
23	2	00:11:00	00:06:00
24	2	00:42:00	00:20:00
25	C	00:37:00	00:15:00
26	C	00:30:00	00:18:00
27	C	00:31:00	00:18:00
28	C	00:33:00	00:16:00
29	C	00:31:00	00:18:00
30	C	00:28:00	00:25:00
31	C	00:30:00	00:10:00
32	C	00:22:00	00:17:00
33	C	00:21:00	00:12:00
34	C	00:21:00	00:10:00
35	C	00:19:00	00:14:00
36	C	00:21:00	00:12:00

Source: Author.

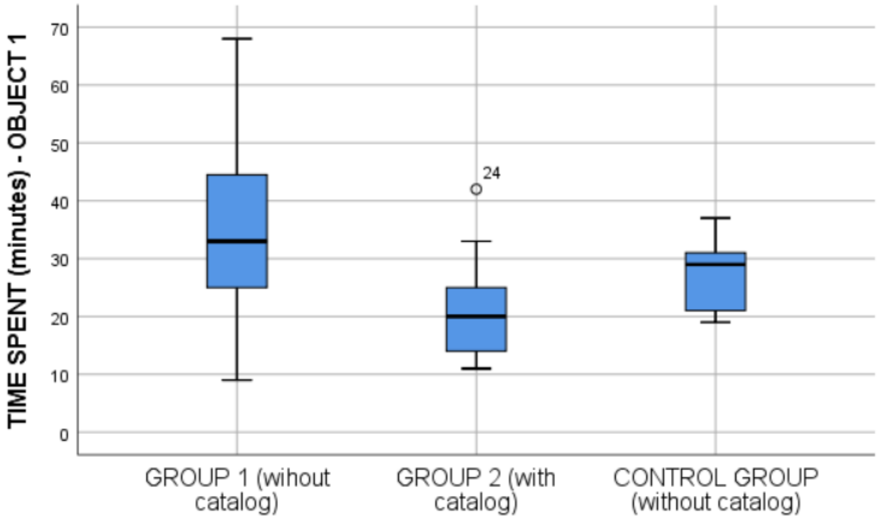
Table 43 – Descriptive Statistics for CE-RQ2

	Group	Mean	Median	Mode	St. Dev.
TS_OBJECT1	1	34,25	33	33	15,89
	2	21,42	20	20	9,07
	C	27	29	21	5,9
TS_OBJECT2	1	10,83	11,5	12	3,46
	2	13,5	13,5	16	4,68
	C	15,42	15,5	18	4,25

Source: Author.

overview of results regarding TS in Object 1. It is possible to see that the only outlier detected was in Group 2, participant 24. Although the participant 8 in group 1 seems an outlier, the IBM SPSS tool did not consider it as an outlier according to their mathematical formula, then this work agrees with the tool.

Figure 75 – Boxplot - Time Spent in Object 1 (AutomaGREat)



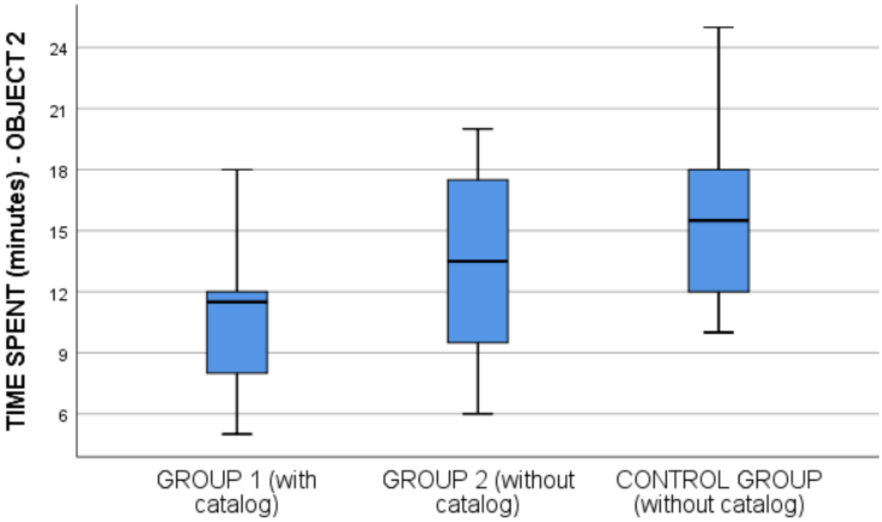
Source: Author

Comparing the Control Group, which is also another group that did not use the catalog, it is possible to see that Group 2 was also better than the Control Group in mean, median and mode.

With regard to the time spent in Object 2 (TS_OBJECT2), the group with the catalog (Group 1) was faster than the two groups without catalog (Group 2 and Control Group). However, the difference was not as high as in the first object. It is possible to see that all groups have decreased their times, even the control group, who did not use the catalog in any object. This may indicate that participants can be faster as they learn to do the tasks, regardless of the catalog. However, the group using the catalog kept being faster even when performing tasks in the second object.

Figure 76 presents an overview of the results regarding TS in Object 2. It is possible to see that no outlier was detected in all groups. Finally, the group executing the tasks with the catalog (Group 1) had better results regarding the time spent.

Figure 76 – Boxplot - Time Spent in Object 2 (GREatBus)



Source: Author

6.5.3.2 Normality and Hypothesis Testing

To test the null hypothesis of Efficiency, defined in Equation 6.5, Time Spent (TS) should have significant differences (p<0,05) in both objects (AutomaGREat - OBJECT1 and GREatBus - OBJECT2).

$$H_{0-CE-RQ2} : TSwithCatalog = TSwithoutCatalog \tag{6.5}$$

As performed to Efficacy, the tests for Efficiency should be executed by comparing three set of groups: *(i)* Group 1 and Group 2; *(ii)* Control Group and Group 1; and *(iii)* Control Group and Group 2. These three combinations of groups should be analyzed regarding Time Spent in each object (AutomaGREat - OBJECT1 and GREatBus - OBJECT2). In this way, six normality and hypothesis tests should be executed.

Shapiro-Wilk test was used as it was in the previous question (CE-RQ1). The results indicated that this data set for the three combinations of groups and for all measures (TP in Object 1, TN in Object 1, TP in Object 2 and TN in Object 2) follow a normal distribution (See Table 44), since the values are bigger than 0,05. Therefore, a parametric test can be used.

Table 44 – Shapiro-Wilk tests for CE-RQ2

Measures / Samples	Group 1 and Group 2	Control Group and Group 1	Control Group and Group 2
TS_OBJECT1	0,860	0,870	0,595
TS_OBJECT2	0,345	0,330	0,789

Source: Author.

The T-Test was applied in this case because it is a parametric test suggested by (WOHLIN *et al.*, 2012) for the type of design in this experiment and it is present in the SPSS IBM tool. Table 45 presents the results (p-values) of the hypothesis tests.

Table 45 – T-Test for CE-RQ2

Measures / Samples	Group 1 and Group 2	Control Group and Group 1	Control Group and Group 2
TS_OBJECT1	0,023	0,159	0,088 / 0,010
TS_OBJECT2	0,127	0,008	0,305

Source: Author.

Regarding the measure TS in Object 1 (TS_OBEJCT1) at groups 1 and 2, the p-value was 0,023, which was bellow 0,05. Therefore, the p-value indicates a statistically significant difference between these groups, noting that Group 1 did not use the catalog and Group 2 used it.

Concerning the measure TS in Object 1 (TS_OBEJCT1) at Control Group and Group 1, the p-value was 0,159. This result indicates that there is no statistically significant difference between the groups. Both groups did not use the catalog in Object 1 . Therefore, nothing could indicate a difference between them. It is interesting to note that because it strengthens the conclusion that the only difference between groups is the usage of the catalog.

Regarding the measure TS in Object 1 (TS_OBEJCT1) at Control Group and Group 2, the p-value was 0,088, not being bellow 0,05. Therefore, the p-value does not indicate a statistically significant difference between the groups. Group 2 used the catalog and Control Group did not use it. However, taking a closer look at results for Group 2 in Object 1, there is a outlier, which is an abnormal or false data point (WOHLIN *et al.*, 2012). Figure 75 shows this outlier detected by the SPSS IBM tool, which is Subject 24. He/she was much above the other participants. When excluding this participant from dataset, the p-value is 0,010, being bellow 0,05.

Regarding the measure TS in Object 2 (TS_OBEJCT2) at Groups 1 and 2, the p-value was 0,127, being above 0,005. Therefore, the p-value does not indicate a statistically significant difference between the groups, noting that Group 1 used the catalog and Group 2 did not use it in Object 2.

Regarding the measure TS in Object 2 (TS_OBEJCT2) at Control Group and Group 1, the p-value was 0,008. This result indicates that there is a statistically significant difference between the groups. Control Group did not use the catalog and Group 1 used the catalog in Object 2.

Regarding the measure TS in Object 2 (TS_OBEJCT2) at Control group and Group 2, the p-value was 0,305. This result indicates that there is no statistically significant difference between the groups. Both groups did not use the catalog in Object 2.

In summary, not all p-values resulted as expected. In Table 45, there is one value that do not favor the rejection of the null hypothesis for CE-RQ2. A possible reason why the time spent decreased in the second object, even without the catalog, is that the participants learned how to do the tasks and could be more efficient.

6.5.4 CE-RQ3: Satisfaction

The research question related to Satisfaction is "Will the participants in the role of requirements engineers feel more satisfied with using a catalog compared to when they are not using it?". This characteristic was evaluated regarding a set of statements where the participants should use the five-point Likert scale to rate their feelings: Strongly Agree, Partially Agree, Neither Agree nor Disagree, Partially Disagree and Strongly Disagree. Next subsection presents the descriptive statistics, then, the hypothesis testing is presented.

In this work, Satisfaction is related to the users feelings that the catalog made the experiment tasks fast and easy. As no supporting tool was used, the ease of use of the catalog was not considered. The catalog was represented as a table, that was given on a sheet to the participants as a artifact to help them in making decisions.

All participants had to answer questions after the execution of the experiment in each object. In total, 6 questionnaires were answered. Table 46 introduces all statements for each group and object.

Four statements were asked for all groups in all objects: 1. I easily identified the impacts, 2. I quickly identified the impacts, 3. I easily made my decision and 4. I quickly made my decision. Asking these same questions for all tasks gives the possibility to make comparisons between the treatments, and through them, the hypothesis could be statistically tested.

Additionally, each group had few different statements. For example, the Control Group did not use the catalog in neither objects. Therefore, two statements were added: *If there was a catalog, it would be easier* and *If there was a catalog, it would be faster*. The participants of this group were aware of what is a correlation catalog. Group 1 also had these two additional questions since participants performed tasks in Object 1 without the correlations catalog.

When executing tasks in Object 2 with the catalog, participants from Group 1 had to

Table 46 – Statements to measure Satisfaction for CE-RQ3

Group	***Object 1***	***Object 2***
Control	1. I easily identified the impacts 2. I quickly identified the impacts 3. I easily made my decision 4. I quickly made my decision **5. If there was a catalog, it would be easier** **6. If there was a catalog, it would be faster**	1. I easily identified the impacts 2. I quickly identified the impacts 3. I easily made my decision 4. I quickly made my decision **5. If there was a catalog, it would be easier** **6. If there was a catalog, it would be faster**
1	1. I easily identified the impacts 2. I quickly identified the impacts 3. I easily made my decision 4. I quickly made my decision **5. If there was a catalog, it would be easier** **6. If there was a catalog, it would be faster**	1. I easily identified the impacts 2. I quickly identified the impacts 3. I easily made my decision 4. I quickly made my decision **5. I think the catalog made my decision easier** Why? **6. I would recommend using the catalog for decision making** Why?
2	1. I easily identified the impacts 2. I quickly identified the impacts 3. I easily made my decision 4. I quickly made my decision **5. I think the catalog made my decision easier** Why? **6. I would recommend using the catalog for decision making** Why?	1. I easily identified the impacts 2. I quickly identified the impacts 3. I easily made my decision 4. I quickly made my decision **5. I think the absence of a catalog made my decision harder** Why? **6. I think the catalog would made my decision easier** Why? **7. I would recommend using a catalog for decision making** Why?

Source: Author.

answer these following additional statements: *I think the catalog made my decision easier* and *I would recommend using the catalog for decision making*. Group 2 also had these two additional questions when participants performed tasks in Object 1 using the correlations catalog.

Then, when participants of Group 2 had to perform the tasks in Object 2, the following statements were asked: *I think the absence of a catalog made my decision difficult*; *I would recommend using the catalog for decision making*; and *I would recommend using a catalog for decision making*.

Furthermore, open questions in the form of "Why?" were asked to stimulate the participants to give their opinion regarding the usage of the catalog.

6.5.4.1 Descriptive Statistics and Qualitative Answers

Figure 77 presents the results for statements 1, 2, 3 and 4 in Object 1 (AutomaGREat), where Control Group and Group 1 did not use the catalog and Group 2 used it. In general, Group

2 provided better results in comparison with the other groups. In statements 1 and 2 (*I easily identified the impacts*), only Group 2 had participants that strongly agreed. Furthermore, most of them partially agreed in both statements.

Group 2 and Group 1 obtained the same quantity of answers for Partially Agree in statement 3 and for Strongly Agree in statement 4. Regarding negative options (Partially and Strongly Disagree), Group 2 had one answer to Strongly Disagree in statements 1 and 2, being worse than Group 1. However, when comparing to Control Group, Group 2 always obtained less answers for the disagreements options.

Figure 78 presents the results for statements 1, 2, 3 and 4 in Object 2 (GREatBus), where Control Group and Group 2 did not use the catalog and Group 1 used it. In this turn, Group 1 obtained much better results regarding the other groups in all statements. Furthermore, participants in Group 2 that used catalog in Object 1 and could not use it again on Object 2, were not satisfied with the fact that the catalog was not available anymore.

In the second round of the experiments, the use of the catalog obtained better satisfaction results. Participants could feel the consequences of having or not having a catalog to help with their decisions. These results show how important is to do more than one round of experimentation. The real feeling would not manifest itself in the numbers if the second round was not performed.

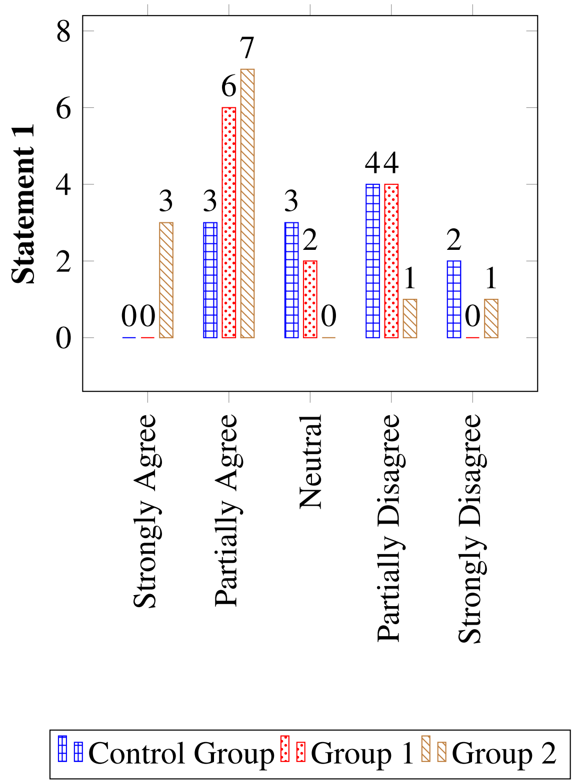
Table 47 presents the descriptive statistics, where 5 is “Strongly Agree”, 4 is “Partially Agree”, 3 is “Neither Agree nor Disagree”, 2 is “Partially Disagree” and 1 is “Strongly Disagree”.

In general, the median and mode are better for Group 2 in statements for Object 1 (AutomaGREat). This group was using the catalog in this object. In Object 2 (GREatBus), Group 1 obtained better medians and modes than the other groups. Not surprisingly, this was the group using the catalog. Furthermore, participants could get a better feeling of the difference of using the catalog and not using the catalog when performing tasks for the second time in Object 2.

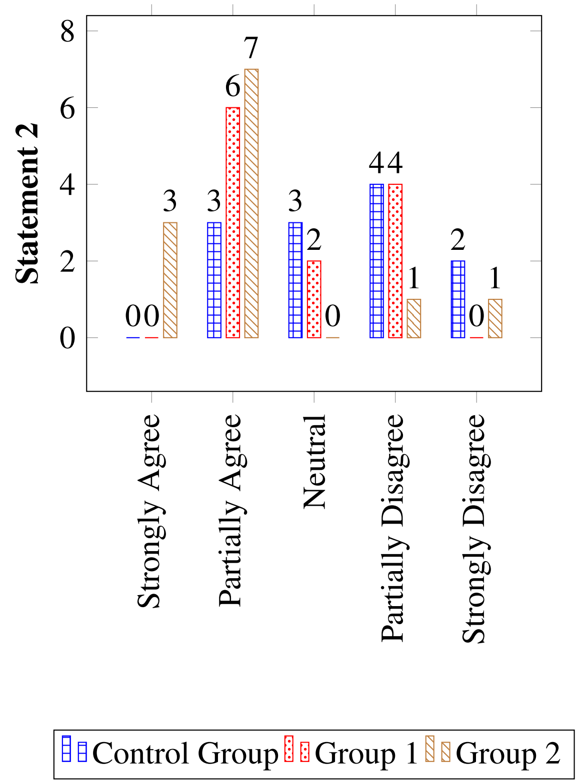
Figure 79 presents the results regarding the other statements for participants who did not use the catalog, which were Control Group in Object 1 (CG - O1), Control Group in Object 2 (CG - O2) and Group 1 in Object 1 (G1 - O1). Most of them strongly agreed that the use of a correlation catalog would make the decision easier and faster.

Figure 80 presents the results regarding the statements for participants who used the catalog, which were Group 1 in Object 2 (G1 - O2), Group 2 in Object 1 (G2 - O1) and Group 2

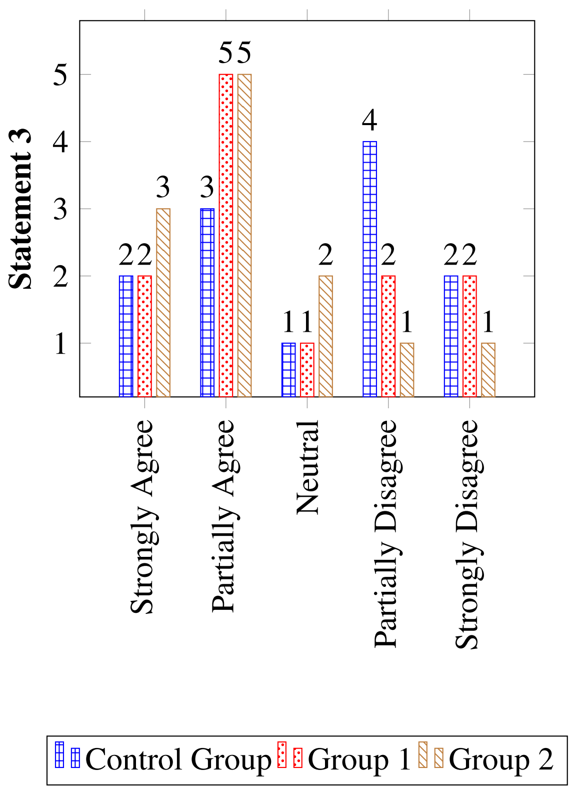
Figure 77 – Results of Satisfaction in Object 1 - AutomaGREat



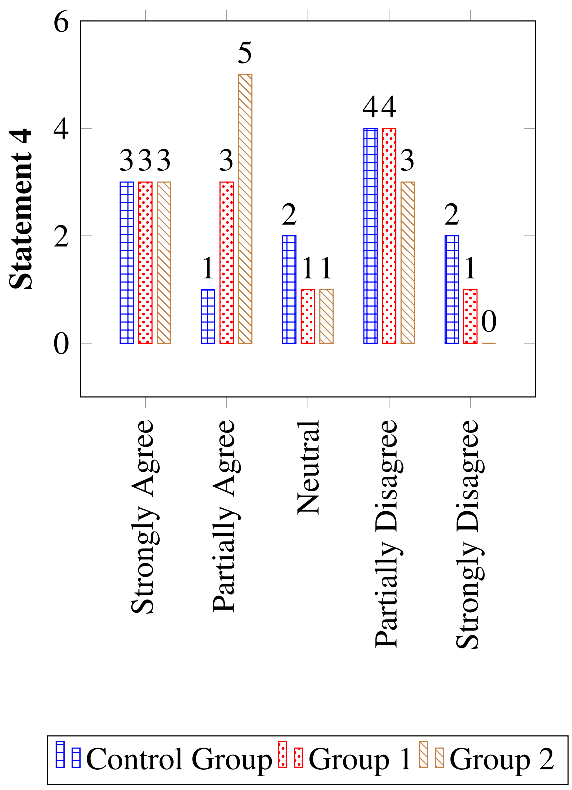
(a) I easily identified the impacts



(b) I quickly identified the impacts



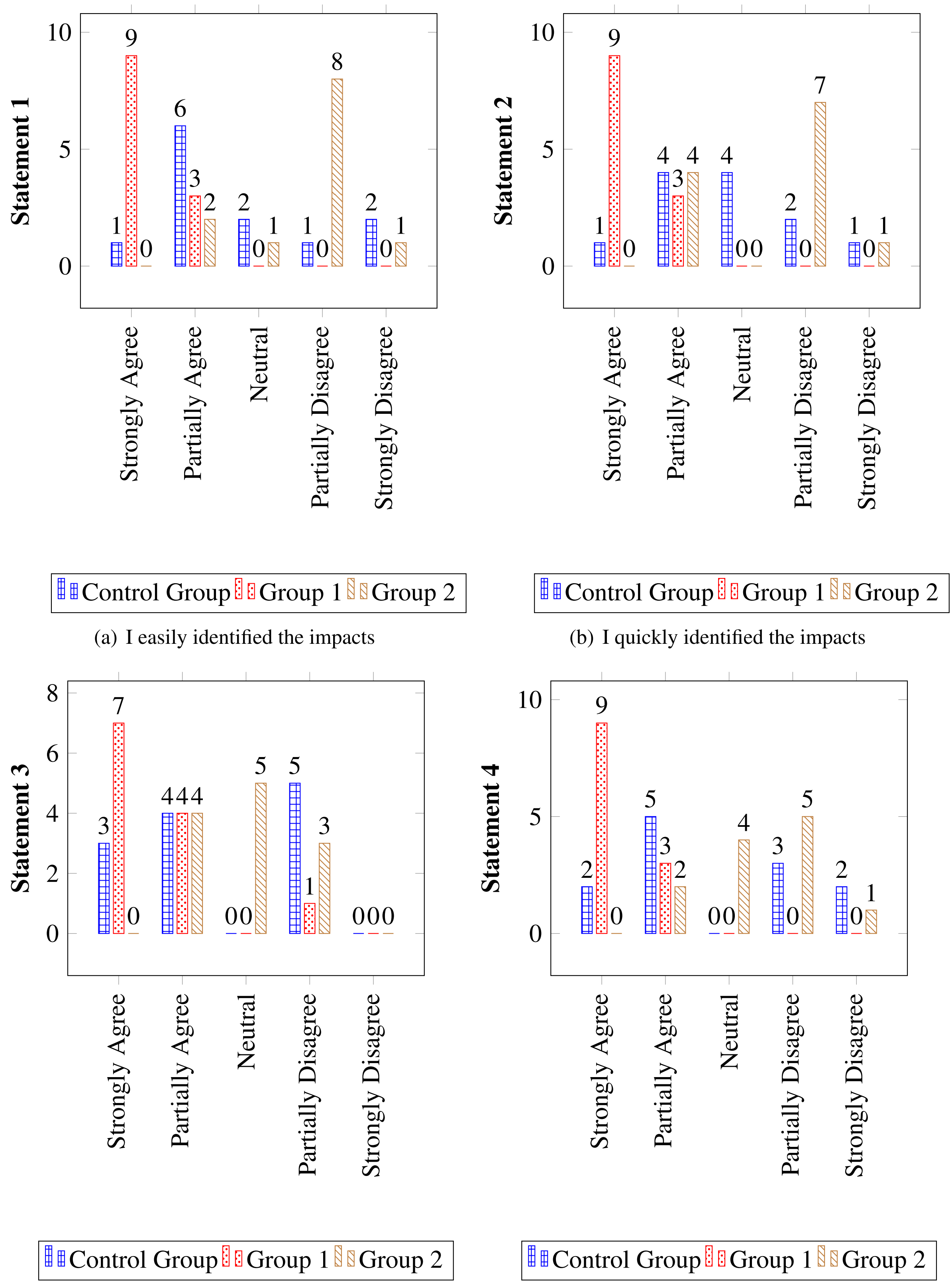
(c) I easily made my decision



(d) I quickly made my decision

Source: Author

Figure 78 – Results of Satisfaction in Object 2 - GREatBus



(a) I easily identified the impacts

(b) I quickly identified the impacts

(c) I easily made my decision

(d) I quickly made my decision

Source: Author

Table 47 – Descriptive Statistics for CE-RQ3

Object and Statement	*Group*	*Median*	*Mode*
O1 - I easily identify the impacts	1	3	4
	2	4	4
	C	2	2
O1 - I quickly identify the impacts	1	2	2
	2	4	4
	C	2	2
O1 - I easily made my decision	1	4	4
	2	4	4
	C	2	2
O1 - I quickly made my decision	1	3	2
	2	4	4
	C	2	2
O2 - I easily identify the impacts	1	5	5
	2	2	2
	C	4	4
O2 - I quickly identify the impacts	1	5	5
	2	2	2
	C	3	3
O2 - I easily made my decision	1	5	5
	2	3	3
	C	4	2
O2 - I quickly made my decision	1	5	5
	2	2	2
	C	4	4

Source: Author.

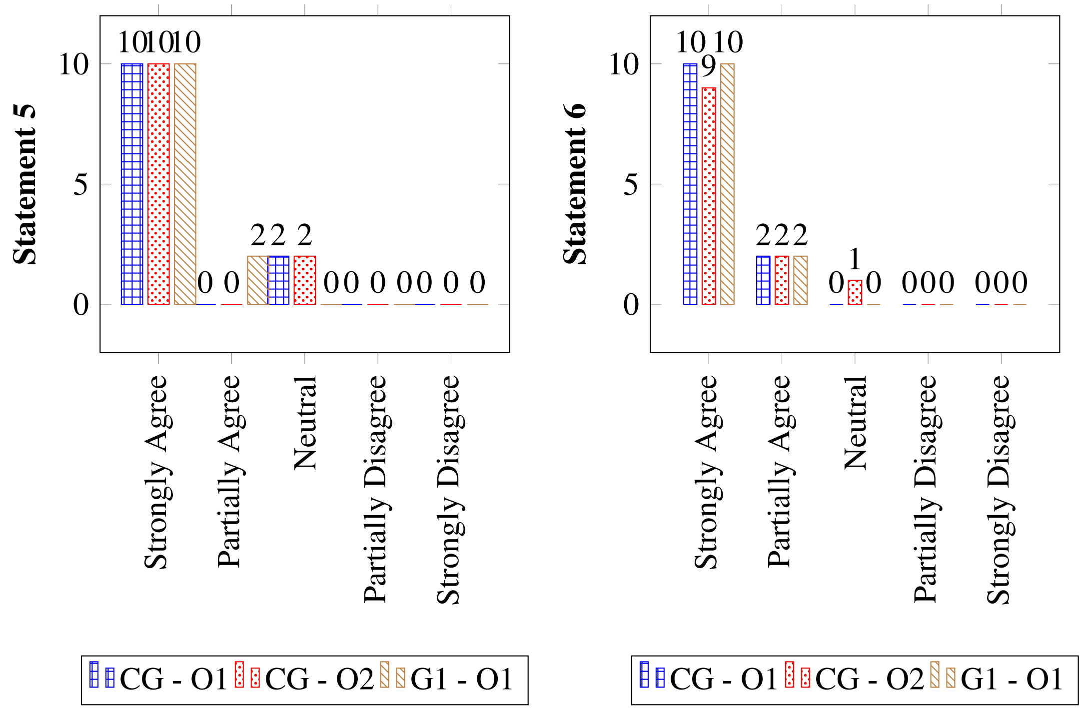
in Object 2 (G2 - O2). Most of them strongly agreed that the use of a catalog made the decision easier and that they would recommend it.

Furthermore, all of the participants who experienced the usage of the catalog in the first object, and then could not use it in the second object, strongly agreed with the statement: *I think the absence of a catalog made my decision harder*.

The post-task forms also had open questions to the tasks that were performed with the help of the catalog. Most participants reported that the usage of the catalog was fundamental to take more informed decisions and that they were not aware of the impacts since is from UbiComp and IoT area. Some of the comments are presented as follows:

- *"Without the catalog, I would not know 80% of the impacts"*
- *"Decreases the time for reflection on the impacts of each strategy."*
- *"It takes less time to decide and avoids speculation about the strategy."*
- *"The catalog helped because I have a lack of knowledge in the area to accurately specify the impacts"*
- *"Without the catalog, I was not sure, I worked on assumptions that I barely know"*

Figure 79 – Results of Satisfaction when Participants did not use the Catalog

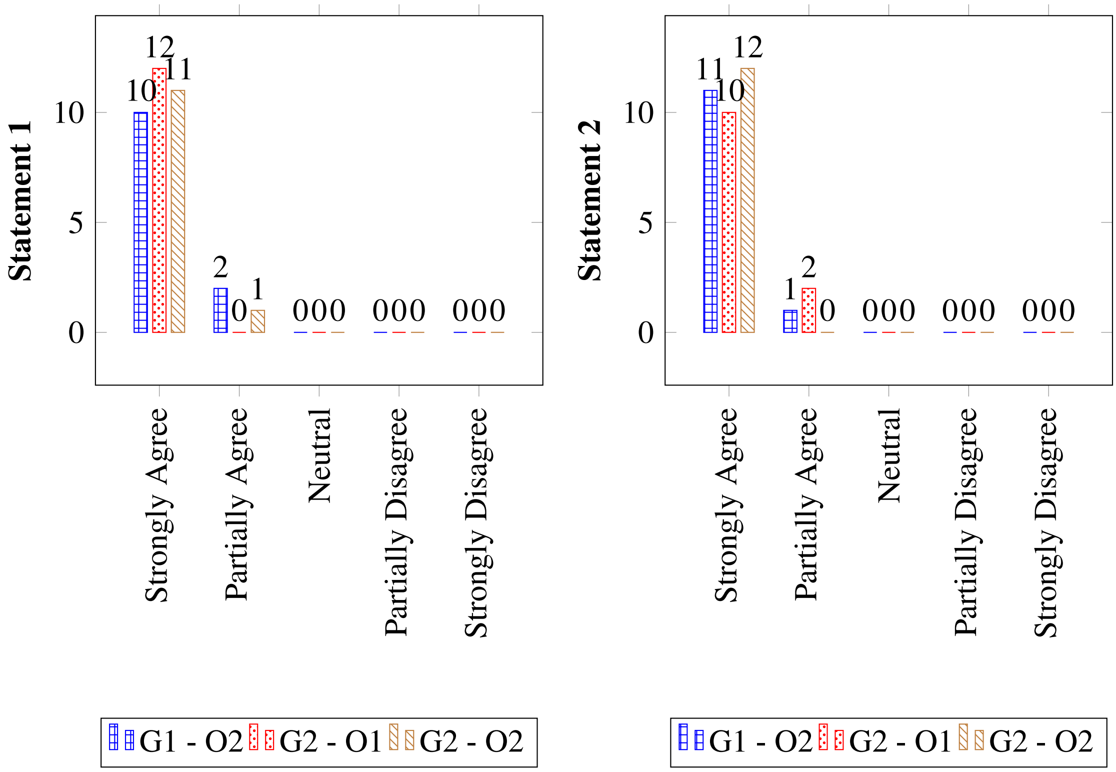


(a) If there was a catalog, it would be easier

(b) If there was a catalog, it would be faster

Source: Author

Figure 80 – Results of Satisfaction when Participants used the Catalog



(a) I think the catalog made my decision easier

(b) I would recommend using the catalog for decision making

Source: Author

6.5.4.2 Cronbach's and Hypothesis Testing

The Cronbach's alpha test was applied to measure the reliability of the obtained answers, resulting in 74,4% of reliability, a value considered acceptable by the literature (GLIEM; GLIEM, 2003).

To test the null hypothesis of Satisfaction, defined in Equation 6.6, the Mann–Whitney U test was used. Normality tests were not executed, because Likert scale does not follow a normal distribution, since it is an ordinal scale (JAMIESON, 2004), as mentioned before.

$$H_{0-CE-RQ3} : SatisfactionWithCatalog = SatisfactionWithoutCatalog \tag{6.6}$$

Table 48 presents the results of the hypothesis tests. The statistical differences between Group 1 and Group 2 in all statements of Object 1 were not significant (p > 0,05). However, the difference was significant in Object 2 (p < 0,05).

Table 48 – Hypothesis Testing for CE-RQ3

	Group 1 and Group 2	***Control Group and Group 1***	***Control Group and Group 2***
S1-OBJECT1	0,101	0,219	0,12
S2-OBJECT1	0,242	0,410	0,89
S3-OBJECT1	0,514	0,630	0,219
S4-OBJECT1	0,514	0,630	0,242
S1-OBJECT2	0,000	0,001	0,78
S2-OBJECT2	0,000	0,000	0,219
S3-OBJECT2	0,001	0,068	0,514
S4-OBJECT2	0,000	0,003	0,291

Source: Author.

Regarding Control Group and Group 1 in Object 1, there were not significant statistical differences between them in all statements (p > 0,05). However, this result was expected since both groups did not use the catalog in Object 1. On the other hand, in Object 2, the groups had statistical differences between them (p > 0,05) at statements 1, 2 and 4.

The results for Control Group and Group 2 in both objects have no statistical differences. This result was expected for Object 2 since both groups did not have the proposed catalog to perform this task. However, this result was not expected for Object 1 due to the fact that the participants in Group 2 used the catalog.

In summary, not all p-values resulted as expected when comparing results between groups. Therefore, the general null hypothesis for Satisfaction cannot be rejected. However, the results showed that the consequences of not having the catalog could be felt in Object 2 (GREatBus).

6.6 Discussion

With this experiment, it was possible to obtain interesting findings about the usage of NFRs catalogs by participants in the role of novice requirements engineers, which was presented in the answers for each research question (CE-RQs). A synthesis of these findings is presented and discussed in Subsection 6.6.1. The threats to validity to this experiment is presented in Subsection 6.6.2.

6.6.1 Synthesis of the Results

The evaluation of the correlation catalog focused on investigating three questions:

- CE-RQ1 - Is the set of selected operationalizations suitable to maximize the positive impact and minimize the negative impact of the required NFRs when the catalog is used?
- CE-RQ2 - Is the time spent to make decisions towards NFRs lower when the correlations catalog is used?
- CE-RQ3 - Will the participants in the role of requirements engineers feel more satisfied with using a catalog compared to when they are not using it?

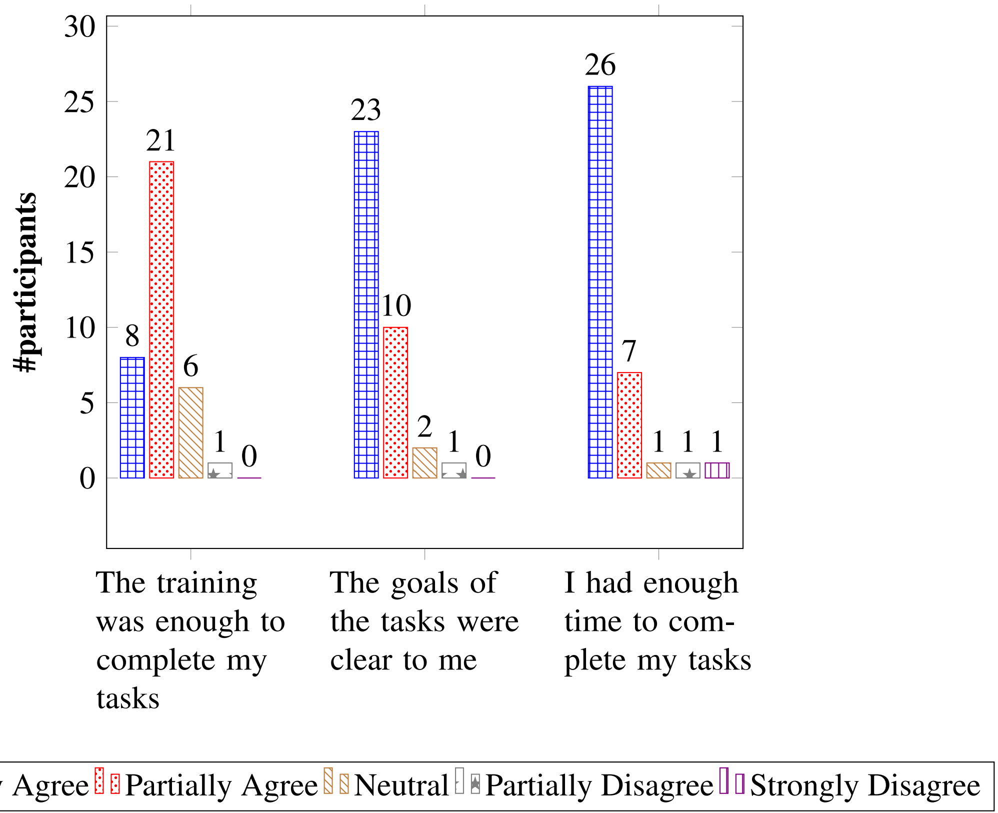
Regarding CE-RQ1, this work aimed to evaluate if the proposed catalog helps in making better decisions using the catalog than not using the catalog, which means only using the participant's experience. Better decisions mean choosing a set that has maximizes the positive impacts and minimizes the negative impacts for the chosen quality characteristics. Two measures were needed to evaluate this question: one is to measure if the choices of the participants maximized the NFRs and another one is to measure the minimization of negative aspects. Both measures obtained expected results. Therefore, the null hypothesis was completely rejected. Furthermore, this work can say that the correlations catalog minimizes the negative impact and maximizes the positive impact of the required NFRs.

Regarding CE-RQ2, the goal was to assess the quality focus "efficiency", which means checking if the participants spent more time using the catalog or not using the catalog. Results indicated that the null hypothesis could not be rejected. The reason why this happened may be due to the fact that the participants learned how to do the experiment's tasks quickly, even when not using the catalog.

Regarding CE-RQ3, this work investigated the participant's satisfaction regarding the usage of a catalog. Results showed that participants in general felt more satisfied regarding performance and easiness in analyzing impacts and making decisions. When comparing results between groups, the null hypothesis could not be rejected, although it was possible to see a statistical difference when participants performed the tasks for the second time.

Furthermore, when all participants finished tasks in all objects, a post-experiment questionnaire was applied. The results are presented in Figure 81. They show that most participants were satisfied with trainings, goals of the tasks and duration.

Figure 81 – Results of Post-Experiment Questionnaire



Source: Author.

6.6.2 *Threats to Validity*

This section discusses the threats to the validity of the experiment results as suggested by (WOHLIN *et al.*, 2012) regarding: *(i)* Conclusion Validity; *(ii)* Internal Validity; *(iii)* Construct Validity; and *(iv)* External Validity.

Conclusion validity is concerned with the relationship between the treatment and the outcome. Some decisions were made in this study to obtain results that represent the real relationship:

- The evaluation of the normality of the data through Shapiro-Wilk tests before choosing a hypothesis test;
- The selection of participants presenting a homogeneous profile; and
- The execution of pilot tests to improve the reliability of the instrumentation[4].

Internal validity is concerned about influences that can affect the independent variable with respect to causality, without the researcher's knowledge. These influences threat the conclusion about a possible causal relationship between treatment and outcome. Some decisions made in this study to minimize influences on the independent variable:

- All activities in the groups were executed at the same time in one day. Therefore, there is no risk that history affects the experimental results;
- The participants were split in three groups, one of them is a Control Group. Then, several combinations were made to analyze data and give more reliable results; and
- The experiment's tasks were executed twice for each group. Thus, more data were collected so the researcher could be more certain about the relationship between treatment and outcome.

Construct validity concerns to the treatments and outcomes. They should have a good reflect on the cause and effect of the experiment. The following are strategies used to minimize the risks of this kind of threat:

- All measures were defined before the experiment took place. Therefore, the theory was clear enough, and hence the experiment was sufficiently ready to be performed;
- A threat is if an experiment is conducted with a single document as object, the cause construct is under-represented. In this work, two objects were used, then two rounds of the tasks were performed; and
- Another possible threat is the representation of the construct, since only parts of Invisibility SIG and Correlation Catalog were used in the experiment. Larger documents could produce more reliable data, but it could not be possible to execute the experiment tasks in an acceptable time. Therefore, this work accepted this risk.

External validity is related to conditions that limit the ability to generalize the results of our experiment to industrial practice.

---

4 However, the pilot tests were not executed with undergraduate students since they all were used in the experiment tests

- There is a threat of having participants not representative of the population, *i.e.*, the wrong people participating in the experiment. Although this work did not used practitioners from industry, the participants were all students from a requirements engineering course, where this activity (decision making regarding NFRs) is required. Furthermore, there are many studies in literature supporting experiments with students (SALMAN *et al.*, 2015) (FALESSI *et al.*, 2018).

6.7 Chapter Summary

This Chapter presented an evaluation of the LEAD usage, which was performed using a controlled experiment[5].

The main purpose of a correlation catalog is to support developers and requirements engineers in making better and faster decisions towards strategies that will be selected to benefit required NFRs. LEAD was developed to the Invisibility characteristic and has correlations not yet cataloged and that could not be known by software engineers, especially novice ones.

Therefore, the treatments evaluated in this experiment were the usage of the catalog (Treatment 1) and the not usage of the catalog (Treatment 2). The main goal was to analyze the usage of LEAD with respect to efficacy, efficiency and satisfaction.

Thirty-six subjects participated in this experiment and provided data that supported the comparison of using or not using the catalog.

Regarding the quality focus Efficacy, the results pointed out evidence that LEAD minimizes the negative impacts and maximize the positive effects for the required NFRs, which could be statistically validated.

Concerning the quality focus Efficiency, the results indicated that participants could be faster when they used the catalog. However, this result was not completely proved statistically.

Finally, concerning the quality focus Satisfaction, participants felt more satisfied with the proposed correlation catalog when they performed the tasks in the second object. However, the null hypothesis could not be rejected.

[5] Detailed results can be found at https://github.com/great-ufc/LEAD_Experiment

7 CONCLUSION

This research aimed to investigate correlations among NFRs that may impact the development of UbiComp and IoT applications. To better understand how a correlations catalog could be defined, an exploratory study using a systematic mapping was performed. The results showed the need to establish a process to organize the steps and approaches that researchers and developers could use to define the catalog. Then, the CORRELATE process was proposed together with its instruments and approaches. From the first execution of this process, the LEAD catalog was defined for the Invisibility characteristic, providing a proof of concept of the process. Next, this catalog was evaluated experimentally by comparing efficacy, efficiency and satisfaction with and without its usage to make decisions in the development of UbiComp and IoT applications.

This Chapter concludes this thesis by revisiting what was achieved in the doctoral work and what can come ahead. In short, the research questions that are introduced in Section 1 are revisited in Section 7.1. After that, in Section 7.2, the thesis hypothesis is discussed and checked regarding how much it is supported by the results presented in this thesis. Section 7.3 summarizes the contributions achieved in this doctoral work. Then, Section 7.5 outlines the work that may continue in the future.

7.1 Revisiting the Research Questions

This thesis defined four research questions, which are discussed in the next subsections in terms of their main results.

7.1.1 RQ1 - Which are the existing NFRs catalogs and how they are defined?

This question dealt with how is the state of the art of NFRs catalogs before starting the definition of the proposed catalog of correlations. Chapter 3 presented the answers to this question, which were gathered through a systematic mapping study, and they are summarized in this section.

As a result, 102 NFRs catalogs were found and deeply investigated. Among four research opportunities, there is only one treated this thesis: the definition of an NFR catalog. Different studies use different combinations of techniques to build a catalog. Some papers used well-known research methods and others use the knowledge and experience of the authors.

However, a systematic and reusable process that organizes a step by step with inputs, outputs, and well-defined approaches on how to create a correlation catalog at a specific level was not found in this systematic mapping. Since the lack of such approach makes the definition of catalogs harder, part of this thesis work was first dedicated to propose an approach capable of defining NFRs catalog, which is the concern of RQ2.

7.1.2 RQ2 - How can an NFR catalog for HCI quality characteristics in UbiComp and IoT systems be defined?

This question was stated to define an approach that can capture, analyze and catalog correlations. Chapter 4 presented a proposed process that is the answer to this question.

The proposed process is called CORRELATE and it is composed of four general steps: *(i) Selecting a quality characteristic*; *(ii) Refining the characteristic*; *(iii) Identifying development strategies*; and *(iv) Defining correlations*. These steps are further refined into sub steps, and proposed instruments and approaches are also presented in this work. In Step 1, an instrument based on a questionnaire to prioritize an NFR is introduced. In Step 2, an approach based on Grounded Theory is proposed. This approach is called ARRANGE and uses the literature as a source of knowledge. In Step 3, an instrument based on a questionnaire for collecting development strategies from developers is introduced. In Step 4, an approach called TRACE is proposed to establish correlations regarding development strategies of NFRs.

The proposed process can be used by researchers and developers to generate knowledge that will be cataloged in forthcoming NFRs Catalogs.

7.1.3 RQ3 - To what extent does a specific HCI quality characteristic from UbiComp and IoT impact on user interaction quality characteristics?

Once the proposed process to capture, analyze and catalog correlations was ready, the investigation about correlations between NFRs in UbiComp and IoT was performed and then RQ3 was answered. Chapter 5 presented the answers for this question, which resulted in a catalog of subcharacteristics, strategies and correlations for the Invisibility Characteristic called LEAD. The answer to this question provided a proof of concept of the CORRELATE process.

LEAD comprises 2 subcharacteristics, 12 sub subcharacteristic, 10 general strategies and 56 specific strategies for the Invisibility characteristic represented in a Softgoal Interdependency Graph. From this knowledge, it was possible to establish 110 correlations with 9 NFRs, being 51 HELP and 59 HURT correlations.

On one hand, Invisibility has a more positive impact on Usability, 22 correlations are positively related to this characteristic, followed by Performance, Functional Suitability, Satisfaction, Efficiency and Context Coverage with intermediate values and Security and Reliability with minor values. On the other hand, Invisibility has more negative impact on Security; ten correlations are negatively related to this characteristic, followed by Privacy, Reliability, Performance, Usability, Functional Suitability with intermediate values, and Satisfaction, Efficiency and Context Coverage with minor values.

7.1.4 RQ4 - Does an NFR catalog improve decisions regarding NFRs in UbiComp and IoT systems?

This question is related to both the evaluation of the use of LEAD and the support for developers to make accurate and informed decisions about NFRs.

The evaluation was performed through a controlled experiment with participants who should make decisions regarding what development strategies to use that could maximize the positive impacts and minimize the negative ones among NFRs that were defined. Thirty-six subjects participated in this experiment and provided data that supported the comparison of using or not using the catalog. The results pointed out evidence that LEADS minimize the negative impacts and maximize the positive impacts of the required NFRs, which was statistically proved. General results regarding time spent to make decisions indicated that participants could be faster when using the catalog. However, this result was not statistically proved. Furthermore, participants felt more satisfied when they performed the experiment tasks for the second time.

7.2 Revisiting the Thesis Hypothesis

At the beginning of this research, this work established the following hypothesis:

> **Research Hypothesis (RH)**
>
> Specific HCI quality characteristics (for example, AMICCaS) impact on user interaction quality characteristics negative and positively, due to its design and implementation solutions.

The results indicated that, on one hand, there are strategies of Invisibility that can be used in any application. For example, Google Sign in API, Speech API, Neural Network, among others. On the other hand, there are strategies specific to the UbiComp and IoT environment. For example, a general strategy, such as adapt according to the context, brings specific solutions such as Middleware and Protocols for IoT and UbiComp applications. Regardless of being used just for the context of UbiComp and IoT systems or other systems, these strategies present correlations with Usability, Performance, Security, Reliability, Functional Suitability, Efficiency, Context Coverage, Satisfaction and Privacy. Therefore, the hypothesis of this thesis was confirmed.

7.3 Contributions and Publications

This thesis presents the following main contributions:

- **The CORRELATE process**. This process supports the generation of well-organized knowledge of NFRs. It is composed of four general steps, two instruments, and two approaches. Researchers and developers can use this proposition to define NFRs catalogs.
- **The ARRANGE approach**. This approach supports Step 2 of the CORRELATE process. Then, a researcher can refine an NFR and represent the generated knowledge through Softgoals Interdependency Graphs. Although it is used to support Step 2, this approach can also be used independent of the proposed process, if the researcher or developer is interested in defining a SIG for a specific quality characteristic.
- **The TRACE approach**. This approach supports Step 4 of the CORRELATE process. With it, a researcher can establish correlations regarding development strategies. Although it is used to support Step 4, this approach can also be used independently of the proposed process. If the researcher already has a set of development strategies and wants to investigate them regarding NFRs, TRACE can be used.
- **The LEAD catalog**. This catalog is proposed to the Invisibility characteristic. It comprises subcharacteristics, development strategies and correlations regarding other NFRs. Developers can use this catalog to establish requirements and the high-level design of a UbiComp and IoT systems at the beginning of the development. A controlled experiment showed that by using this catalog, developers could make better decisions.

As secondary contributions, this thesis presented:

- The exploratory study through a systematic mapping generated a dataset containing more than 1000 subcharacteristics, 1113 implementation and design methods, 473 positive correlations, 395 negative correlations for 86 NFRs. This dataset can be useful for

Table 49 – Main Publications of this thesis

Reference	*Qualis*
CARVALHO, R. M.; ANDRADE, R. M. C.; OLIVEIRA, K. M. Correlations between invisibility and usability in ubicomp and IoT applications: partial results. In: Proceedings of the XXXII Brazilian Symposium on Software Engineering. ACM, 2018.	B2
CARVALHO, R. M.; ANDRADE, R. M. C.; OLIVEIRA, K. M.; KOLSKI, C. Catalog of Invisibility Requirements for UbiComp and IoT Applications. In: IEEE International Requirements Engineering Conference, 2018.	A2
CARVALHO, R. M.; ANDRADE, R. M. C.; OLIVEIRA, K. M. Towards a Catalog of Conflicts for HCI Quality Characteristics in UbiComp and IoT Applications: Process and First Results. In: IEEE 12th International Conference on Research Challenges in Information Science, 2018.	B1
CARVALHO, R. M. Dealing with Conflicts between Non-Functional Requirements of Ubicomp and IoT Applications In: Doctoral Symposium at IEEE International Requirements Engineering Conference, 2017.	-
CARVALHO, R. M.; ANDRADE, R. M. C.; BARBOSA, J.; MAIA, A. M.; JUNIOR, B. A.; AGUILAR, P. A.; BEZERRA, C. I. M.; OLIVEIRA, K. M. Evaluating an IoT Application using Software Measures. In: International Conference on Human-Computer Interaction (HCII), 2017.	B2
ANDRADE, R. M. C.; **CARVALHO, R. M.**; ARAUJO, I. L.; OLIVEIRA, K. M.; MAIA, M. E. F.; What changes from Ubiquitous Computing to Internet of Things in Interaction Evaluation? In: International Conference on Human-Computer Interaction (HCII), 2017.	B2
CARVALHO, R. M.; ANDRADE, R. M. C.; OLIVEIRA, K. M.; SANTOS, I. S; BEZERRA, C. I. M. Quality characteristics and measures for human-computer interaction evaluation in ubiquitous systems. Software Quality Journal, 2016.	B1

Source: Author.

developers to search several ways of implementing NFRs and, thus, reuse the knowledge gained from this work; and

- The evaluation work performed through a controlled experiment. Since few studies (only two papers were found in this thesis (CYSNEIROS, 2007) (VELEDA; CYSNEIROS, 2019)) present quantitative evidence about the usage of NFRs catalogs. This thesis provided additional evidence to the literature regarding statistical results about the usage of NFR catalogs.

Furthermore, fifteen papers were published in conferences and journals. Seven of them are a direct result of the research performed in this work (See Table 49). Eight of them are publications not directly related to the thesis but important for acquiring knowledge during this doctoral research (See Table 50).

Table 50 – Secondary Publications of this thesis

Reference	***Qualis***
CARVALHO, R. M.; ANDRADE, R. M. C.; OLIVEIRA, K. M. AQUArIUM - A Suite of Software Measures for HCI Quality Evaluation of Ubiquitous Mobile Applications In: The Journal of Systems and Software , 2018.	A2
ALMEIDA, R. L. A.; MESQUITA, L. B.; **CARVALHO, R. M.**; ANDRADE, R. M. C. When Technology supports Urban Mobility: Improvements for Mobile Applications based on a UX Evaluation. In: International Conference on Human-Computer Interaction (HCII), 2017.	B2
CARVALHO, R. M.; SANTOS, I. S.; MEIRA, R. G.; AGUILAR, P. A.; ANDRADE, R. M. C. Machine Learning and Location Fingerprinting to Improve UX in a Ubiquitous Application. In: Human-Computer Interaction International, Distributed, Ambient and Pervasive Interactions, 2016	B2
ALMEIDA, R. L. A.; MESQUITA, L. B.; **CARVALHO, R. M.**; A. JUNIOR, B. R.; ANDRADE, R. M. C. Quando a Tecnologia apoia a Mobilidade Urbana: Uma Avaliação sobre a Experiência do Usuário com Aplicações Móveis. In: Evaluation Competition at XV Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais (IHC), 2016.	B2
SIEWERDT, F.; **CARVALHO, R. M.**; ANDRADE, R. M. C. Recomendações para Testes de Usabilidade em Aplicações Ubíquas. In: XV Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais (IHC), 2016.	B2
CARVALHO, R. M.; ANDRADE, R. M. C.; OLIVEIRA, K. M. Using the GQM Method to Evaluate Calmness in Ubiquitous Applications. In: HCI International, 2015, Los Angeles, California. Distributed, Ambient, and Pervasive Interactions, 2015	B2
ANDRADE, R. M. C.; SANTOS, I. S.; ARAUJO, I. L.; **CARVALHO, R. M.** Uma Metodologia para o Ensino Teórico e Prático da Engenharia de Software. In: VIII Fórum de Educação em Engenharia de Software (FEES 2015), 2015, Belo Horizonte. Anais do VIII Fórum de Educação em Engenharia de Software, 2015.	-
BEZERRA, C. I. M.; ANDRADE, R. M. C.; **SANTOS, R. M.**, R. M.; ABED, M.; OLIVEIRA, K. M.; MONTEIRO FILHO, J. M. S.; SANTOS, I. S.; EZZEDINE, H. Challenges for usability testing in ubiquitous systems. In: Proceedings of the 26th Conference on l'Interaction Homme-Machine, 2014.	-

Source: Author.

7.4 Limitations

This thesis has potential limitations regarding its proposed solutions: the CORRELATE process and the LEAD catalog.

The CORRELATE process has not been evaluated by other users regarding easy of use. Although the proposed process was built with the concern to deeply describe its steps and approaches, improvements can still be made. Additionally, CORRELATE is performed manually and thus can be considered extensive.

The established correlations in LEAD also have limitations. They were defined based on the interviewed developers' current knowledge about the strategies. However, such strategies may change and so these correlations. For example, a strategy that can hurt Privacy today may no longer hurt in the future. However, it is natural that the knowledge is constantly evolving, then LEAD can be updated.

7.5 Future Work

This research brings future work that is described taking into account the following three topics: *(i)* the CORRELATE process, presented in Chapter 4; *(ii)* the LEAD catalog, presented in Chapter 5; and *(iii)* the exploratory study through the SM, presented in Chapter 3.

First, regarding the proposed process, the following future work related to this can be defined:

- To execute CORRELATE for the other characteristics in AMICCaS: Calmness, Mobility, Attention and Context-Awareness. Investigating the impact of these others characteristics in user interaction quality;
- To execute CORRELATE for other quality characteristics out of the scope of this thesis. Then, AMICCaS could be replaced for another set of characteristics;
- To execute CORRELATE considering other quality characteristics out of the scope of user interaction. During the execution of CORRELATE, it was possible to see that developers cited Maintainability, although that was not the thesis focus. Therefore, CORRELATE could be executed to investigate this kind of correlations; and
- To obtain qualitative and quantitative feedback from other people using CORRELATE to define an NFR Catalog.

Two approaches were defined for supporting CORRELATE process: ARRANGE and TRACE. ARRANGE is based on Literature and Grounded Theory. However, ARRANGE is only one possibility. Other sources of knowledge and techniques can be used. For example, existing ontologies and existing definitions from dictionaires such as Oxford[1] can be use to define catalogs. Therefore, future work can be performed to see the possibility of establishing complementary approaches.

Additionally, steps 1 and 3 use questionnaire as technique to extract information since it is possible to collect large amounts of data in a short time. However, other techniques could be used, such as interview, literature review. Then, future work can be done to investigate the usage of other techniques.

[1] https://www.oxfordlearnersdictionaries.com/us/

In the same way, TRACE was defined by using interviews with developers and content analysis. However, correlations can be discovered from literature or even existing systems. A possible future work is investigating other ways of identifying correlations. An interesting idea is to see the possibility of automatizing the discovering of correlations.

Second, regarding future work resulted from the LEAD catalog, it is possible to delineate the following opportunities:

- To investigate LEAD in a real context of usage, which means make the proposed catalog available to developers working in real projects and then collect qualitative feedback about the usefulness of the catalog;
- To investigate through a controlled experiment if LEAD is useful even for developers who are experienced in IoT and UbiComp development. The investigation performed in this thesis include novice participants, whose experience in IoT is low. However, evaluating the catalog with experienced developers can bring different results;
- To make the catalog available online through a tool where developers can navigate in the catalog. The catalog is proposed in a table format, and a repository of correlations available to the community can facilitate the usage and possible updates of the proposal;
- To investigate the correlations established by LEAD with other sources of knowledge, such as literature;
- One way to visualize correlations is through SIGs. However, when there are many correlations of one softgoal with several others, such as in LEAD, viewing this information can be very difficult or even impossible. In this work, correlations were stored in a table. Then, future work is to investigate the visualization of data when a SIG model is extensive; and
- To investigate the use of the catalog regarding teaching ubiquitous systems. An interesting idea is to see how much students who used the catalog learned about UbiComp and IoT.

Finally, regarding future work from the systematic mapping presented in Chapter 3. Through this study, a high-level understanding of the literature regarding NFRs catalog could be obtained. This scenario represents a valuable addition to the literature, since it lacks studies summarizing NFRs catalogs. Four opportunities were defined. One of them is regarding the definition of catalogs, which is treated in this thesis. The rest of them became future work of this thesis, described as follows.

- To investigate more deeply the identified correlations from the SM (473 positive correlations and 395 negative correlations). There can be duplicated correlations, then, an analysis should be performed in order to exclude them and possibly construct a general catalog;

- To investigate the correlation's representation. It is interesting to study other notations and ways of representing this knowledge and then understand what can be better to support decision-making in a faster way; and
- To define guidelines about how to evaluate NFRs catalogs. Few of them presents detailed evaluation procedures. Thus, a dedicated research work about how to evaluate is a useful future work.

Another future work that is related to this general picture of NFRs Catalogs is to conduct a study in the industry to understand the usage of NFRs catalogs in real scenarios. Then, the current challenges and issues the practitioners have faced in their work could be investigated.

Furthermore, this systematic mapping generated a dataset containing more than 1000 subcharacteristics, 1113 implementation and design methods, 473 positive correlations, 395 negative correlations for 86 NFRs. However, additional searches to possibly expand the dataset of NFRs catalogs can be performed.

BIBLIOGRAPHY

ABDULRAZAK, B.; MALIK, Y. Review of challenges, requirements, and approaches of pervasive computing system evaluation. **IETE Technical Review**, v. 29, n. 6, p. 506, 2012.

ABOWD, G.; MYNATT, E. Designing for the human experience in smart environments. **Smart environments: technologies, protocols, and applications**, Wiley Online Library, p. 151–174, 2004.

ABOWD, G. D. Software design issues for ubiquitous computing. In: IEEE. **Proceedings IEEE Computer Society Workshop on VLSI'98 System Level Design**. [S.l.], 1998. p. 104–109.

ABOWD, G. D.; MYNATT, E. D.; RODDEN, T. The human experience [of ubiquitous computing]. **IEEE pervasive computing**, IEEE, v. 1, n. 1, p. 48–57, 2002.

AFREEN, N.; KHATOON, A.; SADIQ, M. A taxonomy of software's non-functional requirements. In: SPRINGER. **International Conference on Computer and Communication Technologies**. [S.l.], 2016.

ALAN, A. T.; SHANN, M.; COSTANZA, E.; RAMCHURN, S. D.; SEUKEN, S. It is Too Hot: An In-Situ Study of Three Designs for Heating. In: **Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems**. [S.l.: s.n.], 2016. p. 5262–5273.

ANDRADE, R.; CARVALHO, R.; ARAÚJO, I. de; OLIVEIRA, K.; MAIA, M. What changes from ubiquitous computing to internet of things in interaction evaluation? In: **International Conference on Distributed, Ambient, and Pervasive Interactions**. [S.l.: s.n.], 2017.

ANDREOPOULOS, B. Satisficing the Conflicting Software Qualities of Maintainability and Performance at the Source Code Level. In: **WER**. [S.l.: s.n.], 2004.

ARAGAO, A.; MORENO, N.; VIANA, D.; SILVA, F.; SOUSA, T.; RIVERO, L.; TELES, A.; CONCEIçãO, A. da; COSTA, I. Utilizando métricas de qualidade na avaliação de uma aplicação para cidades inteligentes. In: **Anais do IV Workshop sobre Aspectos Sociais, Humanos e Econômicos de Software**. Porto Alegre, RS, Brasil: SBC, 2019. p. 51–60. Disponível em: <https://sol.sbc.org.br/index.php/washes/article/view/6409>.

ASHRAF, M. U.; KHAN, N. A. Software engineering challenges for ubiquitous computing in various applications. In: IEEE. **2013 11th International Conference on Frontiers of Information Technology**. [S.l.], 2013. p. 78–82.

ASHTON, K. That "internet of things" thing. **RFID journal**, Jun, v. 22, n. 7, p. 97–114, 2009.

BADAMPUDI, D.; WOHLIN, C.; PETERSEN, K. Experiences from using snowballing and database searches in systematic literature studies. In: ACM. **Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering**. [S.l.], 2015. p. 17.

BARDLN, L. Análise de conteúdo. **Lisboa: edições**, v. 70, p. 225, 1977.

BEIGL, M.; GELLERSEN, H.-w.; SCHMIDT, A. MediaCups: Experience with Design and Use of Computer- Augmented Everyday Artefacts. **Computer Networks**, v. 35, n. 4, p. 401–409, 1998.

BERANDER, P. *et al.* Software quality attributes and trade-offs. **Blekinge Institute of Technology**, 2005.

BEZERRA, C. I. M. *et al.* Challenges for Usability Testing in Ubiquitous System. In: **l'Interaction Homme-Machine**. [S.l.: s.n.], 2014.

BODEI, C.; DEGANO, P.; FERRARI, G.-L.; GALLETTA, L.; MEZZETTI, G. Formalising security in ubiquitous and cloud scenarios. In: SPRINGER. **IFIP International Conference on Computer Information Systems and Industrial Management**. [S.l.], 2012.

BOEHM, B.; IN, H. Identifying Quality Requirement Conflicts. **IEEE Software**, 1996.

BROCK, D. L. The Electronic Product Code (EPC): A Naming Scheme for Physical Objectsc. **Auto-ID Center White Paper MIT-AUTOID-WH-002**, p. 1–21, 2001.

BROWN, A.; MORTIER, R.; RODDEN, T. MultiNet: Reducing interaction overhead in domestic wireless networks. **Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13**, p. 1569–1578, 2013.

BURGESS, C.; KRISHNA, A.; JIANG, L. Towards Optimising Non-Functional Requirements. In: **International Conference on Quality Software**. [S.l.: s.n.], 2009.

CAMPBELL, R.; AL-MUHTADI, J.; NALDURG, P.; SAMPEMANE, G.; MICKUNAS, M. D. Towards security and privacy for pervasive computing. In: SPRINGER. **International Symposium on Software Security**. [S.l.], 2002. p. 1–15.

CAPPELLI, C.; CUNHA, H.; GONZALEZ, B.; CESAR, J. Transparency versus Security: Early Analysis of Antagonistic Requirements. In: **ACM Symposium on Applied Computing**. [S.l.: s.n.], 2010. p. 298–305. ISBN 9781605586380.

CARDOSO, E.; GUIZZARDI, R. A Method for Eliciting Goals for Business Process Models based on Non-Functional Requirements Catalogues. n. April, 2011.

CARLETTA, J. Assessing agreement on classification tasks: the kappa statistic. **Computational linguistics**, MIT Press, v. 22, n. 2, p. 249–254, 1996.

CARTER, S. A.; MANKOFF, J. **Challenges for ubicomp evaluation**. [S.l.]: Computer Science Division, University of California, 2004.

CARVALHO, R. Dealing with Conflicts between Non-functional Requirements of UbiComp and IoT Applications. In: **Proceedings - 2017 IEEE 25 International Requirements Engineering Conference, RE 2017**. [S.l.: s.n.], 2017.

CARVALHO, R. *et al.* Evaluating an IoT application using software measures. In: **International Conference on Distributed, Ambient, and Pervasive Interactions**. [S.l.: s.n.], 2017.

CARVALHO, R. M.; ANDRADE, R. M. C.; OLIVEIRA, K. M. Towards a Catalog of Conflicts for HCI Quality Characteristics in UbiComp and IoT Applications : Process and First Results. In: **IEEE International Conference on Research Challenges in Information Science**. [S.l.: s.n.], 2018.

CARVALHO, R. M.; ANDRADE, R. M. d. C.; OLIVEIRA, K. M.; SANTOS, I.; BEZERRA, C. I. M. Quality characteristics and measures for human–computer interaction evaluation in ubiquitous systems. **Software Quality Journal**, Springer, v. 25, n. 3, p. 743–795, 2017.

CARVALHO, R. M.; ANDRADE, R. M. d. C.; OLIVEIRA, K. M. de. AQUArIUM - A suite of software measures for HCI quality evaluation of ubiquitous mobile applications. **Journal of Systems and Software**, v. 136, p. 101–136, 2018.

CARVALLO, J. P. Improving Quality Model Construction Through Knowledge Reuse. n. January, 2015.

CARY, E.; JUMPELT, R. Internet of Things (IoT): In a Way of Smart World. In: **Proceedings of the International Congress on Translation**. [S.l.: s.n.], 2016. v. 6, n. 1, p. 9–9.

CHALLOO, R.; OLADEINDE, A.; YILMAZER, N.; OZCELIK, S.; CHALLOO, L. An overview and assessment of wireless technologies and coexistence of ZigBee, bluetooth and wi-fi devices. **Procedia Computer Science**, v. 12, p. 386–391, 2012.

CHARMAZ, K. **Constructing grounded theory: A practical guide through qualitative analysis**. [S.l.]: sage, 2006.

CHO, J. Y.; LEE, E.-H. Reducing confusion about grounded theory and qualitative content analysis: Similarities and differences. **The qualitative report**, v. 19, n. 32, p. 1–20, 2014.

CHONG, M. K.; MAYRHOFER, R.; GELLERSEN, H. A survey of user interaction for spontaneous device association. **ACM Computing Surveys (CSUR)**, v. 47, n. 1, p. 8, 2014.

CHUNG, L.; NIXON, B. A.; YU, E. Using Non-Functional Requirements to Systematically Select Among Alternatives in Architectural Design. In: **Workshop on Architectures for Software Systems**. [S.l.: s.n.], 1995.

CHUNG, L.; NIXON, B. A.; YU, E.; MYLOPOULOS, J. **Non-functional requirements in software engineering**. [S.l.]: Springer Science and Business Media, 2000. v. 5.

CLELAND-HUANG, J.; SOCIETY, I. C.; MARRERO, W. Goal-Centric Traceability : Using Virtual Plumblines to Maintain Critical Systemic Qualities. n. November, 2008.

CONTE, T.; CABRAL, R.; TRAVASSOS, G. H. Aplicando grounded theory na análise qualitativa de um estudo de observação em engenharia de software–um relato de experiência. In: **V Workshop" Um Olhar Sociotécnico sobre a Engenharia de Software"(WOSES 2009)**. [S.l.: s.n.], 2009. p. 26–37.

COSTA, C. A.; YAMIN, A. C.; GEYER, C. F. R. Toward a general software infrastructure for ubiquitous computing. **IEEE Pervasive Computing**, v. 7, n. 1, p. 64–73, 2008.

CUNHA, H.; CESAR, J.; LEITE, P. Reusing Non-functional Patterns in i * Modeling. p. 25–32, 2014.

CUNHA, H.; CESAR, J.; LEITE, P.; DUBOC, L.; WERNECK, V. The Challenges of Representing Transparency as Patterns. p. 12–15, 2013.

CYSNEIROS, L. M. Evaluating the Effectiveness of Using Catalogues to Elicit Non-Functional Requirements. **Workshop in Requirements Engineering**, 2007.

CYSNEIROS, L. M.; CLAUDIA, L.; MAR, S.; BREITMAN, K. K. Querying Software Interdependence Graphs. 2009.

CYSNEIROS, L. M.; LEITE, J. C. S. d. P. Integrating Non-Functional Requirements into Data Modeling. In: **IEEE International Symposium on Requirements Engineering**. [S.l.]: IEEE, 1999.

CYSNEIROS, L. M.; LEITE, J. C. S. do P. Nonfunctional requirements: From elicitation to conceptual models. **IEEE transactions on Software engineering**, IEEE, v. 30, n. 5, p. 328–350, 2004.

CYSNEIROS, L. M.; WERNECK, V. M.; KUSHNIRUK, A. Reusable Knowledge for Satisficing Usability Requirements. 2005.

DEY, A. K.; ABOWD, G. D. **Towards a Better Understanding of Context and Context-Awareness**. [S.l.], 1999.

DEY, A. K.; ABOWD, G. D.; SALBER, D. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. **Human-computer interaction**, L. Erlbaum Associates Inc., v. 16, n. 2, p. 97–166, 2001.

ECKHARDT, J.; VOGELSANG, A.; ECKHARDT, J.; VOGELSANG, A.; FERNÁNDEZ, D. M. Are "Non-functional " Requirements really Non-functional? An Investigation of Non-functional Requirements in Practice. In: . [S.l.: s.n.], 2016.

EGYED, A.; GRUNBACHER, P. Identifying requirements conflicts and cooperation: How quality attributes and automated traceability can help. **IEEE Software**, 2004.

FALESSI, D.; JURISTO, N.; WOHLIN, C.; TURHAN, B.; MUNCH, J.; JEDLITSCHKA, A.; OIVO, M. Empirical software engineering experts on the use of students and professionals in experiments. **Empirical Software Engineering**, Springer, v. 23, n. 1, p. 452–489, 2018.

FEITOSA, D.; AMPATZOGLOU, A.; AVGERIOU, P.; NAKAGAWA, E. Y. Investigating quality trade-offs in open source critical embedded systems. In: ACM. **International ACM SIGSOFT Conference on Quality of Software Architectures**. [S.l.], 2015.

FREITAS, N.; FILHO, D.; BARBOSA, E. F. A Requirements Catalog for Mobile Learning Environments. v. 55, n. 16, p. 1266–1271, 2013.

FRIEDEMANN, M.; FLOERKEMEIR, C. From the Internet to the Internet of Things. In: **From Active Data Management to Event-Based Systems and More**. [S.l.: s.n.], 2011. p. 242–259.

GARCIA, F. P. **Evoluindo um Sistema de Monitoramento Passivo Energeticamente Eficiente para Redes de Sensores Sem Fio**. 3–16 p. Tese (Doutorado) — Federal University of Ceara, 2014.

GARCIA-MIRELES, G. A.; RUBIA, M. A. M. de la; GARCIA, F.; PIATTINI, M. Identifying quality characteristic interactions during software development. In: **International Conference on Evaluation of Novel Approaches to Software Engineering**. [S.l.: s.n.], 2015.

GHAZI, P. **FlexiView: A Physics-based Focus+ context Navigation Technique for Requirements Modeling Tools**. Tese (Doutorado) — Universitat Zurich, 2018.

GHAZI, P.; GLINZ, M. An exploratory study on user interaction challenges when handling interconnected requirements artifacts of various sizes. In: IEEE. **2016 IEEE 24th International Requirements Engineering Conference (RE)**. [S.l.], 2016. p. 76–85.

GHAZI, P.; GLINZ, M. An experimental comparison of two navigation techniques for requirements modeling tools. In: IEEE. **2018 IEEE 26th International Requirements Engineering Conference (RE)**. [S.l.], 2018. p. 240–250.

GINER, P.; CETINA, C.; FONS, J.; PELECHANO, V. Implicit interaction design for pervasive workflows. **Personal and Ubiquitous Computing**, Springer-Verlag, v. 15, n. 4, p. 399–408, 2011.

GLIEM, J. A.; GLIEM, R. R. Calculating, interpreting, and reporting cronbach's alpha reliability coefficient for likert-type scales. In: MIDWEST RESEARCH-TO-PRACTICE CONFERENCE IN ADULT, CONTINUING, AND COMMUNITY .... [S.l.], 2003.

GODAU, R. *et al.* Qualitative data analysis software: Maxqda and maxdictio. **Qualitative Research Journal**, RMIT Publishing, v. 4, n. 1, p. 66, 2004.

GORBIN, J.; STRAUSS, A. **Basics of Qualitative Research: Techniques and procedures for developing grounded theory.** 3 ed.. ed. [S.l.]: Sage Publications, 2008. 393 p.

GRAMATICA, M.; LABUNETS, K.; MASSACCI, F.; PACI, F.; TEDESCHI, A. The role of catalogues of threats and security controls in security risk assessment: an empirical study with ATM professionals. In: **International Working Conference on Requirements Engineering: Foundation for Software Quality**. [S.l.: s.n.], 2015.

HAMMANI, F. Z. Survey of non-functional requirements modeling and verification of software product lines. In: **2014 IEEE Eighth International Conference on Research Challenges in Information Science (RCIS)**. [S.l.]: IEEE, 2014. p. 1–6. ISBN 9781479923939.

HILL, T.; SUPAKKUL, S.; CHUNG, L. Confirming and Reconfirming Architectural Decisions on Scalability : A Goal-Driven Simulation Approach. p. 327–336, 2009.

HILL, T.; SUPAKKUL, S.; CHUNG, L. Run-Time Monitoring of System Performance : A Goal-Oriented and System Architecture Simulation Approach. p. 31–40, 2010.

HO, G.; LEUNG, D.; MISHRA, P.; HOSSEINI, A.; SONG, D.; WAGNER, D. Smart locks: Lessons for securing commodity internet of things devices. In: ACM. **Proceedings of the 11th ACM on Asia conference on computer and communications security**. [S.l.], 2016. p. 461–472.

HOLLER, J.; TSIATSIS, V.; MULLIGAN, C.; AVESAND, S.; KARNOUSKOS, S.; BOYLE, D. **From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence**. [S.l.: s.n.], 2014. 352 p.

HORKOFF, J. *et al.* Goal-Oriented Requirements Engineering: A Systematic Literature Map. In: **IEEE 24 International Requirements Engineering Conference**. [S.l.: s.n.], 2016. p. 106–115.

HSIEH, H.-F.; SHANNON, S. E. Three approaches to qualitative content analysis. **Qualitative health research**, Sage publications Sage CA: Thousand Oaks, CA, v. 15, n. 9, p. 1277–1288, 2005.

HU, H.; ZHANG, T.; TAN, Y.; XIANG, H.; FU, C.; FENG, Y. Semantic modelling and automated reasoning of non-functional requirement conflicts in the context of softgoal interdependencies. v. 9, p. 145–156, 2015.

ISHII, H.; WISNESKI, C.; BRAVE, S.; DAHLEY, A.; GORBET, M.; ULLMER, B.; YARIN, P. ambientROOM: Integrating Ambient Media with Architectural Space. In: **CHI 98 Conference Summary on Human Factors in Computing Systems**. [S.l.]: ACM, 1998. (CHI '98), p. 173–174.

ISO/IEC 25010. ISO/IEC 25010. **Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models**, 2011.

JAFARI, S.; MTENZI, F.; O'DRISCOLL, C.; FITZPATRICK, R.; O'SHEA, B. Measuring privacy in ubiquitous computing applications. **Int. J. Digit. Soc**, v. 2, n. 3, p. 547–550, 2011.

JALALI, S.; WOHLIN, C. Systematic literature studies: database searches vs. backward snowballing. In: IEEE. **Proceedings of the 2012 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement**. [S.l.], 2012. p. 29–38.

JAMIESON, S. Likert scales: how to (ab) use them. **Medical education**, v. 38, n. 12, p. 1217–1218, 2004.

JEWELL, M. O.; COSTANZA, E.; KITTLEY-DAVIES, J. Connecting the Things to the Internet : An Evaluation of Four Configuration Strategies for Wi-Fi Devices with Minimal User Interfaces. In: **Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp '15**. [S.l.: s.n.], 2015. p. 767–778.

KAASINEN, E.; KYMÄLÄINEN, T.; NIEMELÄ, M.; OLSSON, T.; KANERVA, M.; IKONEN, V. A User-Centric View of Intelligent Environments: User Expectations, User Experience and User Role in Building Intelligent Environments. **Computers**, v. 2, n. 1, p. 1–33, 2013.

KARAISKOS, D.; KOUROUTHANASSIS, P.; GIAGLIS, G. M. Towards a Validated Construct for Information Systems Pervasiveness: An Exploratory Assessment. In: **BLED Proceedings. Paper 12**. [S.l.: s.n.], 2009.

KARVONEN, H.; KUJALA, T. Designing and Evaluating Ubicomp Characteristics of Intelligent In-Car Systems. In: **5 International Conference on Applied Human Factors and Ergonomics**. [S.l.: s.n.], 2014. p. 1726–1737.

KEMP, E.; THOMPSON, A.-J.; JOHNSON, R. Interface evaluation for invisibility and ubiquity: an example from e-learning. In: ACM. **Proceedings of the 9th ACM SIGCHI New Zealand Chapter International Conference on Human-Computer Interaction: Design Centered HCI**. [S.l.], 2008. p. 31–38.

KITCHENHAM, B. A.; CHARTERS, S. Guidelines for Performing Systematic Literature Reviews in Software Engineering. In: EBSE TECHNICAL REPORT. **Technical Report. EBSE-2007-01, Keele University,**. [S.l.], 2007.

KO, I.-Y.; KOO, H.-M.; JIMENEZ-MOLINA, A. User-centric web services for ubiquitous computing. In: **Advanced techniques in web intelligence-I**. [S.l.]: Springer, 2010. p. 167–189.

KOUROUTHANASSIS, P. E.; GIAGLIS, G. M.; KARAISKOS, D. C. Delineating the degree of 'pervasiveness' in Pervasive Information Systems: An assessment framework and design implications. In: **Pan-Hellenic Conference on Informatics**. [S.l.: s.n.], 2008.

KUMAR, P.; SUBRAMANIAN, N.; ZHANG, K. Evaluation of Information Visualization Tools Using the NFR Approach. p. 44–55, 2008.

LEAL, A. L. de C.; BRAGA, J. L.; CRUZ, S. M. S. da. Cataloguing provenance-awareness with patterns. In: IEEE. **2015 IEEE Fifth International Workshop on Requirements Patterns (RePa)**. [S.l.], 2015. p. 9–16.

LEE, J. S.; SU, Y. W.; SHEN, C. C. A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi. In: **Industrial Electronics Society, 2007. IECON 2007. 33 Annual Conference of the IEEE**. [S.l.: s.n.], 2007. p. 46–51.

LEITE, J. C. S. d. P.; CAPPELLI, C. Software transparency. **Business and Information Systems Engineering**, Springer, v. 2, n. 3, p. 127–139, 2010.

LIU, C.-L. CDNFRE: Conflict detector in non-functional requirement evolution based on ontologies. **Computer Standards and Interfaces**, Elsevier, 2016.

LÓPEZ, C.; INOSTROZA, P.; CYSNEIROS, L. M.; ASTUDILLO, H. Visualization and comparison of architecture rationale with semantic web technologies. **Journal of Systems and Software**, Elsevier, v. 82, n. 8, p. 1198–1210, 2009.

MACASAET, R. J.; NOGUERA, M.; RODRÍGUEZ, M. L.; GARRIDO, J. L.; SUPAKKUL, S.; CHUNG, L. A Requirements-Based Approach for Representing Micro-business Patterns.

MAIA, M. E.; ROCHA, L. S.; ANDRADE, R. M. Requirements and challenges for building service-oriented pervasive middleware. **International conference on Pervasive services - ICPS**, ACM Press, 2009.

MAIA, M. E. F.; FONTELES, A.; NETO, B.; GADELHA, R.; VIANA, W.; ANDRADE, R. M. C. LOCCAM - Loosely Coupled Context Acquisition Middleware. In: **Proceedings of the 28 Annual ACM Symposium on Applied Computing**. [S.l.: s.n.], 2013. p. 534–541.

MAIA, M. E. F. *et al.* USABle - A communication framework for ubiquitous systems. **Proceedings - International Conference on Advanced Information Networking and Applications, AINA**, n. 481417, p. 81–88, 2014.

MAIRIZA, D.; ZOWGHI, D. An ontological framework to manage the relative conflicts between security and usability requirements. **International Workshop on Managing Requirements Knowledge**, 2010.

MAIRIZA, D.; ZOWGHI, D. Constructing a Catalogue of Conflicts among Non-functional Requirements. **Communications in Computer and Information Science**, 2011. ISSN 18650929.

MAIRIZA, D.; ZOWGHI, D.; GERVASI, V. Conflict characterization and analysis of non functional requirements: An experimental approach. In: IEEE. **International Conference on Intelligent Software Methodologies, Tools and Techniques**. [S.l.], 2013.

MAIRIZA, D.; ZOWGHI, D.; NURMULIANI, N. Managing conflicts among non-functional requirements. In: UNIVERSITY OF TECHNOLOGY, SYDNEY. **Australian Workshop on Requirements Engineering**. [S.l.], 2009.

MAIRIZA, D.; ZOWGHI, D.; NURMULIANI, N. Towards a catalogue of conflicts among non-functional requirements. In: SCITEPRESS. **International Conference on Evaluation of Novel Approaches to Software Engineering**. [S.l.], 2010.

MASHAL, I.; ALSARYRAH, O.; CHUNG, T.-Y.; YANG, C.-Z.; KUO, W.-H.; AGRAWAL, D. P. Choices for interaction with things on internet and underlying issues. **Ad Hoc Networks**, Elsevier, v. 28, p. 68–90, 2015.

MEHTA, R.; WANG, H.; CHUNG, L. Dealing with NFRs for Smart-Phone Applications : A Goal-Oriented Approach. In: **Software Engineering Research, Management and Applications**. [S.l.: s.n.], 2012.

MINERVA, R.; BIRU, A.; ROTONDI, D. **Towards a definition of the Internet of Things (IoT)**. [S.l.], 2015. 1–86 p.

MONTAGUD, S.; ABRAHÃO, S.; INSFRAN, E. A systematic review of quality attributes and measures for software product lines. **Software Quality Journal**, Springer, v. 20, n. 3-4, p. 425–486, 2012.

MONTONI, M. A. Uma investigação sobre os fatores críticos de sucesso em iniciativas de melhoria de processos de software. **UFRJ/COPPE**, 2010.

MORAN, S.; NAKATA, K. Analysing the factors affecting users in intelligent pervasive spaces. **Intelligent Buildings International**, v. 2, n. 1, p. 57–71, 2010.

MOTTA, R. C. **ON CONTEXT-AWARE SOFTWARE SYSTEMS AND INTEROPERABILITY: A DISCUSSION GROUNDED IN DATA**. Tese (Doutorado), 2016.

MYLOPOULOS, J.; CHUNG, L.; LIAO, S.; WANG, H.; YU, E. Exploring alternatives during requirements analysis. n. February, 2001.

NETO, P. A. d. M. S.; MACHADO, I. do C.; MCGREGOR, J. D.; ALMEIDA, E. S. D.; MEIRA, S. R. de L. A systematic mapping study of software product lines testing. **Information and Software Technology**, Elsevier, v. 53, n. 5, p. 407–423, 2011.

NGUYEN, X.; TRAN, H.; BARAKI, H.; GEIHS, K. Optimization of non-functional properties in Internet of Things applications. **Journal of Network and Computer Applications**, Elsevier Ltd, v. 89, n. September 2016, p. 120–129, 2016.

NIELSEN, J. **Usability engineering**. [S.l.]: Elsevier, 1994.

NIXON, B. A. Management of Performance Requirements for Information Systems. **IEEE Transactions on Software Engineering**, v. 26, n. 12, p. 1122–1146, 2000.

OATES, B. J. **Researching information systems and computing**. [S.l.]: Sage, 2005.

OH, J.; LEE, U.; LEE, K. Usability evaluation model for biometric system considering privacy concern based on mcdm model. **Security and Communication Networks**, Hindawi, v. 2019, 2019.

PABLO, J.; VEGA, C.; LARGA, C.; CAPAC, H.; FRANCH, X.; QUER, C. Towards a Unified Catalogue of Non-Technical Quality Attributes to Support COTS-Based Systems Lifecycle Activities. 2007.

PASHAZADEH, S. Modeling Non Functional Requirements in Designing Middleware for Pervasive Healthcare System. **2011 5th International Conference on Application of Information and Communication Technologies (AICT)**, IEEE, p. 1–5, 2011.

PERERA, C.; ZASLAVSKY, A.; CHRISTEN, P.; GEORGAKOPOULOS, D. Context aware computing for the internet of things: A survey. **IEEE Communications Surveys and Tutorials**, v. 16, n. 1, p. 414–454, 2014.

PETERSEN, K.; FELDT, R.; MUJTABA, S.; MATTSSON, M. Systematic Mapping Studies in Software Engineering. In: **Proceedings of the 12 International Conference on Evaluation and Assessment in Software Engineering**. [S.l.]: British Computer Society, 2008. (EASE'08), p. 68–77.

PETERSEN, K.; VAKKALANKA, S.; KUZNIARZ, L. Guidelines for conducting systematic mapping studies in software engineering: An update. **Information and Software Technology**, Elsevier B.V., v. 64, p. 1–18, 2015.

POPPE, R.; RIENKS, R.; DIJK, B. V. Evaluating the Future of HCI: Challenges for the Evaluation of Emerging Applications. **Artifical Intelligence for Human Computing. Lecture Notes in Computer Science**, Springer, v. 4451, p. 234–250, 2007.

PORTUGAL, R. L.; LI, T.; SILVA, L.; ALMENTERO, E.; LEITE, J. C. S. P. NFRFinder : A Knowledge Based Strategy for Mining Non-Functional Requirements. In: . [S.l.: s.n.], 2018.

POSLAD, S. **Ubiquitous Computing: Smart Devices, Environments and Interactions**. [S.l.]: Wiley, 2009.

PRODANOV, C. C.; FREITAS, E. C. de. **Metodologia do trabalho científico: métodos e técnicas da pesquisa e do trabalho acadêmico-2ª Edição**. [S.l.]: Editora Feevale, 2013.

RANGANATHAN, A.; AL-MUHTADI, J.; BIEHL, J.; ZIEBART, B.; CAMPBELL, R. H.; BAILEY, B. Towards a pervasive computing benchmark. In: IEEE. **Third IEEE International Conference on Pervasive Computing and Communications Workshops**. [S.l.], 2005. p. 194–198.

RAZALI, N. M.; WAH, Y. B. *et al.* Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests. **Journal of statistical modeling and analytics**, v. 2, n. 1, p. 21–33, 2011.

RILSTON, F.; PAIM, S.; CASTRO, J. F. B. Enhancing Data Warehouse Design with the NFR Framework. In: **WER**. [S.l.: s.n.], 2002.

ROWLAND, C.; GOODMAN, E.; CHARLIER, M.; LIGHT, A.; LUI, A. **Designing Connected Products**. First edit. [S.l.]: O'Reilly Media, Inc, 2015.

RYU, H.; HONG, G.; JAMES, H. Quality assessment technique for ubiquitous software and middleware. Massey University, 2006.

SADANA, V.; LIU, X. F. Analysis of Conflicts among Non-Functional Requirements Using Integrated Analysis of Functional and Non-Functional Requirements. n. Compsac, p. 0–3, 2007.

SADI, M. H.; YU, E. Accommodating Openness Requirements in Software Platforms: A Goal-Oriented Approach. In: **International Conference on Advanced Information Systems Engineering**. [S.l.: s.n.], 2017.

SAHA, D.; MUKHERJEE, A. Pervasive computing: A paradigm for the 21 century. **IEEE Computer Society**, v. 36, n. 3, p. 25–31, 2003.

SALMAN, I.; MISIRLI, A. T.; JURISTO, N. Are students representatives of professionals in software engineering experiments? In: IEEE. **2015 IEEE/ACM 37th IEEE International Conference on Software Engineering**. [S.l.], 2015. v. 1, p. 666–676.

SANTOS, B. P.; SILVA, L. A. M.; CELES, C. S. F. S.; PERES, B. S.; VIEIRA, M. A. M.; VIEIRA, L. F. M. **Internet das Coisas: da Teoria a Pratica.** [S.l.], 2016.

SANTOS, I. **TESTDAS: Testing Method for Dynamically Adaptive Systems**. Tese (Doutorado) — Federal University of Ceará, 2018.

SANTOS, I.; ANDRADE, R. M. C.; ROCHA, L. S.; MATALONGA, S.; OLIVEIRA, K. M.; TRAVASSOS, G. H. Test Case Design for Context-Aware Applications: Are We There Yet? **Information and Software Technology**, Elsevier, 2017.

SANTOS, R.; OLIVEIRA, K.; ANDRADE, R.; SANTOS, I.; LIMA, E. A quality model for human-computer interaction evaluation in ubiquitous systems. In: SPRINGER. **Latin American conference on human computer interaction**. [S.l.], 2013. p. 63–70.

SARMIENTO, E.; CESAR, J.; ALMENTERO, E. Using Correctness , Consistency , and Completeness Patterns for Automated Scenarios Verification. 2015.

SATYANARAYANAN, M. Pervasive computing: vision and challenges. **IEEE Personal Communications**, 2001.

SCHILIT, B. N.; THEIMER, M. M. Disseminating active map information to mobile hosts. **IEEE Network**, v. 8, n. 5, p. 22–32, 1994.

SCHMIDT, A. Interactive Context-Aware Systems Interacting with Ambient Intelligence. **Ambient intelligence**, p. 159–178, 2005.

SCHOLTZ, J.; CONSOLVO, S. Toward a Framework for Evaluating Ubiquitous Computing Applications. **IEEE Pervasive Computing**, v. 3, n. 2, p. 82–88, 2004.

SERRANO, M. Ubiquitous , Pervasive and Mobile Computing : A Reusable-Models-based Non-Functional Catalogue Objectives of Research. In: **ER@ BR**. [S.l.: s.n.], 2013.

SERRANO, M.; LEITE, J. C. S. d. P. Capturing transparency-related requirements patterns through argumentation. In: IEEE. **Requirements Patterns (RePa), 2011 First International Workshop on**. [S.l.], 2011. p. 32–41.

SHAFER, S.; BRUMITT, B.; CADIZ, J. Interaction issues in context-aware intelligent environments. **Human-Computer Interaction**, L. Erlbaum Associates Inc., v. 16, n. 2, p. 363–378, 2001.

SILVA, A.; PINHEIRO, P.; ALBUQUERQUE, A.; BARROSO, J. A Process for Creating the Elicitation Guide of Non-functional Requirements. In: **Software Engineering Perspectives and Application in Intelligent Systems**. [S.l.]: Springer, 2016. p. 293–302.

SILVA, C.; PINTO, R.; CASTRO, J.; TEDESCO, P. Requirements for Multi-Agent Systems. In: **WER**. [S.l.: s.n.], 2003.

SILVA, D. P.; SOUZA, P. C.; MACIEL, C. Establishing guidelines for user quality of experience in ubiquitous systems. In: SPRINGER. **International Conference on Distributed, Ambient, and Pervasive Interactions**. [S.l.], 2016. p. 46–57.

SOAD, G. W.; BARBOSA, E. F. Quality Evaluation of Mobile Learning Applications. **IEEE Frontiers in Education Conference (FIE)**, 2016.

SOKOLOVA, M.; LAPALME, G. A systematic analysis of performance measures for classification tasks. **Information processing & management**, Elsevier, v. 45, n. 4, p. 427–437, 2009.

SOLINGEN, R. V.; BASILI, V.; CALDIERA, G.; ROMBACH, H. D. Goal question metric (gqm) approach. **Encyclopedia of software engineering**, Wiley Online Library, 2002.

SOMMERVILLE, I.; SAWYER, P. **Requirements Engineering: A Good Practice Guide**. 1. ed. [S.l.]: John Wiley and Sons, Inc., 1997.

SOUSA, G. M. C. D.; CASTRO, J. B. D. Towards a Goal-Oriented Requirements Methodology Based on the Separation of Concerns Principle.

SUBRAMANIAN, N.; CHUNG, L. SA3 A tool for supporting adaptable software architecture generation for embedded systems. v. 25, p. 283–290, 2003.

SUBRAMANIAN, N.; DRAGER, S.; MCKEEVER, W. Designing Trustworthy Software Systems Using the NFR Approach. In: **Emerging Trends in ICT Security**. Elsevier Inc., 2014. cap. Chapter 13, p. 203–226. ISBN 9780124114746. Disponível em: <http://dx.doi.org/10.1016/B978-0-12-411474-6.00013-X>.

SUOMALAINEN, J. Smartphone assisted security pairings for the Internet of Things. **2014 4 International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace and Electronic Systems, VITAE 2014 - Co-located with Global Wireless Summit**, 2014.

SWAN, M. Sensor Mania! The Internet of Things, Wearable Computing, Objective Metrics, and the Quantified Self 2.0. **Journal of Sensor and Actuator Networks**, v. 1, n. 3, p. 217–253, 2012.

TAHVILDARI, L.; KONTOGIANNIS, K. A software transformation framework for quality-driven object-oriented re-engineering. **International Conference on Software Maintenance, 2002. Proceedings.**, IEEE, p. 596–605, 2002.

TAN, L.; WANG, N. Future internet: The Internet of Things. **2010 3 International Conference on Advanced Computer Theory and Engineering (ICACTE)**, p. V5–376–V5–380, 2010.

Telecommunication Union, I. **ITU internet reports 2005: The internet of things**. [S.l.], 2005.

THOMPSON, S. G.; AZVINE, B. No Pervasive Computing Without Intelligent Systems. **BT technology journal**, Springer, v. 22, n. 3, p. 39–49, 2004.

TORRES, R. C.; MARTINS, L. E. G. NFR Catalogues for RFID Middleware. **Journal of Computer Science and Technology**, v. 14, n. 2, p. 102–108, 2014.

TOTIYA, S.; SENIVONGSE, T. Framework to Support Cloud Service Selection Based on Service Measurement Index. In: **World Congress on Engineering and Computer Science**. [S.l.: s.n.], 2017. I. ISBN 9789881404756.

UCHÔA, A. G.; LIMA, L. P.; BEZERRA, C. I.; MONTEIRO, J. M.; ANDRADE, R. M. DyMMer-NFP: Modeling Non-functional Properties and Multiple Context Adaptation Scenarios in Software Product Lines. In: **International Conference on Software Reuse**. [S.l.: s.n.], 2017. ISBN 9783319568560.

VALE, L.; ALBUQUERQUE, A. B.; BESERRA, P. V. The importance of professional quality of requirements analysts for success of software development projects: a study to identify the most relevant skills. In: IEEE. **2011 25th Brazilian Symposium on Software Engineering**. [S.l.], 2011. p. 253–262.

VASSEUR, J.-P.; DUNKELS, A. **Interconnecting Smart Objects with IP**. [S.l.]: Morgan Kaufmann, 2010.

VELEDA, R.; CYSNEIROS, L. M. Towards an ontology-based approach for eliciting possible solutions to non-functional requirements. In: SPRINGER. **International Conference on Advanced Information Systems Engineering**. [S.l.], 2019. p. 145–161.

VILLA, L.; CABEZAS, I.; LOPEZ, M.; CASAS, O. Towards a Sustainable Architectural Design by an Adaptation of the Architectural Driven. v. 2, p. 71–86, 2016.

WANG, H.; SUPAKKUL, S.; CHUNG, L. Rule-based Context-aware Adaptation Using a Goal-Oriented Ontology. p. 67–76, 2011.

WEISER, M. The Computer for the 21th Century. **Scientific Americal Special Issue on Communications, Computers and Networks**, p. 94–104, 1991.

WHITMORE, A.; AGARWAL, A.; XU, L. D. The internet of things—a survey of topics and trends. **Information Systems Frontiers**, Springer, v. 17, n. 2, p. 261–274, 2015.

WIEGERS, K.; BEATTY, J. **Software requirements**. [S.l.]: Pearson Education, 2013.

WIERINGA, R.; MAIDEN, N.; MEAD, N.; ROLLAND, C. Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. **Requirements engineering**, Springer, v. 11, n. 1, p. 102–107, 2006.

WIJAYARATHNA, C.; ARACHCHILAGE, N. A. An empirical usability analysis of the google authentication api. In: ACM. **Proceedings of the Evaluation and Assessment on Software Engineering**. [S.l.], 2019. p. 268–274.

WOHLIN, C. Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering. **International Conference on Evaluation and Assessment in Software Engineering (EASE 2014)**, p. 1–10, 2014.

WOHLIN, C.; RUNESON, P.; HÖST, M.; OHLSSON, M. C.; REGNELL, B.; WESSLÉN, A. **Experimentation in software engineering**. [S.l.]: Springer Science and Business Media, 2012.

WOHLIN, C.; RUNESON, P.; NETO, P. A. d. M. S.; ENGSTRÖM, E.; MACHADO, I. do C.; ALMEIDA, E. S. D. On the reliability of mapping studies in software engineering. **Journal of Systems and Software**, Elsevier, v. 86, n. 10, p. 2594–2610, 2013.

YANG, R.; NEWMAN, M. W. Learning from a Learning Thermostat : Lessons for Intelligent Systems for the Home. In: **Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing (UbiComp 2013)**. [S.l.: s.n.], 2013. p. 93–102.

YE, J.; DOBSON, S.; MCKEEVER, S. Situation identification techniques in pervasive computing: A review. **Pervasive and Mobile Computing**, v. 8, n. 1, p. 36–66, 2012.

YRJONEN, A.; MERILINNA, J. Extending the nfr framework with measurable nonfunctional requirements. In: **NFPinDSML@ MoDELS**. [S.l.: s.n.], 2009.

YU, E. S. Towards modelling and reasoning support for early-phase requirements engineering. In: IEEE. **Proceedings of ISRE'97: 3rd IEEE International Symposium on Requirements Engineering**. [S.l.], 1997. p. 226–235.

YU, P.; MA, X.; CAO, J.; LU, J. Application mobility in pervasive computing: A survey. **Pervasive and Mobile Computing**, 2013.

YUE, W.; WANG, .; WANG, G. Designing Transparent Interaction for Ubiquitous Computing. **International Conference on Human-Computer Interaction. Interaction Design and Usability. Lecture Notes in Computer Science.**, v. 4550, p. 331–339, 2007.

ZAMANI, E.; KOUROUTHANASSIS, P.; KARAISKOS, D.; GIAGLIS, G. Exploring the adoption of ubiquitous information systems within the museum context. In: **Proceedings, 6th Mediterranean Conference on Information Systems, Limassol, Cyprus**. [S.l.: s.n.], 2011.

ZEZSCHWITZ, E. V. *et al.* **Human Computer Interaction in the Internet of Things Era - An overview of current trends, developments, and research in human-computer interaction**. [S.l.], 2015.

ZHU, M.-x.; LUO, X.-x.; CHEN, X.-h.; DASH, D. A non-functional requirements tradeoff model in Trustworthy Software. **Information Sciences**, Elsevier Inc., v. 191, p. 61–75, 2012. ISSN 0020-0255. Disponível em: <http://dx.doi.org/10.1016/j.ins.2011.07.046>.

ZHU, Q.; WANG, R.; CHEN, Q.; LIU, Y.; QIN, W. IOT Gateway: BridgingWireless Sensor Networks into Internet of Things. In: **2010 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing**. [S.l.: s.n.], 2010. p. 347–352.

ZINOVATNA, O.; CYSNEIROS, L. M. Reusing knowledge on delivering privacy and transparency together. In: IEEE. **2015 IEEE Fifth International Workshop on Requirements Patterns (RePa)**. [S.l.], 2015. p. 17–24.

APPENDIX A – PRIMARY STUDIES OF THE SM STUDY

This Appendix lists the primary studies analyzed in the Systematic Mapping of NFRs Catalogs in Tables 51 and 52, earlier addressed in Chapter 3.

Table 51 – List of Primary Papers from the SM Study - Part 1/2

PAPER ID	*REFERENCE*	*SOURCE*	*CATALOG ID*
1	(CARVALHO *et al.*, 2018)	Scopus	C1
2	(SOAD; BARBOSA, 2016)	Scopus	C9
3	(VILLA *et al.*, 2016)	Scopus	C10
4	(HU *et al.*, 2015)	Scopus	C12
5	(ZINOVATNA; CYSNEIROS, 2015)	Scopus	C14, C15, C16, C17
6	(SUBRAMANIAN *et al.*, 2014)	Scopus	C20
7	(SERRANO, 2013)	Scopus	C21
8	(MEHTA *et al.*, 2012)	Scopus	C22
9	(MACASAET *et al.*,)	Scopus	C23, C24
10	(ZHU *et al.*, 2012)	Scopus	C25
11	(CARDOSO; GUIZZARDI, 2011)	Scopus	C26, C27, C28, C29, C30
12	(MAIRIZA; ZOWGHI, 2011)	Scopus	C31
13	(WANG *et al.*, 2011)	Scopus	C32, C33, C34
14	(HILL *et al.*, 2010)	Scopus	C35
15	(YRJONEN; MERILINNA, 2009)	Scopus	C36
16	(CYSNEIROS *et al.*, 2009)	Scopus	C37
17	(BURGESS *et al.*, 2009)	Scopus	C38
18	(LÓPEZ *et al.*, 2009)	Scopus	C58
19	(CLELAND-HUANG *et al.*, 2008)	Scopus	C55, C56
20	(CYSNEIROS, 2007)	Scopus	C45, C46, C47, C48, C49, C50, C51, C52, C53
21	(PABLO *et al.*, 2007)	Scopus	C54
22	(TAHVILDARI; KONTOGIANNIS, 2002)	Scopus	C39, C40, C41
23	(SADI; YU, 2017)	WoS	C2, C3, C4, C5, C6
24	(UCHÔA *et al.*, 2017)	WoS	C7
25	(TOTIYA; SENIVONGSE, 2017)	WoS	C8
26	(LEAL *et al.*, 2015)	WoS	C13
27	(SARMIENTO *et al.*, 2015)	WoS	C11
28	(CUNHA *et al.*, 2014)	WoS	C18
29	(CUNHA *et al.*, 2013)	WoS	C19
30	(HILL *et al.*, 2009)	WoS	C42, C43
31	(KUMAR *et al.*, 2008)	WoS	C44
32	(SUBRAMANIAN; CHUNG, 2003)	WoS	C57
33	(NIXON, 2000)	WoS	C59, C60

Source: Author.

Table 52 – List of Primary Papers from the SM Study - Part 2/2

PAPER ID	***REFERENCE***	***SOURCE***	***CATALOG ID***
37	(FEITOSA *et al.*, 2015)	Snowballing	C72, C73
38	(GARCIA-MIRELES *et al.*, 2015)	Snowballing	C74
40	(HAMMANI, 2014)	Snowballing	C77
41	(FREITAS *et al.*, 2013)	Snowballing	C78
39	(ISO/IEC 25010, 2011)	Snowballing	C75, C76
44	(PASHAZADEH, 2011)	Snowballing	C81
42	(SERRANO; LEITE, 2011)	Snowballing	C79
36	(MAIA *et al.*, 2009)	Snowballing	C66, C67, C68, C69, C70, C71
43	(SADANA; LIU, 2007)	Snowballing	C80
35	(BERANDER *et al.*, 2005)	Snowballing	C63, C64, C65
34	(EGYED; GRUNBACHER, 2004)	Snowballing	C61, C62
47	(MYLOPOULOS *et al.*, 2001)	Snowballing	C90, C91
48	(CYSNEIROS; LEITE, 1999)	Snowballing	C92
45	(BOEHM; IN, 1996)	Snowballing	C82, C83, C84
46	(CHUNG *et al.*, 1995)	Snowballing	C85, C86, C87, C88, C89
49	(RILSTON *et al.*, 2002)	WER	C93, C94, C95, C96, C97
50	(ANDREOPOULOS, 2004)	WER	C98, C99
51	(SOUSA; CASTRO,)	WER	C100
52	(SILVA *et al.*, 2003)	WER	C101
53	(CARVALLO, 2015)	WER	C102

Source: Author.

APPENDIX B – PRIMARY STUDIES FOR CHARACTERIZING INVISIBILITY

This Appendix lists the primary studies collected and analyzed in the second step of CORRELATE first execution in Table 53, earlier addressed in Chapter 5. Therefore, it presents the papers collected to refine Invisibility characteristic.

Table 53 – List of Primary Studies for Invisibility Dataset

PAPER ID	***REFERENCE***	***SOURCE***
1	(KOUROUTHANASSIS *et al.*, 2008)	Start Set
2	(KARAISKOS *et al.*, 2009)	Start Set
3	(SCHOLTZ; CONSOLVO, 2004)	Start Set
4	(KEMP *et al.*, 2008)	Start Set
5	(KARVONEN; KUJALA, 2014)	Start Set
6	(THOMPSON; AZVINE, 2004)	Start Set
7	(KO *et al.*, 2010)	Start Set
8	(RYU *et al.*, 2006)	Start Set
9	(SANTOS *et al.*, 2013)	Start Set
10	(BEIGL *et al.*, 1998)	Backward
11	(ABOWD *et al.*, 2002)	Backward
12	(YUE *et al.*, 2007)	Backward
13	(SHAFER *et al.*, 2001)	Backward
14	(SATYANARAYANAN, 2001)	Backward
15	(RANGANATHAN *et al.*, 2005)	Backward
16	(WEISER, 1991)	Backward
17	(CAMPBELL *et al.*, 2002)	Backward
18	(SAHA; MUKHERJEE, 2003)	Backward
19	(ABOWD, 1998)	Backward
20	(ZAMANI *et al.*, 2011)	Forward
21	(ABOWD; MYNATT, 2004)	Forward
22	(CARTER; MANKOFF, 2004)	Forward
23	(MORAN; NAKATA, 2010)	Forward
24	(KAASINEN *et al.*, 2013)	Forward
25	(ABDULRAZAK; MALIK, 2012)	Forward
26	(JAFARI *et al.*, 2011)	Forward
27	(MORAN; NAKATA, 2010)	Forward
28	(ASHRAF; KHAN, 2013)	Forward
29	(SILVA *et al.*, 2016)	Forward
30	(COSTA *et al.*, 2008)	Forward
31	(GINER *et al.*, 2011)	Scopus

Source: Author.

APPENDIX C – EXPERIMENT INSTRUMENTS

C.1 Consent Form

Hello! I am working on a research that aims to improve the satisfaction of non-functional requirements, which is part of my doctoral research. At this stage, I would like to know what and how students deal with this type of activity. Accordingly, I ask your consent to conduct an experiment that collects data about your choices regarding this type of requirement. To decide on giving or not your consent, it is important that you know the following information about the survey:

- Data collected during the experiment is strictly for analysis and development of a solution for a doctoral project;
- I am committed to disseminating the results of this research to the scientific community. The disclosure of these results is based on respect for your privacy, and the anonymity of the participants will be preserved in any documents we prepare;
- Consent to the experiment is a free choice made by providing all necessary clarifications about the research;
- The experiment can be stopped at any time, according to your availability and will;

You can find me available for contact by email: rainara@ufc.br. With this information in mind, I would like you to choose one:

1. I give my consent to its execution.
2. I do not consent to its execution.

____

<Signature of the participant>

____

<Signature of the researcher>

C.2 Background Form

1. Name:
2. What is your experience with quality characteristics?
    a) I have no knowledge about NFRs
    b) I have basic knowledge about NFRs
    c) I have knowledge and experience with NFRs
3. What is your experience with Softgoals Interdependency Graphs (SIGs)?
    a) I have no knowledge about SIGs
    b) I have basic knowledge about SIGs
    c) I have knowledge and experience with SIGs
4. What is your experience with Internet of Things (IoT) and Ubiquitous Computing (UbiComp)?
    a) I have no knowledge about IoT and UbiComp
    b) I have basic knowledge about IoT and UbiComp
    c) I have knowledge and experience with IoT and UbiComp
5. What is your experience with Invisibility characteristic for UbiComp and IoT systems?
    a) I have no knowledge about Invisibility for UbiComp and IoT
    b) I have basic knowledge about Invisibility for UbiComp and IoT
    c) I have knowledge and experience with Invisibility for UbiComp and IoT

C.3 Object 1 - AutomaGREat and Tasks

AutomaGREat proposes an intelligent environment for the Seminar Room of the GREat lab. In the seminar room, lectures, weekly meetings, defenses and other activities happen. Several objects in this room are handled by the group employees who are using the room, such as air conditioners and lights. In this scenario, a development team proposed the AutomaGreat project, in which the goal is to create an application to facilitate the use of room devices: air conditioners and lamps. Thus, users can manipulate these objects remotely through a mobile application. In addition, the system can automate tasks commonly performed in this environment.

The functional requirements of this application are:

1. The system must allow user authentication
2. The system should allow the user to set their preferences regarding air conditioning and lights
3. The system must allow the user to configure the system operation mode: manual or automatic
4. Manual mode should allow air and light control directly by the user.
    a) The system must allow the user to turn on the seminar room air conditioner
    b) The system must allow the user to turn off the seminar room air conditioner
    c) The system must allow the user to change the seminar room air conditioner temperature
    d) The system must allow the user to turn on the seminar room lights
    e) The system shall allow the user to turn off seminar room lights
    f) The system shall allow the brightness of the seminar room to be manipulated
    g) The system must allow the color of the seminar room lights to be changed
5. Automatic mode should allow air conditioners and lamps to be triggered from room presence detection and user preferences

The non-functional requirements of this application are:

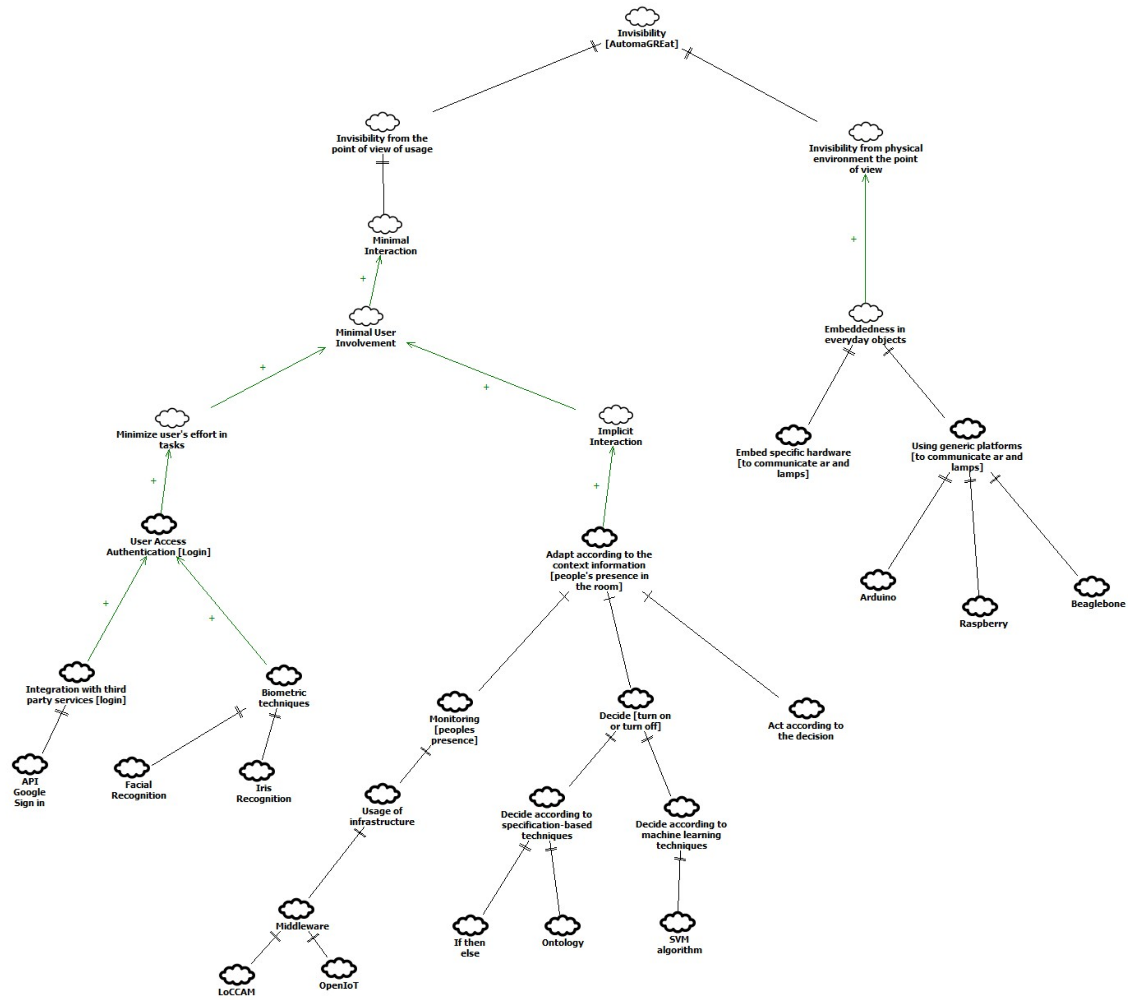
1. Invisibility: refers to merging technology into the user's physical environment or decreasing the interaction workload;
2. Security: degree to which a product or system protects information and data so that people or other products or systems have the degree of access to data appropriate to their types and levels of authorization;
3. Performance: performance against the amount of resources used under established condi-

tions;

4. Interaction efficiency: resources spent on the accuracy and completeness with which users reach goals; and
5. Reliability: degree to which a system, product or component performs specified functions under specified conditions for a specified period of time.

Figure 82 shows the Invisibility SIG constructed for AutomaGREat.

Figure 82 – Invisibility SIG for AutomaGREat System



Source: Author.

Description of the operationalizing softgoals in the last level of the SIG:

1. Google Sign in API: API that allows authentication with Google data;
2. Facial Recognition: Technique to identify the user based on their face;
3. Iris Recognition: Technique to identify the user based on their iris;
4. LoCCAM: Middleware for managing and acquiring context information. It can run on a single device or can be distributed across devices;

5. OpenIoT: A natural extension for cloud computing implementations, allowing access to IoT-based features, and functions as sensor middleware;
6. If-then-else: Modeling and implementation of adaptation decisions;
7. Ontology: Generic, formal and explicit way to capture and specify domain knowledge with its intrinsic semantics through consensual terminology and formal axioms and constraints. Provide a formal way of representing sensor data, context, and situations in well-structured terminology;
8. SVM algorithm: Supervised learning model that analyzes data used for classification and regression analysis.
9. Action: Execution of the decision
10. Embedded hardware: Acting and sensing specific embedded hardware on objects;
11. Arduino: Open source electronic platform based on hardware and software;
12. Raspberry: Small size single card that plugs into a computer monitor or TV and uses a standard keyboard and mouse; and
13. Beaglebone: Low power open source single board computer.

Tasks performed by the participants in AutomaGREat:

- Task 1: Given the set of operationalizations in the last level of the Invisibility SIG, analyze if they have positive and negative impacts with Security, Performance, Efficiency and Reliability.
- Task 2: Choose and describe the operationalizations that maximize the positive impact and minimize the negative impact to the required NFRs.

C.4 Part of the Correlation Catalog for AutomaGREat

This Section lists the part of the correlation catalog received by the participants when they had to perform the experiment's tasks in AutomaGREat

Table 54 – Part of the correlations presented for AutomaGREat tasks

Strategy	Type	Quality Characteristic
Google Sign-in	HELPS	Efficiency
Google Sign-in	HURTS	Privacy
Google Sign-in	HURTS	Security / Confidentiality
Facial Recognition	HELPS	Usability / Accessibility
Facial Recognition	HURTS	Functional Suitability / Functional Correctness
Facial Recognition	HURTS	Privacy
Facial Recognition	HURTS	Performance Efficiency / Time Behavior
Facial Recognition	HURTS	Efficiency
Facial Recognition	HURTS	Security / Authenticity
Iris Recognition	HELPS	Security
Iris Recognition	HELPS	Usability/Accessibility
Iris Recognition	HURTS	Performance Efficiency / Time Behavior
Iris Recognition	HURTS	Efficiency
OpenIoT	HELPS	Functional Suitability
LoCCAM	HELPS	Functional Suitability
LoCCAM	HURTS	Privacy
LoCCAM	HURTS	Performance Efficiency
LoCCAM	HURTS	Security
LoCCAM	HURTS	Reliability
if then else	HURTS	Context Coverage / Flexibility
if then else	HURTS	Reliability
Ontology	HURTS	Performance Efficiency
Arduino	HURTS	Reliability
Arduino	HURTS	Performance efficiency / capacity
Raspberry	HURTS	Reliability
Raspberry	HURTS	Security
Beaglebone	HURTS	Reliability
Beaglebone	HURTS	Security
Embedded hardware	HELPS	Reliability

Source: Author.

C.5 Object 2 - GREatBus and Tasks

GREatBus proposes an intelligent system for passengers and bus drivers. Overall, the project aims to facilitate bus-related tasks. For the driver it is important for example to know if the people who are at the stop will take the bus. For the passenger it is important to know estimates, bus capacity, among others.

The functional requirements of this application are:

1. The system must be able to receive or request information about the number of bus requests per stop;
2. The system shall be able to calculate the estimated bus arrival time based on the distance from the bus to the user and the speed of the vehicle;
3. The system must be able to inform the capacity of the bus; and
4. The system must be able to indicate that at that location there is a passenger requesting the bus.

The non-functional requirements of this application are:

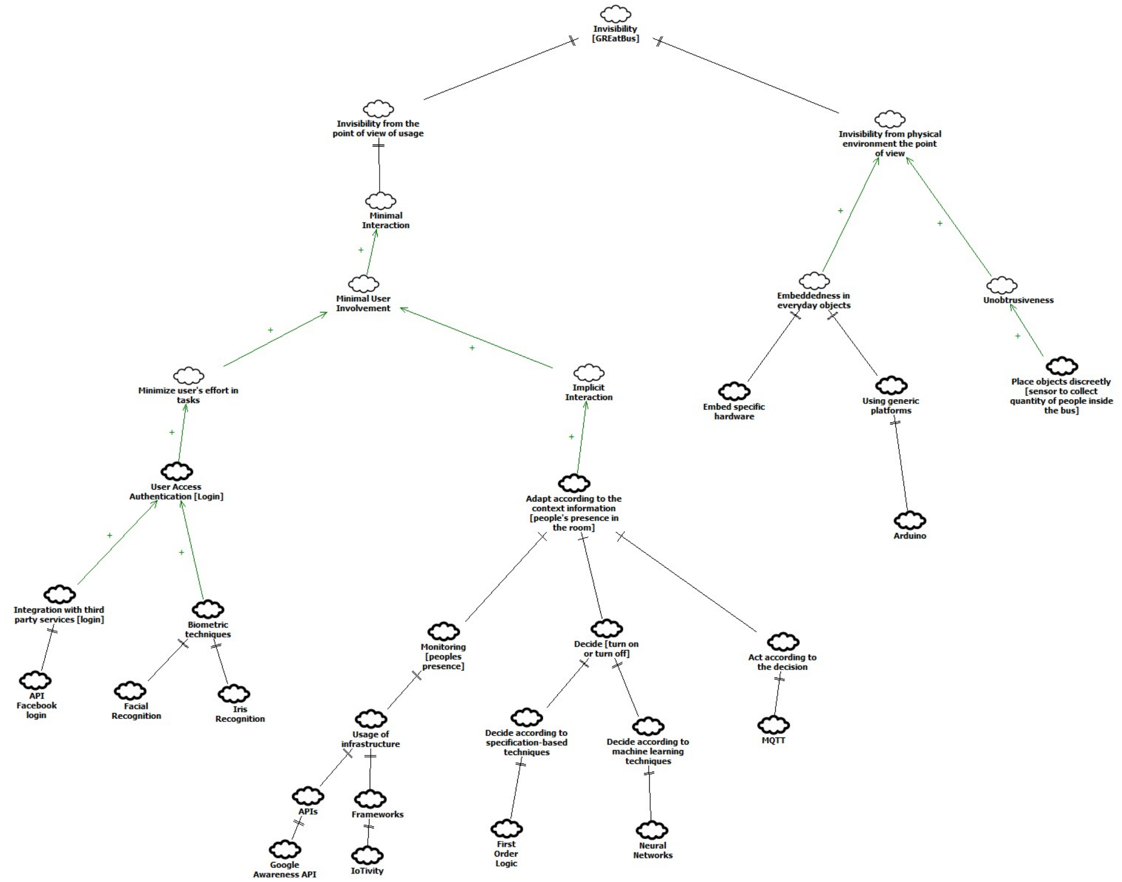
1. Invisibility: refers to merging technology into the user's physical environment or decreasing the interaction workload.
2. Privacy: the state or condition of being free to be observed or disturbed.
3. Accessibility: the degree to which a product or system can be used by people with the widest range of features and capabilities to achieve a specified goal in a specified context of use.

Figure 83 shows an Invisibility SIG constructed for GREatBus.

Description of the operationalizing softgoals:

1. Facebook Log in API: API that allows authentication with Facebook data;
2. Facial Recognition: Technique to identify the user based on their face;
3. Iris Recognition: Technique to identify the user based on their iris;
4. IoTivity: Open source framework that enables device to device connectivity to meet emerging IoT needs;
5. First order logic: Mathematical logic used to specify system states and operators / functions to apply to those states. They provide reasoning support to identify complex contexts and situations;
6. Neural Network: Technique that presents a mathematical model inspired by the neural structure of intelligent organisms that gain knowledge through experience;

Figure 83 – Invisibility SIG for GREatBus System



Source: Author.

7. MQTT: Machine-to-machine (M2M) / "IoT" connectivity protocol. Designed as a publish/subscribe message transport.
8. Embedded hardware: Acting and sensing specific embedded hardware on objects;
9. Arduino: Open source electronic platform based on hardware and software;
10. Place objects discreetly: If hardware devices cannot be fully hidden, they must be discreetly placed in the physical area. Therefore, places where the user does not need to perform actions such as wall and roof corners are ideal.

Tasks performed by the participants in GREatBus:

- Task 1: Given the set of operationalizations in the last level of the Invisibility SIG, analyze if they have positive and negative impact with Privacy and Accessibility.
- Task 2: Choose and describe the operationalizations that maximize the positive impact and minimize the negative impact to the required NFRs.

C.6 Part of the Correlation Catalog for GREatBus

This Section lists the part of the correlation catalog received by the participants when they had to perform the experiment's tasks in GREatBus

Table 55 – Part of the correlations presented for GREatBus tasks

Strategy	Type	Quality Characteristic
Facebook Log-in	HELPS	Efficiency
Facebook Log-in	HURTS	Privacy
Facebook Log-in	HURTS	Security / Confidentiality
Facial Recognition	HELPS	Usability / Accessibility
Facial Recognition	HURTS	Functional Suitability / Functional Correctness
Facial Recognition	HURTS	Privacy
Facial Recognition	HURTS	Performance Efficiency / Time Behavior
Facial Recognition	HURTS	Efficiency
Facial Recognition	HURTS	Security / Authenticity
Iris Recognition	HELPS	Security
Iris Recognition	HELPS	Usability/Accessibility
Iris Recognition	HURTS	Performance Efficiency / Time Behavior
Iris Recognition	HURTS	Efficiency
Awareness	HELPS	Functional Suitability
Awareness	HURTS	Privacy
IoTivity	HELPS	Functional Suitability
First order logic	HURTS	Performance Efficiency
Neural Network	HELPS	Efficiency
Neural Network	HELPS	Performance Efficiency
Neural Network	HELPS	Context Coverage / Flexibility
Neural Network	HURTS	Usability / Learnability
MQTT	HELPS	Performance Efficiency
Embedded hardware	HELPS	Reliability
Place objets discret	HELPS	Satisfaction
Place objets discret	HURTS	Usability / Operability
Arduino	HURTS	Reliability
Arduino	HURTS	Performance efficiency / capacity

Source: Author.

C.7 Post Experiment Questionnaire

1. The training was enough to complete my tasks
    a) Strongly Agree
    b) Partially Agree
    c) Neither Agree nor Disagree
    d) Partially Disagree
    e) Strongly Disagree
2. The goals of the tasks were clear to me
    a) Strongly Agree
    b) Partially Agree
    c) Neither Agree nor Disagree
    d) Partially Disagree
    e) Strongly Disagree
3. I had enough time to complete my tasks
    a) Strongly Agree
    b) Partially Agree
    c) Neither Agree nor Disagree
    d) Partially Disagree
    e) Strongly Disagree