



# Catalog of invisibility correlations for UbiComp and IoT applications

Rainara Maia Carvalho<sup>1</sup> · Rossana Maria de Castro Andrade<sup>1</sup> · Káthia Marçal de Oliveira<sup>2</sup>

Received: 18 December 2020 / Accepted: 12 October 2021 / Published online: 25 November 2021  
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2021

## Abstract

The advance of Ubiquitous Computing (UbiComp) and Internet of Things (IoT) brought a new set of Non-Functional Requirements (NFRs), especially related to Human-Computer Interaction (HCI). Invisibility is one of these NFRs and refers to either the merging of technology in the user environment or the decrease in the interaction workload. This new NFR may impact traditional ones (e.g., Usability), revealing positive correlations, when one NFR helps another, and negative correlations, when a procedure favors an NFR but creates difficulty for another one. Correlations between NFRs are usually stored in catalogs, which is a well-defined body of knowledge gathered from previous experience. Although Invisibility has been recently cataloged with development strategies, the literature still lacks catalogs with correlations for this NFR. Therefore, this work aims at capturing and cataloging invisibility correlations for UbiComp and IoT systems. To do that, we also propose to systematize the definition of correlations using the following well-defined research methods: Interview, Content Analysis and Questionnaire. As a result, we defined a catalog with 110 positive and negative correlations with 9 NFRs. We evaluated this correlation catalog using a controlled experiment to verify if it helps developers when they are making decisions about NFRs in UbiComp and IoT systems. Results indicated that the catalog improved the decisions made by the participants. Therefore, this well-defined body of knowledge is useful for supporting software engineers to select appropriate strategies that satisfy Invisibility and other NFRs related to user interaction.

**Keywords** Invisibility · Non-functional requirement · Catalog

## 1 Introduction

Over the last decades, systems for Ubiquitous Computing (UbiComp) [47] have increasingly been used to support everyday activities [48]. UbiComp and other technologies have given rise to what is today called Internet of Things (IoT), which is a collection of smart objects from our daily lives connected to the Internet. This work considers that IoT is

an extension of UbiComp since many of the technologies and visions from UbiComp directly apply to IoT [45]. Both UbiComp and IoT provide systems capable of accessing and controlling many objects.

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, Invisibility, Calmness, Attention and Synchronicity [12].

Invisibility in UbiComp and IoT applications refers to either a merging 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 tasks. [11]. Also, this NFR was cataloged regarding its subcharacteristics and development strategies for UbiComp and IoT systems, using Softgoal Interdependency Graph (SIG), a well-known notation in the requirements community to analyze and catalog NFRs [11]. In total, two main subcharacteristics, 12 sub-subcharacteristics, ten general strategies, and 56 specific strategies were identified and cataloged.

✉ Rainara Maia Carvalho  
rainaracarvalho@great.ufc.br

Rossana Maria de Castro Andrade  
rossana@great.ufc.br

Káthia Marçal de Oliveira  
kathia.oliveira@uphf.fr

<sup>1</sup> Computer Science Department (DC), Group of Computer Networks, Software Engineering and Systems (GREat), Federal University of Ceará, Fortaleza, Brazil

<sup>2</sup> Laboratory of Industrial and Human Automation Control Mechanical Engineering and Computer Science - LAMIH UMR CNRS 8201, Université Polytechnique Hauts-de-France, Valenciennes, France

Among these strategies, there are APIs, algorithms, middleware, protocols, interfaces, and hardware components.

However, Invisibility for UbiComp and IoT was indicated as an NFR that may impact Usability, negatively [10]. In fact, it is well known that in general NFRs interact with each other [2, 6, 19, 24, 30, 31, 33, 38], revealing positive correlations, when one NFR helps another and negative correlations, when a procedure favors an NFR but creates difficulty for others [15]. A classic example of an interaction between NFRs is Security and Performance. When developers add extra layers of Security in the system, Performance is impacted [49]. Then, the same situation can occur with new NFRs brought by the UbiComp and IoT.

Correlations can be captured from the developers experience and stored in correlation catalogs, a common artifact used by the requirements community to help software engineers avoid conflicting NFRs and select suitable strategies to satisfy different NFRs [15]. The literature has several catalogs that generally focus on correlations that are generic to any system [18, 31, 44, 52], but it lacks catalogs with Invisibility for the domain of UbiComp and IoT systems. This lack complicates the choice of development strategies to satisfy both Invisibility and other NFRs.

Therefore, there is a need to capture and catalog correlations that Invisibility may bring to other NFRs, especially the ones related to the quality of user interactions [9]. NFRs such as Usability, Security and Reliability are essential to the user, since UbiComp and IoT systems are designed to be anywhere and to work anytime for users in their everyday lives. This work is about the perception we could capture from developers who work with solutions for Invisibility. Thus, we answered the question “*How do developers believe Invisibility impacts NFRs related to User Interaction in UbiComp and IoT Systems?*.” As a consequence, a catalog of correlations was defined for developers and researchers.

Existing studies that propose to define correlations usually use knowledge from literature or industry [28, 52]. However, we found them hard to reuse, especially when capturing knowledge from several developers regarding development strategies. Therefore, our work systematizes the definition of our catalog of correlations through a methodology that organizes steps and its inputs, outputs and suggested methods [9]. This systematization allows the reuse of the methodology by others researchers and developers in the definition of correlations for new NFRs.

This paper extends the work presented in Carvalho et al. [9] by including additional information about the correlations and by presenting a controlled experiment to evaluate whether the proposed catalog of correlations 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 the catalog of correlations is used.

The remainder of this paper is organized as follows. We present the key concepts of this work and related studies in Sect. 2. Section 3 introduces the methodology we propose to define our catalog and its main outcomes. Section 4 presents the resulting catalog of correlations and a discussion about how Invisibility impacts other NFRs. Section 5 presents the performed controlled experiment. Section 6 presents the threats to validity of this work. Finally, Sect. 7 concludes with a summary and further work.

## 2 Background

### 2.1 Invisibility in UbiComp and IoT

Invisibility is long seen as an essential characteristic for achieving the goals of UbiComp [16, 27, 39, 41, 42], which can also be taken to IoT systems [1, 8]. This NFR was recently cataloged using the Softgoal Interdependency Graph (SIG) [15]. In this notation, every concept related to the NFR being cataloged is documented as softgoals. The NFR itself and its subcharacteristics are documented as NFR softgoals with light clouds. They are refined until reaching the level of strategies (*i.e.*, solutions), which are documented as Operationalizing Softgoals with dark clouds in this notation [15].

Figure 1 presents part of the SIG created for Invisibility<sup>1</sup>. This NFR is represented by two subcharacteristics: Invisibility from the usage point of view and Invisibility from the physical environment point of view.

In the case of *Invisibility from the usage point of view*, analysts and designers should decrease the workload of the user interaction with the system. The workload reduction can be achieved in two ways: reducing interactions or designing an interaction that is more natural for the user. Therefore, this subcharacteristic is refined into two softgoals: (1) Minimal Interaction, which refers to the system’s ability to design tasks without them being entirely or constantly dependent on explicit user inputs, and (2) Natural Interaction, which refers to supporting more natural and expressively powerful means of interaction by using natural interfaces and letting the user switch between modes of interaction. Figure 1 shows that these softgoals are refined into more softgoals that are strategies to support analysts and designers on how to implement such requirements, for example, Usage of Natural Interfaces.

In the case of *Invisibility from the physical environment point of view*, analysts and designers should merge the technological infrastructure in the physical space to ubiquitously

<sup>1</sup> Detailed description of this SIG can be found in our previous study [11].

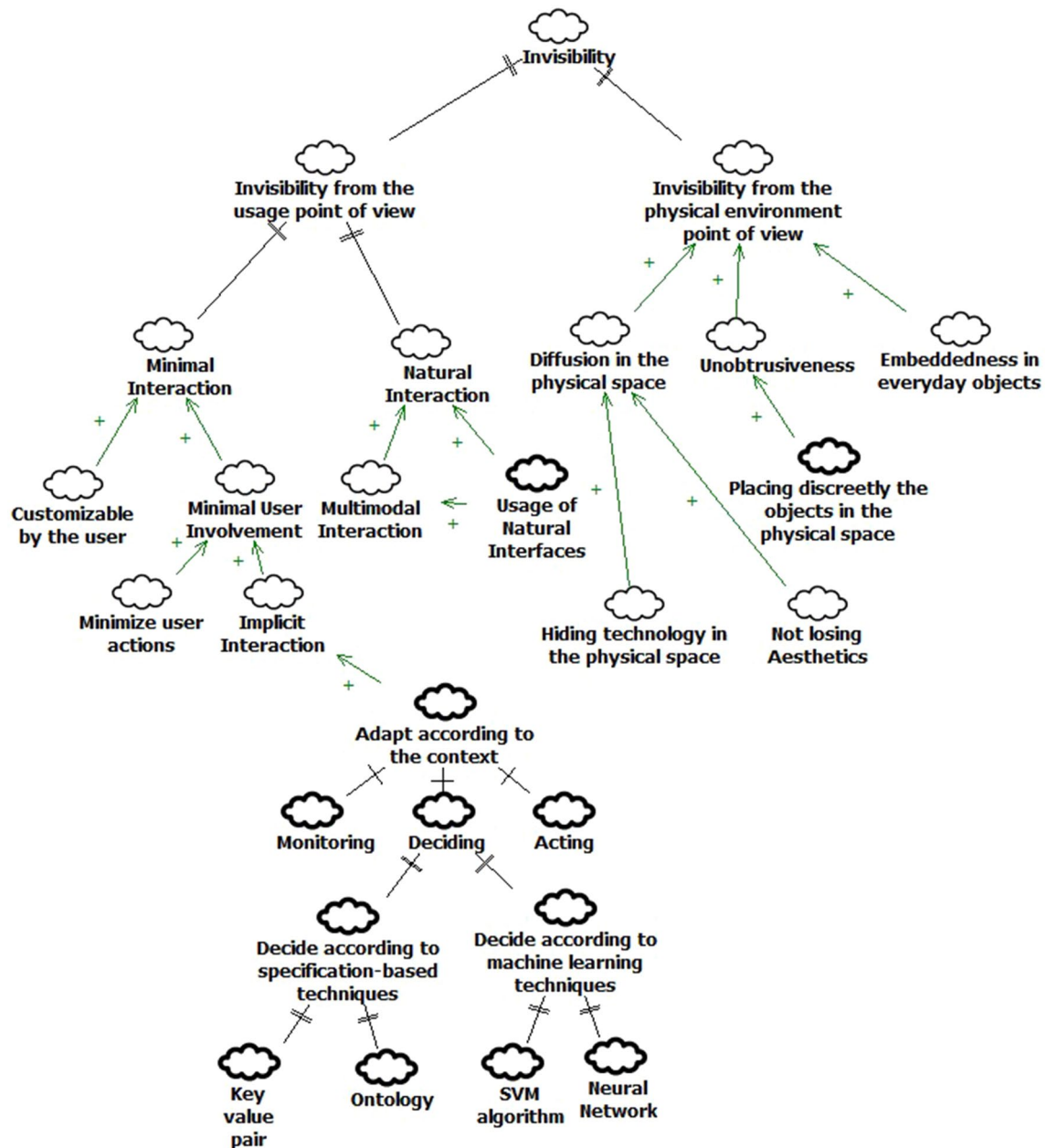


Fig. 1 Part of the Invisibility SIG [11]

support their users. A way of implementing this subcharacteristic is by placing discreetly physical objects such as sensors and actuators in the user space.

Table 1 presents a list of all the softgoals that are present in the last level of the Invisibility SIG (44 in total). They are specific strategies commonly used by developers to make Invisibility a reality. Among them, there are APIs, such as Google Sign-in<sup>2</sup> and Facebook Login<sup>3</sup>, which use an existing

account to log-in the user in the application; and Infrastructures as Middleware, such as LoCCAM [29] and OpenIoT<sup>4</sup>, two middlewares that help developers implement context-aware features. There are also techniques to specify context situations, such as Key-value pair. Moreover, protocols are present in this list, such as MQTT<sup>5</sup> and CoAP<sup>6</sup>. However,

<sup>2</sup> <https://developers.google.com/identity/sign-in/android/>.

<sup>3</sup> <https://developers.facebook.com/products/account-creation>.

<sup>4</sup> <http://www.openiot.eu/>.

<sup>5</sup> <http://mqtt.org/>.

<sup>6</sup> <http://coap.technology/>.

**Table 1** Softgoals to support Invisibility in UbiComp & IoT Systems [11]

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 hardware
S20	Neural Networks	S42	Hide technology
S21	MQTT	S43	Not losing aesthetics
S22	CoAP	S44	Place objects discreetly

it is important to highlight that this list can be updated to include more solutions.

We use all these softgoals to investigate how Invisibility impacts other NFRs so that correlations can be defined.

## 2.2 Catalog of correlations between NFRs

NFRs may conflict or cooperate with each other. Conflicts between them mean that achieving one NFR can negatively

impact another [4]. Cooperation between NFRs means that one NFR can help another [15]. Negative correlations symbolize conflicts and positive correlations represent cooperations.

Most of the researches on correlations among NFRs provide documentation, catalogs, or list of potential conflicts and cooperations among NFRs [32]. For example, a catalog is a body of knowledge that engineers accumulate from previous experience [15] and can store correlations between NFRs. Catalogs are used by software engineers to identify and analyze conflicts among NFRs since the beginning of the development. Thus, this work chooses catalog as a solution to deal with correlations between NFRs.

Figure 2 presents a partial correlation catalog, showing that “Validation” contributes to Confidentiality (plus sign), but affects negatively (minus sign) Response Time.

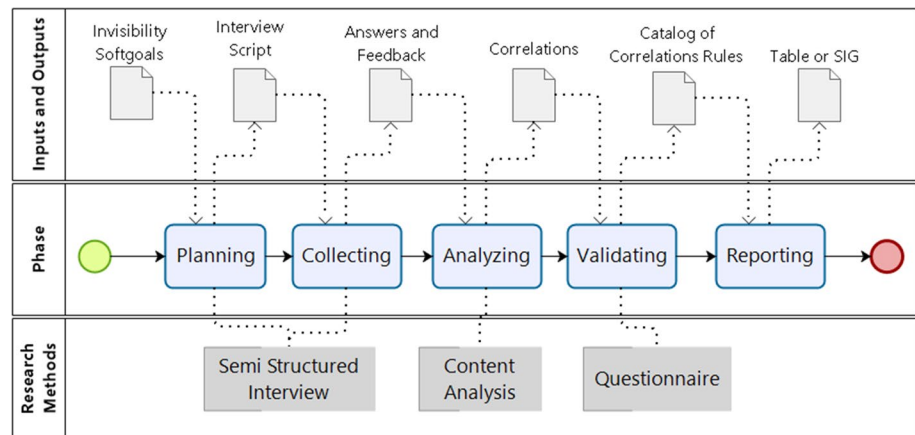
Additionally, correlations can be documented as a rule [15], which is expressed in the following format: **[softgoal] [kind of impact] [NFR or subcharacteristic] [condition]**. Correlations can be written with a condition, that it is a constraint to the rule. An example is given as follows: *FlexibleUserinterface* **HURTS** *Accuracy* **WHEN** *cardinality(User) is greater than 5*.

In this example, it is expressed that a developer might use a flexible user interface, but this can hurt Accuracy, and five users is the acceptable limit [15]. Furthermore, the kind of impact is defined as follows [15]: (i) BREAK correlations (labeled as “ $\llcorner$ ””) mean that a softgoal certainly denies the achievement of another softgoal; (ii) HURT (labeled as “ $\lrcorner$ ”) means that there is a negative partial contribution of a softgoal towards another softgoal; (iii) UNKNOWN correlations (labeled as “?”) mean that there is no knowledge about the relation between two softgoals; (iv) HELP correlations (labeled as “ $\llcorner$ ”) mean that there is a positive partial contribution; and (v) MAKE correlations (labeled as “ $\llcorner$ ”) mean a sufficiently positive contribution to achieve the other softgoal.

**Fig. 2** Example of Correlation Catalog [15]

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		+			-

**Fig. 3** Methodology used to define correlations



### 2.3 NFR catalogs in UbiComp and IoT systems

In addition to the aforementioned Invisibility catalog [11], there exist three other catalogs using Invisibility as a subcharacteristic. These three catalogs are documented in SIG notation, and they come from one study [34].

In this related work, Mehta et al. [34] analyses a set of NFRs for a smart mobile application that was designed to detect falls of elderly, which contains Safety, Ubiquity, Usability, Power-saving, and Cost. Invisibility is a sub-NFR of Ubiquity, and it comprises two subcharacteristics: Mental Invisibility and Physical Invisibility. For Mental Invisibility, there is only one strategy, which is a functional requirement that impacts others. In total, there are three correlations with Accuracy, Fast Relay, and Tolerate Ignorance. They are very specific for one kind of system: health domain.

Hence, there is a need to better investigate the impact of Invisibility regarding other NFRs, such as the ones related to user interaction, which is the focus of our work.

## 3 Defining the catalog of invisibility correlations

Figure 3 presents our methodology to reach a well-defined correlation catalog, which is composed of five phases: planning, collecting, analyzing, validating, and reporting. These phases are supported by three research methods: Interview [35], Content Analysis [14] and Questionnaire [35]. Each one of these phases and its outcomes is better explained in the next subsections.

### 3.1 Planning

The planning phase is concerned with the preparation of a script to guide the interview. The script prepared for this phase in this work followed the recommendations in [35],

containing four parts: introduction, instructions, demography data, and questions of the interview itself.

For the interview part, each softgoal from the lowest level of the SIG generated for Invisibility (44 softgoals presented in Table 1) was linked to the question: *What is the impact of this softgoal on user interaction quality?*, resulting in forty-four questions.

In this work, the user interaction quality is represented by a set of NFRs related to the final user. The ISO/IEC 25010 standard defines two models of quality characteristics [25]. 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 influence the quality in use of the final user: Functional Suitability, Performance Efficiency, Usability, Reliability, and Security. Hence, we consider these NFRs and the five ones in the “Quality in Use” Model as a set of NFRs that is closely related to user interaction quality

Most of these NFRs are refined into subcharacteristics, 40 in total [25]. For example, Usability has six subcharacteristics: Appropriateness Recognizability, Learnability, Operability, User Error Protection, User Interface Aesthetics, and Accessibility. This scheme was explained to the developers in the instruction moment, so they could have an overview of the NFRs taken into account.

Developers answered first what was their experience regarding that softgoal: (a) Not known; (b) Known; or (c) Known and Already Worked On It. Then, they should indicate what impact they think that softgoal has in the



user interaction quality in general, using the impact scale described in Sect. 2.2. Then, only developers who answered (b) Known; or (c) Known and Already Worked On It gave their feedback regarding the impact.

Besides indicating the overall impact using the scale, the developers were asked to give their opinions and feedback regarding that softgoal. Here, the interviewer made an effort and asked the developer if he/she has any comment on that impact, if he/she thinks it may have a positive or negative effect on some other aspect of the NFR. For example, if the developer commented that there is a positive correlation with a particular NFR, the interviewer asked the reason and if he/she thinks there may be some negative relation to some other quality feature or even if there exists a negative impact on some aspect of that NFR, which can indicate a correlation to a subcharacteristic.

The interview script was improved in two rounds of evaluation: first, it was evaluated by two professors and one HCI researcher in order to discover possible ambiguities and problems. Second, a pilot interview was conducted with a developer who works with IoT applications.

### 3.2 Collecting

We selected developers by the convenience sampling technique [51], which means we invited the nearest and most convenient persons. One criterion 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. Then, we performed an initial interview to get to know their experience. At the end, we selected fifteen (15) developers to participate in this study.

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 (See Fig. 11 in “Appendix A” to see more details about each developer).

We conducted most interviews 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.

### 3.3 Analyzing

We started the data analysis by extracting the quantitative data. In total, we obtained 472 answers for the 44 softgoals

among the five-point scale of impact (See Table 15 in “Appendix B”). This number (472) is the sum of answers for each impact item (break, hurt, unknown, help and make) in each softgoal. These answers gave an overview of what developers think in general, but it was not possible to define correlations from them. An observation made from these answers was that for every softgoal, regarding positive impact, the number of answers to HELP was bigger than answers to MAKE. In the same way, regarding the negative impact, the number of answers to HURT was more significant than answers to BREAK. These data showed us that correlations should be defined with “HELP” and “HURT.” Most of the developers stated that BREAK and MAKE are extreme impacts.

Then, we performed a qualitative analysis through the Content Analysis (CA) method [3]. CA is a research method to classify any communication material into identified categories of similar meanings [14]. It is suitable for subjective interpretation of the content of text data through the systematic classification process of coding and identifying patterns [23]. Therefore, this method comes as a strategy to properly analyze the data collected through the interview to define correlations.

There are two ways of conducting qualitative content analysis: inductive approach and deductive approach [14]. 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 concepts derived from the prior relevant literature. In the case of this work, the deductive approach is more appropriate because at this point we already know what we want to analyze, which is the impact of each softgoal in the set of NFRs related to user interaction quality from ISO/IEC 25010 [25] (Effectiveness, Efficiency, Satisfaction, Freedom from Risk, Context Coverage, Functional Suitability, Performance Efficiency, Compatibility, Usability, Reliability, Security, Maintainability and Portability), and its subcharacteristics. Therefore, we already had the categories (the type of correlation), subcategories (the type of quality model), codes (characteristics), and subcodes (subcharacteristics).

Regardless of what approach, coding is the primary procedure for qualitative content analysis. Coding means that segments of data are labeled with concepts, such as subcodes, codes, subcategories, and categories (preconceived or new ones), that depict what each segment is [13].

We performed the coding activity with the support of the MAXQDA tool [22]. In this tool, we could organize all the collected data by softgoal. Also, all the preconceived concepts were added to the tool, and then data coding could start. Figure 4 illustrates part of the initial set of concepts. They correspond to the type of correlations considered in this work (HELP and HURT), the type of quality model, and the set of NFRs related to user interaction quality. The last two are both

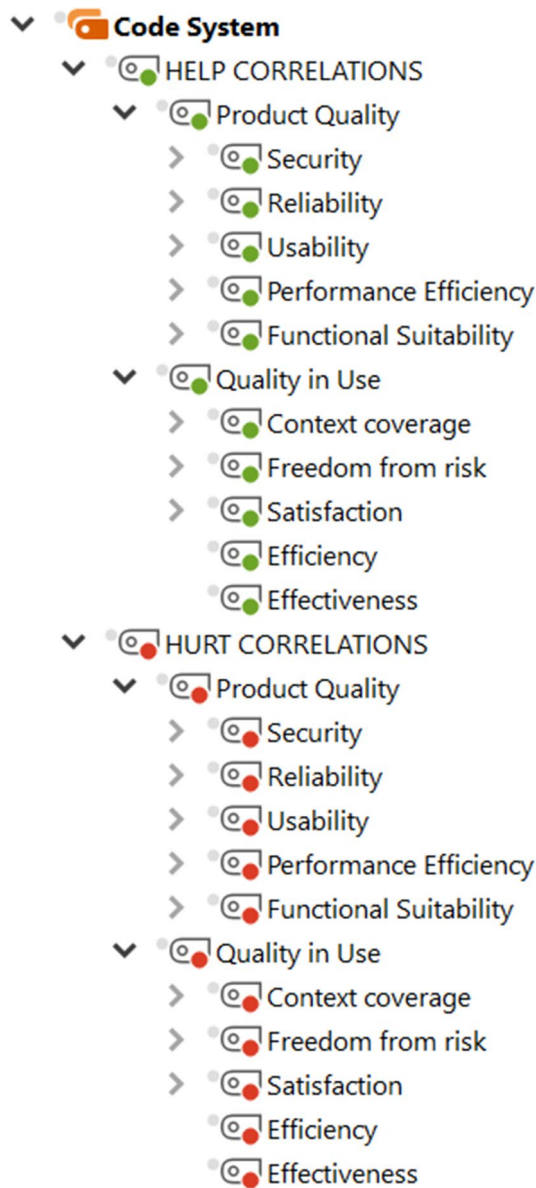


Fig. 4 Predefined Concepts from [25]

from ISO/IEC 25010 [25]. The type of correlations (HELP and HURT) groups the type of quality model, which in turn groups their characteristics and subcharacteristics. They are duplicated because a codification may be performed to the same NFR but in a different impact (HELP or HURT).

Then, one of the authors performed the coding by reading all the feedback from the developers on each softgoal. Every time a sentence seems to have a reference for an NFR or its subcharacteristics, a code representing that characteristic to match the sentence was used. Data that could not be matched into the predetermined concepts, but with another known NFR, was also coded.

Some examples of coded text segments are presented in Table 2. The first one is regarding “Ontology.” Developers

use this strategy to decide and reason on contextual data collected by the system. There are three examples of developers’ comments, which state that this strategy hurts Performance. Moreover, regarding Google’s and Facebook login API’s, three examples are pointing to a positive impact for Efficiency of the interaction.

In total, 329 codifications were performed, 161 of them had positive mentions (HELP correlations) to the NFRs, and 168 had negative mentions (HURT correlations). Table 16 in “Appendix B” shows the distribution of these codifications among the NFR or its subcharacteristic. Also, Table 16 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 positive impact on Security was mentioned by developers five times and for two softgoals. Furthermore, through this table, it is possible to see that most of the predefined NFRs had codifications, except for Freedom from Risk and Effectiveness. Also, three not previously defined NFRs were mentioned during the interviews: Privacy, Maintainability and Cost.

Table 17 presents softgoals of each NFR. For example, Security is impacted by Iris Recognition (S5), positively. The corresponding softgoals for each ID can be seen in Table 1.

Furthermore, each one of these impacts was directly mapped as a correlation rule. For example, “*Facial Recognition (S4) HELPS Security.*” However, while mapping these correlations, we could observe a few conflicts, which means there were negative and positive mentions for the same NFR. Three softgoals (S4–Facial Recognition, S12–Embedded Code, S19–SVM algorithm) presented four conflicts of correlations. The impact with most citations was selected to be present in the correlation rule.

At the end of the mapping, 120 correlation rules were defined. This number corresponds to the sum of numbers (128) in the last column of Table 16 minus four conflicting correlations and four correlations of Maintainability and Cost, since these last two were not the focus of this paper. Privacy was kept because it represents a characteristic essential to the final user [37, 52]. These correlation rules resulting from the data coding were then validated by experts.

### 3.4 Validating and reporting

Validation of the Invisibility correlations has been made to obtain more reliable data. This validation could be made for each mapping between text and code. However, this work generated 329 codifications, which can be quite costly. Thus, we chose to validate the correlation rules with experts through a questionnaire. Each rule was evaluated using a scale: agree, partially agree, disagree. Even though

**Table 2** Example of codifications

Softgoal	Text Segments	Codification
Ontology	<p>“...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”</p> <p>“impacts on the performance, ontology may slow the interaction”</p> <p>“...has a problem at runtime, when you need to infer some information on top of that knowledge, it becomes heavier than other approaches”</p>	HURT / Product Quality / Performance Efficiency
Google Sign-in and Facebook Log-in APIs	<p>“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”</p> <p>“They make it easier to register, shorten the registration time”</p> <p>“...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”</p>	HELP / Quality in Use / Efficiency
LoCCAM	<p>“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.”</p> <p>“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 ...”</p>	HURT / Privacy
MQTT	<p>“...it is a good performance implementation, whatever you build on it will make sure that it is not consuming all the way through.”</p> <p>“...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...”</p>	HELP / Product Quality / Performance Efficiency

the amount of data to be validated 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 NFRs. Therefore, each correlation was validated by exactly one expert. In total, seven experts were consulted and selected by the convenience sampling technique [51]. Two criteria were defined to recruit them: they must have a Ph.D. degree, and they should be from the area of the softgoal or the NFR.

Table 3 presents the experts profile with the number of correlation rules and the softgoals they received to validate. For example, Expert 1 has worked with context-aware and mobile computing, therefore, rules regarding “Adapt according to the context” were sent to Expert 1, 25 of them in total.

Table 4 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 all of them were excluded.

Expert 3 was the one who had the highest rate regarding the disagreement. Analyzing his evaluation about each disagreed rule, we realized that Expert 3 took into account a different definition of Trust, characteristic present in 4 correlations. Expert 3 disagreed even though the evaluation asked to take into account the definition from ISO/IEC 25010 [25]. However, all correlation rules that were disagreed by the experts (including the ones by Expert 3) were excluded since we would like to keep the set of correlation consistent to the Experts' opinion.

The rules with partially agree rates were analyzed to include some condition. For example, the correlation rule: “SVM algorithm hurts 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 called “cold start” [5], which means the system can take a while to infer

**Table 3** Profile of the experts

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



**Table 4** 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

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 Learnability when a technique for minimizing the cold start problem is not used.”

- 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*
- 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*

Finally, the correlations rules agreed by the experts were kept and did not change. In the end, 110 correlation rules were defined and then cataloged. They can be viewed in a SIG or a table. For clarity reasons, the rules in this work will be presented in a table. The next section presents all of them together, thus comprising the proposed correlation catalog.

## 4 Catalog of invisibility correlations

This section presents all resulting correlations in Tables 5 and 6. In total, there are 51 positive and 59 negative correlations.

An ID with a \* indicates that the correlation rule contains a condition. Therefore, 19 rules have conditions, presented as follows:

**Table 5** Catalog of correlations—Part 1/2

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

**Table 6** Catalog of Correlations—Part 2/2

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

Fig. 5 Overview of correlations



Figure 5 presents an overview of the correlations from Invisibility to the NFRs investigated in this work. 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 lower 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 give 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.

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 [25]). 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, especially 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 subcharacteristic, 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 lower part of the graph in Figure 5, 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), and Functional Suitability (8). Satisfaction presents 4 negative correlations, while Efficiency and Context Coverage are the characteristics with fewer negative correlations, 2 and 1, respectively.



The negative correlations with Security are related to softgoals that give another alternative to authentication, such as: Google Sign-in API, Facebook Login API, Facial Recognition. Indeed, a recent study [50] showed that the security of applications using Google's authentication API would depend on how programmers are using it. The study 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 [36]. This way, authenticity is impaired.

Privacy is not a characteristic defined in ISO/IEC 25010 [25]; 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 Log-in 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 [36];
- 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 inferring some information over knowledge modeled with first order or fuzzy logic is heavier than “if then else” logic. The possibilities grow too much, and it can be a long time for the final user until the system can make a decision.

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 users, the user may lose 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 Ubi-Comp 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 cannot 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, and thus new context information or situations will not be considered.

It is possible to see in Figure 5 that Invisibility has 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 such characteristics 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 Evaluation of the invisibility catalog

To investigate how well the catalog supports decision making, we performed the following phases of a controlled experiment [51]: (i) scoping; (ii) planning; (iii) operation; (iv) analysis and interpretation; and (v) presentation. These phases are presented in the following subsections.

### 5.1 Scoping

According to [51], the scope of the experiment is set by defining its goal, which in our case is: “**Analyse** the usage of the correlations catalog; **for the purpose of** characterizing it **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.”

To be able to achieve this goal, the following research questions are defined:

- 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.
- 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 not using the catalog.
- RQ3: Will the participants in the role of novice 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.

### 5.2 Planning

After we specified the scope, the planning starts. In this phase, the following topics should be defined: variables, factor and treatments; hypothesis; subjects; tasks and objects;

design type; and instrumentation [51]. They are described as follows.

#### 5.2.1 Variables, factor, treatment and measures

This experiment needs to study the effect of using a correlation catalog on the decisions of a novice requirements engineer. Therefore, one of the independent variables is the usage of the correlations catalog. Also, the background experience is another independent variable since we want to control this characteristic so that it will not to affect the outcome. The factor is the correlations catalog, and the treatments are using the catalog and not using the catalog.

The dependent variables need to be set to test the effect of changing the treatments (using or not using a correlation catalog). We set efficacy, efficiency and satisfaction as dependent variables.

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

We used the confusion matrix from the machine learning area as inspiration [43] to define metrics to Efficacy, presented as follows.

- True Positive (TP), which refers to the percentage of operationalizations that the participants must choose, because it has a positive effect on the required NFRs. This is measured by Eq. 1.

$$TP = \frac{\#ChosenOperationalizations}{\#PositiveOperationalizations} \quad (1)$$

- True Negative (TN), which refers to the percentage of operationalizations that the participants should not choose because it has a negative effect on the required NFRs. This is measured by Eq. 2.

$$TN = \frac{\#NotChosenOperationalizations}{\#NegativeOperationalizations} \quad (2)$$

For Efficiency, this study evaluates if the catalog supports better decisions in a faster way than when it is not used. It is important to highlight that these decisions do not have to be done in a time-sensitive manner. However, we would like to evaluate if an already developed catalog can facilitate the developer’s decision. We presume that without it, the developer needs to study or consult a specialist. When knowledge is reused, in this case, in a catalog, this activity

can be more efficient. Therefore, the following measure is defined in Eq. 3.

$$TS = \text{Time Spent in minutes to complete the tasks} \quad (3)$$

For Satisfaction, this study evaluates if the participants feel more satisfied when using the catalog. However, Satisfaction is hard to measure since it may be impacted by several other factors [20]. 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.

### 5.2.2 Hypothesis formulation

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

*Hypothesis for RQ1* We conjecture that although some correlations may look obvious, novice engineers of UbiComp and IoT systems do not have enough experience to make a correct decision regarding which strategies to choose. Therefore, this hypothesis evaluates if using the correlation catalog results on better values of True Positive (TP) and True Negative (TN) compared to not using the catalog. Based on this statement, we declare the null and alternative hypotheses as follows.

$$H_{0RQ1} : (TP_{withCatalog} \leq TP_{withoutCatalog}) \wedge (TN_{withCatalog} \leq TN_{withoutCatalog})$$

$$H_{1RQ1} : (TP_{withCatalog} > TP_{withoutCatalog}) \wedge (TN_{withCatalog} > TN_{withoutCatalog})$$

The null hypothesis is the one that the experimenter wants to reject with as high significance as possible. The alternative hypothesis is the one in favor of which the null hypothesis is rejected [51].

*Hypothesis for RQ2* We believe that by using a catalog with the information already defined, participants will not spend time overthinking to make a decision. Then, this hypothesis evaluates if using the correlation catalog results on different values of Time Spent (TS) compared to not using the catalog. The null and alternative hypotheses are presented as follows.

$$H_{0RQ2} : TS_{withCatalog} \geq TS_{withoutCatalog}$$

$$H_{1RQ2} : TS_{withCatalog} < TS_{withoutCatalog}$$

*Hypothesis for RQ3* We suppose that participants will feel more satisfied by using the proposed catalog since they will have support to make the decision. Then, this hypothesis

**Table 7** Experiment design type

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

evaluates if using the correlation catalog results on a different feeling of satisfaction compared to not using the catalog. The null and alternative hypotheses are presented as follows.

$$H_{0RQ3} : \text{SatisfactionWithCatalog} \leq \text{SatisfactionWithoutCatalog}$$

$$H_{1RQ3} : \text{SatisfactionWithCatalog} > \text{SatisfactionWithoutCatalog}$$

### 5.2.3 Subjects

In this work, the population is composed of novice requirements engineers. Convenience sampling [51] was used to select subjects for this population. In this way, 44 undergraduate students from the Requirements Engineering course at the Federal University of Ceará, Brazil, 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.

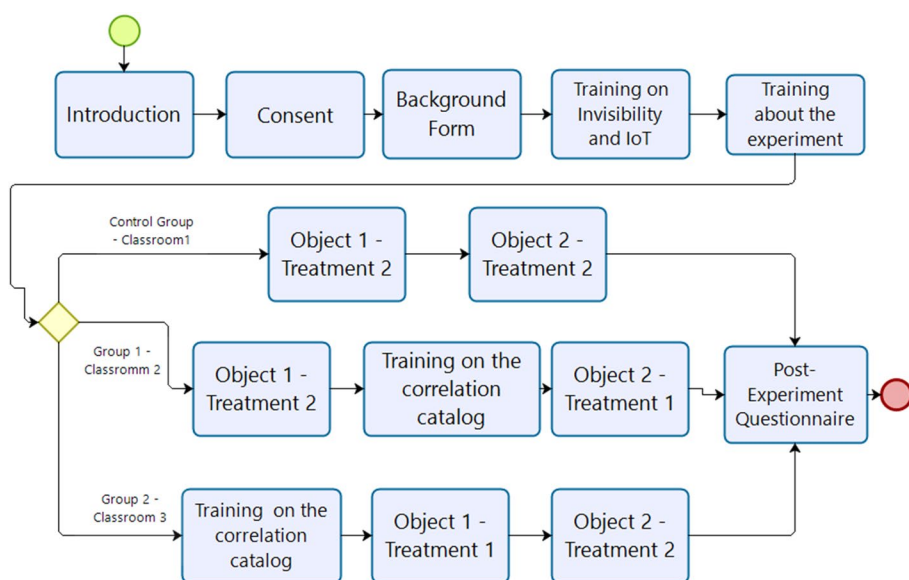
In total, 36 of them participated in the experiment. Most of them (32) had basic knowledge about NFRs, obtained

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. Thirteen of them had basic knowledge, and only one was experienced. Finally, regarding the Invisibility characteristic, the majority did not know about it, and only 2 had basic knowledge.

### 5.2.4 Tasks and objects

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.

Fig. 6 Experiment Operation



The selected objects for this experiment were two Ubi-Comp and IoT systems called AutomaGREat (Object 1) [1] and GREatBus<sup>7</sup> (Object 2). AutomaGREat (Object 1) is an application that proposes an intelligent environment for the Seminar Room of the GREat research laboratory, located at Federal University of Ceará in Brazil. 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 SIG. Also, a set of required NFRs for both systems was defined. For AutomaGREat, besides Invisibility, the NFRs were Security, Performance, Efficiency, and Reliability. For GREatBus, besides Invisibility, two NFRs were Accessibility and Privacy. In this way, the participants received a material containing the description of the system; their functional and non-functional requirements; their Invisibility SIG model with the description of the softgoals in the last level; and the correlation catalog (such materials can be seen in Appendix A). Then, 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 on the required NFRs for the system.
- Task 2: Choose the operationalizations that maximize the positive impact and minimize the negative impact on the required NFRs.

The participants received the correlation catalog only when they were executing the tasks with the “not using the

catalog” treatment. Section 5.3 explains how these tasks were executed by the participants in each object.

Furthermore, 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.

### 5.2.5 Design type

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 it (see Table 7).

This type of design uses the same objects for both treatments and assigns the subjects randomly to each treatment [51]. The Control Group was added to increase the reliability

Table 8 Descriptive statistics to answer RQ1

Measure (TP, TN)/Object	Group	Mean	Median	Mode	SD
TP_OBJECT1	1	0.58	0,67	0.33	0.25
	<b>2*</b>	<b>0.97</b>	<b>1</b>	<b>1</b>	<b>0.09</b>
	C	0.52	0.67	0.67	0.26
TN_OBJECT1	1	0.7	0.71	0.71	0.20
	<b>2*</b>	<b>0.95</b>	<b>1</b>	<b>1</b>	<b>0.09</b>
	C	0.66	0.71	0.71	0.14
TP_OBJECT2	<b>1*</b>	<b>0.83</b>	<b>1</b>	<b>1</b>	<b>0.38</b>
	2	0.33	0	0	0.49
	C	0.16	0	0	0.38
TN_OBJECT2	<b>1*</b>	<b>0.8</b>	<b>0.93</b>	<b>1</b>	<b>0.25</b>
	2	0.52	0.66	0.33	0.26
	C	0.47	0.67	0.67	0.26

\* groups using the catalog

<sup>7</sup> GREatBus is a system in development as part of the project called Smart Bus Stop, accepted by the Institutional Program of Innovation Scholarships at Federal University of Ceará in Brazil.



**Table 9** Mann–Whitney tests for RQ1

Measure(TP,TN)/ Object	Group 1 and Group 2	Control Group and Group 1	Control Group and Group 2
TP_OBJECT1	<b>0.000</b>	0.755	<b>0.000</b>
TN_OBJECT1	<b>0.002</b>	0.671	<b>0.000</b>
TP_OBJECT2	<b>0.039</b>	<b>0.005</b>	0.514
TN_OBJECT2	<b>0.012</b>	<b>0.005</b>	0.977

(cells with values in bold are indicating combinations where one group used a catalog)

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 SIG different from what they received in the first object. This strategy minimizes the possibility of memorizing the correlations, which is essential to Group 2 since the participants started the experiment with the catalog, and then they could not use it anymore. That is the reason why we created two SIGs.

### 5.3 Operation

Once planning is finished and all materials prepared, the experiment can take place, which happened at one day in this work. Figure 6 presents how the execution happened. The activities were based on the design type established in this experiment and also based on an existing experiment performed in Santos et al. [40].

First, we introduced the experiment to the students, and then we asked and registered their consent. After that, a background form was applied to get information about their experience. Then, the training took place. First, we performed a training about IoT and UbiComp systems. After that, another training was performed to explain the tasks the students would be asked to do. In this training, we revisited the concepts about NFRs, SIGs, and tradeoffs.

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

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. In Group 2, the subjects started the tasks in AutomaGREat object with treatment 1—using the catalog.

**Table 10** Descriptive statistics to answer RQ2

Measure(TS)/ Object	Group	Mean	Median	Mode	SD
TS_OBJECT1	1	34.25	33	33	15.89
	<b>2*</b>	<b>21.42</b>	<b>20</b>	<b>20</b>	<b>9.07</b>
	C	27	29	21	5.9
TS_OBJECT2	<b>1*</b>	<b>10.83</b>	<b>11.5</b>	<b>12</b>	<b>3.46</b>
	2	13.5	13.5	16	4.68
	C	15.42	15.5	18	4.25

\* groups using the catalog

**Table 11** T tests for RQ2

Measure(TS)/Object	Group 1 and Group 2	Control Group and Group 1	Control Group and Group 2
TS_OBJECT1	<b>0.023</b>	0.159	<b>0.088 / 0.010</b>
TS_OBJECT2	<b>0.127</b>	<b>0.008</b>	0.305

(cells with values in bold are indicating combinations where one group used a catalog)

Then they performed tasks in Object 2 with treatment 2—not using the catalog.

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. Finally, after finishing all the experiment tasks, we asked participants to fill out the Post-Experiment Questionnaire.<sup>8</sup>

### 5.4 RQ1: Efficacy

The measures used to answer this question were True Positive (TP) and True Negative (TN), described in Section 3. To calculate them, the answers of the participants were compared to a set of predefined answers that we established based on the correlation catalog. The main task of the experiment was about choosing operationalizing softgoals that would have a positive impact on the required NFRs and minimize the negative impact on those NFRs. Thus, participants should choose operationalizing softgoals that HELP the NFR and not choose those that HURT the NFR. For example, regarding GREatBus system, the softgoal Iris Recognition helps one of the NFRs of the system (Accessibility); hence, this softgoal should be chosen.

All raw data to draw conclusions about RQ1 are presented in Table 18 in “Appendix D,” and the descriptive statistics are presented in Table 8.

<sup>8</sup> All materials used in the operation of this experiment can be seen at [encurtador.com.br/hPVZ0](http://encurtador.com.br/hPVZ0).

**Table 12** Statements to measure satisfaction for 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 <b>5. If there was a catalog, it would be easier</b> <b>6. If there was a catalog, it would be faster</b>	1. I easily identified the impacts 2. I quickly identified the impacts 3. I easily made my decision 4. I quickly made my decision <b>5. If there was a catalog, it would be easier</b> <b>6. If there was a catalog, it would be faster</b>
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 <b>5. If there was a catalog, it would be easier</b>  <b>6. If there was a catalog, it would be faster</b>	1. I easily identified the impacts 2. I quickly identified the impacts 3. I easily made my decision 4. I quickly made my decision <b>5. I think the catalog made my decision easier</b> Why? <b>6. I would recommend using the catalog for decision making</b> 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 <b>5. I think the catalog made my decision easier</b> Why? <b>6. I would recommend using the catalog for decision making</b> 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 <b>5. I think the absence of a catalog made my decision harder</b> Why? <b>6. I think the catalog would made my decision easier</b>  Why? <b>7. I would recommend using a catalog for decision making</b> Why?

Notably, groups using the catalog (indicated with an asterisk \*) obtained better results. For example, regarding 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 Control Group was also smaller (0.52).

To test the null hypothesis of Efficacy, both measures (True Positive–TP and True Negative–TN) should have significant differences ( $p < 0.05$ ) in both objects (AutomaGREat–OBJECT1 and GREatBus–OBJECT2).

As the design of this experiment includes three groups using both objects, the tests should be executed to compare three sets 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 were analyzed in each measure (TP and TN) and each object (AutomaGREat—OBJECT1 and GREatBus—OBJECT2). In this way, twelve hypothesis tests were executed.

We selected the Mann-Whitney test for all hypotheses for RQ1 because the Shapiro-Wilk test indicated that the dataset for the three combinations of groups and all measures do not follow a normal distribution. The Mann-Whitney test, on the

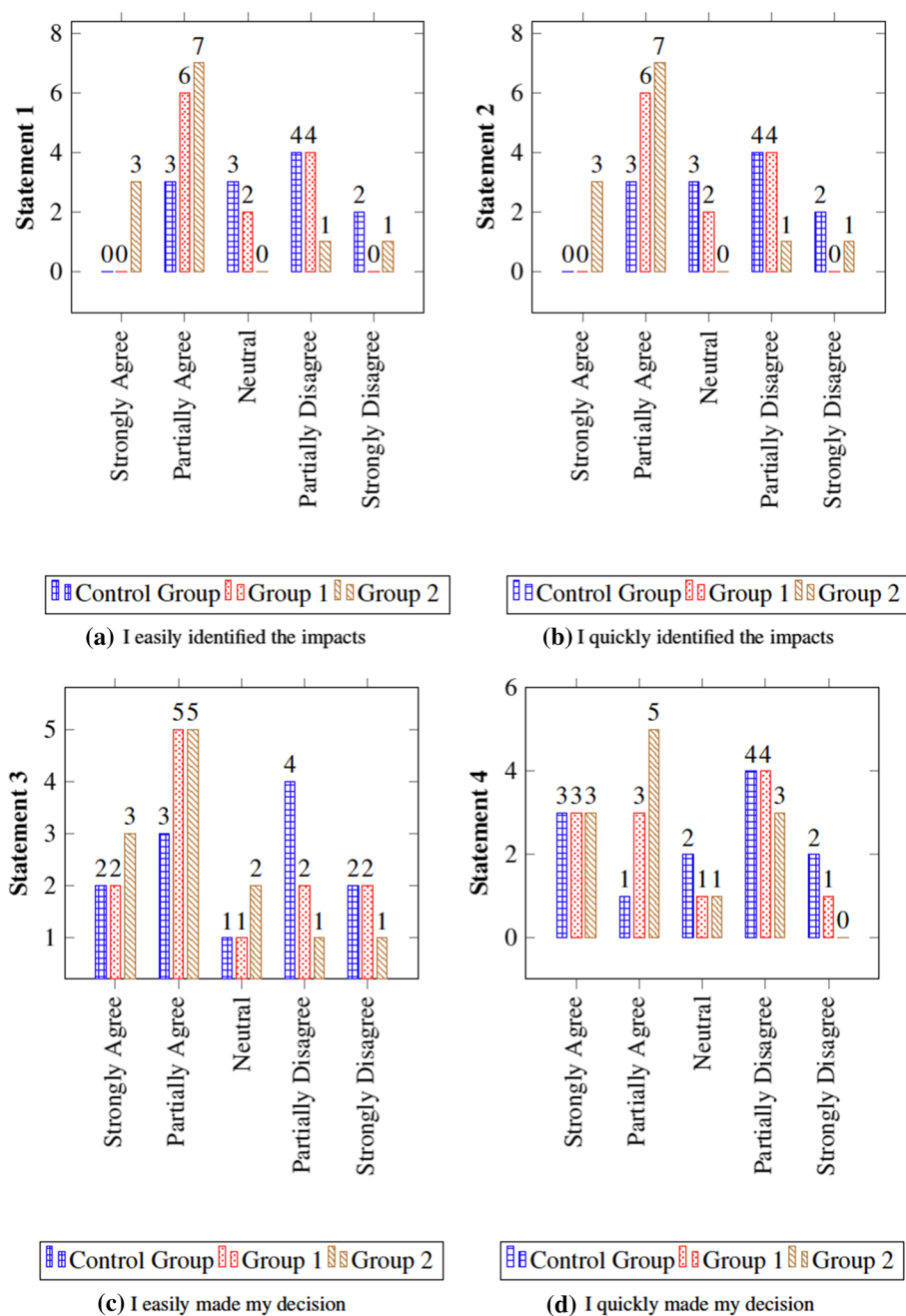
other hand, is a proper nonparametric test for experiments with one factor, two treatments, and randomized design, which is the case for this work, and it is present in the SPSS IBM tool [51]. Table 9 presents the results.

In all combinations where one group was using the catalog and the other group was not, the p values were below 0.05. In combinations where both groups were not using the catalog, the p-value was well above 0.05. For example, the measure TP in Object 1 (TP\_OBEJCT1) at Control Group and Group 1, the p-value was 0,755. 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. Therefore, the null hypothesis stated for RQ1 is rejected, allowing the acceptance of its alternative hypothesis.

### 5.5 RQ2: Efficiency

The measure used to answer this question was Time Spent (TS). All raw data used to draw conclusions in regard to Question 2 are presented in Table 19 in “Appendix D” and the descriptive statistics in Table 10.

**Fig. 7** Results of Satisfaction in Object 1—AutomaGREat, where Group 2 was using the catalog



Participants took less time to perform their tasks when they were using the catalog. The mean time of the groups not using the catalog was bigger than the mean of the groups using the catalog (indicated with an asterisk \*).

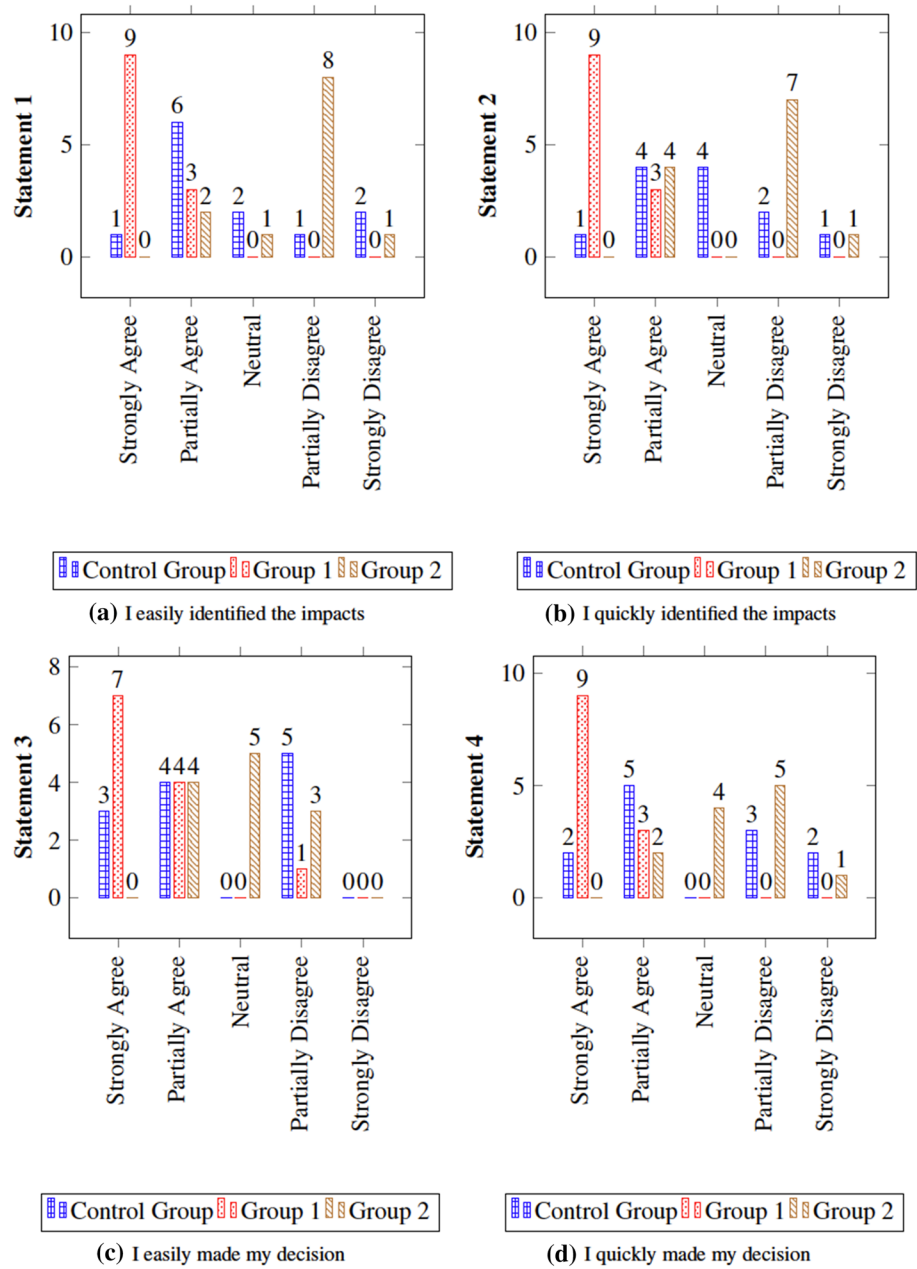
The Shapiro–Wilk test in the case of data from RQ2 indicated that the three combinations of groups and for all measures (TP in Object 1, TN in Object 1, TP in Object 2 and TN in Object 2) follow a normal distribution. Hence, it was necessary to use a parametric test, which was the T test since it is suitable for the type of design in this experiment [51],

and it is present in the SPSS IBM tool. Table 11 presents the results (p-values) of the hypothesis tests.

In summary, not all p-values resulted as expected. The measure TS in Object 2 (TS\_OBEJCT2) at Groups 1 and 2, the p-value was 0.127, being above 0.05. 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 1 (TS\_OBEJCT1) at Control Group and Group 2, the p-value was 0.088, not being below 0.05. Therefore, the p-value does not indicate

**Fig. 8** Results of Satisfaction in Object 2—GREatBus, where Group 1 was using the catalog



a statistically significant difference between the groups. Group 2 used the catalog and Control Group did not use it. However, when we took a closer look at results for Group 2 in Object 1 through the SPSS tool, we discovered there is an outlier, which is an abnormal or false data point [51]. When outliers are identified, it is important to decide what to do with them [51]. If the outlier is caused by a strange or rare event that will never happen again, the point could be excluded. If the outlier is because of a rare event that may occur again, it is not advisable to exclude the value from the analysis. We did not know the exact reason why this data point is an outlier. Therefore, we decided to analyze this data with and without this outlier. So, when excluding

this participant from the dataset, the p-value is 0.010, being below 0,05. Nevertheless, the null hypothesis for RQ2 could not be rejected.

**5.6 RQ3: Satisfaction**

The research question related to Satisfaction is “Will the participants in the role of novice 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



**Table 13** Descriptive statistics for RQ3

Object (O) and Statement (S)	Group	Median	Mode
<i>O1-S1</i> : I easily identify the impacts	1	3	4
	2	4	4
	C	2	2
<i>O1-S2</i> : I quickly identify the impacts	1	2	2
	2	4	4
	C	2	2
<i>O1-S3</i> : I easily made my decision	1	4	4
	2	4	4
	C	2	2
<i>O1-S4</i> : I quickly made my decision	1	3	2
	2	4	4
	C	2	2
<i>O2-S1</i> : I easily identify the impacts	1	5	5
	2	2	2
	C	4	4
<i>O2-S2</i> : I quickly identify the impacts	1	5	5
	2	2	2
	C	3	3
<i>O2-S3</i> : I easily made my decision	1	5	5
	2	3	3
	C	4	2
<i>O2-S4</i> : I quickly made my decision	1	5	5
	2	2	2
	C	4	4

five-point Likert scale to rate their feelings: Strongly Agree, Partially Agree, Neither Agree nor Disagree, Partially Disagree and Strongly Disagree. First the descriptive statistics are presented and then hypothesis testing.

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 an artifact to help them in making decisions.

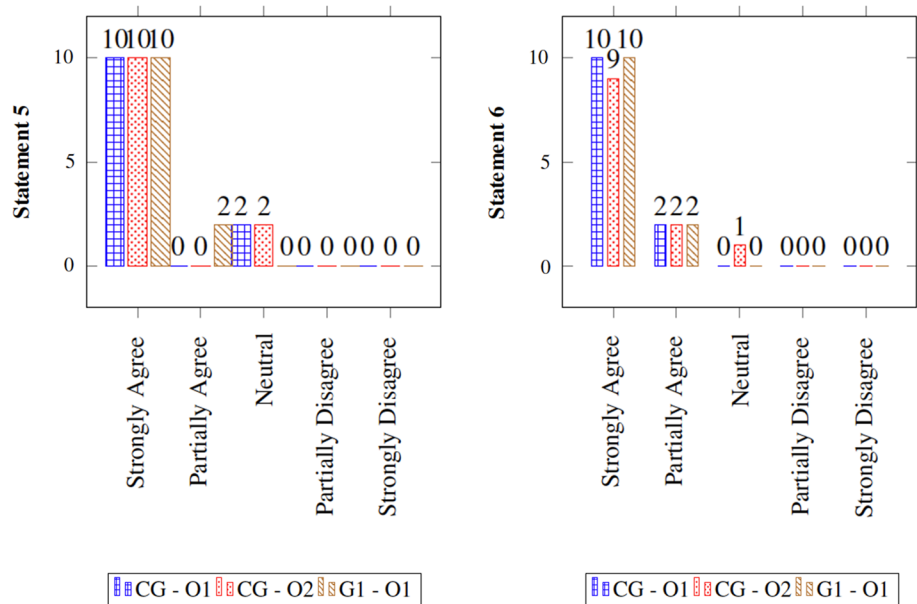
All participants had to answer questions after the execution of the experiment in each object. In total, 6 questionnaires were answered. Table 12 introduces all statements for each group and object.

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 answer these following

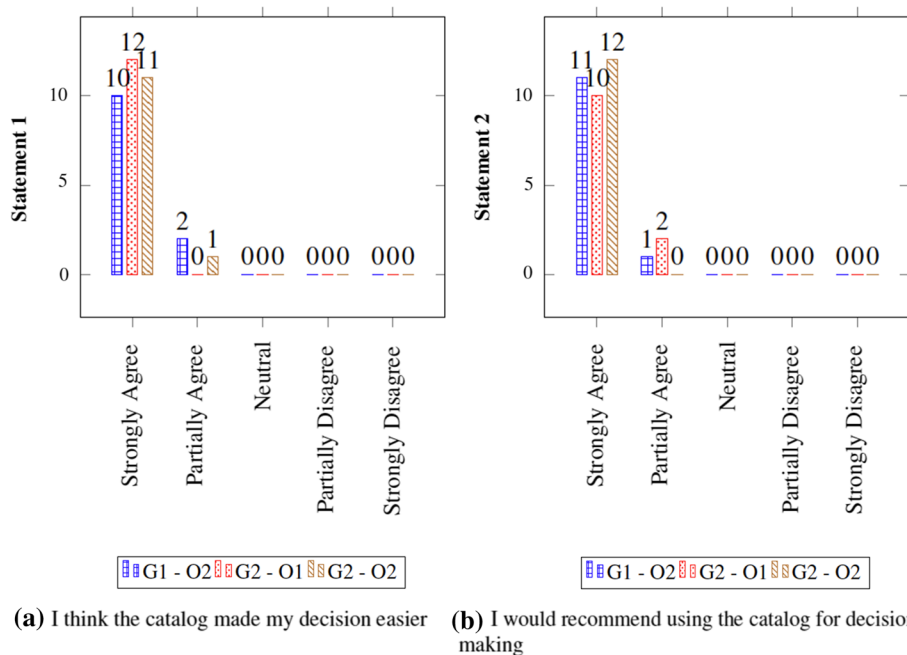
**Fig. 9** Results of satisfaction when participants did not use the catalog



(a) If there was a catalog, it would be easier

(b) If there was a catalog, it would be faster

**Fig. 10** Results of satisfaction when participants used the catalog



**Table 14** Hypothesis testing for RQ3

Statements (S)	Group 1 and Group 2	Control Group and Group 1	Control Group and Group 2
<i>O1-S1</i>	<b>0.101</b>	0.219	<b>0.120</b>
<i>O1-S2</i>	<b>0.242</b>	0.410	<b>0.89</b>
<i>O1-S3</i>	<b>0.514</b>	0.630	<b>0.219</b>
<i>O1-S4</i>	<b>0.514</b>	0.630	<b>0.242</b>
<i>O2-S1</i>	<b>0.000</b>	<b>0.001</b>	0.780
<i>O2-S2</i>	<b>0.000</b>	<b>0.000</b>	0.219
<i>O2-S3</i>	<b>0.001</b>	<b>0.068</b>	0.514
<i>O2-S4</i>	<b>0.000</b>	<b>0.003</b>	0.291

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.

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.

Figure 7 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 compared to Control Group, Group 2 always obtained less answers for the disagreements options.

Figure 8 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 it 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 13 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 9 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 10 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 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.

Besides the hypothesis tests, we also performed the Cronbach’s alpha test to measure the reliability of the obtained answers, resulting in 74,4% of reliability, a value considered acceptable by the literature [21].

We used the Mann–Whitney test for the null hypothesis of Satisfaction, defined for RQ3. However, we did not perform normality tests, because Likert scale does not follow a normal distribution since it is an ordinal scale [26]. Table 14 presents the results. 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 ( $p < 0,05$ ).

In summary, not all p-values resulted as expected when comparing results between groups. Therefore, the general null hypothesis for Satisfaction cannot be rejected.

Furthermore, the post-task forms also had open questions about 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 in the UbiComp and IoT areas. 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”

## 5.7 Discussion

Regarding 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 maximizes the positive impacts and minimizes the negative impacts of 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 the expected results. Thus, 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 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, even though groups using the catalog took less time to perform the tasks. However, we realized that the difference was not so big, especially in the second object. 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 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. In this way, we can see that participants could get a better feeling of the importance of using the catalog when they were performing tasks for the second time in Object 2.

Furthermore, when all participants finished tasks in all objects, a post-experiment questionnaire was applied. The results showed that most participants were satisfied with the training, goals of the tasks and duration.

## 6 Threats to validity

In this work, we have threats related to the definition of the catalog and its evaluation through a controlled experiment. We tried to minimize the threats by considering the four categories of validity [51]: conclusion, internal, construct and external validity. They are described as follows.

**Conclusion validity** is concerned with issues that affect the ability to draw the correct conclusion. Regarding the definition of the catalog, the correlations represent our conclusions. We applied a systematic qualitative method, Content Analysis, to generate them.

A threat that can invalidate the correctness of the correlations is that they were generated by only one researcher during the coding activity, meaning that we had only one annotator in the coding. To minimize the effects of this threat, each correlation is validated by one expert, and thus we decided to exclude the ones that the expert disagreed. Only 8% of the correlations were disagreed, indicating that our set of correlations was well defined. Furthermore, while mapping the correlations, only 4 codifications out of 329 were considered conflicting with each other, indicating that there is a consistency among the opinion of the developers.

The threat that we had only one coder may lead to a number of missed correlations. However, this fact does not impact the correctness of the generated correlations and a good number of correlations was generated (110 correlations). Hence, we accept this consequence and we understand that a good number of correlations was generated (110 correlations).

According to [51], in controlled experiments, threats to the conclusion validity can be (i) violated assumptions of statistical tests, (ii) random heterogeneity of subjects and (iii) reliability of measures and instruments. Therefore, to minimize these threats we performed (i) the evaluation of the normality of the data through Shapiro-Wilk tests before choosing a hypothesis test; (ii) the selection of participants presenting a homogeneous profile; and (iii) the execution of pilot tests to improve the reliability of the instrumentation.

**Internal validity** is concerned about influences that can affect the independent variable with respect to causality, without the researcher's knowledge. These influences threaten the conclusion about a possible causal relationship between treatment and outcome. In the definition of the catalog, it 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. For this work, the interview

script remained the same for all participants during all the study. The interviews were performed in one month, which is a short time that does not require advances in software or hardware to be considered. Furthermore, the participants (developers and experts) did not receive any compensation.

In the experiment, some decisions were made to minimize influences on the independent variable (the undergraduate students). First, 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. Second, the participants were split into three groups; one of them is a Control Group. Then, several combinations were made to analyze data and give more reliable results. Finally, the experiment's tasks were executed twice for each group. Thus, more data were collected so the researcher could be more precise about the relationship between treatment and outcome.

**Construct validity** for the definition of the catalog ensures that the study actually asks what it is supposed to ask. All interviews were conducted by the first author. Moreover, we discussed the interview script with an expert and performed a pilot study to correct any issues. Furthermore, we tried to interview the participants in the same environment. This happened with all of them, except the ones who were interviewed by video conference (three of them—20%). Furthermore, we 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.

In the experiment, construct validity is concerned with the treatments and outcomes. They should have a good reflection on the cause and effect of the experiment. 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. Moreover, another threat is if an experiment is conducted with a single document as an object, in which case the cause construct is under-represented. In this work, two objects were used and two rounds of the tasks were performed. Another possible threat is the representation of the construct since only parts of the Invisibility SIG and Correlation Catalog were used in the experiment. Larger documents could produce more reliable data, but it would not be possible to execute the experiment tasks within an acceptable time. Therefore, this work accepted this risk.

**External validity** of a research means that the results are generalizable. For external validity for the definition of the catalog, 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. Also, we understand that our study is about the experience we could get from developers who work with these solutions. Our findings in the definition of the catalog (i.e., correlations) should be considered as hypotheses, rather than generically valid facts.

A threat to external validity for the experiment is the fact that the participants were all students. To minimize this risk, we select students from a requirements engineering course, where this activity (decision making regarding NFRs) is required. Therefore, we accept this threat. Furthermore, we plan other studies as we are aware that this study should be replicated in other contexts.

Another threat present in both the definition of the catalog and the experiment is the sampling strategy used to select participants: the convenience sample. According to [7], the convenience sampling is the least advantageous regarding generalizability, since results are general only to the sample studied. However, we accept this threat because the other sampling strategies are cost and resource prohibitive for the authors. However, we tried to minimize this threat by making sure that the participants met our criteria, such as knowledge in the softgoals (in the case of the definition of the catalog) and knowledge in SIG (in the case of the experiment).

## 7 Conclusion and future work

Invisibility has been cataloged with a variety of subcharacteristics and development solutions. In this work, we investigated how this NFR impacts other NFRs that are related to the user interaction quality through the experience of developers.

From this investigation, we defined 110 correlation rules of Invisibility with 9 NFRs as the main contribution of this work. A methodology combining five phases and three research methods was used to determine the correlation

rules. We interviewed 15 experienced developers of UbiComp and IoT systems. All feedback was analyzed through a systematic qualitative method, the Content Analysis. Then, through a questionnaire, 7 experts evaluated a set of correlation rules, which means each correlation was evaluated exactly by one expert.

Through these correlations, we could understand the extent to which Invisibility impacts user interaction quality. On the one hand, Usability was the most positively impacted characteristic. On the other hand, Security was most negatively impacted.

Moreover, by carrying on a controlled experiment, we confirmed that the usage of the catalog helps novice requirements engineers when they are making decisions about NFRs in UbiComp and IoT systems. However, results of this experiment did not show statistically that efficiency and satisfaction improve. This experiment can be seen as a secondary contribution since few studies present statistical evidence about the usage of catalogs in general (only two papers were found [17, 46]).

We are currently working on the usage of this catalog in the development of different IoT applications. Furthermore, we suggest as future work to investigate correlations of Invisibility regarding other NFRs not taken into account in this study.

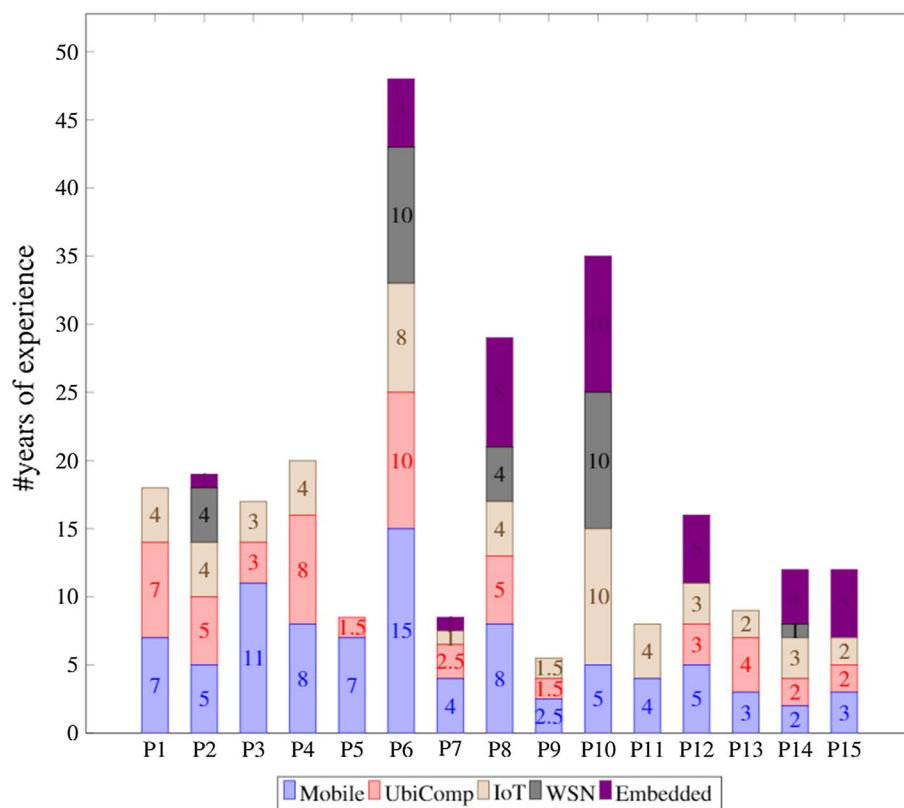
## Appendices

### A Interview to define the catalog: profile of the developers

Figure 11 presents the years of experience of each developer regarding each domain area of expertise. We consider their experience in the development of applications for ubiquitous computing, Internet of Things (IoT), wireless sensor networks (WSN), and embedded systems.



**Fig. 11** Years of Experience and Background of the Developers



## B Results of the analysis to define the correlations

Table 15 presents all quantitative results from developers. Table 16 shows the distribution of the codifications among the NFR or its subcharacteristic. Table 17 presents softgoals of each NFR.

## C Tasks and objects of the experiment

### C.1 Object 1: AutomaGREat

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.

### C.1.1 Functional requirements

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

**Table 15** 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 discretely	0	2	1	9	1

**Table 16** Quantity of codifications by NFR

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

5. Automatic mode should allow air conditioners and lamps to be triggered from room presence detection and user preferences

### C.1.2 Non-functional requirements

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 conditions;

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.

**Table 17** Softgoals of each NFR or subcharacteristic

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

### C.1.3 Softgoal interdependency graph (SIG) for AutomaGREat system

Figure 12 presents the SIG model for AutomaGREat System, received by the participants.

The description of the operationalizing softgoals in the last level of the AutomaGREat SIG is presented as follows.

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

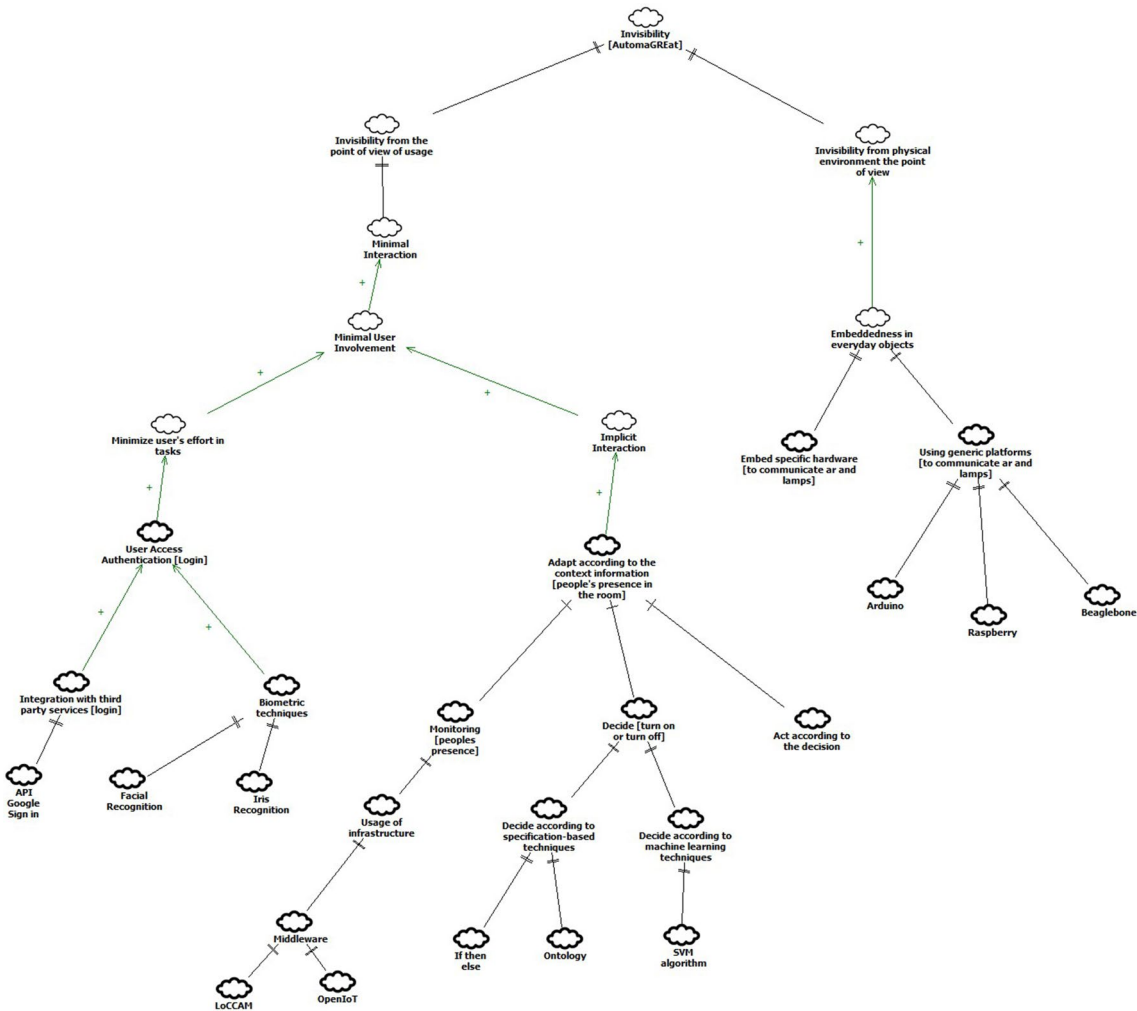


Fig. 12 AutomaGREat SIG

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.

#### C.1.4 Tasks

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.2 Object 2: GREatBus

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.

### C.2.1. Functional requirements

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.

### C.2.2 Non-functional requirements

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.

### C.2.3. Softgoal interdependency graph for GREatBus system

Figure 13 presents the SIG model for AutomaGREat System, received by the participants.

The description of the operationalizing softgoals in the last level of the GREatBus SIG is presented as follows.

1. Facebook Log-in API: API that allows authentication with Facebook data;
2. Facial Recognition: Technique to identify the user based on their face;

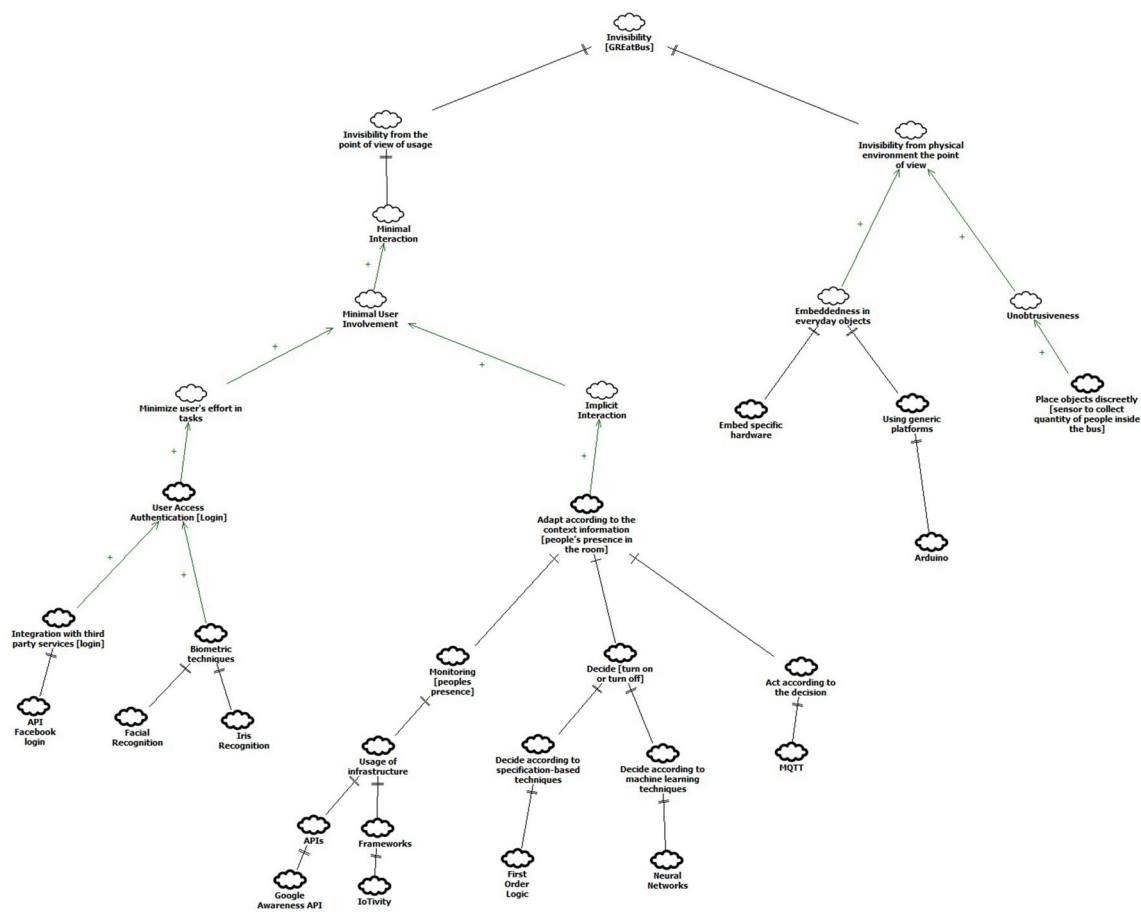


Fig. 13 GREatBus SIG

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;
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.

#### C.2.4 Tasks

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.

#### D Raw data of the experiment

Tables 18 and 19 present raw data of RQ1 and RQ2.

**Table 18** Raw data to answer RQ1

Subject	Group	Object 1		Object 2	
		TP	TN	TP	TN
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

**Table 19** Raw data to answer 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

**Acknowledgements** We thank CNPQ for the Productivity Scholarship of Rossana Maria de Castro Andrade DT-2 (No. 315543/2018—3).

## References

- Andrade R, Carvalho R, de Araújo I, Oliveira K, Maia M (2017) What changes from ubiquitous computing to internet of things in interaction evaluation? In: International conference on distributed, ambient, and pervasive interactions. [https://doi.org/10.1007/978-3-319-58697-7\\_1](https://doi.org/10.1007/978-3-319-58697-7_1)
- Andreopoulos B (2004) Satisficing the conflicting software qualities of maintainability and performance at the source code level. In: Workshop de Engenharia de Requisitos, pp 176–188. Citeseer
- Bardln L (2011) Análise de conteúdo. Almedina, 1st edn. ISBN: 978-8562938047
- Berander P, et al (2005) Software quality attributes and trade-offs. Blekinge Institute of Technology
- Bernardi L, Kamps J, Kiseleva J, Müller MJI (2015) The continuous cold start problem in e-commerce recommender systems. CoRR **abs/1508.01177**. URL <http://arxiv.org/abs/1508.01177>
- Boehm B (2015) Architecture-based quality attribute synergies and conflicts. In: Proceedings of the second international workshop on software architecture and metrics, pp 29–34. IEEE Press. <https://doi.org/10.1109/SAM.2015.18>
- Bornstein MH, Jager J, Putnick DL (2013) Sampling in developmental science: situations, shortcomings, solutions, and standards. Dev Rev 33(4):357–370. <https://doi.org/10.1016/j.dr.2013.08.003>
- Carvalho R, et al (2017) Evaluating an IoT application using software measures. In: International conference on distributed, ambient, and pervasive interactions. [https://doi.org/10.1007/978-3-319-58697-7\\_2](https://doi.org/10.1007/978-3-319-58697-7_2)
- Carvalho RM, Andrade RM, Oliveira KM (2020) How developers believe invisibility impacts nfrs related to user interaction. In: 2020 IEEE 28th international requirements engineering conference (RE), pp 102–112. IEEE. <https://doi.org/10.1109/RE48521.2020.00022>
- Carvalho RM, Andrade RMC, Oliveira KM (2018) 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. <https://doi.org/10.1109/RCIS.2018.8406651>
- Carvalho RM, Andrade RMC, Oliveira KM, Kolski C (2018) Catalog of invisibility requirements for ubicomp and IoT applications. In: International requirements engineering conference - RE, pp 88–99. IEEE. <https://doi.org/10.1109/RE.2018.00019>
- Carvalho RM, Andrade RMc, de Oliveira KM (2018) AQUARIUM - A suite of software measures for HCI quality evaluation of ubiquitous mobile applications. J Syst Softw 136:101–136. <https://doi.org/10.1016/j.jss.2017.11.022>
- Charmaz K (2006) Constructing grounded theory: a practical guide through qualitative analysis. SAGE Publications Ltd. ISBN = 978-0761973539
- Cho JY, Lee EH (2014) Reducing confusion about grounded theory and qualitative content analysis: similarities and differences. Qualit Rep 19(32):1–20
- Chung L, Nixon BA, Yu E, Mylopoulos J (2000) Non-functional requirements in software engineering, vol 5. Springer Science and Business Media, New York
- Costa CA, Yamin AC, Geyer CFR (2008) Toward a general software infrastructure for ubiquitous computing. IEEE Pervasive Comput 7(1):64–73. <https://doi.org/10.1109/MPRV.2008.21>
- Cysneiros LM (2007) Evaluating the effectiveness of using catalogues to elicit non-functional requirements. Workshop in Requirements Engineering
- Cysneiros LM, Werneck VM, Kushniruk A (2005) Reusable knowledge for satisficing usability requirements. <https://doi.org/10.1109/RE.2005.60>
- Egyed A, Grunbacher P (2004) Identifying requirements conflicts and cooperation: How quality attributes and automated traceability can help. IEEE Software. <https://doi.org/10.1109/MS.2004.40>
- Ghazi P, Glinz M (2018) An experimental comparison of two navigation techniques for requirements modeling tools. In: 2018 IEEE 26th international requirements engineering conference (RE), pp 240–250. IEEE. <https://doi.org/10.1109/RE.2018.00032>

21. Gliem JA, Gliem RR (2003) Calculating, interpreting, and reporting cronbach's alpha reliability coefficient for likert-type scales. Midwest research-to-practice conference in adult, continuing, and community ..
22. Godau R et al (2004) Qualitative data analysis software: Maxqda and maxdictio. *Qual Res J* 4(1):66
23. Hsieh HF, Shannon SE (2005) Three approaches to qualitative content analysis. *Qual Health Res* 15(9):1277–1288
24. Hu H, Zhang T, Tan Y, Xiang H, Fu C, Feng Y (2015) Semantic modelling and automated reasoning of non-functional requirement conflicts in the context of softgoal interdependencies 9:145–156. <https://doi.org/10.1049/iet-sen.2014.0153>
25. ISO/IEC 25010 (2011) ISO/IEC 25010. Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models
26. Jamieson S (2004) Likert scales: how to (ab) use them. *Med Educ* 38(12):1217–1218. <https://doi.org/10.1111/j.1365-2929.2004.02012.x>
27. Karvonen H, Kujala T (2014) Designing and evaluating ubicomp characteristics of intelligent in-car systems. In: 5 International conference on applied human factors and ergonomics, July, pp 1726–1737
28. Leite JCSdP, Cappelli C (2010) Software transparency. *Bus Inf Syst Eng* 2(3):127–139
29. Maia MEF, Fonteles A, Neto B, Gadelha R, Viana W, Andrade RMC (2013) LOCCAM - loosely coupled context acquisition middleware. In: Proceedings of the 28 annual ACM symposium on applied computing, pp 534–541. <https://doi.org/10.1145/2480362.2480465>
30. Mairiza D, Zowghi D (2011) Constructing a catalogue of conflicts among non-functional requirements. *Commun Comput Inf Sci*. <https://doi.org/10.1007/978-3-642-23391-33>
31. Mairiza D, Zowghi D, Gervasi V (2013) Conflict characterization and analysis of non-functional requirements: an experimental approach. In: International conference on intelligent software methodologies, tools and techniques. IEEE. <https://doi.org/10.1109/SoMeT.2013.6645645>
32. Mairiza D, Zowghi D, Nurmulliani N (2009) Managing conflicts among non-functional requirements. In: Australian workshop on requirements engineering. University of Technology, Sydney
33. Mairiza D, Zowghi D, Nurmulliani N (2010) Towards a catalogue of conflicts among non-functional requirements. In: International conference on evaluation of novel approaches to software engineering. SciTePress
34. Mehta R, Wang H, Chung L (2012) Dealing with NFRs for smartphone applications: a goal-oriented approach. In: Software engineering research, management and applications. [https://doi.org/10.1007/978-3-642-30460-6\\_8](https://doi.org/10.1007/978-3-642-30460-6_8)
35. Oates BJ (2005) Researching information systems and computing. Sage
36. Oh J, Lee U, Lee K (2019) Usability evaluation model for biometric system considering privacy concern based on mcdm model. *Security and Communication Networks* 2019. <https://doi.org/10.1155/2019/8715264>
37. Peixoto MM, Silva C (2018) Specifying privacy requirements with goal-oriented modeling languages. In: Proceedings of the XXXII Brazilian symposium on software engineering, pp 112–121. ACM. <https://doi.org/10.1145/3266237.3266270>
38. Sadana V, Liu XF (2007) Analysis of conflicts among non-functional requirements using integrated analysis of functional and non-functional requirements (Compsac), 0–3
39. Saha D, Mukherjee A (2003) Pervasive computing: a paradigm for the 21 century. *IEEE Comput Soc* 36(3):25–31. <https://doi.org/10.1109/MC.2003.1185214>
40. Santos IS, Andrade RM, Neto PAS (2015) Templates for textual use cases of software product lines: results from a systematic mapping study and a controlled experiment. *J Softw Eng Res Dev*
41. Satyanarayanan M (2001) Pervasive computing: vision and challenges. *IEEE Personal Communications*. <https://doi.org/10.1109/98.943998>
42. Scholtz J, Consolvo S (2004) Toward a framework for evaluating ubiquitous computing applications. *IEEE Pervasive Comput* 3(2):82–88. <https://doi.org/10.1109/MPRV.2004.1316826>
43. Sokolova M, Lapalme G (2009) A systematic analysis of performance measures for classification tasks. *Inf Process Manage* 45(4):427–437
44. Torres RC, Martins LEG (2014) NFR catalogues for RFID middleware. *J Comput Sci Technol* 14(2):102–108
45. Vasseur JP, Dunkels A (2010) Interconnecting smart objects with IP. Morgan Kaufmann. <https://doi.org/10.1016/B978-0-12-375165-2.00001-6>
46. Velela R, Cysneiros LM (2019) Towards an ontology-based approach for eliciting possible solutions to non-functional requirements. In: International conference on advanced information systems engineering, pp 145–161. Springer
47. Weiser M (1991) The Computer for the 21th Century. *Scientific America* special issue on communications, computers and networks, pp 94–104. <https://doi.org/10.1145/329124.329126>
48. Whitmore A, Agarwal A, Da Xu L (2015) The internet of things—a survey of topics and trends. *Inf Syst Front* 17(2):261–274
49. Wiegers K, Beatty J (2013) Software requirements. Pearson Education
50. Wijayarathna C, Arachchilage NA (2019) An empirical usability analysis of the google authentication API. In: Proceedings of the evaluation and assessment on software engineering, pp 268–274. ACM. <https://doi.org/10.1145/3319008.3319350>
51. Wohlin C, Runeson P, Höst M, Ohlsson MC, Regnell B, Wesslén A (2012) Experimentation in software engineering. Springer Science and Business Media, New York
52. Zinovatna O, Cysneiros LM (2015) Reusing knowledge on delivering privacy and transparency together. In: 2015 IEEE fifth international workshop on requirements patterns (RePa), pp 17–24. IEEE. <https://doi.org/10.1109/RePa.2015.7407733>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.