Chapter

Negative UX-Based Approach for Deriving Sustainability Requirements

Nelly Condori-Fernandez, Marcela Quispe-Cruz, Alejandro Catala, Joao Araujo and Patricia Lago

Abstract

In this chapter, a Negative User Experience (NUX)-based method for deriving sustainability requirements of persuasive software systems is proposed. The method relies on the analysis of NUX assessment, and the exploitation of relationships between the SQ model and the PSD model, which are well-known models for sustainability-quality in software systems and persuasive system design respectively. To illustrate the method, a user study has been conducted involving people in their real working environments while using specific software intended to change their behavior for preventing or reducing repetitive strain injury (RSI). The method allowed us to discover thirteen requirements that contribute to social, technical and economic sustainability dimensions.

Keywords: usability, sustainability-quality model, quality attributes, PSD model, persuasive software systems, UX assessment

1. Introduction

Persuasive technology (PT) can be defined as "design, research, and analysis of interactive computing products created to change people's attitudes or behaviors" [1]. As technology can be used as a promoter of sustainable behavior, many studies have investigated the possibilities to persuade people within the context of environmental sustainability (e.g., increase consumers' awareness of energy consumption [2]. However, most of these studies have shortcomings that limit their long-term effectiveness. Although behavioral models (e.g., Transtheoretical Model of behavior change [3], the Goal-setting Theory [4], the Fogg Behavior Model [5]) are very useful for conceptualizing the impact of persuasive technology, most of them cannot easily be applied to the design or assessment of persuasive systems directly because they do not provide appropriate methodological support [6, 7]. For example, through a user experience assessment of existing persuasive software applications, Condori-Fernandez et al. [7] found that some relevant non-functional requirements had not been addressed, and consequently users experienced negatively in using such kinds of systems.

As the identification and management of non-functional requirements in software projects are challenging [8], various assessment models have been proposed for software product quality (e.g., ISO/IEC 25010 quality model). In the software engineering community, a software sustainability model consists of both sustainability-related requirements and quality requirements (e.g., [9–11]). Lago et al. [12] defined software sustainability based on a four-dimensional model that adds the technical dimension to the social, environmental and economic dimensions that already appear in the Brundtland report [13]. Condori-Fernandez and Lago [9] proposed a Sustainability-Quality (SQ) model for supporting the identification of quality requirements that contribute to the four-dimensional model of software-intensive systems[1]. The multidimensional approach of Becker et al. [15] adds the individual dimension to the four sustainability dimensions [12]. However, Calero et al. [10] define sustainability only in terms of energy consumption, resource optimization and perdurability, and they do not consider the individual, social, and economic dimensions.

According to Assefa and Frostell [16], for a system to be deemed socially sustainable, it should at minimum enjoy wider social acceptance. In this respect, ensuring the quality of User Experience (UX) is important for increasing the likelihood of accepting socially software systems (e.g., [17]).

In order to provide support for discovering non-functional requirements (NFR) that contribute to sustainability dimensions, we present a Negative User Experience (NUX)-based approach for deriving sustainability-quality requirements, with special emphasis on the social and technical dimensions. Then, the approach is applied in existing software applications designed for preventing RSI.

The following sections provide a detailed account of our work. Section 2 describes the SQ model and PSD model on which the NFR discovery approach is based. Section 3 presents the NUX based approach for deriving sustainability requirements. As a result of applying the approach, we present first the design of a user study. And the discovered nonfunctional requirements and features are reported in Section 5. Finally, we draw the conclusions.

2. Background

In this work, we adopted the PSD model [18] as the theoretical framework for our research, and the SQ model [9].

2.1 The PSD model

The PSD model [18] is a recent conceptualization for designing, developing and evaluating persuasive systems. It consists of (a) the premises behind any persuasive system, (b) the persuasion context and (c) the persuasive software system features. Hence, according to the PSD model any persuasive system is based on eight premises detailed in [18] and listed here: P1: Useful; P2 User-friendly; P3: Unobtrusiveness; P4: Open for transparency; P5: Cognitive Consistency; P6: Incremental; P7: Information technology partiality; P8: Direct and indirect routes to persuasion.

The analysis of the *persuasion context* consists of looking into (1) the intent, (2) the event and (3) the strategy. The event comprises the use situation, user's characteristics, technological platform and environment. The strategy includes the message itself and the route to be used to achieve a goal.

The PSD model describes persuasive software system features grouped in four categories: (i) The **primary activity support category** focuses on supporting the activities that lead to achievement of the BCSS goals. These activities include

[1] Systems in which software interacts with other software, systems, devices, sensors and people [14].

reduction, tunneling, tailoring, personalization, self-monitoring, simulation and rehearsal. (ii) **Dialog support** refers to techniques/mechanisms to motivate users to use BCSS. This category includes praises, rewards, reminders, suggestions, similarity, liking and social role features. (iii) The **credibility category** relates to how to design a system so that it is more credible and thereby more persuasive. This category includes the following features: trustworthiness, expertise, surface credibility, real world feel, authority, third party endorsements, verifiability. iv) **The social influence category** describes how to design the system so that it motivates users by leveraging different aspects of social influence. The features that belong to this category are: social learning, social comparison, normative influence, social facilitation, cooperation, competition, recognition.

According to the PSD model, a behavior change can be divided into three categories:

- C-Change - or compliance change, is to make sure that the user complies with the request of the behavior change support system.
- B-Change - or behavior change, is to elicit a more enduring change than simply compliance a couple of times. Short time behavior change is easier to achieve than long-term behavior change.
- A-Change - or attitude change, is to influence the users' attitudes rather than behavior only. Changing the attitude of a user may be the most difficult type of change to achieve by a behavior change support system.

The outcomes of these C, B and A-Changes are the formation, alteration or reinforcement:

- F-Outcome - It means the formulation of a pattern in a situation where it did not exist before.
- A-Outcome - It means a change in the response of a user to an issue.
- R-Outcome – It means the reinforcement of current attitudes

2.2 The SQ model

The SQ model [9] is defined in terms of four *sustainability dimensions*: (i) *Technical dimension* addresses the long-term use of software-intensive systems and their appropriate evolution in an execution environment that continuously changes. (ii) *Economic dimension* focuses on preserving capital and (economic) value. (iii) *Social dimension* focuses on supporting current and future generations to have the same or greater access to social resources by pursuing generational equity. (iv) *Environmental dimension* aims at improving human welfare while protecting natural resources. For software-intensive systems, this dimension aims at addressing ecologic concerns, including energy efficiency and ecologic awareness creation.

Each dimension is characterized by a set of *Quality attributes*, which can be interdependent. Such dependency can be of two types: (i) it is *inter-dimensional* if it relates a pair of quality attributes defined simultaneously in two different dimensions (e.g. security defined in the *technical* dimension can influence security in the *social* dimension); and (ii) it is *intra-dimensional* if a dependency exists between two different quality requirements defined within the same dimension (e.g. in the *technical* dimension, security may depend on reliability).

Our SQ model supports the identification of sustainability design concerns, and the quality assessment of software architecture. The list of measurable attributes of the SQ model and corresponding contributions to the four dimensions can be found at [9], which has been empirically evaluated [19, 20].

3. NUX-based approach for deriving sustainability-quality requirements

In this section, we present the process needed for deriving sustainability-quality requirements from NUX results. As shown in **Figure 1**, the process consists of three stages: (i) UX assessment for understanding user needs, (ii) translating user needs into NFRs, and (iii) deriving sustainability-quality requirements from identified features and NFRs. The first two stages correspond to the NFR discovery approach proposed in [21], which uses the PSD model as a means to identify non-functional requirements (NFR) for a persuasive software system.

3.1 UX assessment for understanding user needs

UX assessment is supported by a wide range of research methods available, ranging from attitudinal evaluations (e.g. UX questionnaire, think-aloud) to behavioral evaluations (e.g. eye-tracking, activity trackers, emotions measurement). In this phase, in contrast to Sonnleitner et al. [22], we focus on negative User Experience (NUX) that is caused by the lack of fulfillment of needs during the interaction with a software product (a RSI software in our case). The effect of NUX impacts on the user attitudes ("what people say") and user behaviors ("what people do"). The outcome of this stage is the user feedback, information used for understanding user needs interacting with any persuasive software application.

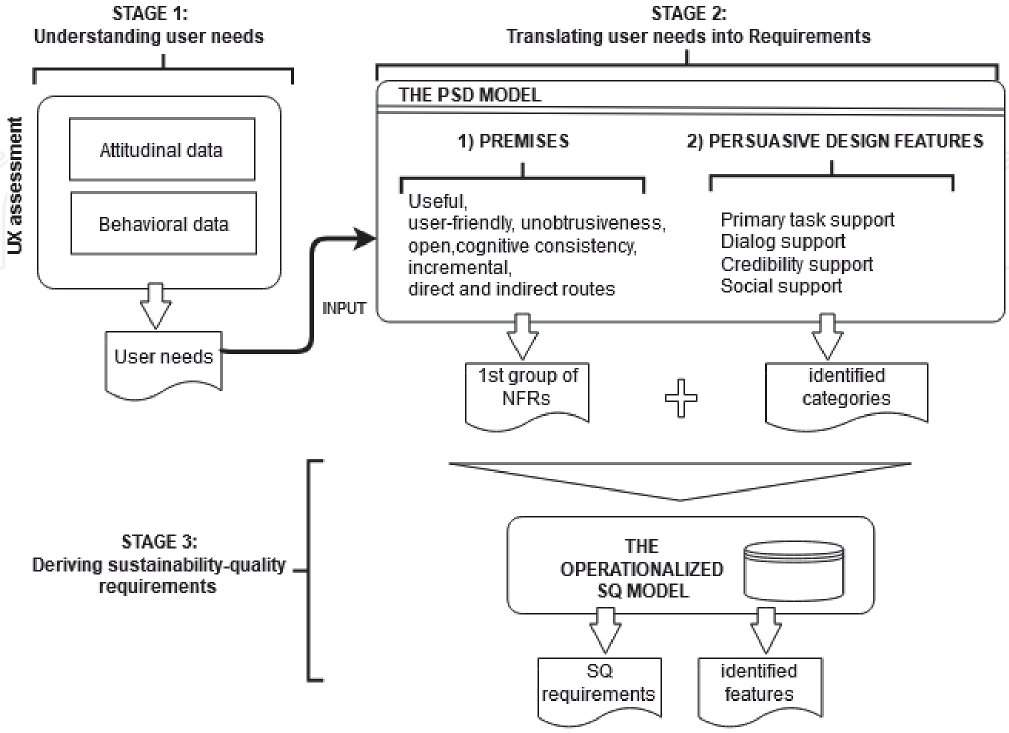


Figure 1.
Deriving sustainability quality requirements from NUX results.

3.2 Translating user needs into NFRs

In this stage, the user feedback from the UX assessment is used through two mapping steps to obtain a first group of NFRs and a set of categories of features respectively:

First step: by mapping the unfulfilled user needs (results) with the PSD model premises, we first discover a group of NFRs, which are directly discovered by translating the corresponding identified premises. Some examples of these NFRs are: usefulness (P1), cognitive consistency (P5), unobtrusiveness (P3), fun and enjoyment (P2), and trustworthiness (P4). Moreover, we also identify the affected categories of features described in the PSD model [18] (i.e., dialog support, primary activity support, perceived credibility, social influence).

Second step: by doing a content analysis, the unfulfilled user needs (expressed as user comments or suggestions) are categorized according to the identified categories.

3.3 Deriving sustainability-quality requirements

In order to facilitate the identification of quality requirements and features that contribute to the sustainability of software systems, a graph database has been created using the Neo4j Bloom tool. Such as shown in **Figure 2**, the data scheme includes elements from the SQ model (i.e., Dimension, Characteristic, Attribute) and the PSD model (i.e., feature and category), as well as the corresponding relationships among these elements. The relationships among the different elements (nodes) are represented by colored edges.

For instance, the edge COMPOSED_OF (green) is used for representing that a quality characteristic is defined in terms of a set of quality attributes of the SQ model. Similarly, there is a composition relationship between the features and categories of the PSD model (briefly introduced in Section 2.1).

Having the graph database, designers will be able to run (predefined) queries in the Neo4j browser. Query results are rendered either as a visual graph or a table format. **Figure 3** shows an example of a query result that displays all the quality attributes related to the usability characteristic and their corresponding contribution to the sustainability dimensions.

For illustrating the application of our approach for deriving sustainability quality requirements from a UX assessment, in the following section we present the design of a user study that aims to assess the experience of existing persuasive software applications for preventing RSI.

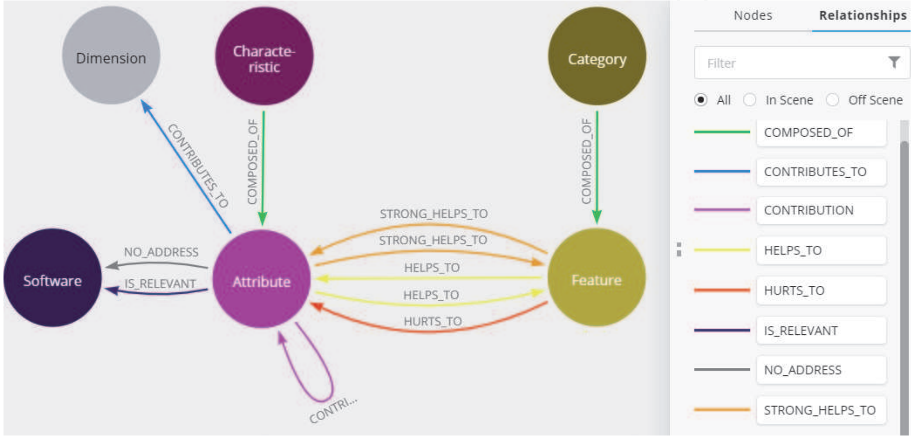


Figure 2.
Scheme with types of nodes and relationships in the graph database.

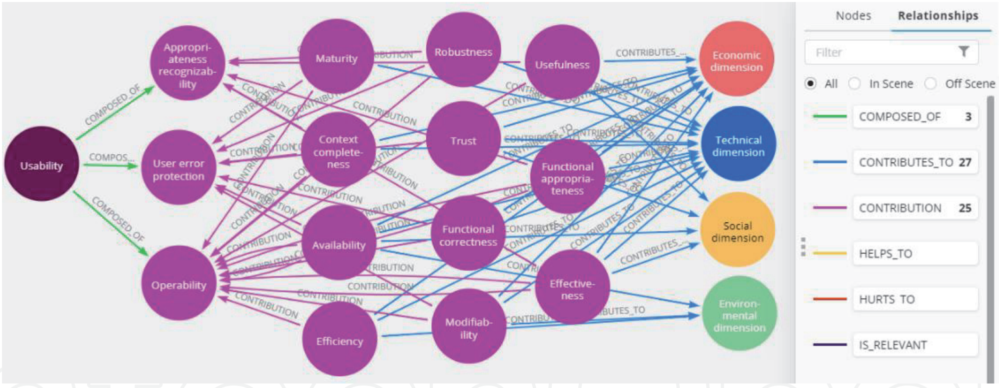


Figure 3.
Example of a query result in Neo4j: Quality attributes related to usability and their contributions to sustainability dimensions.

4. User study design

4.1 Participants

We targeted office employees working with computers (i.e. desktop, laptop) as RSI typically arises among this kind of user. A total of 30 people from four universities working in offices were invited to participate in the study. Twelve participants accepted our invitation. 3 were female and 9 male, whose age ranged from 21 to 45 years old.

4.2 Software and equipment

Table 1 shows the software available in the market, characterized by several commonplace functions, although their implementation can differ between systems:

1. Break reminder (BR) – Remind to take breaks based on several factors like elapsed time, how much/intensely you are working, natural rest patterns, times of day, etc.

2. Tracking (TK) – Track information like break-taking patterns, working hours per day, repetitions (e.g., keystrokes, mouse clicks).

3. Biofeedback (B) – Gain greater awareness of body functions primarily using instruments that provide information on the activity of those same systems.

Name	Features				Operative system
	Break reminder	**Tracking**	**BioFeedback**	**Training**	
Stretch, 2013	X	X		X	Windows
Workrave, 2013	X	X		X	Windows
Pokoy, 2014	X				Linux
Quick Pause, 2014	X	X			Windows
AntiRSI, 2016	X				Mac OS
SmartBreak, 2016	X	X	X		Windows, Mac OS
EyeLeo, 2016	X			X	Windows

Table 1.
Common functionalities of RSI software.

4. Training (TN) – Provide information on topics including workstation setup, body positioning, work-efficiency tips, psycho-social information, etc.

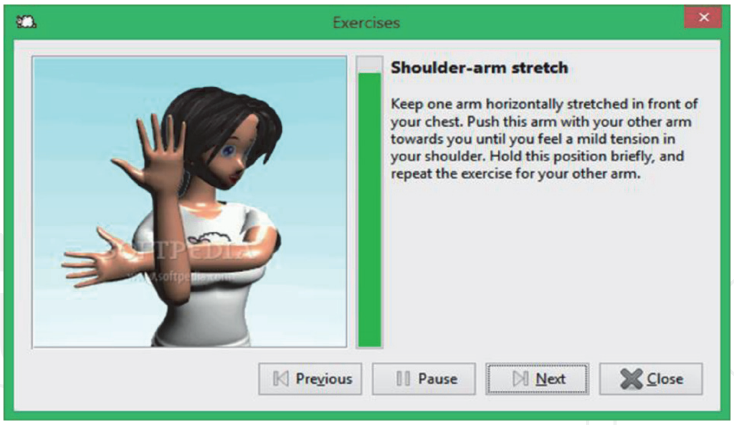
For practical reasons and given that participants self-reported to work with either Windows 10 Pro operating system or MAC OS, they were offered the two most complete options available, Workrave or SmartBreak, so that the study can run in their natural working environment. Although both applications, Workrave and SmartBreak, deliver reminders and enforce to take breaks, there are some differences as explained next.

Workrave is probably one of the most complete applications of its class. It considers micro-pauses, rest breaks, and guidance for exercise routines. This software is based on timers and keyboard/mouse activity, which determine when the actions must be displayed on screen. The user interface offers to configure a good number of parameters and provides a monitor on micro-breaks, rest breaks and working hour limit. The remarkable feature is the Training support, using an animated virtual human to demonstrate the exercises in addition to a textual description (see **Figure 4**), which could have some positive effects on attaining coaching goals [23].

SmartBreak has a minimal user interface consisting of a "stress" level bar, and an overlay message is displayed in the center of screen when a break is suggested. The unique feature supported by SmartBreak is the one on BioFeedback, trying to raise awareness on user's stress state, which in this particular application is based on keyboard and mouse usage only.

According to the PSD model, two types of behavior changes are addressed by Workrave and SmartBreak, as shown in **Table 2**.

The study was conducted in the natural settings of the subjects for 1 week (5 working days). Hence, they used their own computer and had to install by themselves the software application of their choice. The 8 subjects with a Windows computer



Shoulder-arm stretch

Keep one arm horizontally stretched in front of your chest. Push this arm with your other arm towards you until you feel a mild tension in your shoulder. Hold this position briefly, and repeat the exercise for your other arm.

Figure 4.
Training coach demonstrating exercises to stretch body muscles in Workrave.

	B-change	A-change
F-Outcome	Workrave helps people to form healthier habits by doing stretching exercises	
A-Outcome		SmartBreak helps people to get awareness on their stress level determined based on the way how keyboard and mouse is used.

Table 2.
Outcome/changes analysis for Workrave and SmartBreak.

installed Workrave, whereas 4 users who had a laptop with Mac OS installed SmartBreak. Only one participant had used an RSI software in the past (i.e. Workrave).

4.3 Instrumentation

Participants were given the following instruments:

- An UX questionnaire that is based on the User Needs Questionnaire (UNeeQ) [22]. It is composed of two parts described as follows:

  User needs fulfillment: The first part of this questionnaire measures the user experience of a product or product concept based on the needs fulfillment. Given our assessment focus on BCSS for recovering and preventing RSI, we consider the premises of the PSD model for the formulation of 10 items regarding specific user needs (See **Table 3**). All the items were measured on a five-point rating scale (0–4) ranging from “not at all” to “highly”.

  Positive and Negative UX: The second part of UNeeQ consists of six items regarding overall positive and negative UX measured also in a five-point rating scale (0–4). These overall UX items correspond to positive/negative emotions, feelings and experience. In our study with the purpose of avoiding confusions with the items regarding feelings, we decided to remove the items regarding emotions. This is because usually feelings and emotions are used interchangeably, but there are distinct differences between these two words. Feelings are mental associations, whereas emotions create reactions altering physical state. Emotions could be measured more objectively with techniques such as facial recognition, or monitoring physiological data (e.g. Skin conductance).

- Open questions to capture in free form text any additional issue regarding user experiences and feelings, either positive or negative, as well as to allow participants to make suggestions for improvements to the software applications.

- Demographic questions included age, gender, weekly working hours, and more specific questions that allowed us to know if our respondents suffered RSI in the past (i.e. identifying symptoms and possible triggers).

Premise	Items		Day1	Day5
P1	… I was doing something good for my body and mind.	I1	X	X
P2	… joy and pleasure.	I2	X	X
P4	… safe from uncertainties.	I3		X
P1, P6	… that I was more physically active (e.g. stretching exercises).	I4		X
P1, P6	… that I was less mentally tired	I5		X
P3, P2	… disturbed by the RSI software	I6		X
P4, P6	… that I was developing a deeper understanding on how to prevent RSI	I7	X	X
P5	… that I was acting according to my true interest (e.g. moving naturally to well-being)	I8		X
P5	… that the system was supporting me to make my commitments (e.g. changing work habits)	I9		X
P6	… a better sense of physical well-being.	I10		X

Table 3.
Adapted user needs questionnaire for RSI software.

4.4 Procedure

Each participant was asked to install one of the selected RSI software (Workrave or Smartbreak) in his/her own computer, and configure the timing parameters according to the given instructions.

Then, we asked them to use the RSI software while they were working with their computer for 1 week.

During the study, participants were allowed to adjust the values of any parameter (e.g. break times, sound of alarm) whenever they considered necessary. As the study was conducted in their natural working environment, they were allowed to abandon and not to finish the study. They were informed beforehand about the length of the study and the existence of a final questionnaire that should only be filled in if the study rules were met regarding duration and working with the computer normally.

Given UX can change over time, the data collection was carried out at two moments: (i) At the end of the first day, participants were asked to complete a first UX questionnaire. As some items of the UX questionnaire could not be experienced immediately by users, we considered only a subset of items for the first round listed in **Table 3**. (ii) At the end of the study, participants were asked to complete the second UX questionnaire.

5. Deriving sustainability quality requirements from UX assessment

In this section we present the results obtained following the NUX-based discovery process introduced in Section 3.

5.1 First stage: UX assessment for understanding user needs

According to our demographic data collected at the beginning of the study, we found that most of our participants worked more than 40 hours per week. The distribution is as follows: 5 subjects reported to work more than 45 hours per week. 5 subjects worked between 40 and 45 working hours. Only 2 subjects worked between 38 and 40 weekly hours.

Regarding RSI symptoms, most of the participants indicated felt fatigue, aching or shooting pain. 3 subjects did not experience none of the symptoms shown in **Table 4**. One of these 3 subjects decided to drop-out from the study after experiencing with the RSI software for 1 day.

Nine of our subjects considered stress as the main trigger of their RSI symptoms, followed by a bad ergonomic posture (8 subjects). Surprisingly, we found that omitting breaks and maximum exposure to technology were considered by less than half of the subjects that experienced RSI symptoms.

5.1.1 First impressions

At the beginning of the study we found that the RSI software was not enjoyable enough and pleasant. The most positive answers were given by 3 participants using Workrave, who indicated have enjoyed -more or less- with the software. However, despite this non-positive feeling, 7 of out 11 participants felt that using an RSI software was somewhat useful ("more or less" and "significantly high") to support their own wellness (body and mind). Over half of the participants reported that they understood how to prevent RSI during their first interaction with the software. **Figure 5** shows the frequency distribution of the responses in detail regarding these three items that concern about enjoyment, usefulness, and awareness respectively.

RSI symptoms	Responses
Aching or shooting pain.	50% (6)
Fatigue or lack of strength.	50% (6)
Weakness in the hands or forearms.	25% (3)
Tremors, clumsiness and numbness.	17% (2)
Chronically cold hands, particularly the fingertips.	17% (2)
Difficulty with normal activities like opening doors, turning on a tap.	8% (1)

Table 4.
RSI symptoms experienced by participants.

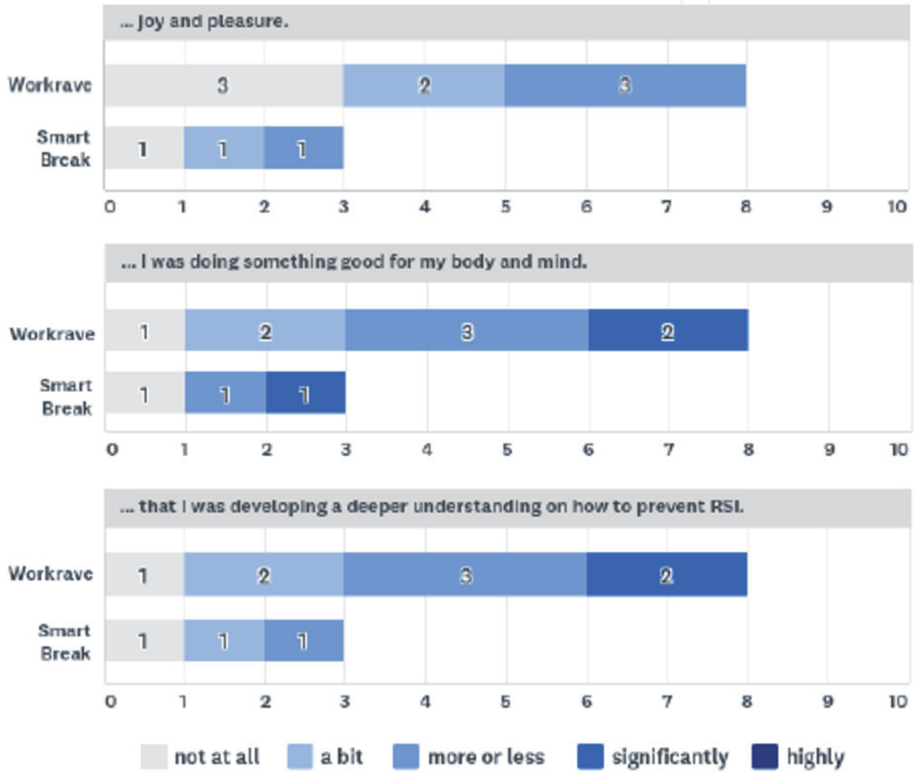


Figure 5.
User experience after day 1 (N = 11).

5.1.2 UX based on the need's fulfillment

A total of 9 participants completed the study and filled in the final questionnaire after day 5. **Table 5** shows the answers' frequency to the user experience questionnaire items and the weighted average. After several days of use, it is confirmed that the extent of "enjoyment and pleasant" is low. The "safe from uncertainties" score reinforces the idea that participants were not fully sure on how the break reminder and monitoring features work (e.g. how the breaks based on the stress level is determined by SmartBreak).

Most of the participants perceived that the software provided "a bit" support for items related to the physical and mental well-being. This feeling was confirmed by the last two items, where participants did not show to have the impression that the software was supporting their commitments and supporting behavior change in a natural way; therefore, more feedback from the software could be needed along with a different strategy to communicate how the software works and help to attain goals. Overall, it seemed that participants have got a moderate belief of "doing

Items	0	1	2	3	4	W-Avg
Joy and pleasure.	4	3	2	0	0	0.77
a better sense of physical well-being.	3	3	2	1	0	1.11
I was doing something good for my body and mind.	0	3	3	3	0	2
Safe from uncertainties.	6	0	2	1	0	0.77
I was more physically active (e.g. stretching exercises).	2	4	1	2	0	1.33
I was less mentally tired	4	3	1	1	0	0.89
Disturbed by the RSI software	0	4	2	1	2	2.11
I was developing a deeper understanding on how to prevent RSI	0	3	3	3	0	2
I was acting according to my true interest (e.g. moving naturally to well-being)	1	3	2	3	0	1.78
The system was supporting me to make my commitments (e.g. changing work habits)	2	3	2	2	0	1.44

Table 5.
UX based on the needs fulfillment on day 5 (N = 9).

something good for their body and mind", and have been developing some moderate understanding on how to prevent RSI.

5.1.3 UX variation analysis

In order to understand the possible variations of the user experience reported after the first day, we carried out an individual analysis with the participants (S1, S2, S5, S6, S9, S11, S12) who finished the study.

What stands out in **Table 6** is that there was not any variation in enjoyment and pleasant from day 1 to day 5. Similarly occur for the other two items concerning

Items		Day 1	Day5
joy and pleasure.	4		
	3		
	2	S1, S6	S6
	1	S2, S12	S1, S2, S12
	0	S5, S9, S11	S5, S9, S11
I was doing something good for my body and mind.	4		
	3	S1, S6	S2, S6, S9
	2	S5, S12	S1, S12
	1	S9	S5, S11
	0	S2, S11	
I was developing a deeper understanding on how to prevent RSI	4		
	3	S6	S2, S6, S9
	2	S1, S12	S1
	1	S2, S9	S5, S11, S12
	0	S5, S11	

Table 6.
User experience variations along the study.

usefulness and awareness. However, what we can remark is that there were two few positive variations (colored in green), which correspond to subjects S2 and S9. Both subjects started rating negatively, but at the end of the study, they considered that the software was good for their own wellness, and to understand better on RSI prevention. Both continued considering their level of enjoyment as negative though.

This was mainly because breaks used to be considered as interruptions. (S2) *"Interruptions while typing, and (interruptions) of the reasoning flow while working"*. But then these breaks were understood as a time for resting. (S9) *"It reminds me that I must relax"*. Unfortunately, this change of attitude occurred only in these two subjects.

We observed also a negative variation (colored in red) that was detected in subject S1, and subject S12. (S1) *"When focusing on a task, being required to take a break resulted to be from time to time counterproductive"*.

5.1.4 Subjective rating of positive and negative UX

Participants rated the four items of in second part of the UX questionnaire that correspond to positive/negative feelings and experience (i.e. "I had a positive experience", "I had a positive feeling", "I had a negative experience", "I had a negative feeling"). The collected responses at day 1 and day 5 (see **Appendix A**) have been processed to facilitate their assessment, leading to overall indicators of UX presented in **Tables 7** and **8**. **Table 7** depicts the frequency for both positive and negative UXs, in such a way that the overall UX is calculated by aggregating the experience and feeling single items. It reports the distribution of the overall UX at the beginning and at the end of the study. It clearly shows that there were a mix of positive and negative user experiences.

In order to calculate the overall UX scores, we averaged the values assigned by the respective participants using Workrave (WR) or SmartBreak (SB) along the study, which are depicted in **Table 8**. It shows the corresponding overall UX of both RSI software was somewhat negative.

On the overall positive UX, data revealed that software applications did not manage to raise a significant or very positive UX. At the end, the ratings for positive experiences suggest that systems were better considered than they were at the beginning, low to moderate though.

Overall negative UX frequencies suggest that at the beginning there were not much negative UX (notice that 46% replied *Not at all*), but participants manifested that negative experiences and feelings became more intense at the end of the study.

	Highly		More or less		Not at all	Not at all		More or less		Highly
At the beginning			**Negative**					**Positive**		
Workrave	9%	18%	9%	5%	32%	23%	14%	18%	9%	9%
Smart Break	0	0	9%	5%	14%	9%	9%	9%	0	0
Total	9%	18%	18%	10%	46%	32%	23%	27%	9%	9%
At the end										
Workrave	0	22%	17%	11%	17%	11%	11%	22%	22%	0
Smart Break	0	0	6%	28%	0	11%	0	17%	6%	0
Total	0	22%	22%	39%	17%	22%	11%	39%	28%	0

Table 7.
Frequency distribution for overall positive and negative UX.

	Day 1	Day5
Positive	1.4	1.73
Negative	2.68	2.51

Table 8.
Overall UX scores on day 1 and day 5.

5.2 Second stage: translating user needs into requirements

In order to discover requirements that should be addressed by a persuasive software application (RSI software in particular) designed for helping people change their behavior to achieve healthier habits, our analysis focuses on the UX results and open-ended descriptive user responses (comments and suggestions).

By mapping user needs (first step of second stage) that were not fulfilled and what people would like to have (results presented in Section 5.1), we found that the premises P1 (Useful), P2 (user friendly), P3 (Unobtrusiveness), P4(Open), P5 (Cognitive consistency), P6(Incremental) were affected, which were accordingly translated into usefulness, pleasure, unobtrusiveness, transparency, cognitive consistency, and awareness requirements. For instance, from the questionnaire (I4, I5) some participants showed their perceptions that the RSI software was not so useful for keeping them more physically active and less mentally tired. It is also important to remark that the usefulness perception of a software system could vary along the time. (i.e. at the beginning some users expected the system would help to change their habits positively but at the end this did not happen).

Analyzing these unfulfilled user needs, we found that both RSI software apps lack some features related to dialog support, primary activity support and credibility categories (second step of second stage). For example, **Table 9** shows the discovered requirements like transparency, awareness, and consistency, which are helpful for implementing features from the *dialog support* category (i.e., providing relevant, motivating and adequate feedback to its users). Other important requirements from the primary activity support category are pleasure, usefulness and unobtrusiveness. And transparency that helps also to the features from the credibility category.

5.3 Third stage: deriving sustainability-quality requirements

Considering the relations of the SQ model for each NFR(attribute) identified in the previous stage, we found out other relevant requirements that contribute to the sustainability dimensions.

For example, regarding the *primary activity support*, pleasure is positively affected by adaptability. This attribute is relevant for contributing to the social and technical sustainability of the RSI software. But as shown in **Table 9**, this requirement was not adequately addressed. If the RSI software had more information about the current situation, an adapted modality of breaks reminder could be delivered, which could influence positively on UX too.

It similarly occurred for usefulness, a key quality attribute that contributes to the economic, social and technical dimensions (see **Appendix B**), and it is positively related to learnability and effectiveness attributes (both from social dimension). From the UX assessment (**Table 9**), we corroborated that learnability and effectiveness were not satisfied by users. For instance, although Workrave provides an animated coach to teach a series of body movements, the UX could be affected if the content itself was not easy to learn/understand (learnability), or users were not able to change some of the exercises proposed by the system (tailorability).

Category of features	UNeeQ Item	User comments (C) and suggestions (S)	Premises	NFR
Dialog Support	I7, I10	C: "I cannot really say I may be satisfied with RSI prevention as I do not know about the state of my condition." S: "Give supporting messages that create awareness on RSI (why it is important)..."	P4, P6	**Transparency, Awareness**
	I8, I9	C: "the RSI software made me realize that I am frequently taking brakes even without tool support"	P5	**Cognitive Consistency**
Primary activity support	I2	C: "I was always annoyed by it, especially when it freezes the screen while I am in the middle of some work." C: "the RSI software resulted to be quite annoying when a break was triggered" C: "The animation starts the exercise but the instructions are in a long hard to read text, so at the beginning it is annoying to read and see how the animation started and you cannot follow it as still reading"	P2	**Pleasure** Usability (operability)
	I1, I4, I5	C: "It reminds me that I must relax".	P1, P6	**Usefulness**, Effectiveness
	I6	S: "..somehow having a do not disturb mode in which it does not interrupt you but uses a more subtly easy to inform you" C: "Interruptions while typing and (interruptions) of the reasoning flow while working."	P3, P2	**Unobtrusiveness**
	I6	C: "I was busy sending some important emails, and I had to wait for a couple of minutes for the computer to unlock." C: "I tried to follow the instructions of the break, but sometimes is not possible for me because you cannot interrupt your activity of you are completing something urgent or you are discussing our taking with someone else using the screen..."	P3	Timeliness
		S: "It would also nice to be able to choose your set of exercises, or changing them for time to time." S: "blocking interruption stressed me. So, I preferred the non-blocking interruption"	P6, P8	Tailorability
		C: "Suggested exercises are not clear and I had a small period of time to understand them"	P8	Learnability
		C: "The RSI software resulted to be quite annoying when a brake was triggered during a Skype call." C: "The software actually did not do anything. I did not get any alert, any break, etc. (I was following a course on coursera, which meant mostly watching a video) so maybe that is the reason."		Adaptability
Credibility	I3	C: "The system showed me a stress bar when I did not feel to be stressed"		Trust
		C: "I better understood the meaning of the countdown in the breaks (of the mouse is touched then it is stopped...). I had the feeling that when I skipped breaks, there were appearing often, and therefore interrupting my working activity is not nice."	P4	**Transparency**

Table 9.
NFR found from the UX assessment.

Given preferences can vary significantly over users with different profiles (ages, interests, etc.), tailorability is an important requirement that must also be considered. From user suggestions, we found that this requirement was not fully addressed. We also found that the break reminder could have a negative effect on

NFR	Sustainability Dimension	Candidate Features
Usefulness	Economic, Social and Technical	Self-monitoring (HELPS_TO–>)
Unobtrusiveness	Social	Tunneling and Reduction (<—-HELPS_TO)
Timeliness	Social	Rehearsal and Tunneling (<—-HELPS_TO)
Tailorability	Social and Technical	Tailoring and Personalization (<—-HELPS_TO)
Transparency	Social	Suggestion, Real-world feel, Praise, Verifiability, Authority, Reminders, Third-party endorsements, Trustworthiness, Rewards, Expertise, Surface credibility (<—-HELPS_TO)

Table 10.
Example of candidate features and sustainability dimension covered by discovered NFR.

the user's experience due to that this reminder sometimes occurred in a not favorable time (lack of timeliness).

Finally, the third category, *credibility*, consists of two requirements that relates to the confidence on software systems that behave as intended (trust) and transparency of the system (implications and consequences of functionality such as skipping or postponing breaks or when they will be offered should be clear to the user regardless of the internal details on how they are being handled or implemented). Trust, an attribute from the social and technical dimension of the SQ model, was derived thanks to the positive relation with the pleasure attribute from the social dimension.

Table 9 presents the 6 NFRs discovered at the second stage (text in bold in the last column) and 7 NFR derived from the SQ model at the third stage. By means of our approach, in this stage we can also determine (i) the new potential features that should be considered in further versions of the software for RSI prevention, and the sustainability dimensions that can be covered with the implementation of the discovered NFRs. Some of the identified features as potential improvements are shown in **Table 10**. For example, the self-monitoring feature might help to address the usefulness of applications because the system would be able to provide other means to track user status through self-monitoring.

6. Threats to validity

In this section, we discuss the threats to the validity of our user study.

6.1 Internal validity

A threat related to the *instrumentation* could be caused by the questionnaires used during the study. Aimed at mitigating this threat, the online questionnaires were carefully reviewed and tested before running the user study. Given the type of collected data (e.g. actual number of working hours), we decided to make responses anonymous (IP were not recorded) and participants were informed of such anonymity. But for our study, responses traceability between the first and second questionnaire was required. Thus we asked participants to create a fictitious username that still kept their responses anonymous.

Considering our user study took five working days and it was conducted in a real setting with a null control, the threat of having a high number of dropouts (*mortality*) could not be reduced. But we tried to reduce it, by informing to the potential participants beforehand about the length of the study, and the existence of a final questionnaire that should only be filled in if the study rules were met, regarding duration and having worked with the computer normally during that time. We got twelve participants from thirty invitees, who accepted our invitation. Once the study was running we sent a couple of emails to remind deadlines for completing the respective UX questionnaires. At the beginning of the study we had one participant who dropped out from the study. And two more at the end of the study.

6.2 External validity

The first threat to consider is concerned with the *interaction of selection and treatment*. This is the effect of having non-representative subjects. We attempt to mitigate this threat by inviting office employees working with computers (i.e., senior researchers, programmers), as potential users interested in preventing RSI. Regarding the size of our sample (2 partial responses, and 9 full responses), it could be considered as small. But according to Bevan et al. [24], 80% of usability findings are discovered after five participants. It is important to point out that our user study focuses on the user experience and not on the effectiveness of the RSI software for changing the behavior, and therefore we consider that the size of our sample is still good enough for illustrating the application of our NUX-based approach. Indeed, if the focus were on the behavior change effects, studies would necessarily be longer.

The second threat is concerned with the *interaction of setting and treatment*. We mitigate this threat by conducting the user study in their natural working office environment. Although originally our data collection plan was for 2 weeks, the study was executed only for 1 week due to negative experiences already experienced during the first 5 working days. Another threat is related to the representativeness of the selected experimental objects (RSI software), although we focus on two desktop applications, they cover common features of RSI software. However, we cannot generalize the results to any persuasive software system since according to the PSD model, the social influence category was not covered. Thus, this limitation can be mitigated by means of further replications including other types of persuasive software systems (e.g. activity trackers to encourage physical activity in non-working environments at leisure time) where other requirements could be discovered.

7. Conclusions

The present research presents an approach for deriving sustainability quality requirements from negative user experience. It starts by understanding the fulfillment of user needs through a UX assessment (first stage). We used the PSD model as the theoretical framework for designing our empirical research of persuasive systems. A user study with 12 subjects working on their natural office environment was carried out. Our UX assessment focused on two popular software systems for preventing RSI (i.e. Workrave and Smartbreak). This study revealed that generally most of the subjects had a negative UX with both RSI software at the end of the study. It is important to remark that from our UX variation analysis, UX of the participants was not much negative at the beginning (46% replied “Not at all”), but along the study their experiences and feelings were changing to be more negative.

From the UX results based on needs fulfillment and user comments/suggestions, 6 NFRs of RSI software (i.e., usefulness, pleasure, unobtrusiveness, transparency, cognitive consistency, and awareness) were discovered (second stage). Then 7 additional NFRs were derived from the SQ model by means of the existing predefined relations among quality attributes that contribute to the social, technical, economic, and environmental dimensions (third stage). All these requirements are helpful for identifying candidate features related to dialog support, primary activity support and credibility categories. Addressing requirements such as awareness, pleasure and consistency, RSI software apps will be in a better capability to provide relevant, motivating and adequate feedback to users. The second group of requirements such as unobtrusive, timely, learnability, usability (operability), usefulness, adaptability will enable RSI software apps to provide a better activity support for the system goal achievement (i.e. reduction and prevention of RSI, main goal of our selected software). The third group of requirements relates to the perceived credibility of the software system, where transparency and trust are very important requirements that positively could affect users to continue using the software system.

In order to support the derivation of requirements and features for improving the sustainability of persuasive systems (RSI software apps in our case), our SQ model along with its relationships with the PSD model has been implemented in Neo4j Graph platform, allowing us to search for potentially relevant attributes by querying and navigating the model interactively.

Appendix A: data collection

At the beginning	0	1	2	3	4
I had a positive experience					
Workrave	3	1	2	1	1
Smart Break	1	1	1	0	0
I had a positive feeling					
Workrave	2	2	2	1	1
Smart Break	1	1	1	0	0
I had a negative experience					
Workrave	3	1	1	2	1
Smart Break	2	0	1	0	0
I had a negative feeling					
Workrave	4	0	1	2	1
Smart Break	1	1	1	0	0

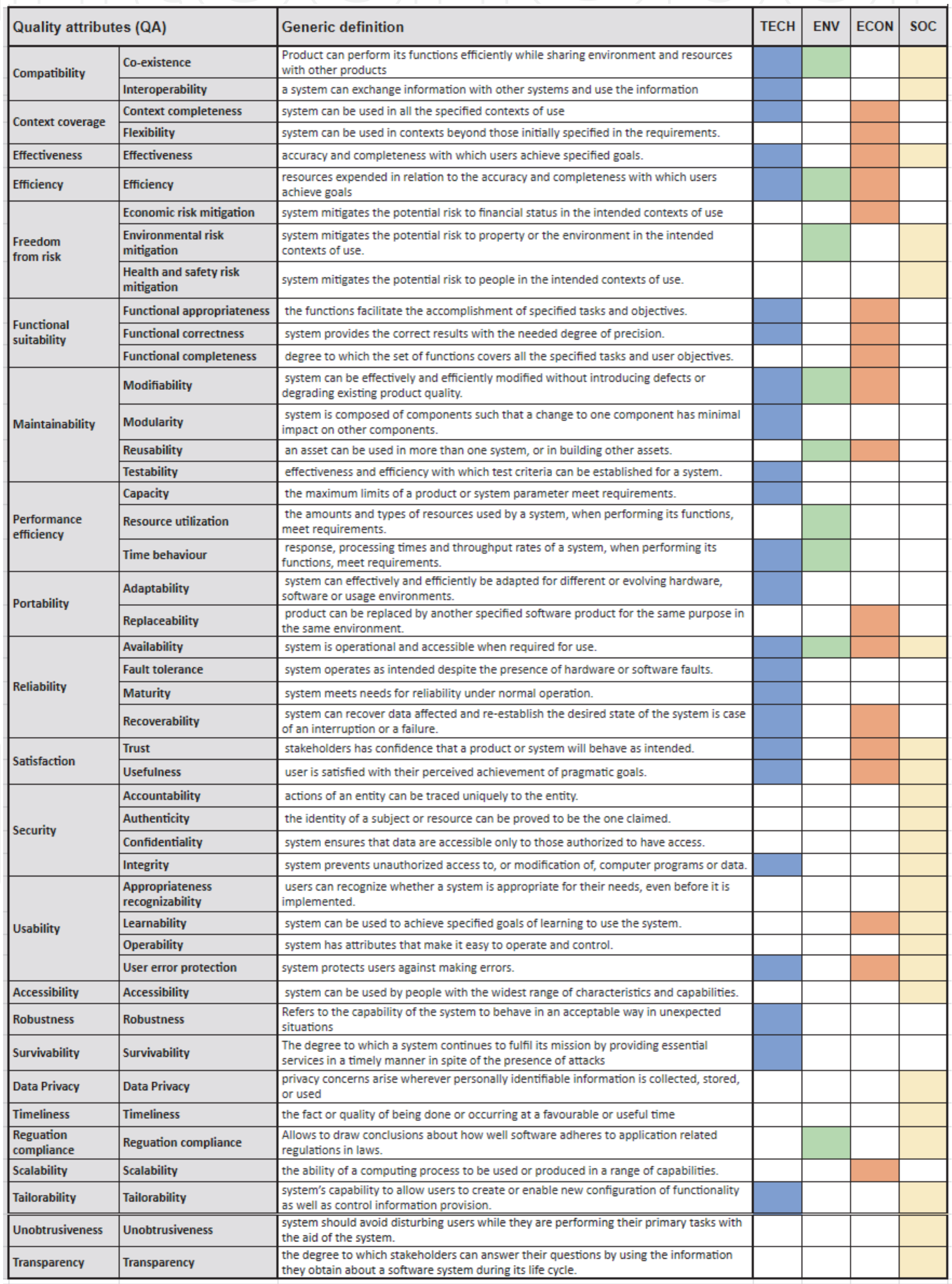
Table 11.
Overall UX at the beginning of the study (N = 11). Data collected from the RSI case presented in [7].

At the end	0	1	2	3	4
I had a positive experience					
Workrave	1	2	1	2	0
Smart Break	1	0	1	1	0
I had a positive feeling					
Workrave	1	0	3	2	0
Smart Break	1	0	2	0	0
I had a negative experience					
Workrave	2	1	1	2	0
Smart Break	0	3	0	0	0

At the end	0	1	2	3	4
I had a negative feeling					
Workrave	1	1	2	2	0
Smart Break	0	2	1	0	0

Table 12.
Overall UX at the end of the study (N = 9). Data collected from the RSI case presented in [7].

Appendix B: the SQ model



Quality attributes (QA)		Generic definition	TECH	ENV	ECON	SOC
Compatibility	Co-existence	Product can perform its functions efficiently while sharing environment and resources with other products	🟦	🟩		🟨
	Interoperability	a system can exchange information with other systems and use the information	🟦			🟨
Context coverage	Context completeness	system can be used in all the specified contexts of use	🟦		🟧	
	Flexibility	system can be used in contexts beyond those initially specified in the requirements.			🟧	
Effectiveness	Effectiveness	accuracy and completeness with which users achieve specified goals.	🟦		🟧	🟨
Efficiency	Efficiency	resources expended in relation to the accuracy and completeness with which users achieve goals	🟦	🟩	🟧	
Freedom from risk	Economic risk mitigation	system mitigates the potential risk to financial status in the intended contexts of use			🟧	
	Environmental risk mitigation	system mitigates the potential risk to property or the environment in the intended contexts of use.		🟩		🟨
	Health and safety risk mitigation	system mitigates the potential risk to people in the intended contexts of use.				🟨
Functional suitability	Functional appropriateness	the functions facilitate the accomplishment of specified tasks and objectives.	🟦		🟧	
	Functional correctness	system provides the correct results with the needed degree of precision.	🟦		🟧	
	Functional completeness	degree to which the set of functions covers all the specified tasks and user objectives.			🟧	
Maintainability	Modifiability	system can be effectively and efficiently modified without introducing defects or degrading existing product quality.	🟦	🟩	🟧	
	Modularity	system is composed of components such that a change to one component has minimal impact on other components.	🟦			
	Reusability	an asset can be used in more than one system, or in building other assets.		🟩	🟧	
	Testability	effectiveness and efficiency with which test criteria can be established for a system.	🟦			
Performance efficiency	Capacity	the maximum limits of a product or system parameter meet requirements.	🟦			
	Resource utilization	the amounts and types of resources used by a system, when performing its functions, meet requirements.		🟩		
	Time behaviour	response, processing times and throughput rates of a system, when performing its functions, meet requirements.	🟦	🟩		
Portability	Adaptability	system can effectively and efficiently be adapted for different or evolving hardware, software or usage environments.	🟦			
	Replaceability	product can be replaced by another specified software product for the same purpose in the same environment.			🟧	
Reliability	Availability	system is operational and accessible when required for use.	🟦	🟩	🟧	🟨
	Fault tolerance	system operates as intended despite the presence of hardware or software faults.	🟦			
	Maturity	system meets needs for reliability under normal operation.	🟦			
	Recoverability	system can recover data affected and re-establish the desired state of the system is case of an interruption or a failure.	🟦		🟧	
Satisfaction	Trust	stakeholders has confidence that a product or system will behave as intended.	🟦		🟧	🟨
	Usefulness	user is satisfied with their perceived achievement of pragmatic goals.	🟦		🟧	🟨
Security	Accountability	actions of an entity can be traced uniquely to the entity.				🟨
	Authenticity	the identity of a subject or resource can be proved to be the one claimed.				🟨
	Confidentiality	system ensures that data are accessible only to those authorized to have access.				🟨
	Integrity	system prevents unauthorized access to, or modification of, computer programs or data.	🟦			🟨
Usability	Appropriateness recognizability	users can recognize whether a system is appropriate for their needs, even before it is implemented.				🟨
	Learnability	system can be used to achieve specified goals of learning to use the system.			🟧	🟨
	Operability	system has attributes that make it easy to operate and control.				🟨
	User error protection	system protects users against making errors.	🟦		🟧	🟨
Accessibility	Accessibility	system can be used by people with the widest range of characteristics and capabilities.				🟨
Robustness	Robustness	Refers to the capability of the system to behave in an acceptable way in unexpected situations	🟦			
Survivability	Survivability	The degree to which a system continues to fulfil its mission by providing essential services in a timely manner in spite of the presence of attacks	🟦			
Data Privacy	Data Privacy	privacy concerns arise wherever personally identifiable information is collected, stored, or used				🟨
Timeliness	Timeliness	the fact or quality of being done or occurring at a favourable or useful time				🟨
Reguation compliance	Reguation compliance	Allows to draw conclusions about how well software adheres to application related regulations in laws.		🟩		🟨
Scalability	Scalability	the ability of a computing process to be used or produced in a range of capabilities.			🟧	
Tailorability	Tailorability	system's capability to allow users to create or enable new configuration of functionality as well as control information provision.	🟦			🟨
Unobtrusiveness	Unobtrusiveness	system should avoid disturbing users while they are performing their primary tasks with the aid of the system.				🟨
Transparency	Transparency	the degree to which stakeholders can answer their questions by using the information they obtain about a software system during its life cycle.				🟨

Figure 6.
Quality attributes of the SQ model defined according to [9, 19].

Author details

Nelly Condori-Fernandez[1,2]*, Marcela Quispe-Cruz[3], Alejandro Catala[4], Joao Araujo[5] and Patricia Lago[2,6]

1 Database Laboratory, University of A Coruña, A Coruña, Spain

2 Vrije Universiteit Amsterdam, The Netherlands

3 National University of Saint Agustin, Arequipa, Perú

4 Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS), Universidade de Santiago de Compostela, Spain

5 Universidade Nova de Lisboa, Portugal

6 Chalmers University of Technology, Sweden

*Address all correspondence to: n.condori.fernandez@udc.es
and n.condori-fernandez@vu.nl

References

[1] Fogg, B. Persuasive Technology. Using Computers to Change What We Think and Do. CA, USA: Morgan Kaufmann Publishers Inc. San Francisco; 2003.

[2] Froehlich, J. Promoting energy efficient behaviors in the home through feedback: The role of human computer interaction. In: HCIC 2009 Winter Workshop

[3] Prochaska, JO, DiClemente, CC. The transtheoretical approach. Administration and Policy in Mental Health and Mental Health Services Research. 2005; 147–171. DOI: 10.1093/med:psych/9780195165791.003.0007

[4] Locke, E, Latham, G. A theory of goal setting & task performance. The Academy of Management Review. 1991; 16: 480–483

[5] Fogg, B. A behavior model for persuasive design. Proceedings of the 4th International Conference on Persuasive Technology - Persuasive '09. 2009; 1

[6] Harjumaa, M, Oinas-Kukkonen, H. An analysis of the persuasiveness of smoking cessation websites. In: The Second International Symposium on Medical Information and Communication Technology; 2007

[7] Condori-Fernández, N, Bolos, AC, Lago, P. Discovering requirements of behaviour change software systems from negative user experience. In: Proceedings of the 40th International Conference on Software Engineering: Companion Proceedings (ICSE 2018); May 27–June 03 2018; Gothenburg, Sweden; 2018. p. 222–223

[8] Ameller, D, Franch, X, Gómez, C, Martínez-Fernández, S, Araujo, J, Biffl, S, Cabot, J, Cortellessa, V, Méndez, D, Moreira, A, Muccini, H, Vallecillo, A, Wimmer, M, Amaral, V, Bühm, W, Bruneliere, H, Burgueño, L, Goulao, M, Teufl, S, Berardinelli, L. Dealing with non-functional requirements in model-driven development: A survey. IEEE Transactions on Software Engineering. 2019; 1–1. DOI: 10.1109/TSE.2019.2904476

[9] Condori-Fernandez, N, Lago, P. Characterizing the contribution of quality requirements to software sustainability. Journal of systems and software. 2018; 137: 289–305. DOI: 10.1016/j.jss.2017.12.005

[10] Calero, C, Moraga, MA, Bertoa, MF. Towards a software product sustainability model. CoRR. 2013; abs/1309.1640

[11] Venters, C, Jay, C, Lau, L, Griffiths, MK, Holmes, V, Ward, R, Austin, J, Dibsdale, CE, Xu, J. Software sustainability: The modern tower of babel. In: RE4SuSy: Third International Workshop on Requirements Engineering for Sustainable Systems. 2014

[12] Lago, P, Kocak, SA, Crnkovic, I, Penzenstadler, B. Framing sustainability as a property of software quality. Communications of the ACM; 2015. 58 (10): 70–78

[13] Brundtland, G, Khalid, M, Agnelli, S, Al-Athel, S, Chidzero, B, Fadika, L, Hauff, V, Lang, I, Shijun, M, Morino de Botero, M, Singh, M, Okita, S. Our common future (brundtland report). Technical report, World Commission on Environment and Development; 1987.

[14] Hölzl, M, Rauschmayer, A, Wirsing, M. Engineering of software-intensive systems: State of the art and research challenges. In Wirsing, M., Banâtre, J.P., Hölzl, M., Rauschmayer, A., eds.: Software-Intensive Systems and New Computing Paradigms. Springer-Verlag, Berlin, Heidelberg. 2008; 1–44

[15] Becker, C, et al. Requirements: The key to sustainability. IEEE Software. 2016; 33(1): 56–65

[16] Assefa G., Frostell B., Social sustainability and social acceptance in technology assessment: A case study of energy technologies, Technology in Society, Volume 29, Issue 1, 2007, Pages 63–78.

[17] Condori-Fernandez N., "HAPPYNESS: an emotion-aware QoS assurance framework for enhancing user experience," 2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C), Buenos Aires, 2017, pp. 235–237, doi: 10.1109/ICSE-C.2017.137.

[18] Oinas-Kukkonen, H, Harjumaa, M. Persuasive systems design: Key issues, process model, and system features. Communications of the Association for Information Systems. 2009; 24: 485–500

[19] Condori-Fernandez N, Lago P, Luaces MR, Places ÁS. An Action Research for Improving the Sustainability Assessment Framework Instruments. *Sustainability*. 2020; 12(4): 1682.

[20] Condori-Fernández N., Lago P., Towards a Software Sustainability-Quality Model: Insights from a Multi-Case Study. 1–11 2019 RCIS https://doi.org/10.1109/RCIS.2019.8877084

[21] Condori-Fernández, N, Araújo, J, Catalá, A, Lago, P. Towards a non-functional requirements discovery approach for persuasive systems. 2020 SAC; 2020. p. 1418–1420. DOI: 10.1145/3341105.3374059

[22] Sonneleitner, A, Pawlowski, M, Kässer, T, Peissner, M. Experimentally Manipulating Positive User Experience Based on the Fulfilment of User Needs. 14th International Conference on Human-Computer Interaction (INTERACT); Sep 2013; Cape Town, South Africa: Springer, LNCS-8120 (Part IV); 2013. p.555–562

[23] Bickmore, TW, Silliman, RA, Nelson, K., Cheng, DM, Winter, M, Henault, L., Paasche-Orlow, MK. A randomized controlled trial of an automated exercise coach for older adults. J Am Geriatr Soc. 2013 Oct; 61 (10):1676–83. DOI: 10.1111/jgs.12449.

[24] Bevan N., Barnum C., Cockton G., Nielsen J., Spool J., and Wixon D., 2003. The "magic number 5": is it enough for web testing? In CHI '03 Extended Abstracts on Human Factors in Computing Systems (CHI EA '03). ACM, New York, USA, 698–699