Supporting Interactive System Development following MR-MPS-SW with HCI Approaches

Taisa Guidini Gonçalves, Káthia Marçal de Oliveira and Christophe Kolski
LAMIH CNRS UMR 8201
Université Polytechnique Hauts-de-France
Valenciennes, France
taisaguidini@gmail.com, kathia.oliveira@univ-valenciennes.fr, christophe.kolski@univ-valenciennes.fr

***Abstract*— The MPS for Software reference model (MR-MPS-SW) is one of the models currently used in Brazilian industry. This software process capability maturity (SPCM) model describes Software Engineering (SE) practices that are applied in the software development by the use of approaches (such as methods, techniques and so on). The Human-Computer Interaction (HCI) community has defined specific approaches (methods, techniques, patterns, and standards) for the conception, design, implementation, and evaluation of interactive systems development. Nevertheless, the literature points out that those approaches are not sufficiently used in industry. Taking advantage of the large use of SPCM models, we have identified HCI approaches that could support SPCM practices related to five processes (Requirements Development, Design and Construction of the Product, Product Integration, Verification and Validation) advocated by the MR-MPS-SW. Analyzing the HCI literature and MR-MPS-SW, we identified a set of HCI approaches for interactive systems development. We interviewed twenty HCI experts to validate and improve this initial set. Thus, we identified 14 HCI categories with examples of methods, techniques, patterns, and standards adequate for performing SE practices of the MR-MPS-SW when developing interactive systems.**

Keywords*— *Human-Computer Interaction, Interactive system, HCI approaches, Software Process Capability Maturity model, MR-MPS-SW.

I. Introduction

The implementation of Software process capability maturity (SPCM) models in Brazilian industry is well established after little more than a decade [1]. MPS for Software reference model (MR-MPS-SW) is the SPCM Brazilian model [2], and it is a collection of software engineering best practices, organized in processes, which help companies to improve their software process. In Brazil, more than 600 official appraisals are reported on the Brazilian model created in 2005 [1]. Another model widely used in Brazil is the Capability Maturity Model Integration for Development (CMMI-DEV) [3]. Methods, procedures, standards, tools, techniques, and so on are applied to perform what is proposed in SPCM models.

Undoubtedly, Human-Computer Interaction (HCI) engineering is inherently related to software engineering whilst applying to system development. According to Helms et al. [4] usability engineering and software engineering (SE) share common goals: the understanding of customer and user needs; the transformation of needs into system requirements; the design that satisfy the requirements; and the test to assure the final product.

Moreover, several works have discussed the integration of HCI and SE domains for developing/improving usable and useful systems [5] [6] [7] [8]. However, some authors show that HCI/usability approaches are not used or little used in industry [9] [10]. Taking into account that MR-MPS-SW is widely used in Brazilian industry, we consider that indicating which HCI approaches support the application of MR-MPS-SW practices (expected results) can encourage greater application of HCI approaches (any method, technique, standard or pattern for HCI) in the industry. In this context, we defined the following question: What are the HCI approaches that could integrate MR-MPS-SW engineering processes to support the interactive system development?

To address the problem raised, we performed the following activities: (i) analysis of MR-MPS-SW practices for engineering processes (requirements development, design and construction of the product, product integration, verification, and validation) and HCI literature; (ii) validation and improvement of the HCI approaches with experts. After that, we describe in a methodological guide how these approaches can be used with MR-MPS-SW. In previous work [11], we performed the same analysis considering CMMI-DEV. The contribution of this paper is the investigation of HCI approaches concerned with MR-MPS-SW and also the methodological guide generated as a result of this investigation. This methodological guide follows the idea of implementation guidelines provided to the users of MR-MPS-SW [12].

The remainder of this paper is organized as follows. Section 2 presents the main concepts of SPCM models. Section 3 presents the steps and the results of this study. Section 4 presents a part of the methodological guide. Section 5 presents the related work. Finally, in section 5 we present our conclusion and future works.

II. SPCM models

In the last two decades, SPCM models have been developed. Fifty of the 52 models identified by Wangenheim et al. [13] are defined based on CMM [14]/CMMI [3] models. These models are composed of software engineering best practices and they aim to support organizations in defining an evolutionary improvement path from immature to disciplined processes [3].

The MPS.BR is the Brazilian Software Process Improvement Program [2], coordinated by SOFTEX. The MPS.BR defines models for the improvement and evaluation of software and services processes. The MR-MPS-SW model discusses the concepts of maturity levels and process capability focusing in evaluation and improvement of the software/service quality and productivity [2]. This model was developed on the basis of ISO/IEC 12207 [15] and CMMI-DEV [3]. The general guide describes the model according to the following aspects (see Fig. 1): MPS **maturity levels** that are a combination of processes and their capability; **processes**, as well as its purposes and **expected results**; and **process attributes (PA)** that define the level of process capability for each maturity level.

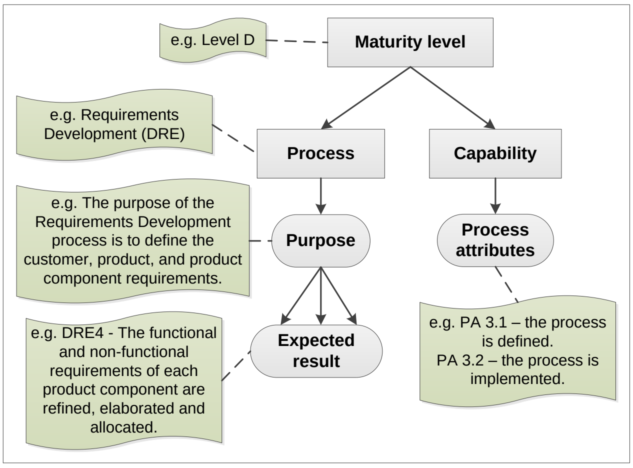

Fig. 1. MR-MPS-SW structure (adapted from [16])

It provides seven sequential and cumulative maturity levels, with a maturity scale starting at **level G** and progressing to **level A**, as well as nineteen processes that are shared between the seven maturity levels. Each maturity level represents a combination of processes and its capabilities. Table I shows each maturity level.

TABLE I. MR-MPS-SW MATURITY LEVELS

MR-MPS-SW maturity levels
A - Optimizing
B - Quantitatively Managed
C - Defined
D - Widely Defined
E - Partially Defined
F - Managed
G - Partially Managed

The **process** definition of the MR-MPS-SW model follows the requirements of a process reference model (purpose/goal and expected results) described in ISO/IEC 15504-2 [17]. The objective expected with the execution of the process represents the **purpose/goal**. The **expected results** represent the objectives that must be achieved with the actual execution of the process. The process capability is represented by a set of attributes (see Fig. 1) that measures it. To have a certain maturity level the organization must meet all the process attributes required for all processes related to the maturity level. Fig. 1 shows the elements of the MR-MPS-SW model.

Since our focus is to support interactive systems development, we decided to concentrate our study on the processes related to the engineering (Requirements Development (RD), Design and Construction of the Product (DCP), Product Integration (PI), Verification (VER), and Validation (VAL)). In this study we used the translation of the original text (in Portuguese) of the MR-MPS-SW model [2]; and to guarantee its equivalence with the CMMI-DEV international model [3] we used the document [16] published by SOFTEX that presents the equivalences between the models.

III. INDENTIFICATION OF HCI APPROACHES

We have investigated how we could support MR-MPS-SW users with the identification and integration of HCI approaches in the engineering processes (expected results) of MR-MPS-SW model. These approaches can be used to define potential work products for each expected result. Our study followed two main phases.

In the first phase, we analyzed the MR-MPS-SW practices (expected results) of five processes[1] (RD, DCP, PI, VER, and VAL). We also analyzed the HCI literature and we proposed a set of HCI approaches organized into categories for the MR-MPS-SW practices. Next, in the second phase we interviewed HCI experts to evaluate and improve our proposition; and we analyzed and synthesized the results and proposed a final set of HCI approaches organized into categories.

A. Phase 1 - Analysis of MR-MPS-SW practices and literature

This phase consisted of reading in the MR-MPS-SW guides [2] [12] the description of each **purpose** and **expected results** (practices) of five MR-MPS-SW processes looking for any citation regarding to HCI issues. The aim was to identify where HCI approaches should be used to implement the expected results. In addition, we studied the HCI literature to identify the main HCI approaches related to the HCI issues found in the analysis of MR-MPS-SW processes. As an outcome of this phase we have a proposition of HCI approaches (methods, techniques, standards, and patterns) organized into HCI categories to support the expected results of the engineering processes of MR-MPS-SW.

We analyzed any **explicit citations** (citations which make a direct reference to any term, technique or method related to HCI engineering[2]) such as: (i) HCI keywords (e.g. end-user, prototype); (ii) examples of HCI techniques or methods (e.g. end-user task analysis); and (iii) examples of work products (e.g. interface design specifications, user manual). Next, we looked for **implicit citations** (citations which do not make direct reference to HCI engineering but the context indicate that it can be interpreted and applied by HCI). These (explicit and implicit) citations were highlighted in the text documentation and reviewed together by the authors of this paper.

[1] In Portuguese these processes are DRE, PCP, ITP, VER and VAL, respectively. In this paper we used the two forms (English and Portuguese - between parentheses) to explicit their equivalence.

[2] Each term, technique or method identified was interpreted according to HCI literature. We do not defined specific terms, but we analyzed all terms in MR-MPS-SW documentation.

Table II shows an example of explicit citation identified for an expected result (DRE1) of RD (DRE) process. This result shows the importance of the end-user in the requirements elicitation ("end-user tasks analysis", "prototypes and models") related to HCI for this purpose. It also shows an example of implicit citation (DRE3) of DRE process. This result points out the importance of non-functional requirements ("quality requirements", such as usability requirements).

TABLE II. ANALYSIS OF EXPECTED RESULTS OF THE RD PROCESS [12]

Expected result	HCI issues
DRE1 - The needs, expectations and constraints of the customer, both the product and its interfaces, are identified.	- *Description of the expected result:* "The achievement of this expected result involves the use of appropriate methods to identify needs, expectations, constraints and interfaces of the customer." - *Suggested techniques:* "interviews, questionnaires, construction of operational scenarios, and end-user task analysis" and "prototypes and models"
DRE3 - A set of functional and non-functional requirements of the product and product components that describe the solution to the problem to be solved is defined and maintained from the customer's requirements.	- *Description of the expected result:* "Non-functional requirements are sometimes known as restrictions or quality requirements."

We analyzed 38 expected results regarding to five processes (RD (DRE), DCP (PCP), PI (ITP), VER, and VAL). We found that 23 expected results (7 from DRE, 5 from PCP, 1 from ITP, 5 from VER, and 5 from VAL) address HCI issues (explicit or implicit). We did not find any explicit or implicit citation for:

- One expected result (DRE5) from RD (DRE) process – this result is related to the internal interface between functional components;
- Three expected results (PCP4, PCP5 and PCP8) from DCP (PCP) process – the result PCP4 refers to the interface between two functional components; the result PCP5 is related to the choice of criteria and the analysis of designs to determine if the product components should be developed, reused, or purchased; and the result PCP8 refers to the criteria defined to maintain the product documentation;
- Almost all expected results from PI (ITP) process (we found citation for one (ITP1)– this result is related to the strategy definition to perform the product integration;
- One expected result (VER5) from VER process – this result refers to the identification and recording of defects found in the verification procedure of the product components and product;
- Two expected results (VAL5 and VAL7) from VAL process – the result VAL5 refers to the identification and recording of problems found in the validation procedure of the product components and product; and the result VAL7 is related to the evidences that show that the developed product is ready for use.

All citations were organized by categories (based on the literature) with the objective to identify the main HCI approaches. As a result of the analysis of all citations regarding to RD (DER) expected results, we identified five HCI categories: (i) **methods of end-user tasks analysis**, for all citations that mention methods or the analysis about the interaction with users; (ii) **detailed operational concept and scenarios**; (iii) **standards and guidelines for design interfaces**, for all citations that talk about quality attributes and criteria; (iv) **techniques for requirements validation**, for the explicit citation that talk about techniques and implicit citation (requirements validation); (v) **prototyping**, for any mention of prototypes in any expected result.

For DCP (PCP) and PI (ITP) expected results, in addition to the HCI categories already identified, two new categories were defined: (i) **architecture patterns**, to represent architectural decisions to develop the HCI design; and (ii) **design patterns**, to implement design patterns in the HCI product design.

Finally, analyzing citations for VER and VAL, one new category was identified: **evaluation methods**, for all kinds of evaluation techniques and methods (such as peer review, inspection, and tests) used for verification, validation, and testing. In this analysis the prototype is related to the final validation, thus we split the category **prototyping** into two: (i) **prototype for HCI requirements**, which could include prototypes in papers, mockups etc.; and (ii) **functional prototype to validate HCI**, to represent the executable prototypes.

After this first analysis, we collected from literature examples of HCI approaches (methods, techniques, patterns, and standards) for all categories. The **evaluation methods** category was refined in two new categories, following the software engineering classical classifications: (i) **evaluation methods for HCI review**, to include techniques such as inspections, reviews, and so on; and, (ii) **evaluation methods for HCI verification tests**, to include all kinds of test. The main results of this phase were: the definition of ten HCI categories (see Table III); the identification of examples of HCI approaches for each category; and 33 propositions that integrate HCI categories and approaches in 23 expected results.

TABLE III. HCI CATEGORIES

HCI categories	
Task Analysis Methods for HCI	Architecture Patterns for HCI
Prototype for HCI requirements	Design patterns for HCI
Operational Concepts and Scenarios Specification for HCI	Functional Prototype to validate HCI
Standards and Guidelines for design and documentation of HCI	Evaluation methods for HCI verification tests
Techniques to validate HCI requirements	Evaluation methods for HCI review

B. *Phase 2 - Validation and Improvement with experts and Analysis and Synthesis of the HCI approaahes*

In this phase, firstly we validated and improved our proposition. We planned and performed a validation with twenty experts in HCI domain (they have Ph.D. degree in Computer Science domain). The objective was to validate if the proposed set of HCI categories and approaches support the expected results. For each expected result the experts should answer if they **agree**, **partially agree** or **do not agree** that the associated HCI category and approaches support the expected result. When they disagree or partially agree, they should justify. The interviewer takes notes about the explanations performed by experts during the interview.

Secondly we analyzed the quantitative and qualitative results for each proposition (33) to define the improvement of our proposal. All experts' justifications were organized in a single form for each expected result and the associated HCI category/categories. We analyzed each justification and with this analysis we decided the improvements (inclusion or exclusion of one category), integrating as much as possible all propositions respecting the opinion and experience of the experts. Thus, we analyzed the results of each process in an integrated way (quantitative and qualitative analysis). As a general result (for the five processes) we had 59% agree, 33% partially agree and 8% do not agree.

We proposed five HCI categories to Requirement Development process (DRE). This is the process with the highest level of partially agreements (46%). The results of the interview are presented in Fig. 2 where we can note that:

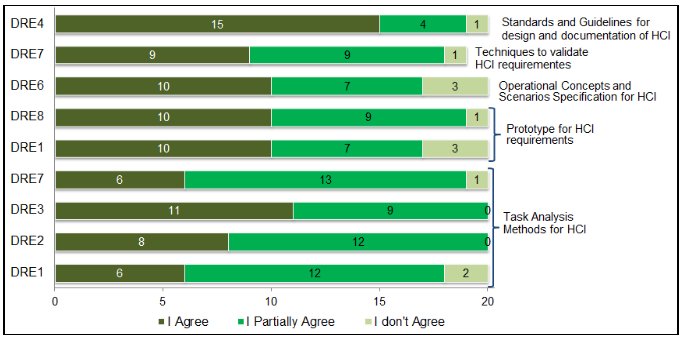

Fig. 2. Results for Requirement Development process (DRE)

- the category **Task analysis Methods for HCI** was considered adequate for DRE2 and DRE3 without disagreement. Since we had several partial agreements some modifications were proposed. For DRE1 and DRE7, the 3 disagreements brought no change;
- the category **Prototype for HCI requirements** had 3 (15%) disagreements for DRE1. We analyzed all experts' justifications and we concluded that this category should be excluded; but we kept this category for DRE8;
- the category **Operational Concepts and Scenarios Specifications for HCI** also had 10 agreements against 3 (15%) disagreements;
- the category **Techniques to validate HCI requirements** can be considered adequate for DRE7. However, 9 partial agreements indicate that improvements should be made; and,
- the category **Standards and guidelines for design and documentation of HCI** presented the greatest amount of agreements for DRE 4 (15 experts - 75%).

Analyzing all experts' justifications (partial agreement and disagreement) and we concluded that:

- the category **Standards and Guidelines for HCI design category** can be included for DRE3;
- the category **Operational Concepts and Scenarios Specification for HCI** should be excluded from DRE6. The proposed approaches of this category and others were placed in a new category **Techniques to identify user and organizational requirements** that was therefore associated with DRE6;
- the category **Techniques to validate requirements** can be included for DRE8.

The Design and Construction of the product process (PCP) received the proposition of five HCI categories. In general, 57% of the answers were agreed. Fig. 3 presents the results and we can note that:

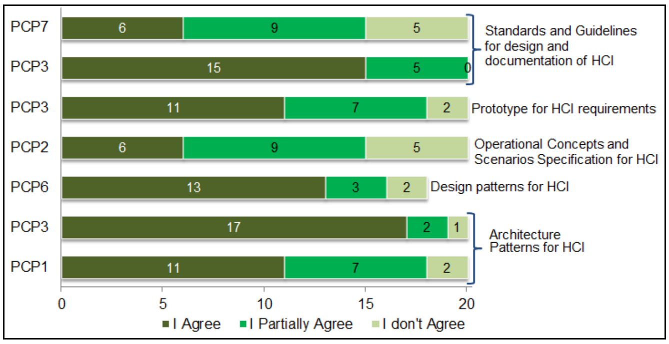

Fig. 3. Results for Design and Construction of the product process (PCP)

- the category **Architecture Patterns for HCI** should be kept for PCP3. However, for PCP1 partial agreements suggested the inclusion of the category **Techniques for interaction modeling**;
- the category **Design patterns for HCI** should be kept for PCP6 and the category **Architecture patterns for HCI** can be included to support PCP6 following the suggestion of two experts;
- the category **Prototype for HCI requirements** should also be kept for PCP3;
- the category **Standards and Guidelines for Design and Documentation of HCI** should also be kept for PCP3. Contrariwise, it can be excluded for PCP7. The experts' justifications allow us to create a new category specifically for documentation (**Techniques for HCI documentation**). The previous category was renamed **Standards and Guidelines for Design**;
- the category **Operational concepts and scenarios specifications for HCI** was excluded for PCP2 according to experts' justifications.

The Product Integration process (ITP) had only two propositions to be evaluated. This process presented the highest level of disagreement (28%). Fig. 4 shows the results. After analysis we concluded that:

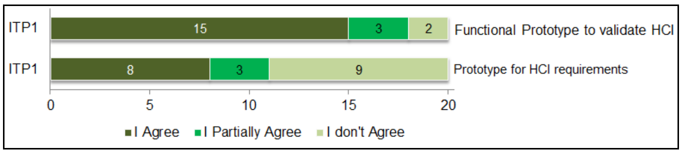

Fig. 4. Results for Product Integration process (ITP)

- the category **Functional Prototype to validate HCI** should be kept. The three partial agreements only suggested some recommendations;
- the category **Prototype for HCI requirements** had 9 disagreements and 3 partial agreements and the analyzed justifications determine the exclusion of this category.

We proposed four categories to Verification process (VER). This process presented the highest level of agreement (68%). The Fig. 5 presents our results and we can note that:

- the category **Standards and guidelines for design and documentation of HCI** (VER3) and the category **Evaluation methods for HCI review** (VER2, VER4 and VER6) had no disagreement, which means that they should be kept;
- the category **Functional prototype to validate HCI** (VER1) had 3 disagreements. However, the experts who partially agreed considered that the prototype should be the first version of the system. Thus, we decided to rename the category **Iterative and Evolutionary prototype (system versions)**.

Analyzing all experts' justifications for the category **Evaluation methods for HCI verification tests**, we observe that two experts who disagreed and one who partially agreed suggested the inclusion of classical verification tests from software engineering.

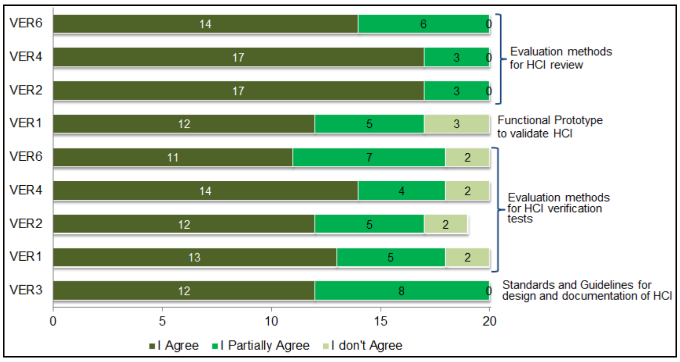

Fig. 5. Results for Verification process (VER)

The category **Evaluation methods for HCI verification tests** was to split into two categories named: **Evaluation methods for HCI verification** and **Evaluation methods for HCI validation**. The category **Evaluation methods for HCI verification** replaced the previous one for VER1, VER2, VER4 and VER6.

The Validation process (VAL) received the proposition of three HCI categories. In general, 64% of the answers were agreed. Fig. 6 presents the results where four out of five propositions were accepted and the category **Functional prototype to validate HCI** had 2 disagreements.

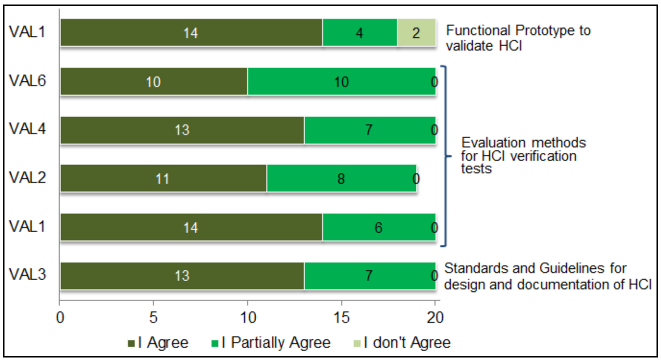

Fig. 6. Results for Validation process (VAL)

Analyzing the partial agreements and disagreements to VAL process we decided:

- to rename the category **Functional Prototype to validate HCI** as **Iterative and Evolutionary prototype (system versions)** since the experts who partially agreed had the same justification as they gave for VER process, saying that it should be the initial version of the system; and
- to replace the category **Evaluation methods for HCI verification tests** by **Evaluation methods for HCI validation**, with appropriate techniques for validation suggested by the experts.

As a conclusion, at the beginning we had 10 HCI categories (Table III) composed of 33 propositions related to 23 expected results. After the analysis of all experts' justifications, we obtained 14 HCI categories composed of 39 propositions related to 22 expected results, as presented in Table IV. Table V presents all HCI categories and approaches, and also the purpose of each HCI category.

C. Threats to validity

Threats to validity [18] for the validation and improvement of our first proposition were analyzed as follows.

The threat to **construct validity** is related to the elaboration of the proposal. To minimize this threat, we elaborated the proposal using the original text extracted from the official documentation of MR-MPS-SW, and the document [15] that presents the equivalences between MR-MPS-SW and CMMI-DEV. In addition, the proposition of HCI categories and approaches were collected from literature.

An **internal validity** threat is associated with the experts involved in the evaluation. They were selected by convenience. We assumed that the experts knew all the proposed approaches, since we selected only people who have experience in HCI domain and have a Ph.D. degree in Computer Science domain. However, we could not control that the experts did not give their real opinion. Therefore, we accepted this risk.

The threat to the **conclusion validity** is related to the HCI categories and approaches associated to each expected result.

TABLE IV. HCI CATEGORIES X PROCESSES AND EXPECTED RESULTS OF MR-MPS-SW

HCI categories \ Processes and Expected results	RD [a]							DCP [a]				PI [a]	VER					VAL				
	DRE1	DRE2	DRE3	DRE4	DRE6	DRE7	DRE8	PCP1	PCP3	PCP6	PCP7	ITP1	VER1	VER2	VER3	VER4	VER6	VAL1	VAL2	VAL3	VAL4	VAL6
Techniques to identify user needs	■																					
Techniques to identify user and organizational requirements	■	■			■																	
Task Modeling	■	■	■			■																
Prototype for HCI requirements	■						■															
Standards and Guidelines for HCI		■	■	■					■						■					■		
Techniques to validate HCI requirements						■	■															
Architecture Patterns for HCI								■	■	■												
Design patterns for HCI										■												
Techniques for interaction modeling								■														
Techniques for HCI documentation											■											
Iterative and Evolutionary Prototypes (system versions)									■			■	■					■				
Evaluation methods for HCI verification													■	■		■	■					
Evaluation methods for HCI review														■		■	■					
Evaluation methods for HCI validation																		■	■		■	■
Total of propositions - 39	4	3	2	1	1	2	2	2	3	2	1	1	2	2	1	2	2	2	1	1	1	1
	15							**8**				**1**	**9**					**6**				

[a] RD (DRE), DCP (PCP), PI (ITP)

TABLE V. HCI CATEGORIES AND APPROACHES TO SUPPORT MR-MPS-SW

HCI Categories	Purpose	HCI Approaches
Techniques to identify user needs	To use techniques with the goal to elicit user needs.	Brainstorming; Surveys/Questionnaires; Interviews; Card Sorting; Focus Groups; Field Studies.
Techniques to identify user and organizational requirements	To use techniques with the goal to elicit user needs and to transform these needs in user requirements.	Persona; Scenario; User stories, User profile; Task analysis; Context-of-use analysis; Storyboards; Requirements specification templates (e.g. VOLERE, IEEE, RESCUE).
Task Modeling	To use task modeling methods with the goal to elicit user needs, to transform these needs in user requirements, to establish user interface requirements, and to analyze the user and user interface requirements.	CTT; K-MAD; HTA; SADT; GTA; Task Model Standard – W3C; HAMSTERS notation.
Standards and Guidelines for HCI design	To use standards and guidelines to establish and maintain a prioritization of user quality attribute requirements, to design the user interface, and to establish verification and validation criteria.	Ergonomic criteria; ISO/IEC 9126-1; ISO 9241-11; ISO/IEC 25000; Accessibility standards and guidelines (WAI - W3C); Nielsen’s Heuristics; Golden Rules of Interface Design.
Prototype for HCI requirements	To use prototypes to transform user needs into user requirements and to validate user requirements.	Paper Prototyping/Sketches; Storyboards; Wireframes; Mockups; Wizard of Oz; Video prototyping.
Techniques to validate HCI requirements	To use techniques to analyze user requirements and to validate user requirements.	ProtoTask; Task Model Simulator; Focus Group to validate requirements; Thinking aloud.
Architecture Patterns for HCI	To define architectural decisions and to select architecture patterns to design and implement the user interface.	Arch Model; Language Model; Seeheim Model; PAC Model; PAC-AMODEUS Model; MVC Model; CAMELEON-RT; Frameworks.
Design patterns for HCI	To use design patterns to implement the design the user interface.	A Pattern Language for Human-Computer Interface Design; A Pattern Approach to Interaction Design; Pattern Languages in Interaction Design: Structure and Organization; Designing interfaces: Patterns for Effective Interaction Design.
Techniques for interaction modeling	To build interactive models to help in the choice of the design solution.	MoLIC; UAN; TAG.
Techniques for HCI documentation	To use techniques to produce the end user documentation.	Style guide; Architecture for help; Training Program.
Iterative and Evolutionary Prototypes (system versions)	To use iterative and evolutionary prototypes to help in: product and/or product component design, integration, validation and verification.	User interface toolkits; User interface builders; User interface development environments.
Evaluation methods for HCI verification	To select and use evaluation methods for products and product components verification.	Unit test; Integration test; System test; Acceptance test; Installation test.
Evaluation methods for HCI review	To select and use review methods for products and product components review.	Semiotic inspection; Heuristic evaluation; Cognitive walkthrough; Groupware Walkthrough; Guidelines review; Consistency inspection; Metaphors of human thinking (MOT); Formal usability inspection.
Evaluation methods for HCI validation	To select and use evaluation methods for products and product components validation.	Usability testing; Communicability test; Standardized usability questionnaires; Post-experience interviews; User experience evaluation.

To reduce this risk, we decided to perform interviews individually (experts were asked to justify their opinion and include any other proposals) and not using survey. The final set of propositions resulted in the majority from the agreement or partial agreements and the experts' justifications.

Finally, one threat to the **external validity** is associated with a biased result related to the domain of expertise. We chose experts from different expertise and also with large experience recognized by the HCI community (e.g., program chair or member of program committee of HCI conferences). Also, they are well known for working on different technologies (e.g., web applications, critical systems, tabletop applications). We could not ensure cover all kind of technologies and application domains. Thus, we asked them to indicate approaches that could be used in general for any kind of interactive system. As consequence, we accepted the risk of potential bias in their evaluation. Therefore, it is not possible to generalize this result.

IV. METHODOLOGICAL GUIDE

The first version of the methodological guide describes how to use our integration proposal of HCI approaches (methods, techniques, standards and patterns) into the engineering processes of MR-MPS-SW model. The HCI approaches were integrated into the expected results as HCI categories. We highlight that the communication between the members of Software Engineering and Human-Computer Interaction development teams is fundamental so that the integration proposal produces a useful and usable iterative system.

We will briefly describe the purpose of one process (**Requirements Development** (RD/DRE)) as well as its expected results, and how HCI approaches (see Table V) should be to support their implementation for interactive system development. When possible, experts' suggestions collected during interviews regarding to each practice are presented (in boxes). The **Requirements Development** process is intended to define the requirements of the customer, product, and product components. The activities related to this process are [3] [12]:

- elicit customer needs, expectations, constraints and interfaces and translate them in customer requirements;
- refine and describe customer requirements in technical terms, giving rise to the functional and non-functional requirements of the product and the product components;
- elaborate a definition of requirements;
- elaborate a detailed definition of the scenarios and operational concepts that allow the accomplishment of technical design and the construction of the software solution;
- analyze, validate and manage the requirements throughout the development or maintenance life cycle of a product.

MR-MPS-SW emphasizes that the expected results of this process are related to the results of three other processes: PCP, VER and VAL. The requirements produced by DRE are the work product required for the beginning of the PCP process [12]. VAL presents a direct intersection with DRE regarding to the requirements validation.

A. DRE1 - The customer needs, expectations and restrictions from both, the product and its interfaces, are identified

In this expected result the needs, expectations, constraints and user interfaces are identified for all phases of the product life cycle.

The category **Techniques to identify user needs** was integrated in this result with the objective to help in the elicitation of the user interfaces needs. The literature presents different techniques that can be used for this purpose, for example, brainstorming, interviews, surveys/questionnaires, card sorting, focus groups, and field studies/observation [19].

Another category that was integrated in this result is: **Techniques to identify user and organizational requirements**. Techniques as scenario, user stories, storyboards, task analysis, persona, context-of-use analysis, user profile [20] [19], and requirements specification templates can be used to document and to refine the needs, expectations, constraints and user interfaces. One example of application of the technique user profile for open source software development process can be found in [21]. This technique allows the definition of the different user profiles that a system can have.

The third category that was integrated in this result is **Task Modeling**. The goal of this category is to produce task models that represent the needs of the different users that a system can have. Examples of methods that can be found in this category are: CTT - Concur Task Tree [22]; K-MAD - Kernel of Model for Activity Description or MAD - Model for Activity Description [23]; HTA - Hierarchical Task Analysis [23]; Task Model Standard - W3C [24]. A Task Modeling application for reengineering processes for mobile learning user interfaces can be found in [25], a task modeling approach for safety-critical systems is presented in [26], and an example regarding the use of task modeling to specify the requirements of an interactive system can be found in [27].

B. DRE2 - A defined set of customer requirements is specified and prioritized from the needs, expectations, and constraints identified

In this expected result the needs, expectations, constraints and user interfaces identified in DRE1 are translated into user requirements. The prioritization of the requirements assists in determining the project scope, iteration or increment [12]. In addition, it ensures that critical requirements (both functional and non-functional) are handled quickly [3]. In the case of interactive systems, prioritization of critical (especially the non-functional) requirements is essential to ensure a usable and useful system.

The category **Techniques to identify user and organizational requirement** was integrated in this result with the objective to prioritize the user requirements. In this level, the techniques of the category **Techniques to identify user and organizational requirement** previously used in DRE1 are revisited to evolve the records in order to define and prioritize user interface requirements. The literature presents many techniques that can be used to prioritize user interface requirements [20] [19]. Persona is an example of HCI technique that gathers information about users to understand their characteristics [28]. An application of this technique in the software development requirements phase can be found in [28].

The second category integrated in this result was **Task Modeling**. The objective is to evolve (initially produced in DRE1) or to product task models that represent the user requirements. Task model is a model-based approach to user interface design where the results are models that describe the activities that should be performed in order to reach users' goals [29]. Task models can be useful in different phases of the development of interactive applications: requirements analysis, design of the user interface, usability evaluation, documentation and others [29]. A joint application of UML diagrams and Task Modeling can be found in [29]. A tool for specifying task models (Responsive CTT) can be found in [30] and this tool can be accessed through touch-based mobile devices, such as smartphones and tablets.

The category **Prototype for HCI requirements** integrated in this result has as objective the construction of a prototype that can be used to discuss the prioritization of the requirements with the end user. Examples of prototype techniques that can be used for this purpose are: Paper Prototyping/Sketches; Storyboards; Wireframes; Mockups; Wizard of Oz; Video prototyping [20]. The prototyping include different steps: (i) study of user requirements; prototype construction; and their validation by users [12].

Experts suggestion:
In this level, the prototypes are used to support the discussion about the design with the team. They cannot use the prototype as a final system but they can build different versions of the prototype.
One expert (with 30 years of experience) suggests the use of techniques, such as FRAM - Functional Resonance Accident Model [31] and STAMP - System-Theoretic Accident Model and Processes [32] to identify organizational context issues for critical interactive systems.

The category **Standards and Guidelines for HCI design** also was integrated in this result with the objective to establish and maintain a prioritization of user quality attribute requirements. Quality attributes such as usability could be considered since the identification of user needs, as well as during the elicitation and prioritization of the user requirements. Criteria, guidelines heuristics and rules are used in this moment according to the selected quality attribute requirements. We can cite as example, ergonomic criteria [33] [34]; ISO Standards - ISO 9241-11 [35] and ISO/IEC 25000 [36]; accessibility standards and guidelines – WAI/W3C [37]; Nielsen's heuristics [38]. In addition, these criteria, guidelines heuristics and rules are used in the evaluation phase.

C. DRE3 - A set of functional and non-functional requirements of the product and product components that describe the solution to the problem being solved is defined and maintained from the customer's requirements

In this expected result the user needs, expectations and constraints identified in DRE1 and DRE2 are translated in a set of functional and non-functional requirements of the product (user interface) and product components.

The **Task Modeling** category was integrated in this result with the objective to produce or evolve task models that represent the user interface requirements. The use of Task Modeling and the construction of task models for context-sensitive user interfaces are discussed in [39].

The category **Standards and Guidelines for HCI design** was also integrated for this result with the aim to capture critical quality attributes. Quality attributes should be considered when defining the non-functional user interface requirements.

In addition, the work products (such as scenario, user stories, storyboards, persona, paper prototyping) produced by the categories **Techniques to identify user and organizational requirement**, **Techniques to identify user and organizational requirement** and **Prototype for HCI requirements** are used in this practice to achieve the goal of this practice.

Experts suggestion:
One expert (with 30 years of experience) suggests the use of Worth-Centered Design [40] to design interactive systems with the finality to deliver worth in the real world and decrease the gap between the user and product. One example of the use of this approach in a mobile and context-aware application can be found in [41].

D. DRE4 - The functional and non-functional requirements of each product component are refined, elaborated and allocated

In this expected result a definition of quality attributes is established. The **Standards and Guidelines for HCI design** category was included in this practice to help the identification and definition of the quality attributes for interactive systems. The quality attributes can be defined based on an analysis of the scenarios previously produced. That is, the work products produced by the category **Techniques to identify user and organizational requirement** in the previous practice are used to support the definition of the quality attributes.

E. DRE6 - Operational concepts and scenarios are developed

In this expected result the operational concepts and scenarios are developed for the product (user interface) and the product components. The category **Techniques to identify user and organizational requirement** was included in this result with the objective to construct scenarios that define the interaction of the user interface, the end user and the environment. In addition, the work products produced by the **Task Modeling** category are used to support the construction of the scenarios.

F. DRE7 - Requirements are analyzed, using defined criteria, to balance stakeholder needs with existing constraints

In this expected result the user and user interface requirements are analyzed to ensure that they are necessary and sufficient in relation to the needs of those interested.

The **Task Modeling** category was included in this result with the aim to perform an analysis of user and user interface requirements, in order to remove conflicts encountered in relation to the user interface. In addition, the work products produced by the category **Techniques to identify user and organizational requirement**, as well as scenarios and detailed definitions of the requirements, can be used in this analysis [3]. The quality attributes defined above and supported by the **Standards and Guidelines for HCI design** category should also be considered in the requirements analysis.

Also, in this result the user and user interface requirements are analyzed in a way to balance stakeholder needs with design constraints. The **Techniques to validate HCI requirements** category was integrated in this result with the goal of analyzing user and user interface requirements in order to balance stakeholder needs with design constraints and minimize the risk of user interface development. The literature proposes models, simulators and techniques that can be used for this type of analysis. For example, Task Model Simulator for CTT [42]; Focus group to validate HCI requirements [43]; and Thinking aloud [44].

In addition, the work products produced by the **Task Modeling** category, that is, task models, can be used to support this analysis. The quality attributes previously defined and supported by the **Standards and Guidelines for HCI design** category should also be considered in this analysis.

G. *DRE8 - The requirements are validated*

In this expected result the user interface requirements are validated using appropriate techniques, thus ensuring that the user interface will perform adequately when installed in the user environment.

The **Prototype for HCI requirements** category integrated in this result aims to provide a first prototype or the evolution of the prototype built in DRE2, which helps in the validation of requirements with the end user. The prototypes are useful for evaluating critical or complex requirements [12]. We remember that the prototypes do not represent the final version of the system, but rather tools to discuss and evaluate requirements and design.

The **Techniques to validate HCI requirements** category was integrated in this result to support the validation of the user and user interface requirements. For example, Focus group and Thinking aloud techniques can be used in conjunction with the prototype for the requirements validation.

In addition, the work products produced by the **Task Modeling** category, that is, task models, can be used in conjunction with their simulators (for example, ProtoTask for K-MAD and Task Model Simulator for CTT) to validate the user interface requirements.

This methodological guide was reviewed by two implementers of MR-MPS-SW.

V. RELATED WORK

We found two works similar to ours in literature. In the first one, Nogueira and Furtado [6] performed a case study where they chose and applied some HCI techniques (from a literature review). They indicate the use of these techniques to support four MR-MPS-SW processes (requirements development, design and construction of the product, verification, and validation). This proposition shows the possibility to suggest HCI techniques to support a generic SPCM model. However, it is only a very specific case study (probably the techniques were chosen for this specific kind of application); and it limits the example of techniques. Despite the similarity with our proposition, this work does not focus on all MR-MPS-SW engineering processes, since it does not consider Product Integration.

In another work, Peres et al. [7] proposed an initial study focused on a reference model for integrating agile methods and user experience (UX) in the software development cycle. This proposition is in line with CMMI-DEV and MR-MPS-SW. The model suggests specific practices, recommendations, and techniques to support some process areas from CMMI-DEV Level 2 (such as, project planning integrated with project monitoring and control). The similarity with our work is the proposition of techniques to support CMMI-DEV process areas in the system development projects (in this case UX projects). However, they suggest techniques for the process areas placed in CMMI level 2 that does not consider the engineering process areas (which are part of CMMI level 3). Moreover, this work is at an initial stage and had no validation, being a simple proposition of the authors.

VI. CONCLUSION

This paper presented a study that aims to identify HCI approaches which can support practices, prescribed by engineering processes of the MR-MPS-SW model. The initial integration proposal was composed of 10 HCI categories and approaches. After interviewing 20 experts, we obtained 14 HCI categories and approaches. New categories were created, others were excluded, and several examples of approaches were included according to experts' justifications and our analysis. In addition, we defined a first version of a methodological guide for DRE process which proposes where and how (purpose) the HCI approaches can be applied

Our on-going work is the definition of an agile software development process for interactive tabletop application with tangible objects using the defined methodological guide. This software process will propose when the HCI approaches can be used and how to integrate the HCI approaches with other approaches known in SE/development domain. Then, we plan to use this software process to develop an interactive tabletop application, and to collect evidence about the effective use of our proposal in practice.

ACKNOWLEDGMENT

We would like to acknowledge the financial support granted by CAPES - Science without Borders Program and the HCI experts who participated in this study.

REFERENCES

[1] M. Kalinowski, K. Weber, G. Santos, N. Franco, V. Duarte, and G. Travassos, "Software Process Improvement Results in Brazil Based on the MPS-SW Model," *Software Quality Professional*, vol. 17, no. 4, pp. 15–28, 2015.

[2] Softex, "MPS.BR - Brazilian Software Process Improvement, MR-MPS-SW - General Guide," 2016.

[3] CMMI Product Team, “CMMI® for Development,” Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, USA, CMU/SEI-2010-TR-033, 2010.
[4] J. W. Helms, J. D. Arthur, D. Hix, and H. R. Hartson, “A field study of the Wheel—a usability engineering process model,” *J. Syst. Softw.*, vol. 79, no. 6, pp. 841–858, 2006.
[5] K. Nebe and D. Zimmermann, “Aspects of Integrating User Centered Design into Software Engineering Processes,” in *Proceedings of 12th HCI International*, vol. Part I, J. A. Jacko, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 194–203.
[6] N. P. Nogueira and E. Furtado, “Integration of IHC techniques in software processes based on a model of process improvement,” in *Proceedings of XXXIX Latin American Computing Conference (CLEI)*, 2013, pp. 1–8.
[7] A. L. Peres, T. S. D. Silva, F. S. Silva, F. F. Soares, C. R. M. D. Carvalho, and S. R. D. L. Meira, “AGILEUX Model: Towards a Reference Model on Integrating UX in Developing Software Using Agile Methodologies,” in *Proceedings of the 2014 Agile Conference*, Washington, DC, USA, 2014, pp. 61–63.
[8] W. Silva, N. M. C. Valentim, and T. Conte, “Integrating the Usability into the Software Development Process - A Systematic Mapping study,” in *Proceedings of the 17th ICEIS*, Portugal, 2015, vol. 3, pp. 105–113.
[9] C. Ardito, P. Buono, D. Caivano, M. F. Costabile, and R. Lanzilotti, “Investigating and promoting UX practice in industry: An experimental study,” *Int. J. Hum.-Comput. Stud.*, vol. 72, no. 6, pp. 542–551, 2014.
[10] A. Salgado, L. A. Amaral, A. P. Freire, and R. P. M. Fortes, “Usability and UX Practices in Small Enterprises: Lessons from a Survey of the Brazilian Context,” in *Proceedings of the 34th ACM International Conference on the Design of Communication*, New York, NY, USA, 2016, pp. 18:1–18:9.
[11] T. G. Gonçalves, K. M. Oliveira, and C. Kolski, “Identifying HCI Approaches to support CMMI-DEV for Interactive System Development,” *Comput. Stand. Interfaces*, vol. 58, pp. 53–86, 2018.
[12] Softex, “MPS.BR - Brazilian Software Process Improvement, Implementation Guide - Part 4: Rationale for the Implementation of Level D of the MR-MPS-SW:2016,” 2016.
[13] C. G. Wangenheim, J. C. R. Hauck, C. F. Salviano, and A. von Wangenheim, “Systematic Literature Review of Software Process Capability/Maturity Models,” in *Proceedings of International Conference SPICE*, 2010.
[14] M. C. Paulk, C. V. Weber, B. Curtis, and M. B. Chrissis, *The Capability Maturity Model: Guidelines for Improving the Software Process*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1995.
[15] ISO/IEC, “Systems and software engineering – Software life cycle processes (ISO/IEC 12207:2008),” Geneva, Switzerland, 2008.
[16] Softex, “MPS.BR - Brazilian Software Process Improvement, Implementation Guide - Part 11: Implementation and Evaluation of MR-MPS-SW: 2016 in conjunction with CMMI-DEV v1.3,” 2016.
[17] ISO/IEC, “Information Technology – Process Assessment – Part 2: Performing an assessment (ISO/IEC 15504-2: 2003),” Geneva, Switzerland, 2003.
[18] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering*. Springer-Berlin Heidelberg, 2012.
[19] C. Courage and K. Baxter, *Understanding your users: a practical guide to user requirements, methods, tools, and techniques*, 1st Edition. San Francisco, CA: Morgan Kaufmann Publishers, 2005.
[20] M. Maguire, “Methods to support human-centred design,” *Int. J. Hum.-Comput. Stud.*, vol. 55, no. 4, pp. 587–634, 2001.
[21] L. Llerena, N. Rodríguez, J. W. Castro, and S. T. Acuña, “Adoption of the User Profiles Technique in the Open Source Software Development Process,” in *Trends and Applications in Software Engineering: Proceedings of CIMPS 2016*, J. Mejia, M. Muñoz, Á. Rocha, T. San Feliu, and A. Peña, Eds. Cham: Springer International Publishing, 2017, pp. 201–210.
[22] F. Paternò, C. Mancini, and S. Meniconi, “ConcurTaskTrees: A Diagrammatic Notation for Specifying Task Models,” in *International Conference on Human-Computer Interaction.*, S. Howard, J. Hammond, and G. Lindgaard, Eds. Boston, MA: Springer US, 1997, pp. 362–369.
[23] Q. Limbourg and J. Vanderdonckt, “Comparing Task Model for User Interface Design,” in *The Handbook of Task Analysis for Human-Computer Interaction*, D. Diaper and N. A. Stanton, Eds. Mahwah, NJ: Lawrence Erlbaum Associates, 2004, pp. 135–154.
[24] World Wide Web Consortium, “MBUI - Task Models,” *W3C Working Group Note*, 2014. [Online]. Available: https://www.w3.org/TR/task-models/.
[25] A. I. Molina, M. A. Redondo, and M. Ortega, “Applying Task Modeling and Pattern-based techniques in Reengineering Processes for Mobile Learning User Interfaces: A case study,” *J. Comput.*, vol. 2, no. 4, pp. 23–30, 2007.
[26] M. Giese, T. Mistrzyk, A. Pfau, G. Szwillus, and M. von Detten, “AMBOSS: A Task Modeling Approach for Safety-Critical Systems,” in *Second Conference on HCSE*, P. Forbrig and F. Paternò, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 98–109.
[27] T. G. Gonçalves, K. M. Oliveira, and C. Kolski, “The use of task modeling in interactive system specification,” *Cogn. Technol. Work*, vol. 19, no. Issue 2–3, pp. 493–515, 2017.
[28] S. T. Acuña, J. W. Castro, and N. Juristo, “A HCI technique for improving requirements elicitation,” *Inf. Softw. Technol.*, vol. 54, no. 12, pp. 1357–1375, 2012.
[29] F. Paternò, “Towards a UML for Interactive Systems,” in *Proceedings of the 8th IFIP International Conference on Engineering for Human-Computer Interaction*, London, UK, UK, 2001, pp. 7–18.
[30] D. Anzalone, M. Manca, F. Paternò, and C. Santoro, “Responsive Task Modelling,” in *Proceedings of the 7th ACM EICS*, New York, NY, USA, 2015, pp. 126–131.
[31] R. Woltjer and E. Hollnagel, “Modeling and evaluation of air traffic management automation using the functional resonance accident model (FRAM),” in *Proceedings of 8th International Symposium of the Australian Aviation Psychology Association.*, Sydney, Australia, 2008.
[32] W. Young and N. G. Leveson, “An Integrated Approach to Safety and Security Based on Systems Theory,” *Commun. ACM*, vol. 57, no. 2, pp. 31–35, 2014.
[33] J. Vanderdonckt, *Guide ergonomique des interfaces homme-machine*. Namur, Belgium: Presses Universitaires de Namur, 1995.
[34] D. L. Scapin and J. M. C. Bastien, “Ergonomic criteria for evaluating the ergonomic quality of interactive systems,” *Behav. Inf. Technol.*, vol. 16, no. 4–5, pp. 220–231, 1997.
[35] ISO, “Ergonomic requirements for office work with visual display terminals (VDTs) – Part 11: Guidance on usability (ISO 9241-11:1998),” Geneva, Switzerland, 1998.
[36] ISO/IEC, “Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Guide to SQuaRE (ISO/IEC 25000:2014),” Geneva, Switzerland, 2014.
[37] World Wide Web Consortium, “WAI Guidelines and Techniques,” *Web Accessibility Initiative (WAI)*, 2015. [Online]. Available: https://www.w3.org/WAI/guid-tech.html.
[38] J. Nielsen, “Usability Inspection Methods,” J. Nielsen and R. L. Mack, Eds. New York, NY, USA: John Wiley & Sons, Inc., 1994, pp. 25–62.
[39] C. Pribeanu, “An approach to Task Modeling for User Interface Design,” *Int. J. Comput. Electr. Autom. Control Inf. Eng.*, vol. 1, no. 5, pp. 1398–1401, 2007.
[40] G. Cockton, “Designing Worth is Worth Designing,” in *Proceedings of the 4th Nordic Conference on HCI: Changing Roles*, New York, NY, USA, 2006, pp. 165–174.
[41] F. Camara and G. Calvary, “Worth-Centered Design in Practice: Lessons from Experience and Research Agenda,” in *Proceedings of 15th International Conference Human-Computer Interaction - Part IV*, J. Abascal, S. Barbosa, M. Fetter, T. Gross, P. Palanque, and M. Winckler, Eds. Cham: Springer International Publishing, 2015, pp. 123–139.
[42] F. Paternò, “ConcurTaskTrees: An Engineered Notation for Task Models.,” in *The Handbook of Task Analysis for Human-Computer Interaction*, D. Diaper and N. A. Stanton, Eds. Mahwah, NJ: Lawrence Erlbaum Associates, 2004, pp. 483–501.
[43] J. Nielsen, “The Use and Misuse of Focus Groups,” *IEEE Softw.*, vol. 14, no. 1, pp. 94–95, 1997.
[44] B. Shneiderman, C. Plaisant, M. Cohen, and S. Jacobs, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 5th Edition. Pearson, 2009.