



HAL
open science

HCI Engineering Integrated with Capability Maturity Models

Taisa Guidini Gonçalves, Káthia Marçal de Oliveira, Christophe Kolski

► **To cite this version:**

Taisa Guidini Gonçalves, Káthia Marçal de Oliveira, Christophe Kolski. HCI Engineering Integrated with Capability Maturity Models. 10th International Conference on Research Challenges in Information Science, IEEE RCIS 2016, Institute of Electrical and Electronics Engineers (IEEE), May 2016, Grenoble, France. pp.633-644, 10.1109/RCIS.2016.7549319 . hal-03350385

HAL Id: hal-03350385

<https://uphf.hal.science/hal-03350385v1>

Submitted on 14 Feb 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

HCI Engineering integrated with Capability Maturity Models: a study focused on Requirements Development

Taisa Guidini Gonçalves, Káthia Marçal de Oliveira, Christophe Kolski
UVHC, LAMIH CNRS UMR 8201
Le Mont Houy, 59313 Valenciennes cedex 9, France
{taisa.guidinigoncalves, kathia.oliveira, christophe.kolski}@univ-valenciennes.fr

Abstract - Even though coming from different communities, software engineering (SE) and Human-Computer Interaction (HCI) engineering have similar goals: the definition of methods, techniques and standards to support the development of software systems, and, in particular for HCI, of interactive systems. Currently, several SE approaches are largely applied in practice thank to the widespread use of capability maturity models in companies. To benefit from this situation and taking advantage of the similarity of SE and HCI goals, we have studied how to support the developers that use these models with approaches from HCI while developing interactive systems. In this paper we present the part of this study regarding requirements development process, an undoubtedly important area from software life cycle. This study involves a detailed analysis of one of the most known models (the CMMI-DEV – Capability Maturity Model Integration for Developers) and interview with twenty experts in HCI domain. As result, a set of HCI approaches to support the developers in the requirements development process area is presented.

Keywords – Capability maturity model, CMMI, Human-computer Interaction

I. INTRODUCTION

In the last years, we have experienced a large production of methods, techniques, guidelines and standards for the conception, design, implementation and evaluation of interactive systems focusing on Human-Computer Interaction (HCI) issues. We can quote for instance the collections by Sears and Jacko [1][2], Richter and Flückiger [3], and Seffah et al. [4][5]. Despite this, it is well known that HCI approaches are not or are insufficiently used in a large number of enterprises ([6][7][8]).

Software engineering community has also proposed several approaches that are commonly used in industry (e.g., Unified Modeling Language [9], design patterns [10], function point analysis [11], and so on). The dissemination¹ of software capability and maturity models in industry has made software engineering approaches even more used. These models are collection of software engineering best practices that help organization to improve their software processes. One of the most known models is CMMI-DEV (Capability Maturity

¹More than 10,000 official appraisals using the international model CMMI (Capability Maturity Model Integration) are reported from over 82 countries. The MR-MPS-SW Brazilian model reports more than 600 officially appraisals since 2005.

Model Integration for developers [12]), an international model widely used in several countries. National models are also developed and used in different countries (e.g., the MR-MPS-SW Brazilian model [13] and the MoProSoft Mexican model [14]).

We believe that if HCI approaches were explicitly integrated to CMMI, one could get better results about their use in industry. This belief motivated us to investigate how we could support the users from capability maturity models (typically software developers) in the development of interactive systems with the proposition of HCI approaches. To that end, we analyzed the documentation of the international capability maturity model (CMMI) and interviewed twenty experts from HCI domain. This paper presents in detail the part of this study concerned with the requirements development process area, that we considered the most relevant of this study. The paper is, therefore, mainly focusing on the process of establishing the list of adequate HCI approaches that could support software developers of CMMI-DEV.

In the next section we present a background where we briefly describe the main features of capability maturity models, HCI engineering and its main related works with software engineering. In section III, we present our study, from the used methodology to the synthesis of the results. Finally, in section IV we conclude the paper by describing our on-going works.

II. BACKGROUND

A. Capability Maturity Models

Capability Maturity models aim to support organizations to define an evolutionary improvement path from immature to disciplined processes, maturity processes with improved quality and effectiveness [12]. To that end, they are composed of software engineering best practices of effective processes for areas of interesting (such as requirements, management, quality, etc.). In the last two decades several capability maturity models have been developed. Von Wangenheim et al. [15] identified 52 models that cover different domains (such as software engineering, e-commerce, and security).

Capability Maturity Model Integration for Development (CMMI-DEV) [12] is an international model for software engineering domain widely used in several countries. In their

study von Wangenheim et al. [15] concluded that 50 from the 52 identified capability maturity models are defined based on CMM/CMMI models. Other models are defined to be compatible with CMMI-DEV (for instance, the Brazilian model MR-MPS.BR [13] holds that when used in companies imply that CMMI-DEV is being also implemented and can be formally appraised). Due to this importance, we chose to perform our study based on CMMI.

The core element of CMMI-DEV is the process area. A process area is a cluster of related practices in an area that, when implemented collectively, satisfies a set of goals considered important for making significant improvement in that area. CMMI-DEV [12] version 1.3 consists of 22 process areas organized in four categories: engineering, support, project management and process management. A process area has 1 to 3 Specific Goals (SG). The SG is composed of Specific Practices (SP), which describe software engineering best practices that are specific to a single process area. Generic goals and generic practices are defined to be applied in all process areas. To support the interpretation and implementation of the practices, typical work products and guidance, defined as subpractices, are provided. Fig. 1 illustrates the CMMI model components.

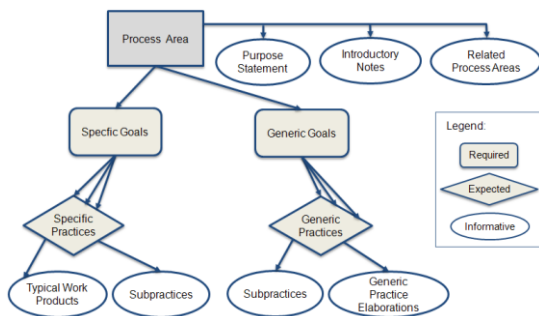


Fig. 1. CMMI model components [12]

Moreover, CMMI-DEV uses the concepts of levels to describe the evolutionary path for an organization that wants software process improvement. Two types of levels are defined: capability and maturity levels. The capability levels enable organization to incrementally improve an individual/ or group of process area. The maturity levels enable the organizations to improve pre-processes by incrementally addressing successive set of predefined process areas.

For the best of our knowledge there is no work that integrates HCI approaches to capability maturity models for software engineering. However, several capability maturity models are defined specifically for usability. In this context a large survey ([16], [17]) presented thirteen usability capability and/or maturity models. These models are based and used the structure of the CMM (Capability Maturity Model) [18], CMMI [12] and ISO 13407 [19]. According these authors, the usability models are “methods for developing user-centered design processes in companies in order to facilitate usability methodologies for creating usable products”. They identified that, in general, the models cover different organizational areas such as: performance of usability processes (user analysis, task analysis, and usability evaluations), and management of usability processes in development projects.

In a more recent work, Raza et al. [20] proposed a specific usability maturity model for open-source projects defining eleven key-factors for usability (such as, user’s requirements, user’s feedback, usability learning, user-centered-design methodology, etc.). The assessment is performed for all key-factors and the rating of the organization considers five levels of maturity. However, the authors recognize that the model does not provide explicit guidelines for improving the usability of projects, only a structure of evaluation.

All these works deal with HCI issues (focused on usability) isolated; that means, with no integration with the software engineering activities as defined in CMMI. Moreover, they were defined as research projects and are not really applied in industry (as CMMI).

B. HCI and Software Engineering

HCI Engineering defined over the years various techniques, methods, life cycles, guidelines and standards to support the development of interactive systems. Some of the most known life cycles are: the star lifecycle [21]; Nielsen usability engineering [22], and Mayhew usability engineering [23]. Furthermore, software engineering classical life cycles (as v-model, spiral, etc.) were also adapted with HCI design activities (see for instance [24][25]).

Moreover, usability standards ([26][27][28]), and specific approaches regarding HCI issues are proposed to support the analysis, design and evaluation of interactive systems ([1][29]). Particularly for requirements and analysis, we can quote the methods for user requirements elicitation (see for instance [30]) and task modeling formalisms ([31]).

Some proposals have investigated how to integrate HCI techniques in the software development process, as follows:

- Ferre et al. [32][33] defined a framework that integrates practices of usability in the software process. To that end they identified the main activities of a user-centered software process (such as: specification of the context of use, usability specifications, prototyping, and usability evaluation) and a set of 35 techniques to support these activities.
- Silva et al. [34] executed a systematic review where they identified, categorized and summarized technologies that have been used for improve the usability within software development process. For them a technology can be a method, technique, model, tool, approach, and other proposal created by the areas of HCI and SE.
- Fischer [35] and Fischer et al. [36][37], propose the integration of usability engineering and HCI through the analysis of standards (ISO 12207 and ISO 9241-210). The result of analysis presents lists of activities, artifacts and correlations of HCI and software engineering, which were validated by HCI experts.

Although the existence of these works, some authors shows that HCI/usability approaches are not used or little used in industry. Bevan [7] argues that usability standards are not used in industry because of the complexity of their documentation, being not easy for designers to use them. Hao

and Jaafar [6] reports a survey with Malaysia companies' practitioners performed to understand and evaluate the practice of usability. They concluded that although most of the respondents know about usability, no budget supports the implementation of usability work in the companies. A recent empirical study [8] investigates the importance of usability issues for small and medium size software companies in Germany. They concluded that they rarely used usability methods.

III. DEFINING HCI APPROCHES TO CMMI

Based on the background previously presented we were motivated to investigate how we could support software developers that use CMMI-DEV with the identification of approaches that could support the implementation of the specific practices of the process areas of this model [38]. These approaches can be introduced as subpractices in each specific practice.

As previously presented, CMMI-DEV is composed of five categories of process areas. Considering that our focus was the engineering activity, that means activities to support the development of interactive systems, we decided to concentrate our study in the process areas of the engineering category. Moreover, the other categories are more generic for any kind of system and support all processes.

Our study followed three main phases as presented in Fig. 2. In the first phase, we analyzed CMMI-DEV process areas. From this analysis and considering the HCI literature we proposed a set of HCI approaches organized into categories for the process areas of engineering. Then, in the second phase we interviewed experts in the HCI domain to evaluate the propositions and to identify other approaches. The last phase was the analysis of the results and synthesis of the final set of HCI approaches organized into categories.

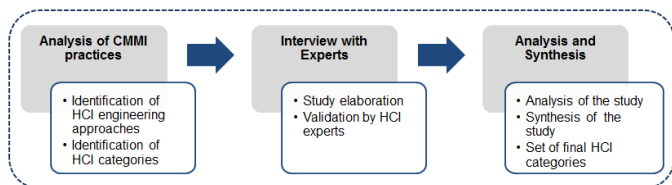


Fig. 2. Research Methodology

The engineering category of CMMI-DEV is composed of five process areas (Fig. 3): Requirements Development (RD), Technical Solution (TS), Product Integration (PI), Validation (VAL) and Verification (VER). All phases of our study were performed for these five process areas. In this paper, we detail only the study for Requirements Development. As can be noted in Fig. 3 this process area is essential in the engineering category since all the other process areas depend of it. Next sections present each one of the phases of our study focusing on Requirement Development practices presented in Fig. 4.

A. Analysis of CMMI

The analysis of CMMI documentation consisted of reading the description of the specific goals and practices of the Requirements Development process area looking for any citation of HCI issues.

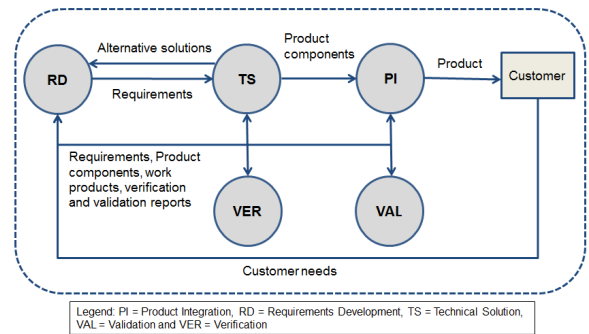


Fig. 3. Engineering process areas [12]

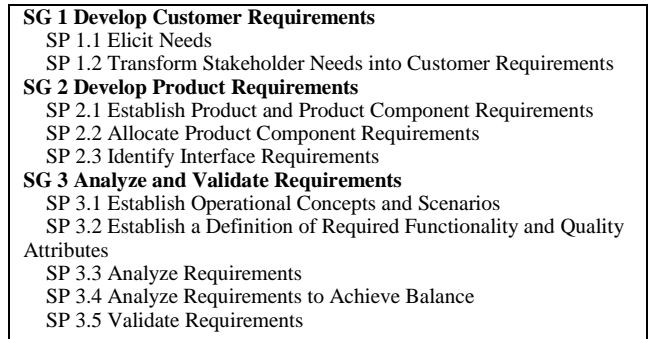


Fig. 4. Requirements development process area [12]

We start seeking any *explicit* citations of HCI engineering by looking for: (i) HCI keywords (for example, external interface, end user, prototype); (ii) examples of techniques or methods of HCI (e.g. end-user task analysis, HCI models); and (iii) examples of work products (e.g. interface design specifications, user manual). Then, we looked for citations that were not directly related to HCI Engineering but that we could interpret in benefit of the use of it. We classify this information as *implicit* citations. Explicit and implicit citations were highlighted in the text and reviewed together by the three authors of this paper.

Fig. 5 shows an example of explicit citation identified for the RD process area. We note the importance of involving the end-user in the requirements elicitation and the use of techniques that are related to HCI for this purpose. These citations are highlighted as explicit since it mentions the words "end-user tasks analysis" and "prototypes and models" which are approaches used in HCI.

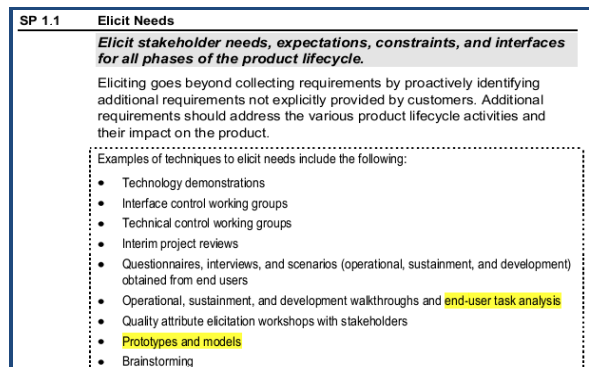


Fig. 5. Analysis of Requirements Development process area (extract from [12])

Table I presents our findings for requirements development process area.

TABLE I. ANALYSIS OF CMMI-DEV – SPECIFIC PRACTICES OF RD

Requirements Development (RD)		Type of citation
Specific Practice	Information of HCI	
SP 1.1: Elicit stakeholder needs, expectations, constraints, and interfaces for all phases of the product lifecycle.	Subpractice 1: “Engage relevant stakeholders using <u>methods for eliciting needs, expectations, constraints, and external interfaces.</u> ” Examples of techniques: “Questionnaires, interviews, and <u>scenarios obtained from end users</u> ”, “ <u>end-user task analysis</u> ” and “ <u>prototypes and models</u> ”	Explicit
SP 1.2: Transform stakeholder needs, expectations, constraints, and Interfaces into prioritized customer requirements.	Subpractice 2: “Establish and maintain a prioritization of customer functional and <u>quality attribute requirements.</u> ”	Implicit
SP 2.1: Establish and maintain product and product component requirements, which are based on the customer requirements.	Subpractice 3: “Develop <u>architectural requirements</u> capturing <u>critical quality attributes</u> and quality attribute measures necessary for establishing the product architecture and design.”	Implicit
SP 2.2: Allocate the requirements for each product component.	-	-
SP 2.3: Identify interface requirements.	-	-
SP 3.1: Establish and maintain operational concepts and associated scenarios.	Subpractice 4: “Develop a detailed <u>operational concept</u> , as products and product components are selected, <u>that defines the interaction of the product, the end user, and the environment</u> , and that satisfies the operational, maintenance, support, and disposal needs.”	Explicit
SP 3.2: Establish and maintain a definition of required functionality and quality attributes.	Subpractice 2: “Identify desirable functionality and <u>quality attributes.</u> ”	Implicit
SP 3.3: Analyze requirements to ensure that they are necessary and sufficient.	Subpractice 1: “ <u>Analyze stakeholder needs, expectations, constraints, and external interfaces</u> to organize them into related subjects and remove conflicts.”	Implicit
SP 3.4: Analyze requirements to balance stakeholder needs and constraints.	Subpractice 1: “ <u>Use proven models, simulations, and prototyping</u> to analyze the balance of stakeholder needs and constraints.”	Explicit
SP 3.5: Validate requirements to ensure the resulting product will perform as intended in the end user's environment.	Subpractice 3: “Assess the design as it matures in the context of the <u>requirements validation</u> environment to identify validation issues and expose unstated needs and customer requirements.” Examples of technique: “ <u>Prototyping</u> ”.	Explicit

We analyzed 10 practices and we found that 8 practices address HCI approach. For each specific practice it is shown the exact transcription of the documentation where explicit or implicit citations were identified (underlined information). We note explicit citations that mention HCI approaches, such as the examples of techniques for requirement development (SP1.1 end-user task analysis, prototypes), prototyping use for requirement development (SP3.4 and SP3.5). Implicit citations can also be found. The need of quality attributes requirements (SP1.2), for instance, implies consider usability (ISO 9241-11 [39]), in the case of interactive systems.

As can be noted in Table I we did not find any explicit or implicit citation for two practices: SP2.2 and SP2.3. Indeed, these practices are more related to functional aspects of the system. SP2.2 refers to the allocation of functional requirements to software components. SP 2.3 is related to the internal interface between functional components not associated to HCI itself.

After identifying all citations, we organized them separately to identify the main approaches related to HCI engineering. Fig. 6 presents a summary of this analysis, where we show the main citations (implicit and explicit) for RD process area. Analyzing the citations we identified five groups of approaches, that we call categories: prototyping, techniques to validate requirements, methods of end-user tasks analysis, detailed operational concept and scenarios, standards and guidelines for design interfaces and end-user documents. For example, the citations in the practices SP1.1, SP1.2, SP2.1 and SP3.3 refers to requirements analysis and documentation that are usually supported for approaches like CTT, HTA, K-MAD [31].

After this first analysis, we refined the categories considering the literature and our own experience in HCI. Table II presents the final set of HCI approaches indicating to which specific practices should be used when implementing CMMI-DEV for the design of interactive systems.

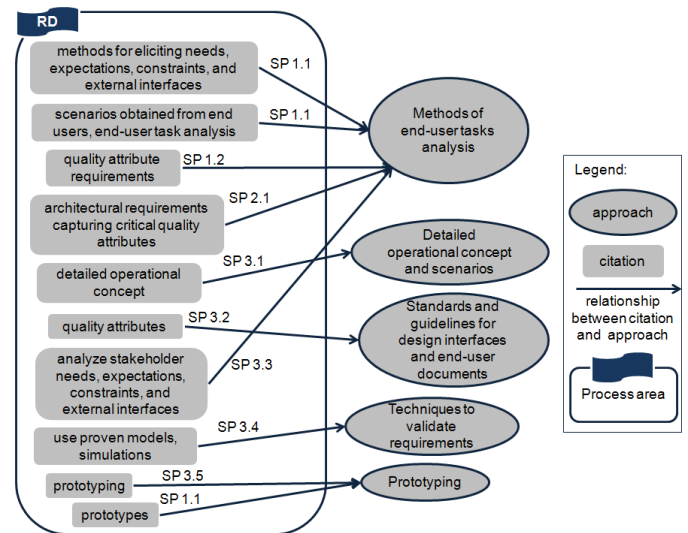


Fig. 6. Analysis of the citations (implicit and explicit)

TABLE II. HCI ENGINEERING TO SUPPORT CMMI

Specific Practice	HCI Engineering Approaches to support CMMI
SP 1.1 SP 1.2 SP 2.1 SP 3.3	Task Analysis Methods for HCI Examples: • CTT (Concur Task Tree) • K-MAD (Kernel of Model for Activity Description) • HTA (Hierarchical Task Analysis) • SADT (Structured Analysis and Design Technique) • GTA (Groupware Task Analysis)
SP 1.1 SP 3.5	Prototype for HCI requirements Examples: • Rapid Prototyping <ul style="list-style-type: none"> □ Offline techniques: Paper and pencil (paper sketches, storyboards), Mockups, Wizard of Oz, Video prototyping □ Online techniques using software tools: Non-interactive simulations, Interactive simulations, Scripting languages
SP 3.1	Operational Concepts and Scenarios Specification for HCI Examples: • Context awareness • Adapting to context • User profile • Persona • Use cases
SP 3.2	Standards and Guidelines for design and documentation of HCI Examples: • Ergonomic Criterion ([40]) • ISO/IEC 9126-1 (2001) • ISO 9241-11 (1998) • ISO/IEC 25000 (2014)
SP 3.4	Techniques to validate HCI requirements Examples: • Proto Task (K-MAD) • Task Model Simulator (CTT) • Focus Group to validate requirements

B. Interview with Experts

With the set of approaches identified, organized into categories and associated to each specific practice of RD, we planned a validation of this first proposal with experts in HCI domain. In this section we present the planning and execution of the interviews.

1) Planning

To plan the interviews we elaborated the following protocol:

- Object of the study - Set of HCI approaches and practices of requirement development process area of the CMMI-DEV model.
- Objective - Validate if the proposed set of HCI approaches support the practices of requirement development process area of the CMMI-DEV model.
- Subjects - HCI experts from different countries. They were selected by convenience from the research contacts of one of the authors and from their reputation recognized by the community.
- Instrumentation - A specific questionnaire (see part of the questionnaire in Fig. 7) was elaborated in three languages: English, French and Portuguese. For each specific practice, the experts should answer if they

agree, partially agree or do not agree that the associated HCI approach supports the practice.

- Results Validity - To analyze the results we considered the threats of validity proposed by Wohlin et al. [41], that is construct validity, internal validity, conclusion validity, and external validity. We analyzed each one of them trying to define some mitigation as described below.

Threats to the construct validity illustrate from the relation between theory and observation, and the questions of whether the treatment adequately reflects the cause; whether the result adequately reflects the effects. To minimize this threat we built the questionnaire using the original text extracted from the official documentation of CMMI-DEV (in the three languages). Moreover, we had the official documentation during the interview to be consulted in case of doubts.

Threats to the internal validity draw from influences that can affect the independent variables with respect to causality without the researchers' knowledge. In our case study this threat is associated with the experts involved in the evaluation. The experts were selected by convenience, researchers known by the authors. The experts should evaluate if they agree that the HCI could support the practices and therefore we assumed that the experts knew the proposed approaches. To minimize this risk, we selected only professionals (professors or people from industry) that have experience in HCI domain and have a PhD degree level. We decide that it was not necessary the experts know CMMI since the practices of process areas from engineering category are typical in the production of software systems. Moreover, we decided to get the opinions by interview where we could clarify each doubt they could have while answering the questionnaire. The authors that made the interviews know CMMI and already participated in official CMMI appraisals.

Threats to the conclusion validity are those that affect the ability to draw the correct conclusion about the relation between the treatment and the outcome of our study (in our case the HCI approaches associated to each specific practice). To reduce this risk we decide to perform interviews individually and not using survey. In this way when the experts disagrees or partially agree with the associated approaches, they were asked to justify their opinion and if other proposals could be made. We decided also, to classify the experts in seniors (with experience greater than or equal to 19 years) and juniors (with experience less than 19 years). Each suggestion of modification should be confirmed at least by one senior expert. The more senior experts suggest a change, the more accurate is the proposition of change. The chose of experts by convenience could also be a threat in the sense that they could feel not comfortable to discord of the propositions. To mitigate this bias the author who knows some of the experts did not participate in any interview and the other authors that did the interviews explained clearly that the propositions were only an initial not validated set that needed to be improved.

Finally, threats to the external validity are conditions that limit our ability to generalize the results of our experiment outside the scope of our study. To minimize this risk we

decide to perform the interview with experts from different countries. However, since the subjects were selected by convenience, we had more experts from the countries from the origin of the authors. We accepted this risk considering that the experts are well-known by the HCI community.

2) Execution

HCI experts answered the questionnaires during individual face meetings (either in person or by video conference). Fig. 8 presents an activity diagram that represents how the interview was performed. We start by presenting the goal of the study, and explaining CMMI-DEV in general. Then, we asked their personal information (time work in HCI (years), if they know CMMI model, their experience and education). After that, for each specific practice we asked their opinion about the proposed approach associated to each practice. When they disagree or partially agree, they should justify with a description in the justification column (see Fig. 7). In the same time, the interviewer takes notes of verbal observations/explanations performed by experts during the interview.

Twenty experts were interviewed. The interviews were performed for the five process areas from engineering category. They were not recorded, and all the information collected was registered in the questionnaire by the expert when in person or by the interviewer and/or the expert when by videoconference. Moreover, the interviewer took informal

notes of interest comments made by the experts (references, sites do be later consulted, example of tools, etc.). Each interview took on average slightly more than one hour (we had 24 hours and 52 minutes of interview in total).

Table III presents the demographic data from the experts (seniors (s) and juniors (j)). We note that in average they have 19 years of experience (from 7 to 40 years). They have different nationalities: 12 from France, 5 from Brazil, 1 from Belgium, 1 from Tunisia, and 1 from Algeria. All of them have teaching experience, 14 have experience in industry, and 7 come from software engineering domain as well as HCI. Moreover, 6 know CMMI model (although this was not required for answering the questionnaire and we consider has no impact for the results).

C. Analysis and Synthesis

After finishing all interviews we performed the descriptive quantitative and qualitative analysis of the data. First of all, we analyze the general results for all practices of Requirements Development. As previously describe 8 practices were object of our study. For 7 of them we had proposed one category of HCI and for one of them (SP.1.1) we proposed two categories of HCI approaches. As consequence, we should have 180 answers to analyze (since we performed 20 interviews). However, for the practice SP3.4 one expert preferred to not give his opinion since he did not know some of the presented techniques. As consequence, we got 179 responses to analyze.

Process Area and Specific Goal (SG)	Specific Practice (SP)	Methods, techniques, standards and patterns of HCI	Answer			Justification
			I agree	I partially agree	I don't agree	
Requirements Development <i>SG 1 Develop Customer Requirements</i> Stakeholder needs, expectations, constraints, and interfaces are collected and translated into customer requirements.	SP 1.1 Elicit Needs Elicit stakeholder needs, expectations, constraints, and interfaces for all phases of the product lifecycle.	<i>Task Analysis Methods for HCI</i> Examples: • CTT (Concur Task Tree) • K-MAD (Kernel of Model for Activity Description) • HTA (Hierarchical Task Analysis) • SADT (Structured Analysis and Design Technique) • GTA (Groupware Task Analysis)			x	<i>"Task Analysis is further modeling activity than an elicitation activity."</i>
		<i>Prototype for HCI requirements</i> • Rapid Prototyping □ Offline techniques: Paper and pencil (paper sketches, storyboards), Mockups, Wizard of Oz, Video prototyping □ Online techniques using software tools: Non-interactive simulations, Interactive simulations, Scripting languages -		x		<i>"Since that has not focused on finding the solution. It may be too early."</i>
	SP 1.2 Transform Stakeholder Needs into Customer Requirements Transform stakeholder needs, expectations, constraints, and interfaces into prioritized customer requirements.	<i>Task Analysis Methods for HCI -</i> Examples: • CTT (Concur Task Tree) • K-MAD (Kernel of Model for Activity Description) • HTA (Hierarchical Task Analysis) • SADT (Structured Analysis and Design Technique) • GTA (Groupware Task Analysis)		x		<i>"It is high level of abstraction. To add Scenarios and textual representations."</i>

Fig. 7. Extract of the questionnaire

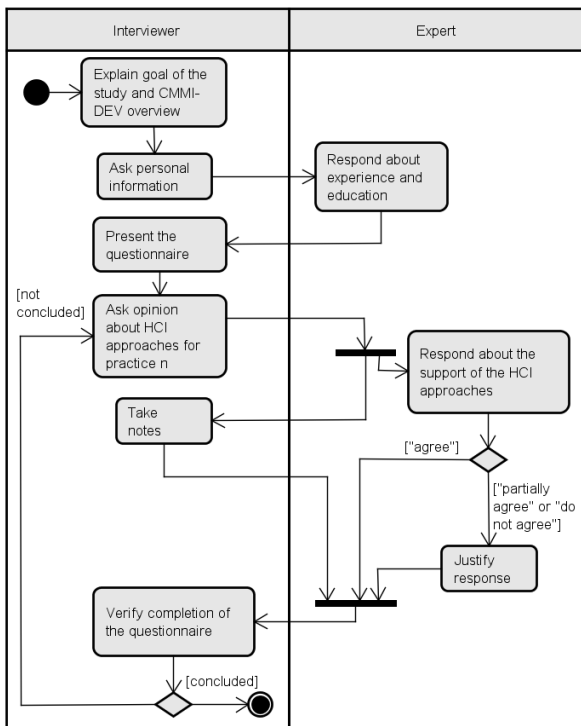


Fig. 8. Interview sessions

TABLE III. LIST OF EXPERTS

Expert	Time work in HCI (years)	Know CMMI model	Industry experience	PhD domain	Origin
E1j	13	No	No	HCI	France
E2s	25	No	Yes	HCI	France
E3j	8	No	No	HCI	France
E4j	8	No	Yes	SE and HCI	France
E5s	25	Yes	Yes	SE and HCI	France
E6s	26	No	No	HCI	France
E7s	27	Yes	Yes	SE and HCI	Belgium
E8s	20	No	Yes	HCI	Brazil
E9j	10	No	No	HCI	Brazil
E10s	25	No	No	HCI	France
E11s	20	No	Yes	SE and HCI	France
E12s	40	Yes	Yes	SE and HCI	France
E13j	12	No	Yes	SE and HCI	France
E14j	7	Yes	Yes	SE and HCI	France
E15j	10	No	No	HCI	Brazil
E16s	30	Yes	Yes	Computer Science and HCI	France
E17s	27	No	Yes	Computer Science and HCI	Tunisia
E18s	21	No	Yes	Computer Science and HCI	Brazil
E19j	10	Yes	Yes	Computer Science and HCI	Brazil
E20s	27	No	Yes	Computer Science and HCI	Algeria

* j= junior, s= senior

Fig. 9 presents the general results for all practices considering the three possible answers: 47% agree (85 responses), 46% partially agree (82 responses) and 7% disagree (12 responses) with the propositions.

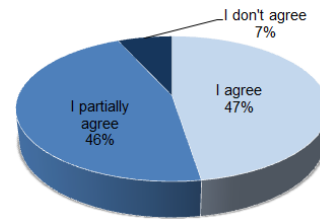


Fig. 9. General results

Fig. 10, Fig. 11 and Fig. 12 present the details of these results by practice and considering the profile of the expert. A joint analysis of the three graphs shows the following remarks:

- 6 practices from the 8 presented at least one response “I don't agree”; where the minimum was 1 disagree for SP3.2, SP3.3, SP3.4 and SP3.5, and the maximum was 5 disagrees for SP1.1 (2 for the approach Task analysis and 3 for prototype). Despite of this, there were very few disagreements (7 from junior experts and 5 from senior experts).
- For two practices (SP1.2 and SP2.1), the approaches proposed were confirmed either completely (“I agree”) or partially (“I partially agree”) with no disagreement. The partially agree was equilibrated among seniors and juniors experts. The justifications can generate some modifications.
- SP3.3 was the practice with more agreements (15). The 4 partially agree come half from seniors and half from junior experts (an analysis of the justifications will later presented). This result indicates that the category proposed for this practice is adequate and should be kept.
- All practices presented “I partially agree” as response; where the minimum was 4 responses for SP3.2 and 17 responses for SP1.1 (12 for Task Analysis category and 7 for prototype). Most of the answers were given by senior experts (except for SP3.3 where we have the same number of seniors and juniors, for all other propositions we had much more seniors that partially agree than juniors). For sure, the justifications for these answers should be carefully analyzed in order to identify modifications of the set of categories).

To complete our descriptive analysis we also compute the mode and median value presented in Table IV. Since almost all practices had 20 answers (except for practice 3.4 as previously described) the median value was defined with the 10th and 11th element. When they are different the decision was considered based on the mode. The practice 3.4 was bimodal and the decision was based on the median. Analyzing the decision result of Table IV we note we have 5 agreements, 4 partially agreements, and no disagreement which indicates our proposition is valid but should be improved. In this way, we conducted a qualitative analysis of all justifications.

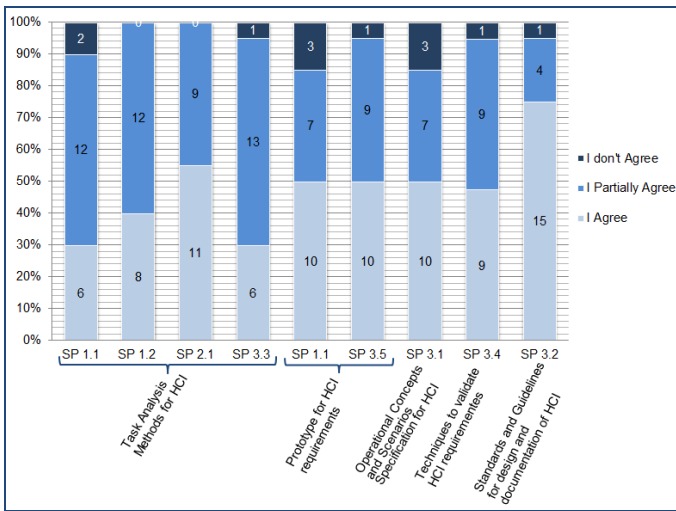


Fig. 10. Quantitative results by practice

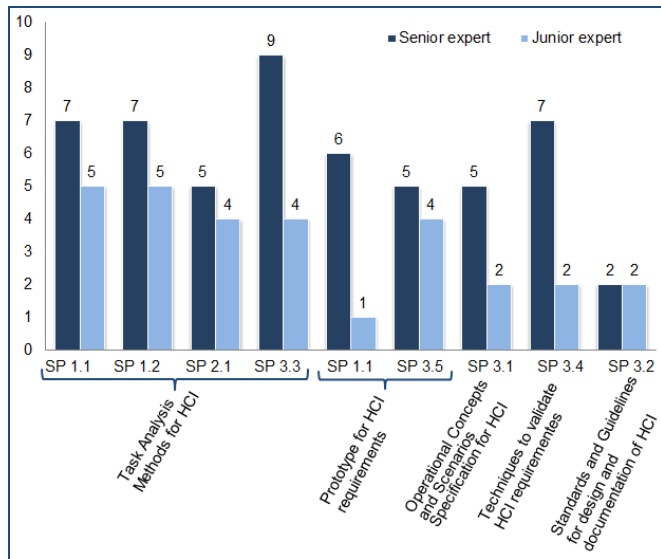


Fig. 11. Quantitative results to "I partially agree"

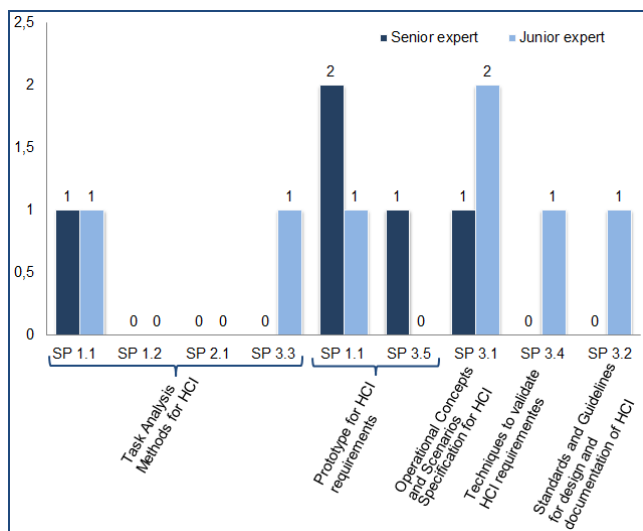


Fig. 12. Quantitative results to "I don't agree"

For qualitative analysis we organized in a single form all comments wrote by each expert for each practice and associated approach. For this part we analyzed each comment for each individual answer according to the following steps: (i) first, all comments were analyzed by an author in order to identify if there were proposal for improvements or exclusions of categories or new approaches proposed; (ii) second, the comments were organized in groups of similarity of propositions; (iii) third, the groups of comments were reviewed by two authors; (iv) finally, the propositions were reviewed by all authors in a three-hours meeting.

To better explain this process we show an example of the analysis for the practice SP1.1. Table V presents the part of form for the practice SP1.1. The comments of each expert were transcribed exactly equal from their questionnaire to the column in which it was given the answer (I agree, I partially agree, I do not agree).

TABLE IV. MODE AND MEDIAN OF THE RESULTS

Practice	HCI Approaches	Mode	Median	Decision
SP1.1	Task Analysis Methods for HCI	Partially agree	Partially agree	Partially agree
SP1.1	Prototype for HCI requirements	Agree	Partially agree and Agree	Agree
SP1.2	Task Analysis Methods for HCI	Partially agree	Partially agree	Partially agree
SP2.1	Task Analysis Methods for HCI	Agree	Agree	Agree
SP3.1	Operational Concepts and Scenarios Specification for HCI	Agree	Partially agree and Agree	Agree
SP3.2	Standards and Guidelines for design and documentation of HCI	Agree	Agree	Agree
SP3.3	Task Analysis Methods for HCI	Partially agree	Partially agree	Partially agree
SP3.4	Techniques to validate HCI requirements	Partially agree and Agree	Partially agree	Partially agree
SP3.5	Prototype for HCI requirements	Agree	Partially agree and Agree	Agree

In the first step (i), one of the authors analyzed all the comments. For Task analysis methods for HCI, 12 experts (7 seniors and 5 juniors) partially agree and 2 experts (1 senior and 1 junior) disagree. All comments indicate, in general, that these methods were not enough, and others should be used. The two *disagreements* suggest that elicitation techniques should be used before task modeling methods. For "I partially agree" answer, the experts suggested some techniques.

In the second step (ii) the suggested techniques were separately organized in a different document (see Table VI). Thus, based on literature [29][30], the techniques were grouped in two different categories: Techniques to identify user needs (techniques marked in gray in Table V – 3 seniors and 5 juniors) and Techniques to identify user and organizational requirements (the one marked in black – 6 seniors and 2 juniors).

TABLE V. EXAMPLE OF ANALYSIS FOR THE PRACTICE SP1.1.

Category	Practice	Answer – I agree	Answer - I partially agree	Answer – I don't agree	Conclusion
Task Analysis Methods for HCI	SP 1.1	<p>E1j E2s E5s- “To include a new category Interview.” E6s E13j E20s</p>	<p>E3j- “It is not enough. I suggest defining a new category Interview.” E4j- “It is not enough. I suggest techniques for description of requirements.” E7s- “It is not enough. It does not express requirements. I propose to include VOLERE, RESCUE. See methods in ISO 24744.” E8s- “To add other representations of tasks such as: scenarios, persona, storyboard (descriptive representations). The task model represents the HOW, other techniques represent WHAT. Suggests the inclusion of a new category with methods such as: field study, interviews, brainstorming.” E10s- “It is not enough. There is a step between obtaining the needs and tasks analysis. I suggest including personas and scenarios.” E11s- “It is not enough. I suggest the inclusion of Questionnaires, Focus Group, Scenarios and Personas.” E14j- “It is not enough. There is a step before. I suggest including scenarios, Focus Group and Questionnaires.” E15j- “I Suggests to include techniques to requirements elicitation: Brainstorming, Questionnaires, Interview, Observation, and Ethnography. Also, to include Scenarios.” E16s- “Before the model we need to do the tasks analysis. I suggest the inclusion of a new category Methods to task analysis.” E17s- “It does not integrate the contextual aspect.” E18s- “To add Scenarios and Personas.” E19j- “Include techniques to bring the vision of all stakeholders for development (Interviews).”</p>	<p>E9j- “Task Analysis is further modeling activity than an elicitation activity.” E12s- “Use informal techniques.”</p>	<p>Inclusion of 2 new categories: (i) “Techniques to identify user needs” (proposed by 3 senior experts: E5, E8, E11, and 5 junior experts: E3, E4, E14, E15, E19) – Total: 8 experts (ii) “Techniques to identify user and organizational requirements” (proposed by 7 senior experts: E7, E8, E10, E11, E16, E17, E18, et 2 junior experts: E14, E15) – Total: 9 experts</p>
Prototype for HCI requirements	SP 1.1	<p>E1j E2s E4j E5s E6s E7s E9j E13j E14j E19j</p>	<p>E8s- “Since that has not focused on finding the solution. It may be too early.” E11s- “Only if it is to discuss the design with the team.” E15j- “To consider Experience prototype for the developer.” E16s- “Before the model we need to do the tasks analysis. I suggest the inclusion of task analysis.” E17s- “It does not integrate the contextual aspect.” E18s- “Since that has not focused on finding the solution. To present different versions.” E20s- “I think it is early to use online techniques.”</p>	<p>E3j- “I think it is early to build a prototype.” E10s- “It's not the right time.” E12s- “The prototype should only be used as a form of user reaction.”</p>	<p>Exclusion of the category “Prototype for HCI requirements” (proposed by 4 senior experts: E8, E10, E12, E20 and 1 junior expert: E3) – Total: 5 experts</p> <p>The suggestions will be considered in the recommendations format for our study. (E11, E15, E18)</p> <p>E16, E17 (senior experts) – Their suggestions were considered in the new category “Techniques to identify user and organizational requirements”</p>

While analyzing the techniques and considering our notes from the interviews, we observed that the approaches cited in Task analysis methods are related only to the task modeling, and that task analysis was an approach to identify users and organization requirements suggested by 2 senior experts. As consequence, we decided to rename the category *Task Analysis Methods for HCI* to *Task Modeling*.

In the third step (iii) two of the authors re-analyzed the comments and confirmed the need of creation of the two categories for identification users' needs and requirements. Analyzing the comments from disagreement and partial agreement, we noted that they suggest the same idea: methods of elicitation should be performed to support this practice. A senior and a junior expert disagree with the use of task modeling approaches. The junior suggests eliminate it but the senior suggested only to integrate others. Analyzing the 12 partially agreements, we observed that they argued that task modeling approaches are not enough. In our notes from the interviews, we found that they said the task modeling in this phase might help when designed in a very high-level of abstraction to understand the general sequence of tasks. Since we had 12 partially agreement that only suggests new approaches, and 6 agreements, we decided to keep this category to support this practice in our final analysis meeting (fourth step (iv)).

TABLE VI. ANALYSIS OF THE PRACTICE SP1.1

Expert \ Technique	Expert											
	E3junior	E4junior	E7senior	E8senior	E10senior	E11senior	E14junior	E15junior	E16senior	E17senior	E18senior	E19junior
Description of Requirements												
Interview												
Questionnaire												
Brainstorming												
Field study												
Scenarios												
Personas												
Storyboard												
Focus group												
Ethnography												
Task analysis												
Organizational Context												

In the analysis of the category “Prototype for HCI requirements” in SP1, we note that we had only three agreements. Moreover, for the responses (2 senior experts and 1 junior expert) as “*I don't agree*” we note that the experts argue it is early to build a prototype or this cannot have the purpose of finding the solution. We note that the 7 responses classified as “*I partially agree*” (6 senior experts and 1 junior expert), the experts argues in the same way than the experts that responded as “*I don't agree*”. Therefore, we concluded in our final analysis meeting that this category should be excluded from this practice.

In the same way, we analyzed all practices. We identify some modifications as follows:

- As previously described the *Task analysis methods for HCI* was renamed to *Task modeling*. The category *Standards and Guidelines for design and documentation of HCI* was simplified to consider only the design since they consider the documentation is not covered in the standards.
- For SP1.2 we had 12 partially agreement and no disagreement. In the justifications 4 experts (3 seniors and 1 junior) suggest the use of techniques to user requirements already identified previously in the category *Techniques* to identify user and organizational requirements. Moreover 3 experts suggest including prototype and 3 to use the standards as a way to prioritize the quality requirements.
- For SP2.1, we had 9 partially agreements and no disagreements. 4 experts (1 senior and 3 junior) suggest including the use of the standards to establish requirements for user interfaces.
- For SP3.1, The category *Operational Concepts and Scenarios Specification for HCI* was eliminated since the experts justified their partially agreement showing that this category was a miscellaneous of approaches. However, the relevant approaches were placed in the new category *Techniques to identify user and organizational requirements* and therefore associated to this practice.
- Finally for SP3.5, we had 9 partially agreement (5 from senior and 4 from junior experts). From them 5 experts suggest including the category of *Techniques* to validate requirements to support this practice.

Table VII presents the summary with the proposition submitted for evaluation of the experts and the final set of categories for each specific practice.

TABLE VII. SET OF CATEGORIES (BEFORE AND AFTER VALIDATION WITH EXPERTS)

Specific Practice (SP)	Methods, techniques, standards and patterns of HCI (Before)	Methods, techniques, standards and patterns of HCI (After)
SP 1.1 Elicit Needs	- Task Analysis Methods for HCI - Prototype for HCI requirements	- Task Modeling - Techniques to identify user needs - Techniques to identify user and organizational requirements
SP 1.2 Transform Stakeholder Needs into Customer Requirements	- Task Analysis Methods for HCI	- Task Modeling - Techniques to identify user and organizational requirements - Prototype for HCI requirements - Standards and Guidelines for HCI design
SP 2.1 Establish Product and Product Component Requirement	- Task Analysis Methods for HCI	- Task Modeling - Standards and Guidelines for HCI design

TABLE VII (CONT.). SET OF CATEGORIES (BEFORE AND AFTER VALIDATION WITH EXPERTS)

Specific Practice (SP)	Methods, techniques, standards and patterns of HCI (Before)	Methods, techniques, standards and patterns of HCI (After)
SP 3.1 Establish Operational Concepts and Scenarios	- Operational Concepts and Scenarios Specification for HCI	- Techniques to identify user and organizational requirements
SP 3.2 Establish a Definition of Required Functionality and Quality Attributes	- Standards and Guidelines for design and documentation of HCI	- Standards and Guidelines for HCI design
SP 3.3 Analyze Requirements	- Task Analysis Methods for HCI	- Task Modeling
SP 3.4 Analyze Requirements to Achieve Balance	- Techniques to validate HCI requirements	- Techniques to validate HCI requirements
SP 3.5 Validate Requirements	- Prototype for HCI requirements	- Prototype for HCI requirements - Techniques to validate HCI requirements

Based on this analysis and synthesis, our final set of categories to support requirement development practices from CMMI-DEV is presented in Table VIII.

TABLE VIII. HCI ENGINEERING TO SUPPORT CMMI AFTER VALIDATION OF EXPERTS

Specific Practice (SP)	HCI Engineering Approaches to support CMMI
SP 1.1	Techniques to identify user needs Examples: • Brainstorming • Interviews • Surveys/Questionnaires • Card Sorting • Focus Groups • Field Studies
SP 1.1 SP 1.2 SP 3.1	Techniques to identify user and organizational requirements Examples: • Persona • Scenario • User stories • User profile (detailed) • Task analysis • Context-of-use analysis • Storyboards • Requirements specification templates (e.g. VOLERE, IEEE, RESCUE)
SP 1.2 SP 2.1 SP 3.3	Task Modeling Examples: • CTT (Concur Task Tree) • K-MAD (Kernel of Model for Activity Description) or MAD (Model for Activity Description) • HTA (Hierarchical Task Analysis) • SADT (Structured Analysis and Design Technique) or SADT coupled with Petri Nets • GTA (Groupware Task Analysis) • Task Model Standard (W3C)

TABLE VIII (CONT.). HCI ENGINEERING TO SUPPORT CMMI AFTER VALIDATION OF EXPERTS

Specific Practice (SP)	HCI Engineering Approaches to support CMMI
SP 2.1 SP 3.2	Standards and Guidelines for HCI design Examples: • Ergonomic criteria ([40][42]) • ISO/IEC 9126-1 (2001) • ISO 9241-11 (1998) • ISO/IEC 25000 (2014) • Accessibility standards and guidelines (WAI-W3C) • Nielsen's Heuristics • Golden Rules of Interface Design
SP 3.4	Techniques to validate HCI requirements • Proto Task (K-MAD) • Task Model Simulator (CTT) • Focus group to validate HCI requirements • Thinking aloud

IV. CONCLUSION

This paper has presented a study of the capability maturity model for developer (CMMI-DEV) that aims to identify how HCI approaches can support engineering practices, advocated by process area of requirements development from this model. Our initial proposition was composed of five categories of HCI approaches. After interview with twenty HCI experts, we obtained six categories of HCI approaches (where one category was eliminated, two new categories were created and several examples of approaches were included).

As on-going work we have been analyzing the results of the other process areas from engineering category. Our future work includes defining methodological guidance to support CMMI-DEV users. This guidance will be used by students from the master program of the UVHC in the requirements specification of an interactive system.

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] A. Sears, J. A. Jacko, Human-Computer Interaction: Development Process. NW: CRC Press, 2009.
- [2] A. Sears, J. A. Jacko, Human-computer interaction: Design issues, solutions, and applications. NW: CRC Press, 2009.
- [3] M. Richter, M. Flückiger, User-Centred Engineering: Creating Products for Humans. Springer Berlin Heidelberg, 2014.
- [4] A. Seffah, J. Gulliksen, M. C. Desmarais, Human-Centered Software Engineering - Integrating Usability in the Software Development Lifecycle, Human-Computer Interaction Series, Volume 8. Springer Netherlands, 2005.
- [5] A. Seffah, J. Vanderdonck, M. C. Desmarais, Human-Centered Software Engineering - Software Engineering Models, Patterns and Architectures for HCI, Human-Computer Interaction Series. Springer London, 2009.
- [6] H. M. Hao, A. Jaafar, "Usability in Practice: Perception and Practicality of Management and Practitioners," in International Conference on Pattern Analysis and Intelligent Robotics, pp. 211-216, 2011.
- [7] N. Bevan, "International Standards for Usability Should Be More Widely Used," Journal of Usability Studies, 4 (3), pp. 106-113, 2009.

- [8] F. Scheiber, D. Wruk, A. Oberg, J. Britsch, M. Woywode, A. Maedche, F. Kahrau, H. Meth, D. Wallach, M. Plach, "Software Usability in Small and Medium Sized Enterprises in Germany: An Empirical Study," in *Software for People - Fundamentals, Trends and Best Practices*, pp 39-52, 2012.
- [9] J. Rumbaugh, I. Jacobson, G. Booch, *Unified Modeling Language Reference Manual*. Addison Wesley, 1997.
- [10] E. Gamma, R. Helm, R. Johnson, J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1994.
- [11] A. J. Albrecht, "Measuring Application Development Productivity," *IBM Application Development Symposium*, pp. 83-92, 1979.
- [12] CMMI Product Team, "CMMI® for Development (CMMI-DEV), V1.3, (CMU/SEI-2010th-TR-033 ed.), Pittsburgh, PA, USA: Carnegie Mellon University, 2010.
- [13] SOFTEX, "MPS.BR - Brazilian Software Process Improvement, MR-MPS-SW - General Guide," retrieved January, 2016 from <http://www.softex.br/mpsbr>, 2016. (in portuguese)
- [14] H. Oktaba, C. A. Esquivel, A. S. Ramos, A. M. Martínez, G. Q. Osorio, M. R. López, F. L. L. Hinojo, M. E. R. López, M. J. O. Mendoza, Y. F. Ordóñez, M. A. F. Lemus, "Modelo de Procesos para la Industria del Software MoProSoft," Versión 1.3, 2005. (in spanish)
- [15] C. G. von Wangenheim, J. C. R. Hauck, C. F. Salviano, A. von Wangenheim, "Systematic Literature Review of Software Process Capability/Maturity Models," in *International SPICE Conference on Process Improvement and Capability determination*, 2010.
- [16] T. Jokela, "Usability Maturity Models: Making your Company User-Centered," *User Experience Magazine*, 9 (1), 2010.
- [17] T. Jokela, M. Siponen, N. Hirasawa, J. Earty, "A survey of usability capability maturity models: implications for practice and research," *Behaviour & Information Technology*, 25 (3), pp. 263-282, 2006.
- [18] M. Paulk, C. Weber, S. Garcia, M. Chrissis, M. Bush, "The Capability Maturity Model: Guidelines for Improving the Software Process," Reading, MA: Addison-Wesley, 1995.
- [19] ISO, "ISO 13407:1999 Human-Centred Design Process for Interactive System", 1999.
- [20] A. Raza, L. F. Capretz, F. Ahmed, "An open source usability maturity model (OS-UMM)," *Computers in Human Behavior*, 28, pp. 1109-1121, 2012.
- [21] D. Hix, H. Hartson, *Developing user interfaces: ensuring usability through product and process*. NY: John Wiley & Sons, 1993.
- [22] J. Nielsen, *Usability Engineering*. NY: Academic Press, 1993.
- [23] D. Mayhew, *The Usability Engineering Lifecycle: a practitioner's handbook for user interface design*. CA: Morgan Kaufmann, 1999.
- [24] A. Valentin, G. Valléry, R. Lugongsang, *L'évaluation ergonomique des logiciels, une démarche itérative de conception*. Montrouge: ANACT, 1993. (in French)
- [25] C. Kolski, *Interfaces homme-machine, applications aux systèmes industriels complexes*. Paris: Hermès Science Publications, 1997. (in French)
- [26] ISO/TR, "ISO/TR 18529:2000 Ergonomics of human-system interaction - Human-centred lifecycle process descriptions," 2000.
- [27] ISO/TR, "ISO/TR 16982:2001 Ergonomics of human-system interaction - Usability methods supporting human centred design," 2001.
- [28] ISO, "ISO 9241-210:2010 Ergonomics of human-system interaction - Part 210: Human-centred design for interactive systems," 2010.
- [29] M. Maguire, "Methods to support human-centred design." *International Journal of Human-Computer Studies*, 55, pp. 587-634, 2001.
- [30] C. Courage, K. Baxter, *Understanding your users: a practical guide to user requirements, methods, tools, and techniques*. San Francisco, CA: Morgan Kaufmann Publishers, 2005.
- [31] D. Diaper, N. A. Stanton, *The Handbook of Task Analysis for Human-Computer Interaction*. Lawrence Erlbaum Associates, 2003.
- [32] X. Ferre, N. Juristo, A. M. Moreno, "Which, when and how usability techniques and activities should be integrated," in [4], pp. 173-200.
- [33] X. Ferre, N. Juristo, A. M. Moreno, "Framework for Integrating Usability Practices into the Software Process," in *International Conference on Product-Focused Software Process Improvement*, pp. 202-215, 2005.
- [34] W. Silva, N. M. C. Valentim, T. Conte, "Integrating the Usability into the Software Development Process - A Systematic Mapping Study," in *International Conference on Enterprise Information Systems*, pp. 105-113, 2015.
- [35] H. Fischer, "Integrating usability engineering in the software development lifecycle based on international standards," in *Symposium on Engineering Interactive Computing Systems*, pp. 321-324, 2012.
- [36] H. Fischer, K. Nebe, F. Klompaker, "A Holistic Model for Integrating Usability Engineering and Software Engineering Enriched with Marketing Activities," in *International Conference on Human-Computer Interaction*, pp. 28-37, 2011.
- [37] H. Fischer, "Towards a holistic tool for the selection and validation of usability method sets supporting human-centered design," in *DUXU/International Conference on Human-Computer Interaction*, pp. 252-26, 2013.
- [38] T. G. Gonçalves, K. M. Oliveira, C. Kolski, "HCI engineering integrated with capability and maturity models," in *27th Conference on l'Interaction Homme-Machine (IHM 2015)*, Toulouse, France, ACM, 2015.
- [39] ISO, "ISO 9241-11:1998 Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11: Guidance on usability," 1998.
- [40] D. L. Scapin, J. M. C. Bastien, "Ergonomic criteria for evaluating the ergonomic quality of interactive systems," *Behaviour & Information Technology*, 16, pp. 220-231, 1997.
- [41] C. Wohlin, P. Runeson, M. Host, M. C. Ohlsson, B. Regnell, A. Wesslén, *Experimentation in Software Engineering: An Introduction*. Kluwer Academic Publishers, Ed. 1, 2000.
- [42] J. Vanderdonck, *Guide ergonomique des interfaces homme-machine*. Belgique: Presses Universitaires de Namur, 1995.