Author version, published in: Computer Standards & Interfaces, 58, pp. 53-86, 2018.

Identifying HCI Approaches to support CMMI-DEV for Interactive System Development

Taisa Guidini Gonçalves, Káthia Marçal de Oliveira, Christophe Kolski

University of Valenciennes and Hainaut-Cambrésis, LAMIH CNRS UMR 8201
Le Mont Houy cedex 9 - 59313 Valenciennes, France
taisa.guidinigoncalves@etu.univ-valenciennes.fr, {kathia.oliveira, christophe.kolski}@univ-valenciennes.fr

Abstract

Software process capability maturity models are currently widely used in industry. To perform the practices defined in these models, software engineering approaches are applied. We have experimented the definition of a large number of methods, techniques, patterns, and standards for the conception, design, implementation, and evaluation of interactive systems focusing on Human-Computer Interaction (HCI) issues. Nevertheless, it is well-known that HCI approaches are not largely used in industry. In order to take advantage of the widespread use of capability maturity models, we have worked on the identification of appropriate HCI approaches for each practice of the engineering advocated by the most known model - the CMMI-DEV (Capability Maturity Model Integration for Development). By exploring the CMMI-DEV and the literature, we identified a set of HCI approaches for the development of interactive systems. Twenty HCI experts were interviewed for the validation and improvement of this initial set. As a result, we identified 14 HCI categories of approaches with examples of methods, techniques, patterns, and standards adequate for performing engineering practices of the CMMI-DEV when developing interactive systems.

Keywords: Capability Maturity Model Integration; CMMI-DEV; Human-Computer Interaction; Interactive systems.

1. Introduction

Software process capability and maturity (SPCM) models are nowadays well established in the industry [33] [128]. These models are a collection of software engineering best practices, organized in process areas, which help companies to improve their software process. A large number of official appraisals using these models indicate that software engineering practices are currently being used in industry. For instance, more than 10,000 official appraisals [19] using CMMI (Capability Maturity Model Integration) [18] are reported covering more than 80 countries. Other national SPCM models (such as the MR-MPS-SW Brazilian model [115], the MoProSoft Mexican model [89], and the Spanish maturity model [41]) are also being widely used in industry; for instance, there are more than 600 official appraisals on the Brazilian model created in 2005 [65]. To perform what is proposed in those models, software engineering approaches (methods, procedures, standards, tools, techniques, etc.) are applied. One can suppose that as software engineering practices are used, Human-Computer Interaction (HCI) approaches essential for the development of interactive systems are also used when necessary. Nevertheless, it is well-known that HCI approaches are not used in industry or are done so insufficiently [10] [48] [107].

Undoubtedly, HCI engineering is inherently related to software engineering whilst applying to the interactive system projects. Jokela and Lalli [63] point out, for instance, that several process areas from CMMI-DEV (Capability Maturity Model Integration for Development) [18] have a direct relationship with usability practices, and, therefore, HCI engineering. Helms *et al.* [52] argue that usability engineering and software engineering share common goals, such as: trying to understand customer and user needs; transforming needs into system requirements; designing to satisfy those requirements and testing to help assure their realization in the final product. Moreover, several works have discussed HCI life cycles and the integration of HCI and Software Engineering (SE) domains for perfecting / improving usable and useful systems [46] [62] [79] [81] [82] [83] [87] [97] [101] [102] [109] [112]. Considering that CMMI-DEV is widely used in industry, we believe that indicating which HCI approaches support the application of CMMI-DEV practices in the development of the interactive systems may favor a greater application of HCI issues in the industry. We considered any method, technique, standard or pattern for HCI to be an HCI approach. In this context, we raise the following question: what are the approaches that could concretely integrate CMMI-DEV process areas in interactive system development?

To answer this question, we have performed a study to identify which HCI approaches should support the engineering practices of CMMI-DEV in the development of interactive systems. Preliminary results were presented in [43] [44]. In [43], a work-in-progress paper for a conference on HCI; we presented a general view of the steps that we described to address this problem and some initial results. In [44] we presented general results for the requirements development area defined in CMMI-DEV.

This paper presents the complete study and its results in detail. To address the problem raised, we performed the following activities: (i) analysis of HCI literature and CMMI-DEV practices for engineering process areas (requirements development, technical solutions, product integration, verification, and validation); (ii) validation and improvement of the HCI approaches with experts; (iii) analysis and synthesis of the HCI approaches, and (iv) validation in practice. The main contribution of this paper, related to the existing literature is the proposition of HCI approaches for the practices of the five engineering process areas (requirements development, technical solution, product integration, verification, and validation) of CMMI-DEV validated by twenty HCI experts. The application of all these propositions in real projects is a long-term study. Working towards this goal, we present the first results related to some approaches proposed for requirements development.

In the next section and subsections, we briefly describe the main features of software process capability and maturity models and main related work. In section 3, we present our research methodology, giving an overview of all decisions taken to perform our work. Sections 4 to 8 describe how we performed each phase of our research methodology. Then, in section 9 we present a discussion of the threats to validity. Finally, in section 10 we present our final remarks and some planned future works.

2. Background

In the following sections, we describe the main elements of software process capability and maturity models. Then, we present the works related to this study.

2.1. Software Process Capability and Maturity Models

SPCM models aim to support organizations in defining an evolutionary improvement path from immature to disciplined processes, maturity processes with improved quality and effectiveness [18]. To that end, they are composed of software engineering best practices of effective processes for areas of interest (such as requirements, management, quality, etc.). In the last two decades, several capability and maturity models have been developed. Wangenheim *et al.* [128] identified 52 models that cover different domains (such as software engineering, e-commerce, security). Fifty of the 52 models are defined based on CMM [94]/CMMI [18] models.

CMM – Capability Maturity Model [94] is a process improvement model, defined by the SEI (Software Engineering Institute) during the 1990s as requested by the US Department of Defense. This Institute has developed different models for several disciplines (e.g. Systems Engineering, Software Engineering, and Software Acquisition) that describe a scalable improvement approach, enabling us to move from immature processes to mature and better processes [94].

CMMI – Capability Maturity Model Integration [18] was an initiative of members working in industry, the US government and the SEI, that represents an evolution of CMM models. The CMMI is composed of several models and provides best practices to help organizations to improve their processes. These CMMI models provide guidance (best practices) to use when developing processes but they are not processes or process descriptions. They are used for the implementation of any type of product (or system). It is however in the development and maintenance of software that it is most used (CMMI for Development, CMMI-DEV). Usually, CMMI-DEV is the basis for the definition of the software process to be used in the development/maintenance of a specific software system. CMMI-DEV is currently in version 1.3 [18]. Therefore, due to the importance of CMM/CMMI [128] and the reported use of CMMI in more than 80 countries [19], we chose to perform our study based on CMMI-DEV.

CMMI-DEV model components (see Figure 1) are grouped into three categories: (i) required – components (generic and specific goals) from this category are essential to achieving process improvement in a given process area; (ii) expected – these components (generic and specific practices) describe the activities that are important in achieving a required component; and (iii) informative – these components (sub-practices, example boxes, notes, references, sources, example work products, etc.) help users of the model to understand the required and expected components and give suggestions to apply the activities.

The core element of CMMI-DEV is the process area (see Figure 1 – e.g. Requirements development), that is a cluster of related practices in an area that, when implemented collectively, satisfies a set of goals considered important for making a significant improvement in that area.

A process area has 1 to 3 *Specific Goals* - SG (see Figure 1 – e.g. SG1 Develop Customer Requirements - described as stakeholder needs, expectations, constraints, and interfaces which are collected and translated into customer requirements). SG describes the unique characteristics that must be present to satisfy the process area. It is composed of *Specific Practices* - SP (see Figure 1 – SP1.1 Elicit Needs - elicit stakeholder needs, expectations, constraints, and interfaces for all phases of the product lifecycle) that describe the activities expected to result in achievement of the specific goals of a process area.

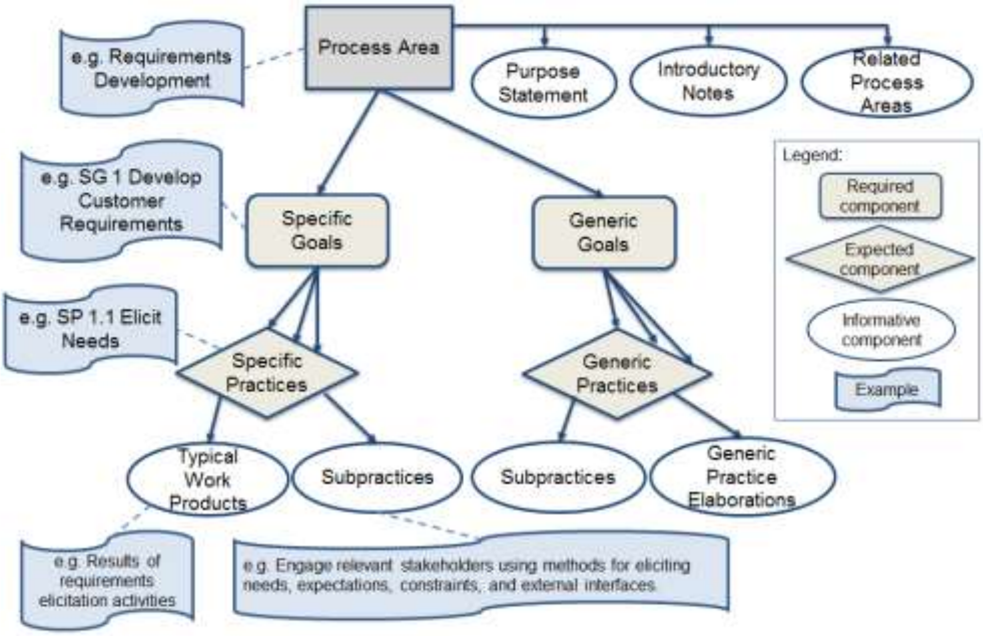


Figure 1. Components of CMMI (adapted from [18])

Generic goals and generic practices are also defined to be applied in all process areas. Moreover, CMMI-DEV uses the concept of levels to describe the evolutionary path for an organization that wants software process improvement. Two types of level are defined: capability level (the improvement is in an individual/or group of process areas) and maturity level (the improvement is through a successive set of predefined process areas). Table 1 presents the names of the different capability and maturity levels. The maturity level is the one used more often in industry and each level is composed of a set of process areas.

Table 1. Capability and Maturity levels of CMMI models

Levels	**Capability levels**	**Maturity levels**
Level 0	Incomplete	-
Level 1	Performed	Initial
Level 2	Managed	Managed
Level 3	Defined	Defined
Level 4	-	Quantitatively managed
Level 5	-	Optimizing

CMMI-DEV version 1.3 [18] contains 22 process areas organized into four categories: project management, process management, engineering, and support. According to CMMI-DEV, engineering process areas cover the development and maintenance activities that are shared across engineering activities. Since our focus is to support the development of interactive systems, we decided to concentrate our study on the process areas of the engineering category. Moreover, the other categories are more generic for any kind of system and support all process areas. The engineering category of CMMI-DEV is composed of five process areas: Requirements Development (RD), Technical Solution (TS), Product Integration (PI), Verification (VER), and Validation (VAL). In general, requirements are developed (RD) from customer needs, and design solutions are designed, coded and integrated (TS and PI). Products and product components are verified (VER) and validated (VAL) when necessary. With regard to the maturity levels (see Table 1), all engineering process areas (focus of this work) are placed in level 3 (defined).

Figure 2 presents specific goals (SG) and specific practices (SP) from all engineering process areas. Each practice is described, presenting its main goal and explaining what is expected to perform that practice. In some cases, informative components are proposed (see typical work products in Figure 1) even if others can also be applied (for instance, for this example storyboards [121] are also used). However, in most of the practices based on the description of what is expected, software developers choose some approaches (techniques, methods, standards, patterns) from the large software engineering workbench. For instance, to establish product and product component requirements (SP2.1 from Requirement Development in Figure 2), software engineers use software modeling approaches (such as business case analysis or suitable Unified Modeling Language diagrams) and traditional standards and guidelines (such as ISO 25000 [60]).

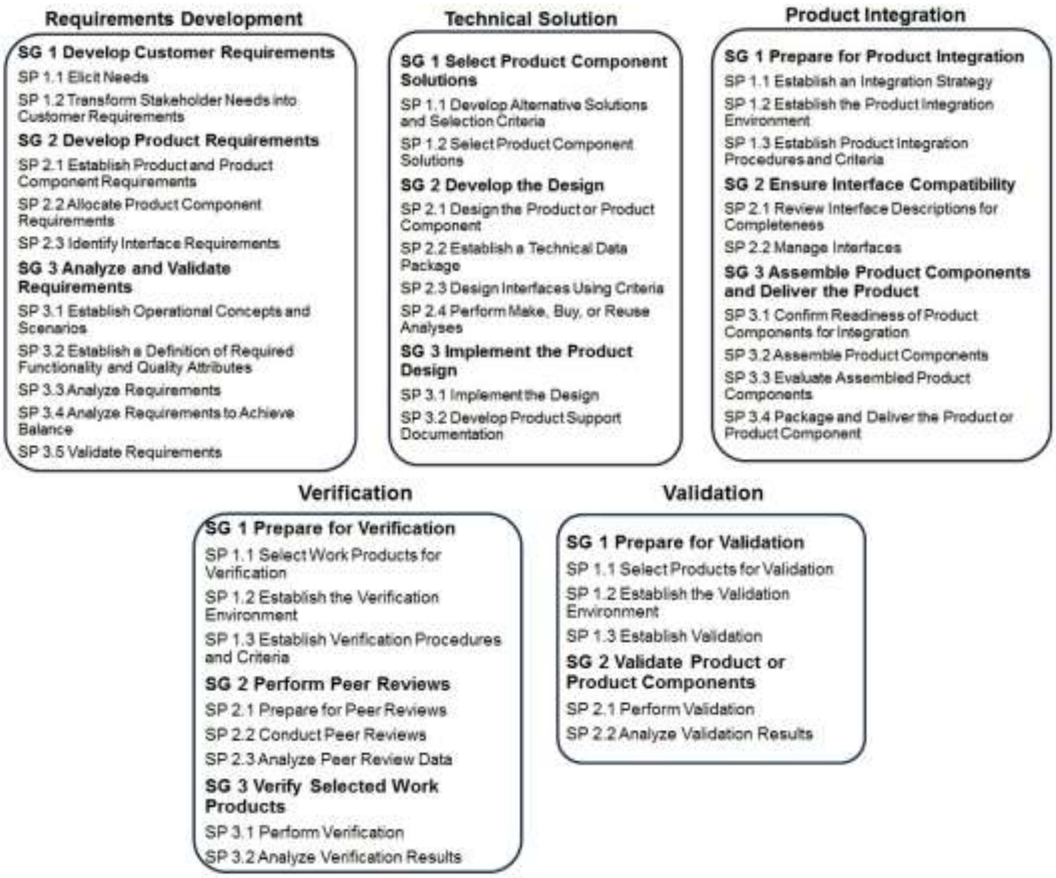


Figure 2. Process areas of the engineering category [18]

2.2. Related work

We found two works [87] [95] in literature with a similar purpose to ours.

Nogueira and Furtado [87], from a literature review, chose some techniques from HCI and used them in a case study. From this application, they indicate the use of these approaches to support four processes in the Brazilian model (MR-MPS-SW [115]) (requirements development, design and construction of the product, verification, and validation). This work is interesting and shows that it is possible to concretely suggest HCI techniques to support a generic SPCM. However, this proposition is based on the application of approaches in a specific case study (which means the techniques were probably chosen for that specific kind of application); it limits the example of techniques (for example, for verification and validation only one technique is suggested) and it targets a national SPCM model. Despite the fact that the Brazilian model claims to be compatible with CMMI-DEV, this work does not focus on all engineering process areas of CMMI-DEV, since it does not consider Product Integration, a process area that is part of the software development life cycle.

Peres *et al.* [95] proposed an initial study towards to a reference model for integrating agile methods and user experience (UX) in the software development cycle. This model is in line with CMMI-DEV, MR-MPS-SW, and ISO18529. The model focuses on Level 2 of CMMI-DEV by suggesting specific practices, recommendations, and techniques to support some areas from this level (project planning integrated with project monitoring and control; requirements management; process and product quality assurance; measurement and analysis). The idea of proposing techniques to support a process area of CMMI-DEV in the development of specific kinds of project (in this case UX projects) is similar to our work. However, they suggest techniques for the process areas as a whole and not for the practices to be applied, their focus is on agile projects, and for CMMI level 2 does not consider the engineering process areas (which are part of CMMI level 3). Moreover, this work is at a very initial stage and had no validation, being a simple proposition of the authors.

Besides these works, we found several others that are not similar but are related to our research. We grouped them into three groups: (i) some propositions for integrating HCI approaches in software development processes or standards; (ii) several usability capability and/or maturity models; and (iii) relevant systematic literature reviews about HCI integration with software engineering.

In the first group, we can quote five main works. Analyzing deeply these works and considering their goals, the software process activities that they support, the HCI approaches suggested, and the performed validation, we note some weakness (see Table 2). As can be observed in Table 2, only one work quoted HCI techniques to be applied in some phases of a software process (focusing on requirements and final evaluation) [36] [37]. Gross [46] proposed his own technique for user centered-design and its application in some phases of the software process. Both works are propositions of the authors with no validation. The other works had different goals, not being interested in the definition of techniques to support software process phases. Moreover, all of them focused on the definition of a software process not working with SPCM, which is more generic than a specific software process and are usually used as good practices in the definition of the software development process for specific projects.

Table 2. Analysis of related work

Reference	Proposal	Process activities/phases	HCI/usability techniques or methods	Validation	Main weakness
Gross [46]	This work proposed a user-centered process model considering existing process models (such as the waterfall model, the spiral model, the Unified Process, the star model and the standard process ISO 9241-210:2010).	- Understand and Define Users, Tasks, and Contexts - Specify System Requirements - Design User Tasks, and User Interactions - Develop the system (implementation and test of the system) - Evaluate the system	Defined a set of specific techniques based on Use-case 2.0. No technique proposed in literature was used.	Not validated	Use of the techniques defined by the authors and the proposition was not validated.
Fischer et al. [39]; Fischer [38]	These authors proposed the integration of usability engineering and HCI through the analysis of standards (ISO 12207 and ISO 9241-210), by defining a list of activities, artifacts, and correlations of HCI and software engineering.	- Requirements Analysis - Architectural design - Qualification testing	Suggested 15 usability methods that can be used.	Validated by usability experts (the number of experts was not presented)	The 15 usability methods are not presented. The paper only discusses the correlation of HCI and software engineering.
Nebe and Zimmermann [82]; Nebe and Zimmermann [83]; Nebe et al. [84]	These authors defined a general framework to integrate software engineering and usability engineering, going from standards to an operational process where close collaboration must be achieved between the two disciplines.	- Requirement analysis - Software specification - Software design and implementation - Software validation - Evaluation	No techniques are proposed.	Validated by interview with Usability Engineering experts (the number of experts was not presented)	The approach did not present HCI techniques.
Helms et al. [52]	This work proposed a usability engineering process model based on HCI and software engineering life cycles. This process should be instantiated according to the need of the company.	- Analyze - Design - Implement - Evaluate	Techniques: user/task model, usage scenarios, screen designs, lo-fi prototype, hi-fi prototype, global usability evaluation.	Validated with a case study	Quoted HCI techniques without specifying which phase they should be used for. Although the focus of the paper is the definition of a generic HCI software process.

Reference	Proposal	Process activities/phases	HCI/usability techniques or methods	Validation	Main weakness
Ferre et al. [36] [37]	These authors proposed a framework for integrating usability practices into the software process.	- Requirements elicitation, analysis, and negotiation - Requirement specification - Interaction design - Requirements validation - Usability evaluation	34 HCI techniques.	Not validated	The choice of techniques for each phase is defined based on the interpretation of the authors not being analyzed by others. The main focus is requirements and final evaluation.

In the second group, Jokela et al. [61] [64] had performed a survey resulting in eleven usability capability/maturity models. The authors claim that these usability models are "methods for developing user-centered design processes in companies in order to facilitate usability methodologies for creating usable products". Some models (ISO/TR 18529 [56], ISO/TS 18152 [57], UMM-P [27], HFIPRA [29]) use the format and requirements of the process assessment models used in software engineering (ISO 15504 [59]). The models are also based on ISO 13407 [55] (UMM-HCS [28], UMM-P [27], ISO/TR 18529 [56], DATech-UEPA [24], KESSU [62], HFIPRA [29]), ISO/TS 18152 [57]), or in previous version CMM [94] of CMMI [18] (Trillium [7], (HPA) [47]). Trillium [7] and (HPA) [47]. They use the structure of CMM/CMMI-DEV, that is process area, goals and practices, but they do not consider any process area or practice defined on CMM. They are well documented but are focused only on process management aspects (e.g. the inclusion of usability activities in a project plan, follow-up implementation of the plan during the project, etc.).

Recently, Lacerda and Wangenheim [69] performed a systematic literature review to identify capability/maturity models that focus usability engineering and that assist in the assessment process. They found fifteen usability capability/maturity models of which four models were also presented in [64], previously mentioned. According to the authors, although most of the models are in conformance with other models, such as CMMI or ISO/IEC 15504, they do not provide support to be applied in practice, it being necessary to seek other sources or make arrangements of different models and methods.

Table 3 presents a summary of all usability capability/maturity models quoted by [61] [64] [69]. The five more recent works that proposed usability capability/maturity models are:

- Kieffer and Vanderdonckt [66] proposed the STRATUS model composed of a questionnaire for cost-effectively assessing strategic usability. The aim of the model is to analyze and describe the current state of strategic usability according to five dimensions (usability awareness, usability expertise, usability resources, management of usability, and attitude towards usability (behavior)) and related indicators. The model proposes 3-level usability.
- Salah et al. [102] proposed a maturity model for integrating agile processes and user-centered design that is composed of process, practices and work products. The model [81] provides assessment tools and presents six maturity levels. This proposition was performed based on the usability maturity model proposed by [62] and the standard ISO 13407 [55].

Table 3. Analysis of usability capability/maturity models

Survey reference	Model	Domain	Validation
Jokela et al. [64]	Trillium [7]	Telecom Product Development	Not validated
	Usability Leadership Management Maturity (ULMM) [98] [99]	Usability	Not validated
	HumanWare Process Assessment (HPA) [47] [80]	Humanware	Not validated
	User Centered Design Maturity (UCDM) [31]	User-Centered Design	Not validated
	DATech-UEPA [24]	Usability engineering	Not validated
	Human-centred design – Process Capability Model (HCD-PCM design and HCD-PCM visioning), found in [64]	Human-centered design	Not validated
	Human Factors Integration Process Risk Assessment (HFIPRA), found in [64]	Human Factors	Not validated
Jokela et al. [64]; Lacerda and Wangenheim [69]	Usability Maturity Model: Human-Centeredness Scale (UMM-HCS) [28]	Usability	Not validated
	Usability Maturity Model: Processes (UMM-P) [27]; ISO/TR 18529 [56]	Usability	Not validated
	KESSU Usability Design Process Model [62]	Usability	Validated with case studies
	ISO/TS 18152 [57]	Human-system interaction	Not validated
Lacerda and Wangenheim [69]	Standardized Usability/User-Experience Maturity Model [76]	Usability/User-Experience	Not validated
	Human factors integration capability maturity model [30]	Human factors	Validated with experts
	Introducing usability engineering into the CMM model: an empirical approach [125]	Usability	Not validated
	Making user experience a business strategy [117]	User-Experience	Not validated
	Corporate user-experience maturity model [123]	User-Experience	Not validated
	Open source usability maturity model (OS-UMM) [97]	Open source usability	Validated with case studies
	Health Usability Model [116]	Usability in healthcare	Reviewed by experts (no information is provided in the paper)
	A Maturity Model for Integrating Agile Processes and User Centered Design [102] [81]	Agile and usability	Validated with experts
	UX Maturity Model [17]	User Experience	Not validated
	AGILEUX model [95]	Agile and User-Experience	Validated with experts
	STRATUS: a questionnaire for strategic usability assessment [66]	Usability	Validated with case studies

- Chapman and Plewes [17] created a model for assessing the level of UX maturity of organizations with the goal of identifying what the organization already knows/uses in terms of UX before adopting UX practices. To that end, they define indicators of maturity (e.g., timing of initial UX, availability of resources, leadership and culture) and their relationship with the respective UX maturity levels.
- Peres *et al.* [95], described in the beginning of this section; and
- Raza *et al.* [97] 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 (such as CMMI) for improving the usability of projects, but only a structure of evaluation.

According to [61] [64] [69], the main limitations of the models found in the surveys are:

- the lack of information on how the models were developed;
- the lack of validation of the models;
- the lack of guidance for the use of the models in practice; and,
- the unavailability of an evaluation process model.

We highlighted that all the propositions from Table 3 focused on the definition of usability maturity/capability models that do not consider the engineering process areas of CMMI-DEV.

In the third group, we included two systematic literature reviews (SLR)[1] [112] [53] related to the integration of SE and HCI domains.

In the first work, Silva *et al.* [112] executed an SLR where they identified, categorized and summarized technologies (method, technique, model, tool, approach, and other proposals created by the areas of HCI and software engineering) that have been used to improve the usability within software development process. The results show that several technologies support the improvement of usability and they can be integrated into the software process model of interactive applications. Although this work does not consider CMMI-DEV practices, it is directly related to our research.

The second work [53] performed a tertiary study[2] (an SLR of SLR) about agile software development. They identified several research areas related to agile development. One of them is related to the use of CMMI, but specifically with agile software development quoting two works [15] [111]. Chagas *et al.* [15] were interested in the characteristics of agile project management and, therefore, focused on a project planning process area from CMMI-DEV from the project management category. They concluded that the area "still lacks detail on how to perform software development activities, what techniques can be used to meet issues not directly addressed by agile methods without losing the desired agility, what tools can be used to facilitate the combination of approaches" [15]. Moreover, they recognize the lack of approaches to support the process. Silva *et al.* [111] evaluated and synthesized the results related to benefits and limitations of the use of the CMMI in combination with the agile software development. According to the authors, the companies have used agile methodologies to reduce their efforts to reach maturity levels 2 and 3 of CMMI. Although they indicate several benefits (such as improvements in organizational aspect, team and customer satisfaction, cost reduction, and process assimilation), they suggest that an in-depth analysis of specific process areas of CMMI can help to define proposals and guidelines to assist the combination with agile practices.

Despite the existence of these works, some authors show that HCI/usability approaches are not used or little used in industry. Bevan [10] argues that usability standards are not used in industry because of the complexity of their documentation, it not being easy for

[1] A systematic literature review [67] "is a means of evaluating and interpreting all available research relevant to a particular research question, topic area, or phenomenon of interest".

[2] A tertiary study "is a study that involves a review of existing secondary studies (such as systematic literature reviews and systematic mappings) that is expected to answer wider research questions and uses the same methodology as the systematic literature review" [53].

designers to use them. Hao and Jaafar [48] report a survey with Malaysian company 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 [107] investigates the importance of usability issues for small and medium size software companies in Germany. They concluded that they rarely used usability methods.

Regarding the use in industry, we argue that usability or HCI approaches should support the models already in practice. All usability capability models previously presented are still research projects not applied in industry. Moreover, they only deal with HCI issues (focused on usability), that means with no integration with the software engineering activities as defined by the classical SPCM. This scenario motivated us to work on the integration of HCI in an SPCM model largely used in industry: the CMMI-DEV.

3. Research Methodology for the Integration of HCI approaches in CMMI-DEV

Based on the background previously presented, we looked at how to integrate HCI with CMMI-DEV by the identification of HCI approaches that could be used while applying the specific practices of the process areas from engineering [43] [44] in the context of interactive systems. These approaches can be introduced as sub-practices or used to define potential work products for each specific practice. Our study followed four main phases presented in Figure 3.

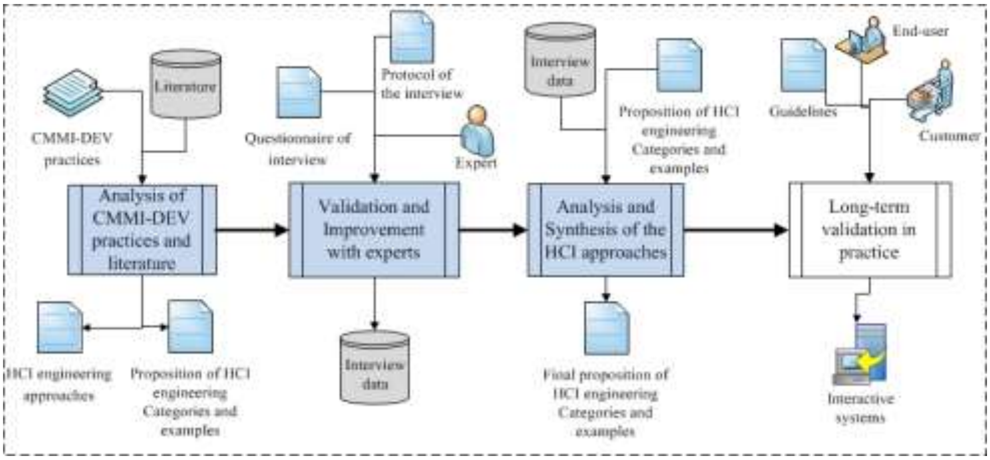


Figure 3. Research Methodology

In the first phase (Analysis of CMMI-DEV practice and literature), an in-depth analysis of the CMMI-DEV documentation (engineering process areas) was performed to identify where HCI approaches should be used to implement the practices. We also studied the HCI literature to identify the main HCI approaches. The results of this phase were a list of HCI approaches (methods, techniques, standards, and patterns) that were organized into categories and a proposition of HCI approaches to support the specific practices of the engineering process area from CMMI-DEV. This phase is presented in section 4.

Once we have the proposition, we should validate it using some evaluation approach. According to [51] [104], an evaluation can be classified into three types: author evaluation, domain expert evaluation, and practical setting evaluation (for instance, case studies). An author evaluation is conducted only by the authors of the proposition; the evaluation can be done based on their knowledge. In domain expert evaluation, the person responsible is an expert in the domain who is intended to improve the propositions; interviews, surveys, or simulated assignments are carried out in this type of evaluation. A practical setting evaluation is conducted through real activities where the proposition is used in a practical setting.

Since the authors reviewed together the propositions generated from the analysis of CMMI-DEV (author evaluation) and because the use of HCI in practice is not large (see section 2.2), we decided to start by doing an evaluation with HCI experts. Moreover, some studies also confirmed the reliability of expert judgment [68] [70] and several other studies in the same domain (see Tables 1 and 2) also used experts for validation of their proposition. We were not expecting all propositions to be perfectly correct, but thought that they could be used as a starting point for an evaluation by the experts and improved with other experts' suggestions.

We therefore planned the third phase (Validation and Improvement with Experts). To that end, we prepared a questionnaire (see Appendix A) where the experts were to answer about his/her level of agreement with the proposition using an ordinal scale (*I agree*, *I partially agree*, *I do not agree*). They were to give some justification when they answered one of the two last options (*partially agree*/ *I do not agree)*. Considering that we could have a lot of items to validate (since we associate HCI categories with CMMI practices) and that we would like also to improve the proposition, we decided to interview the HCI experts instead of simply asking them to answer the questionnaire. We set as a profile that the experts should have a Ph.D., experience in HCI and should be well-known in the HCI community (e.g., be program chair or a member of a program committee of HCI conferences). The detailed procedure and results of the interviews are presented in section 5.

At the beginning, in the first interviews with the experts, we were confronted with the following constraints: the experts imposed a limited time of two hours maximum for an interview and they indicated that they would not feel comfortable with recorded interviews. With these constraints and the kind of validation we wanted (a structured questionnaire of a set of propositions), we concluded we should do semi-structured interviews [71] where the questionnaire was used to support the discussion in each interview section. One could think about the use of the Delphi technique[3] [1] [73] [133], usually used to perform interviews. However this technique was not adequate for our case since (i) this technique is normally used for gaining judgments on complex issues where exact information is not available [73] [133]; that was not our case since the proposed HCI approaches were available and consolidated in the literature; (ii) usually with the Delphi technique we have a series of questionnaires [1] [73] [133] with different kinds of question, and in our case we had one questionnaire with 33 items to be validated (see Appendix A); (iii) the Delphi technique takes a lot of time [1] [73] [133] and requires several sections for the result be effective, and in our case we could only plan two-hour interviews.

With this in mind, we decided therefore to contact other experts and propose *Skype* meetings. In this way, some experts from other countries accepted our invitation and others proposed to have a time slot during a conference in HCI. This request fitted our two-hour interview restriction. Finally, twenty experts from five different countries accepted to take part in an interview.

[3] Delphi is "a method for structuring a group communication process so that the process is effective in allowing a group of individuals, as a whole, to deal with a complex problem" [73].

During the interviews, we explained that the questionnaire was a support for the interview, and that they should answer each question and justify, if necessary. We also explained that after all the interviews we would analyze the results seeking to better integrate the all the interview output. This implied our third phase (Analysis and Synthesis of HCI approaches).

In the third phase, we analyzed quantitatively and qualitatively all results of the interviews to make a synthesis of the proposition. Because the experts work with different kinds of interactive system (e.g. critical systems, serious game, interactive tabletops, etc.) and, therefore, have different experiences, we were not expecting to get a total consensus. Therefore, we planned first to have an overview of the evaluation proposition and then analyse all the comments of the experts to decide on the improvement in such a way as to integrate as much as possible all propositions respecting the opinion of the experts.

To have an overview of the evaluation of the propositions, we computed the mode and median value[4] for each practice of each process area. As the median is the value of the middle-ranked answer, we chose the 10th and 11th answer when we had 20 responses; the 9th answer for 19 responses, and the 9th and 10th for 18 responses). When mode and median are different, the decision was considered to be based on the mode, since the mode is the value of the most commonly occurring answer (previous analysis with this procedure can be found in [44]).

To analyse all comments in order to identify the propositions of improvements we considered the following:

- the full agreements (*I agree*) mean that the expert agrees with the proposition. When comments were included by the experts, they were also analysed;
- the partial agreements (*I partially agree*) mean that the expert agrees but he suggests some modification; for instance, inclusion of other examples or, splitting the category into two new more specialized categories or renaming the category to better express its meaning. When a modification was proposed by at least two experts the suggestion was accepted and the improvement was made; and,
- the disagreement (*I don't agree*) means that the expert proposes to exclude the proposition for the associated practice. Nevertheless, since we had very few disagreements, we decided to analyze the suggestions carefully and exclude only when we found enough confirmation that the proposition was not very adequate in the comments from other experts (who answered with a partial agreement).

We planned to analyse the comments in several steps. First, one of the researchers organized all responses in the same document, analyzed them and synthesized some improvements based on this analysis. After that, a second researcher reviewed all the experts' comments and the first synthesis of improvements, performing some modifications. Finally, all authors reviewed the comments and the synthesis of improvements in a three-hour meeting. The analysis and synthesis of the results are presented in sections 6 and 7 respectively.

We argue that the experts' opinion is quite reliable to accept that the HCI approaches help in the effectiveness of the practices. In addition, being aware that we could have a lot of propositions to validate since we were analyzing all practices of the engineering process, we accepted that applying all the approaches would be a *long-term validation in practice* (fourth phase). To confirm the effectiveness of each one, we should probably compare the results of the application using the approaches and not using the approaches themselves. We also recall that our motivation in this research was that by suggesting HCI approaches

[4] Mode and median are usually recommended when we work with nominal and ordinal scales (see, for instance, Fenton and Pfleeger [35] , p. 57)

for each practice of CMMI-DEV, we would instigate the use of these approaches in practice once CMMI-DEV is widely used in industry. Therefore, the validation of our proposition was also to verify if the practitioners use the approaches more when they are aware of which approach to use at each moment of the software process. Section 8 presents our results for this phase.

4. Analysis of CMMI-DEV practices and literature

Considering that CMMI-DEV is a generic model that can be used to support the development of any kind of system, the first phase focuses on analyzing CMMI documentation to identify where HCI approaches should be used to implement the practices. To that end, HCI literature was also studied. From the analysis of CMMI-DEV documentation (presented in section 4.1), we proposed a set of HCI categories and examples of HCI approaches identified from literature (presented in section 4. 2).

4.1. Analysis of the CMMI-DEV documentation

The analysis of CMMI-DEV documentation consisted of reading the description of the model components (see section 2.1), i.e., the required components (specific goals), expected components (specific practices) and informative components (sub-practices, example boxes, notes, references, sources, example work products; see Figure 4) for each process area.

While reading CMMI-DEV documentation, we looked for any citation of HCI issues. As stated by [63] the CMMI does not impose any requirements for usability; however, it includes 'hooks' where usability activities can be integrated. For the authors, the usability influences the process areas but it is optional. Our goal is to indicate in these 'hooks' explicitly which HCI approach may be used while developing interactive systems. For instance, when we found citations such as prototype or patterns, we analyzed them considering that we could use specific HCI approaches to produce them while developing interactive systems.

We started, therefore, by seeking any explicit citation (see Figure 5 (a)) that can be interpreted from an HCI engineering point of view, by looking for: (i) HCI keywords (e.g. external interface, end user, prototype); (ii) examples of techniques or methods of HCI placed in example boxes (e.g. end-user task analysis, HCI models); and (iii) example 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 towards the use of it. We classify this information as implicit citations (see Figure 5 (b)).

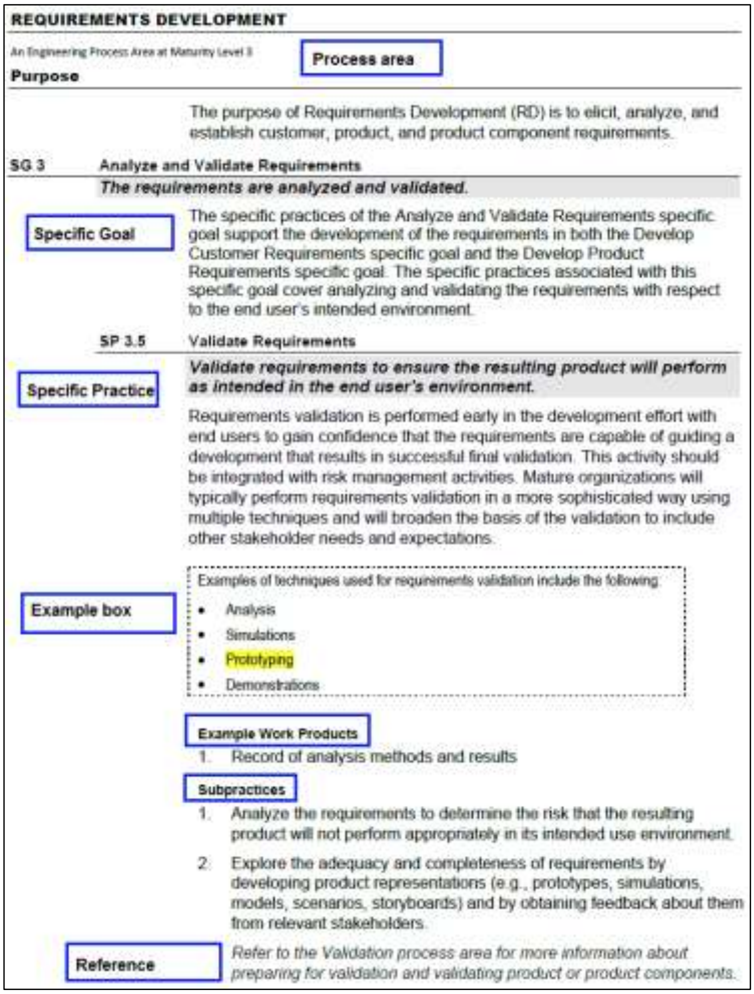


Figure 4. Example of the CMMI model components (extract from [18])

Table 4 presents some examples of explicit and implicit citations for the five analyzed process areas. For each specific practice, it shows the exact transcription of CMMI-DEV documentation where explicit or implicit citations were identified (underlined words). We can note ***explicit citations*** that mention HCI approaches, such as the examples of techniques for requirement development (RD - SP1.1 end-user task analysis, prototypes), criteria to evaluate the design (TS - SP2.1 usable), prototype use for product integration strategy (PI - SP1.1), and prototyping use for verification and validation of systems (VER - SP1.1 and VAL - SP1.1). ***Implicit citations*** are also presented: the identification of architecture patterns to develop the design of the product (TS - SP1.1); use of verification and validation criteria to assess the user interface (VER - SP1.3 and VAL - SP1.3). We analyzed 40 practices (10 of RD, 8 of TS, 9 of PI, 8 of VER and 5 of VAL) and we identified 27 practices (8 of RD, 5 of TS, 1 of PI, 8 of VER and 5 of VAL) that have some citation dealing with HCI issues.

SP 3.5	Validate Requirements

Validate requirements to ensure the resulting product will perform as intended in the end user's environment.

Requirements validation is performed early in the development effort with end users to gain confidence that the requirements are capable of guiding a development that results in successful final validation. This activity should be integrated with risk management activities. Mature organizations will typically perform requirements validation in a more sophisticated way using multiple techniques and will broaden the basis of the validation to include other stakeholder needs and expectations.

Examples of techniques used for requirements validation include the following:

- Analysis
- Simulations
- Prototyping
- Demonstrations

(a)

SP 2.1	Establish Product and Product Component Requirements

Establish and maintain product and product component requirements, which are based on the customer requirements.

...

Subpractices

...

3. Develop architectural requirements capturing critical quality attributes and quality attribute measures necessary for establishing the product architecture and design.

Examples of quality attribute measures include the following:

- Respond within 1 second
- System is available 99% of the time
- Implement a change with no more than one staff week of effort

(b)

Figure 5. Analysis of Requirements Development – explicit (a) and implicit (b) citation (extract from [18])

We did not find any explicit or implicit citation for:

- Two practices from requirements development (SP2.2 and SP2.3) – these practices are more related to functional aspects of the system. SP2.2 refers to the allocation of functional requirements to software components. SP2.3 is related to the internal interface between functional components not associated to HCI itself.
- Three practices from Technical Solution (SP2.2, SP2.3, and SP2.4) - The practice SP2.2 refers to establishing a technical data package for the project. SP2.3 refers to the interface between two functional components. SP2.4 is related to developing criteria for the reuse of product component designs and conducting analysis designs to determine if product components should be developed, reused, or purchased. The decision-making is based on criteria or a specific approach of the organization.
- Almost all practices of Product integration (we only found citations for one SP1.1, we did not find any for the other eight practices) – the scope of this process area is to achieve complete product integration through progressive assembly of product components (i.e., service, service systems and their components) according to defined strategy and management of the internal and external interface between these product components. In this way, we found citations only in the definition of the strategy to perform the product integration. All the other practices are concerned with the integration of the product components.

Table 4. Examples of the CMMI-DEV analysis

	Specific Practice	HCI Information	Citation
RD	**SP1.1:** Elicit stakeholder needs, expectations, constraints, and interfaces for all phases of the product lifecycle.	*Subpractice 1:* "Engage relevant stakeholders using methods for eliciting needs, expectations, constraints, and external interfaces." *Examples of techniques:* "Questionnaires, interviews, and scenarios obtained from end users", "end-user task analysis" and "prototypes and models."	Explicit
	SP1.2: Transform stakeholder needs, expectations, constraints, and Interfaces into prioritized customer requirements.	*Subpractice 1:* "Translate stakeholder needs, expectations, constraints, and interfaces into documented customer requirements." *Subpractice 2:* "Establish and maintain a prioritization of customer functional and quality attribute requirements." *Example Work Products:* "Prioritized customer requirements"	Implicit
TS	**SP1.1:** Develop alternative solutions and selection criteria.	*Subpractice 4:* "Identify reusable solution components or applicable architecture patterns."	Implicit
	SP2.1: Develop a design for the product or product component.	*Subpractice 1:* "Establish and maintain criteria against which the design can be evaluated." *An example of quality attribute:* "Usable".	Explicit
PI	**SP1.1:** Establish and maintain a product integration strategy.	*Additional information:* "A product integration strategy addresses items such as: using models, prototypes, and simulations to assist in evaluating an assembly, including its interfaces."	Explicit
VER	**SP1.1:** Select work products to be verified and verification methods to be used.	*Subpractice 4:* "Define verification methods to be used for each selected work product." *Additional information:* "Verification for systems engineering typically includes prototyping, modeling, and simulation to verify adequacy of system design (and allocation)."	Implicit / Explicit
	SP1.3: Establish and maintain verification procedures and criteria for the selected work products.	*Subpractice 2:* "Develop and refine verification criteria as necessary." *An example of a source for verification criteria:* "Standards."	Implicit
VAL	**SP1.1:** Select work products to be verified and verification methods to be used.	*Subpractice 4:* "Select the evaluation methods for product or product component validation." *Examples of validation methods:* "Discussions with end users perhaps in the context of a formal review, Prototype demonstrations."	Explicit
	SP1.3: Establish and maintain procedures and criteria for validation.	*Subpractice 2:* "Document the environment, operational scenario, procedures, inputs, outputs, and criteria for the validation of the selected product or product component." *An example of a source for validation criteria:* "Standards."	Implicit

4.2. Identification of HCI categories and approaches

After identifying all citations, we organized them separately to identify the main approaches related to HCI engineering and group them into HCI categories [43] [44]. The category names were proposed based on the information collected from the literature. Figure 6 presents the main keywords of the explicit and implicit citations found, indicating which one helped in the identification of each defined HCI category.

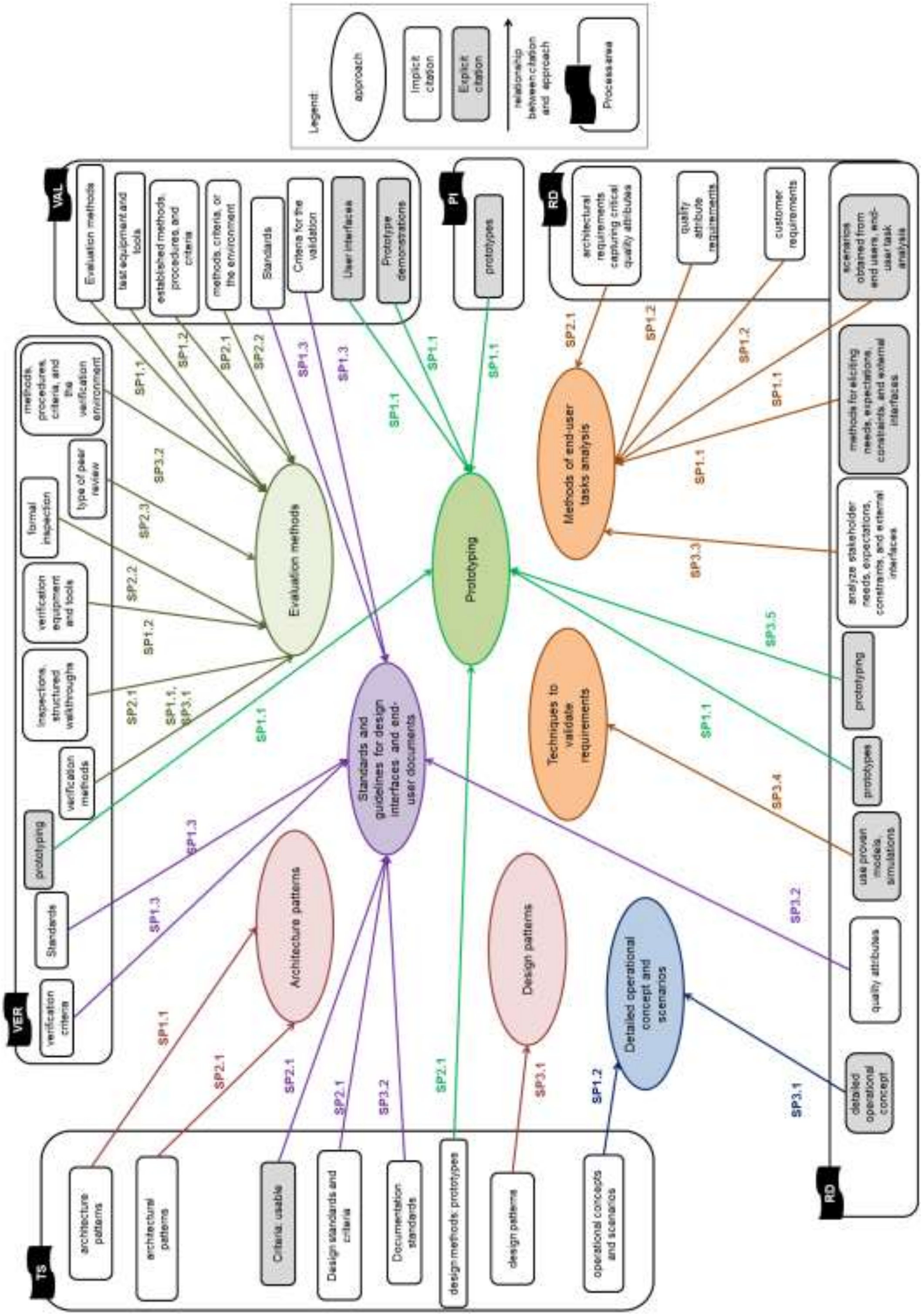


Figure 6. Analysis of the citations (implicit and explicit) to all process areas

From the analysis of all citations in the RD practices, we identified five HCI categories [43] [44]: (i) *methods of end-user tasks analysis*, for all citations that mention methods or the need of the analysis about the interaction with users (e.g. methods to elicit needs, scenarios obtained from end-users, etc.); (ii) *detailed operational concept and scenarios*, identified in the practice 3.1 aimed to "establish and maintain operational concepts and associated scenarios" [18]; (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 of techniques (such as simulation) and implicit citation of requirements validation; *(v) prototyping*, for any mention of prototypes in any practice.

For TS and PI practices, 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*, implementing design patterns to develop the HCI design of the product.

Finally in analyzing citations for Verification and Validation, one new category was identified: *evaluation methods*, for all kinds of evaluation techniques and methods used for verification, validation, and testing, such as peer review, inspection, and tests. Since the prototype in this analysis was related to the final validation, we split the category Prototyping into two: *Prototype for HCI requirements*, which could include prototypes in papers, mockups etc.; and *Functional Prototype to validate HCI*, to represent the executable prototypes. After this first analysis, we collected examples from the literature of HCI approaches (methods, techniques, patterns, and standards) for all categories. Following the software engineering classical classifications, we refined the category of evaluation methods in two groups: *Evaluation methods for HCI review*, to include techniques such as inspections, reviews, and so on; and, *Evaluation methods for HCI verification tests*, to include all kinds of test.

With the identified categories, we examined the literature to identify examples of HCI approaches that can be applied with any software development process and for several types of interactive system. Table 5 presents the ten HCI categories defined and the examples (HCI approaches). Table 6 presents the categories which could be applied to support CMMI-DEV practices in the interactive system development. Each one of the propositions (HCI category for each specific practice) constitutes the main result of this phase that was next used in interviews with experts. It can be seen in Table 6 that we obtained a total of 33 propositions for 27 specific practices. In the next phase of our methodology (see the following section) this set of HCI categories and all these propositions are validated and improved by experts, resulting in a final set of categories and propositions.

Table 5. HCI Engineering approaches to support CMMI-DEV

HCI Category	Examples
Task Analysis Methods for HCI	CTT (Concur Task Tree) [93]; K-MAD (Kernel of Model for Activity Description) [72]; HTA (Hierarchical Task Analysis) [72]; SADT (Structured Analysis and Design Technique) [100]; GTA (Groupware Task Analysis) [126].
Prototype for HCI requirements	Rapid Prototyping [5]: (i) Offline techniques: Paper and pencil (paper sketches, storyboards), Mockups, Wizard of Oz, Video prototyping; (ii) Online techniques using software tools: Non-interactive simulations, Interactive simulations, Scripting languages.
Operational Concepts and Scenarios Specification for HCI	Context awareness [21] [74]; Adapting to context [21] [74]; User profile [21] [74]; Persona [21] [74]; Use cases [18].
Standards and Guidelines for design and documentation of HCI	Ergonomic Criterion [106]; ISO/IEC 9126-1 [58]; ISO 9241-11 [54]; ISO/IEC 25000 [60].
Techniques to validate HCI requirements	ProtoTask (K-MAD) [13]; Task Model Simulator (CTT) [92]; Focus Group to validate requirements [85].
Architecture Patterns for HCI	Arch Model [4]; PAC (Presentation-Abstraction-Control) Model [23]; MVC (Model-View-Controller) Model [42].
Design patterns for HCI	Pattern for Interaction Design [11]; Pattern Languages in Interaction Design: Structure and Organization [124]; Designing interfaces: Patterns for Effective Interaction Design [119].
Functional Prototype to validate HCI	Iterative and Evolutionary Prototypes [5]: User interface toolkits, User interface builders, User interface development environments.
Evaluation methods for HCI verification tests	Usability tests [25] [110]: Exploratory tests, Assessment tests, Validation or verification tests, Comparison tests; Validation by HCI expert(s).
Evaluation methods for HCI review	Heuristic evaluation [20]; Cognitive walkthrough [20]; Groupware walkthrough [20].

Table 6. HCI Categories x CMMI-DEV Practices

	RD								**TS**					**PI**	**VER**								**VAL**				
	SP1.1	SP1.2	SP2.1	SP3.1	SP3.2	SP3.3	SP3.4	SP3.5	SP1.1	SP1.2	SP2.1	SP3.1	SP3.2	SP1.1	SP1.1	SP1.2	SP1.3	SP2.1	SP2.2	SP2.3	SP3.1	SP3.2	SP1.1	SP1.2	SP1.3	SP2.1	SP2.2
Task Analysis Methods for HCI	•	•	•			•																					
Prototype for HCI requirements	•							•			•			•													
Operational Concepts and Scenarios Specification for HCI				•						•																	
Standards and Guidelines for design and documentation of HCI					•						•		•				•								•		
Techniques to validate HCI requirements							•																				
Architecture Patterns for HCI									•		•																
Design patterns for HCI												•															
Functional Prototype to validate HCI														•	•								•				
Evaluation methods for HCI verification tests															•	•					•	•	•	•		•	•
Evaluation methods for HCI review																		•	•	•							
Total of propositions - 33	2	1	1	1	1	1	1	1	1	1	3	1	1	2	2	1	1	1	1	1	1	1	2	1	1	1	1
	9								**7**					**2**	**9**								**6**				

5. Validation and Improvement by Experts

In the second phase, interviews with HCI experts were performed in order to evaluate this proposition and improve it, modifying when necessary, including new examples or new

HCI approaches. This phase of our research had two main goals: (1) to *evaluate* if the proposition previously defined was adequate to be used in the implementation of the correlated practice of CMMI-DEV, and (2) to *improve* the propositions with new examples in the categories or new categories when judged necessary by the experts.

5.1. Planning the validation

As previously presented (section 3), the validation with the experts was planned considering two main constraints: doing two-hour interviews and not recording them. To ensure that the experts had the necessary background, we selected only professionals who had experience (academic and/or industrial) in the HCI domain and had a Ph.D. degree in the HCI domain, Computer Science/HCI domain or Software Engineering/HCI domain. Most of the experts were selected from the research contacts of one of the authors. This author has more than thirty years of experience in HCI, and has participated in numerous conference program committees, journal reviews, and project coordination. Experts from this list suggested other HCI experts. All experts have reputation recognized by HCI community and representativeness in different countries (observed by their active participation in conferences committees and projects financed by national research agencies).

Table 7 presents the background and origin (country) of the 20 experts who participated in the interviews. We note that on average they have 19 years of experience (from 7 to 40 years). The experts are of different nationalities: 12 from France, 5 from Brazil, 1 from Belgium, 1 from Tunisia, and 1 from Algeria. All of them have academic experience (teaching and academic projects), 14 have experience in industry, and 7 come from the software engineering domain as well as HCI.

As can be noted in Table 7, twenty experts were interviewed in total. We noted in the literature that studies with experts go from 11 experts [26] to 30 experts in [32] and samples of twenty experts are relevant to gain expert feedback [6]. Based on these studies, we considered our sample of experts as acceptable to continue to the next phase of our research methodology.

To support the interviews, we elaborated a specific questionnaire (see Appendix A). The questionnaire was composed of some introductory notes about the study, an overview of the CMMI process areas and their main components (goals, specific goals, work products and sub-practices), personal information (name, education, profession, practical experience in industry and number of years working on HCI considering academic and industrial experience), and all the propositions organized by goals and practices of CMMI (see an extract of this part in Figure 7). Since the questionnaire was intended to be the thread of the interviews, it was elaborated in three languages which could be used in the interviews: English, French and Portuguese. For each practice, the expert was asked if they “Agree”, “Partially agree” or “Do not agree” with each proposition, i.e., if the proposed HCI approach supports the related practice. This scale was only used as a starting point for the discussion of each proposition. The expert was motivated to explain and justify his/her answer, mainly for the two last points of the scale.

Table 7. List of Experts

Expert	Background			Origin	Interview	
	Time work in HCI (years)/ Experience	PhD domain	Current interest in interactive systems		Duration	Type
E1	13	HCI	Methods and models for HCI design and evaluation	France	01h30	In person
E2	25	HCI	Tools for the design, realization and evaluation	France	00h55	In person
E3	8	HCI	Agent-based architecture models and HCI evaluation	France	01h00	In person
E4	8	SE-HCI	Interaction and Automatic Reasoning	France	00h50	In person
E5	25	SE-HCI	Methods and tools of systems engineering	France	01h15	In person
E6	26	HCI	HCI	France	00h50	In person
E7	27	SE-HCI	SE and HCI	Belgium	00h50	In person
E8	20	HCI	HCI	Brazil	02h17	Video conference
E9	10	HCI	HCI	Brazil	00h40	Video conference
E10	25	HCI	HCI	France	01h00	In person
E11	20	SE-HCI	User Interfaces Plasticity, Creativity Support Tools, and Persuasive Technology	France	01h45	In person
E12	40	SE-HCI	Innovative interfaces, mobility	France	01h30	In person
E13	12	SE-HCI	Quality of Human-Computer Interfaces	France	00h53	In person
E14	7	SE-HCI	HCI	France	01h00	In person
E15	10	HCI	HCI	Brazil	01h03	Video conference
E16	30	CS-HCI	Interactive critical systems	France	01h36	Video conference
E17	27	CS-HCI	HCI design, Ubiquitous computing	Tunisia	01h26	Video conference
E18	21	CS-HCI	Semiotic engineering, evaluation and design of interfaces	Brazil	01h39	Video conference
E19	10	CS-HCI	Organizational Semiotics, Culture and Values in design	Brazil	01h03	Video conference
E20	27	CS-HCI	Service Design, Ubiquitous Computing, SOA	Algeria	01h50	In person

CS = Computer Science, SE = Software Engineering and HCI = Human-Computer Interaction.

5.2. Performing the Interviews

The interviews were performed as face to face meetings (either in person or by video conference). Each interview took on average slightly more than one hour (we had 24 hours and 52 minutes of interview in total), 13 experts were interviewed in person and 7 by video conference (see Table 7). Figure 8 presents an activity diagram that represents how the interview was performed. We started by presenting the goal of the study and explaining CMMI-DEV in general. Then we asked for their personal information. After that, we "walked through" the questionnaire and for each practice we explained the purpose and asked for their opinion about the proposed HCI approach associated with that practice.

When they partially agreed or disagreed, they were asked to justify with a description in the justification column.

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**	
Technical Solution *SG 1 Select Product Component Solutions* Product or product component solutions are selected from alternative solutions.	*SP 1.1 Develop Alternative Solutions and Selection Criteria* Develop alternative solutions and selection criteria.	*Architecture Patterns for HCI* Examples: • MVC (Model-View-Controller) Model [42] • PAC (Presentation-Abstraction-Control) Model [23] • Arch Model [4]		X		"To complete with Dialog Interaction Model: MOLIC, User Action Notation, Task-Action Grammar. To include the "Prototypes" category."
Requirements Development *SG 3 Analyze and Validate Requirements* The requirements are analyzed and validated.	*SP 3.1 Establish Operational Concepts and Scenarios* Establish and maintain operational concepts and associated scenarios.	*Operational Concepts and Scenarios Specification for HCI* Examples: • Context awareness • Adapting to context • User profile • Persona • Use cases			X	"The Scenarios are made at the beginning."

Figure 7. Extract of a filled questionnaire

At the same time, the interviewer took notes of verbal observations/explanations given by experts during the interview and if necessary, the interviewer consulted the CMMI-DEV document for the specific practice in evaluation and presented the information to the expert. Figure 7 presents an extract of a form filled in by one expert during an interview.

All the information collected was registered in the questionnaire by the expert, when present in person, or by the interviewer and/or the expert when the interview was carried out by video conference. Moreover, as previously mentioned, the interviewer took informal notes of interesting comments made by the experts (references, sites to be consulted later, example of tools, etc.).

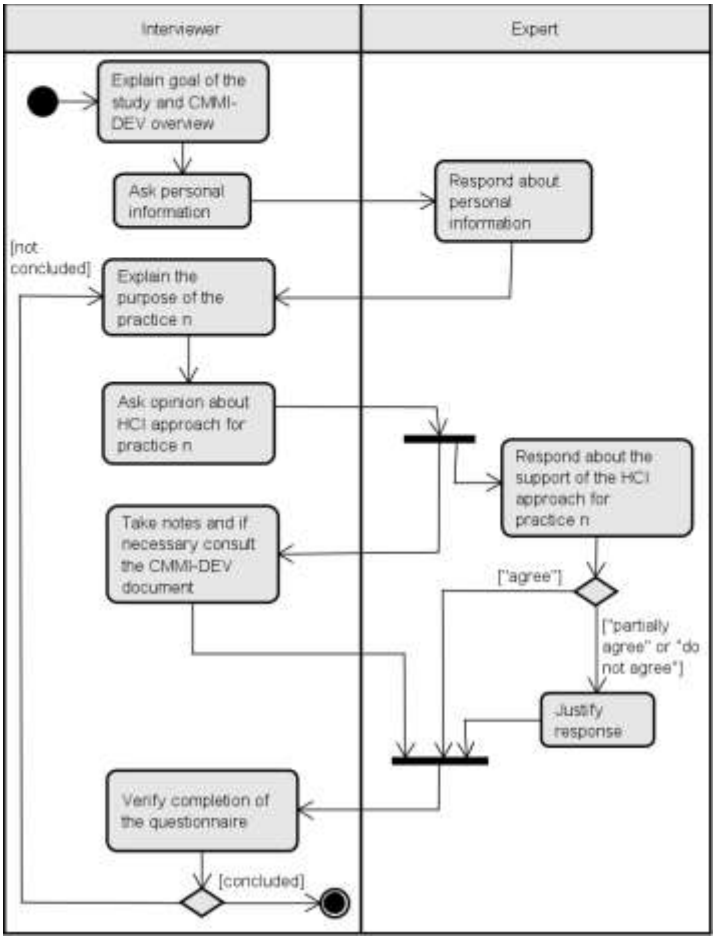


Figure 8. Interview sessions (adapted from [44])

6. Analysis of HCI Approaches

We started our data analysis by looking at the general results for all practices for the five process areas (RD, TS, PI, VER, VAL). As presented in Table 8, we had 33 propositions to be evaluated and improved by experts. Considering that we had 20 experts, we expected 660 responses to analyze. However, three experts did not give their opinion for all the propositions because they considered they did not know enough about them. Expert E4 did not evaluate proposition SP3.4 of RD; expert E10 did not evaluate propositions SP1.2 for VAL and VER, and proposition SP3.1 of TS; and expert E18 did not evaluate proposition SP3.1 of TS. As consequence, we got 655 responses to analyze (see Table 8). Figure 9 presents the detailed results of each process area considering the three possible answers. We note that Validation (VAL) and Product Integration (PI) are the process areas with the fewest (2/119 – 2%) and the most (11/40 – 27%) disagreements respectively. We also note that we got only 8% of disagreement in the whole evaluation.

Table 8. General results about experts' level of agreement by process area

	RD	TS	PI	VER	VAL	TOTAL
# Expected answer (based on each practice associate to each category)	180 (9 * 20)	140 (7 * 20)	40 (2* 20)	180 (9 * 20)	120 (6 * 20)	660 (33 * 20)
# Found answer	179	138	40	179	119	655 (100%)
# "Agree"	85	79	23	122	76	385 (59%)
# "Partially agree"	82	42	6	46	41	217 (33%)
# "Do not agree"	12	17	11	11	2	53 (8%)

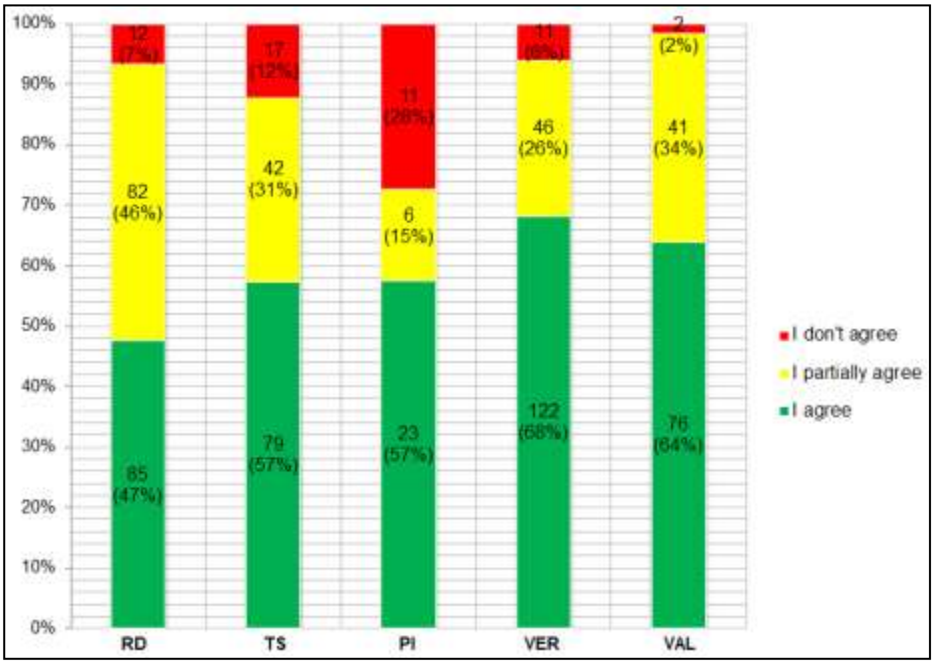


Figure 9. Detailed results by process area

As presented in section 3 (Research Methodology), to have an overview of the evaluation of the propositions, we computed the mode[5] and median values for each practice of each process area and took a decision (Table 9). We recall that when mode and median are different, the decision was considered to be based on the mode, since the mode is the value of the most commonly occurring answer. When the practice has two values for the mode, the decision was based on the median. We note in Table 9 that we have 25 agreements, 7 partially agreements and only 1 disagreement (that probably means this proposition should be eliminated). As a consequence, we concluded that in general our propostion was considered acceptable and improvements should be carried out based on the comments of the experts.

[5] Mode is "the value of the most commonly occurring item" (Fenton and Pfleeger [35], p. 66).

Table 9. Mode and Median of the results (Agree - A, Partially agree - PA, Don't agree - DA)

	HCI Approaches	Practice	Mode	Median	Decision
RD	Task Analysis Methods for HCI	SP1.1	PA	PA	PA
		SP1.2	PA	PA	PA
		SP2.1	A	A	A
		SP3.3	PA	PA	PA
	Prototype for HCI requirements	SP1.1	A	PA and A	A
		SP3.5	A	PA and A	A
	Operational Concepts and Scenarios Specification for HCI	SP3.1	A	PA and A	A
	Standards and Guidelines for design and documentation of HCI	SP3.2	A	A	A
	Techniques to validate HCI requirements	SP3.4	PA and A	PA	PA
TS	Architecture Patterns for HCI	SP1.1	A	A	A
		SP2.1	A	A	A
	Operational Concepts and Scenarios Specification for HCI	SP1.2	PA	PA	PA
	Prototype for HCI requirements	SP2.1	A	A	A
	Standards and Guidelines for design and documentation of HCI	SP2.1	A	A	A
		SP3.2	PA	PA	PA
	Design patterns for HCI	SP3.1	A	A	A
PI	Prototype for HCI requirements	SP1.1	DA	PA	DA
	Functional Prototype to validate HCI	SP1.1	A	A	A
VER	Evaluation methods for HCI verification tests	SP1.1	A	A	A
		SP1.2	A	A	A
		SP3.1	A	A	A
		SP3.2	A	A	A
	Functional Prototype to validate HCI	SP1.1	A	A	A
	Standards and Guidelines for design and documentation of HCI	SP1.3	A	A	A
	Evaluation methods for HCI review	SP2.1	A	A	A
		SP2.2	A	A	A
		SP2.3	A	A	A
VAL	Evaluation methods for HCI verification tests	SP1.1	A	A	A
		SP1.2	A	A	A
		SP2.1	A	A	A
		SP2.2	A and PA	A and PA	PA
	Functional Prototype to validate HCI	SP1.1	A	A	A
	Standards and Guidelines for design and documentation of HCI	SP1.3	A	A	A

To define the improvement, we analyzed in detail the quantitative and qualitative results for each practice. To support this analysis, the justifications were organized in a single form for each practice and associated approach (see an example in the next section – Table 10). With this table and following our research methodology (see section 3), we analysed the results of each process area in an integrated way (quantitative and qualitative analysis) for each process area as presented in the following sub-sections.

6.1. Quantitative and qualitative analysis for Requirements Development (RD)

Figure 10 presents the details of the quantitative results for RD process area, considering the practices and the proposed HCI categories.

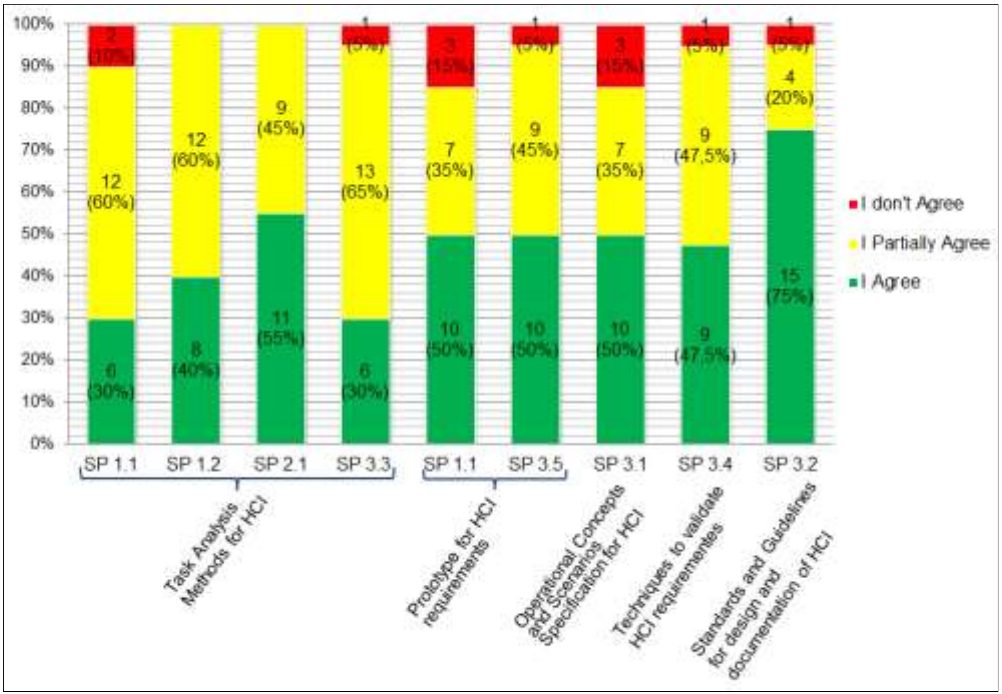


Figure 10. Quantitative results for Requirement Development (category and practice)

Five HCI categories were proposed for RD [44] and for these categories, we observed that:

- *Task analysis Methods for HCI* category was considered adequate for practices SP1.2 and SP2.1 with no disagreement to the propositions. However, several modifications were proposed since we had several partial agreements - 12 (60%) for SP1.2 and 9 (45%) for SP2.1. For practices SP1.1 and SP3.3, we note that only 6 experts agree with the propositions for each practice, and there were 3 disagreements. All the other evaluations were partial agreements.
- The *Prototype for HCI requirements* category had half (10) of the experts who completely agreed with its application for practices SP1.1 and SP3.5, which means it is a good proposition. However, it had 3 (15%) disagreements for SP1.1 which requires a better analysis;

- *Operational Concepts and Scenarios Specifications for HCI* also had 10 full agreements against 3 (15%) disagreements;
- *Techniques to validate HCI requirements* category can be considered adequate for SP3.4 since we had only one disagreement. However, 9 partial agreements indicate that improvements should be made; and,
- *Standards and guidelines for design and documentation of HCI* category obtained the greatest amount of agreements for SP3.2 (15 experts - 75%).

As previously mentioned, all justifications were organized in a single form that supports our analysis. The comments of each expert were transcribed exactly from their questionnaire to the column in which the answer on the ordinal scale was given. We used quotes when the comment was written by the experts on their forms, and brackets when the notes were introduced by us in the justification column during the interview. Table 10 presents an example of this form for the justifications of answers for the association of *Task analysis methods for HCI* with SP1.1 and SP1.2. Another example for RD can be found in [44].

With this form available, one of the authors first analyzed all comments. We observed that, for SP1.1, 6 experts agreed, 12 experts partially agreed and 2 experts disagreed. For SP1.2, 8 experts agreed and 12 experts partially agreed with the propositions. In general, the comments indicate that this category was not enough for the practices, and others should be used. For the two disagreements to SP1.1, the experts suggest that elicitation techniques should be used before task modeling methods. For the partial agreement of both practices, the experts suggested some elicitation techniques.

Thus, the suggested elicitation techniques were organized separately in a different document as presented in Table 11. First, we listed all techniques identifying which expert proposed each technique. Then, based on literature [21] [74], the techniques were grouped into two different categories: *Techniques to identify user needs* and *Techniques to identify user and organizational requirements*. Analyzing the comments, we identified that both categories of techniques are adequate to support SP1.1; and only the second category was adequate for SP1.2 that focuses on the specification of requirements.

While analyzing the techniques and considering our notes from the interviews, we observed that the approaches cited in the category *Task Analysis Methods for HCI* are related only to task modeling, and that task analysis is an approach to identify users and organization requirements suggested by 2 experts (E16 and E17 for SP1.1). As a consequence, we decided to rename the category *Task Analysis Methods for HCI* as *Task Modeling*.

In the final meeting, we re-analyzed the comments and confirmed the need for the creation of the two categories for the identification of 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 both practices. For SP1.1 two experts disagreed with the use of task modeling approaches. One suggested eliminating it, but another suggested only integrating other approaches. Analyzing the 12 partial agreements from both practices, we observed that, in general, 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 partial agreements that only suggest new approaches, and several agreements (6 for SP1.1 and 8 for SD1.2), we decided to keep this category to support both practices.

Table 10. Example of analysis to RD SP1.1 and RD SP1.2

Category	Task Analysis Methods for HCI	**Practice**	SP 1.1
I agree	*E1, E2, E6, E13, E20, E5- [To include a new category with elicitation techniques.]*		
I partially agree	*E3- "It's a little bit strange to conceive the system task without having a clear idea about supported features." [It is not enough. The expert suggests defining a new category Interview.]* *E4- "I think that this method is a help or a communication support for the SP1.1." [It is not enough. The expert suggests techniques for description of requirements.]* *E7- "Requirement engineering is another field. So, RE methods should be used here. But task models do contribute, although they don't express requirement." [It is not enough. The expert proposes to include VOLERE, RESCUE, and methods in ISO 24744.]* *E8- "To add other representations of tasks: scenarios, persona, storyboard (descriptive representations). The task model represents the HOW, other techniques represent WHAT. I suggest a new category with field study, interviews, brainstorming."* *E10- "Other steps and methods are required between obtaining needs and analysis tasks."[The expert suggests including personas and scenarios.]* *E11- "Possible inappropriate use of the model. I suggest the inclusion of Questionnaires, Focus Group, Scenarios, and Personas."* *E14- "I should clarify which task models are based on scenarios" [The expert says that is a step before and suggest including Scenarios, Focus Group, and Questionnaires.]* *E15- "I suggest the inclusion of techniques to requirements elicitation: Brainstorming, Questionnaires, Interview, Observation, and Ethnography. Also, to include Scenarios."* *E16- "Before the model we need to do the tasks analysis. I suggest the inclusion of a new category (elicitation methods: questionnaires, observation, …) for this."* *E17- "It does not integrate the contextual aspect. I suggest observation of the work environment."* *E18- "First of all makes scenarios to validate the understanding of needs with users. Also, use personas."*		
I don't agree	*E9- "Task Analysis is more a modeling activity than an elicitation activity."* *E12- "Customer expressed through the prototype verification." [Use informal techniques.]*		
Conclusion	Replace the category with the two specialized categories: *Techniques to identify user needs* (proposed E5, E8, E11, E3, E4, E14, E15, E19); and *Techniques to identify user and organizational requirements* (proposed by E7, E8, E10, E11, E16, E17, E18, E14, E15).		
Category	Task Analysis Methods for HCI	**Practice**	SP 1.2
I agree	*E1, E2, E3, E5, E8, E12, E19, E6-"The prototyping is important in this practice."*		
I partially agree	*E4- "These methods allow focusing on priority tasks if used to communicate with the customer."* *E7- "Requirement engineering is another field. So, RE methods should be used here. But task models do contribute, although they don't express requirement." [It is not enough. The expert proposes to include VOLERE, RESCUE, and methods in ISO 24744.]* *E9- "It can serve, but I think there are better ways of representing requirements. Note that one of the HCI recommendations is to engage the user. Therefore, it is better to see prototypes of what a model of tasks, for example."* *E10- "Translate the prioritized requirements in functionalities. Brainstorming is needed."* *E11- [It is not enough. The expert suggests the inclusion of quality attributes.]* *E13- "Other techniques to add to prioritization."* *E14- "The model includes critical tasks but does not prioritize the needs."* *E15- "To include Prototype."* *E16- "All aspects must be considered." [The experts suggest new methods for organizational context (FRAM and STAMP) - new category.]* *E17- "It does not integrate the contextual aspect. I suggest observation of the work environment."* *E18- "Only if it is a high level of abstraction. To add Scenarios and textual representations."* *E20- "I do not agree with SADT."*		
I don't agree	-		
Conclusion	Replace the category for the one specialized category *Techniques to identify user and organizational requirements* (proposed by E7, E10, E17, E18, E4). The inclusion of two categories: *Prototype for HCI requirements* (proposed by E6, E9, E15); *Standards and Guidelines for HCI design* (proposed by E11, E13, E14).		

Table 11. Analysis of the suggested techniques (adapted from [44])

Technique \ **Expert**		E3	E4	E7	E8	E10	E11	E14	E15	E16	E17	E18	E19
Techniques to identify user needs and	Description of Requirements (such as: VOLERE, RESCUE)		✓	✓	✓							✓	
	Interview	✓			✓				✓				✓
	Questionnaire						✓	✓	✓				
	Brainstorming				✓	✓			✓				
	Field study				✓				✓				
	Focus group						✓	✓					
	Ethnography								✓				
Techniques to identify user and organizational requirements	Scenarios				✓	✓	✓	✓	✓			✓	
	Personas				✓	✓	✓					✓	
	Storyboard				✓								
	Task analysis									✓			
	Organizational Context										✓		

In the same way, we analyzed all the other justifications for the other practices as follows:

- For SP1.1 we had 7 partial agreements and 3 disagreements for the category *Prototype for HCI requirements*. The three experts who disagreed considered that the time when we elicit needs is too early to use prototypes. The partial agreement justifications were also in the same line of thought, affirming that prototypes can be used, but it is early to use online techniques and also the use of prototypes "cannot have as a purpose the search of a solution". Although we had 10 agreements, we noted that the experts who agreed explained that the use of prototypes for this practice, in the beginning, should be done very carefully since the users can focus on the user interface design and not necessarily explain their needs (the main goal of this practice). Based on all these observations, we concluded that this category should be excluded from this practice.
- For SP2.1 we had 9 partial agreements and no disagreements. Four experts suggested including the use of the standards to establish the non-functional requirements for user interfaces. Thus, the *Standards and Guidelines for HCI design* category was included for this practice.
- For SP3.1 we had 7 partial agreements and 3 disagreements. The category *Operational Concepts and Scenarios Specification for HCI* was eliminated since the experts justified their partial agreement and disagreement by showing that this category was varied in its approaches. Eight experts who registered a partial agreement and the three experts who agreed suggested that several mentioned approaches should be included in a category specific to organizational requirements. These suggestions were considered in the analysis done in Table 11 and placed in the new category *Techniques to identify user and organizational requirements* that was therefore associated to this practice.
- For SP3.2, as mentioned, we got 75% of agreements and the partial agreements were justified only with the inclusion of new examples, which were all considered when proposed by more than two experts. The disagreement was from an expert with 8 years of experience who believes that the standards should be considered apart from requirements analysis. From our experience, and considering the largest agreement (19/20 where 15 were in full agreement), we did not consider this disagreement.
- SP3.3 and SP3.4 were analyzed by the experts in the same way. They considered that all results generated with the approaches defined in the categories associated with SP1.1

and SP1.2 should be used in the analysis (goals from these practices). By considering the results of the approaches, task modeling and validation of the requirements should be performed.

- For SP3.5 we had 9 partial agreements and 1 disagreement. Of them, 4 experts suggested including a category of *Techniques to validate requirements* to support this practice. The inclusion of the category was effectuated. The disagreement just mentioned that prototypes should be simple combining scenarios and screens, rather than functional, which does not justify the exclusion of this category.

6.2. Quantitative and qualitative analysis for Technical Solution (TS)

Based on the quantitative results presented in Figure 11 and justifications in the questionnaires organized in the same way as the one presented in Table 10 for RD, we concluded that:

- *Architecture Patterns for HCI* should be kept for SP2.1 since the partial agreement included only some comments about the use of the practice, observing that the patterns for HCI should be used at the beginning at a high level (for instance, three tiers architecture) which is confirmed by the single disagreement. However, for SP1.1 partial agreements, two experts suggested the use of techniques for interaction dialog (such as UAN (User Action Notion), MoLIC (Modeling Language for Interaction as Conversation) [3], TAG (Task Action-Grammar) [12]). Thus, the category *Techniques for interaction modeling* was created to address this need. Moreover, new examples were proposed for the category *Architecture patterns for HCI*. The two disagreements defended the importance of identifying platforms and interactive components before defining the architecture. We argue that this need can be considered while defining the architecture. We also note that the agreements support that the category should be kept.
- *Design patterns for HCI* should be kept for SP3.1 thanks to the large agreement (13 full agreements and 3 partial that only give new suggestions of approaches); and the justification for the disagreements is that design patterns should be used only for the implementation, which is contrary to the common way of using them from the design phase. *Architecture patterns for HCI* category is also included as a set of approaches to support this practice following the suggestion of two experts.
- *Prototype for HCI requirements* should also be kept for SP2.1, however, to be in coherence with the partial agreements and disagreements, the developers should take care to not consider this prototype as the final product, but it can be used to develop the first version to be developed. Other experts also recommend that the prototypes should be done through participatory design with the users in an interactive development. Those comments will be included as recommendations while using the prototypes.

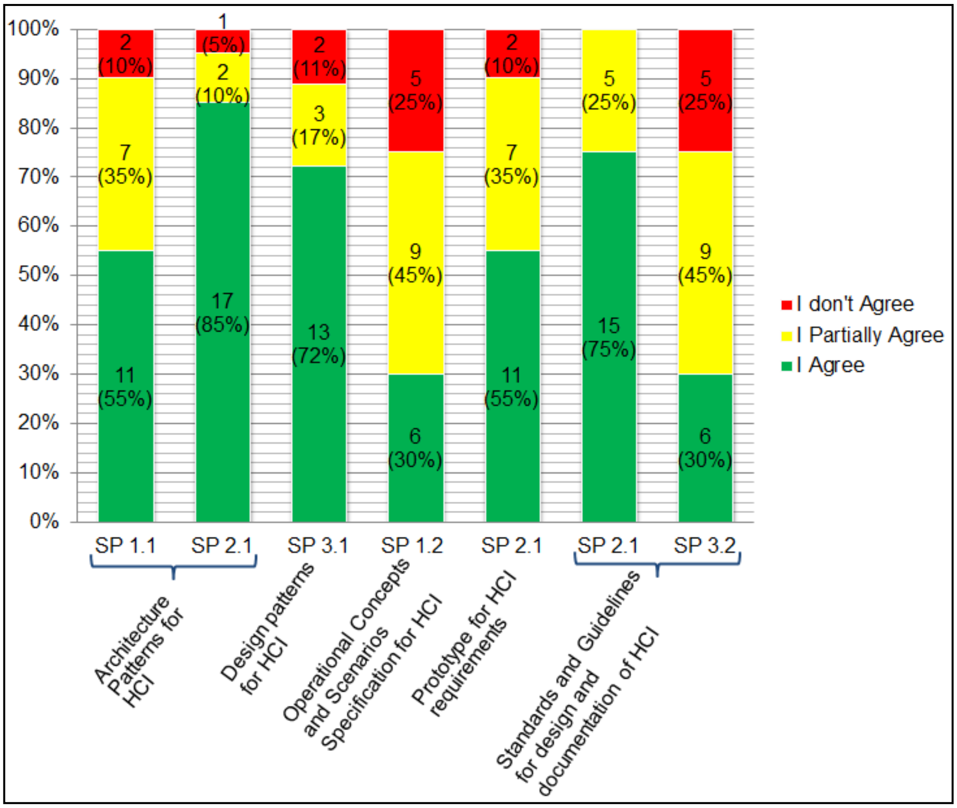


Figure 11. Quantitative results for Technical Solution (category and practice)

- *Standards and Guidelines for Design and Documentation of HCI* should also be kept for SP2.1 since the partial agreements, in general, only comment that the standards are useful but do not reach the final goal. For SP3.2, the justifications for the disagreements and partial agreements converge in the sense that the category does not address the ultimate goal of the practice, which is to develop the product support documentation. Four experts (E16, E18, E13, and E19) proposed some standards, guidelines, and techniques that correspond to the goal of this practice. In this case, we decided to create a new category specifically for documentation (*Techniques for HCI documentation*) that contains the suggestions of the experts, and to rename the previous one *Standards and Guidelines for Design* which is no longer associated to this practice.
- *Operational concepts and scenarios specifications for HCI* – was varied in approaches. The justifications for the disagreement to the proposition for SP1.2 converge in the sense that the category provides examples of techniques - such as scenario, persona and use cases - which should be used earlier in the system development. In addition, other experts (E7, E8) said that the goal of the practice is to use techniques and examples of similar systems for selecting and choosing the best solution, or the use of design rational argumentation [34]. Both suggestions are not specific to HCI and can be used for the selection of any product component solution. We concluded, therefore, that the proposed category for this practice does not respond to the same goal and the category was excluded. As a consequence, this practice had no proposition specific to HCI, being supported by the software engineering approaches currently used to select any product component solution.

6.3. Quantitative and qualitative analysis for Product Integration (PI)

PI process area had only two propositions to be evaluated as presented in Figure 12:

- *Functional Prototype to validate HCI* should be kept (15 agreements) and the three partial agreements only suggested some recommendations (such as the use of a good IDE – Integrated Development Environment). Therefore, it should be kept for the practice SP1.1. Moreover, the disagreements were not valid for its exclusion, but they may be used as a recommendation while using the prototypes since they suggest considering customization, plasticity, context awareness adaptation and design patterns for the functional prototypes.
- *Prototype for HCI requirements* had 9 (45%) disagreements for SP1.1 and was the proposition that had the most disagreements. Moreover, by calculating the median and mode of the experts' opinion we obtained as a result "I don't agree" with this proposition. The disagreements were consensual saying that it is late to make prototypes for requirements and that it does not help in the integration, which was also confirmed by two partial agreements (E8 and E20). As a consequence, despite six experts having agreed with this proposition, the analyzed justifications determine the exclusion of this proposition.

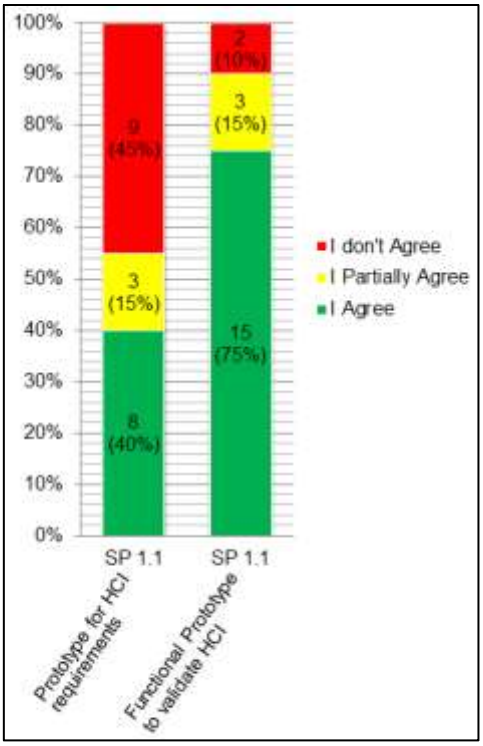


Figure 12. Quantitative results for Product Integration (category and practice)

6.4. Quantitative and qualitative analysis for Verification (VER)

Figure 13 presents the quantitative results for VER process area. In general, we noted that:

- the propositions *Standards and guidelines for design and documentation of HCI* for SP1.3 and *Evaluation methods for HCI verification tests* for SP2.1, SP2.2 and SP2.3 had no disagreement, which means they should be kept. The partial agreements were only suggestions of new examples in each category that were, therefore, included.
- *Functional prototype to validate HCI* for SP1.1: the 3 experts who disagreed argued that the assessment should be carried out with the final product and not a prototype. However, the experts who partially agreed considered that the prototype to be used should be the first version zero of the system. Based on these comments, we decided to rename the category *Iterative and Evolutionary prototype (system versions)*.

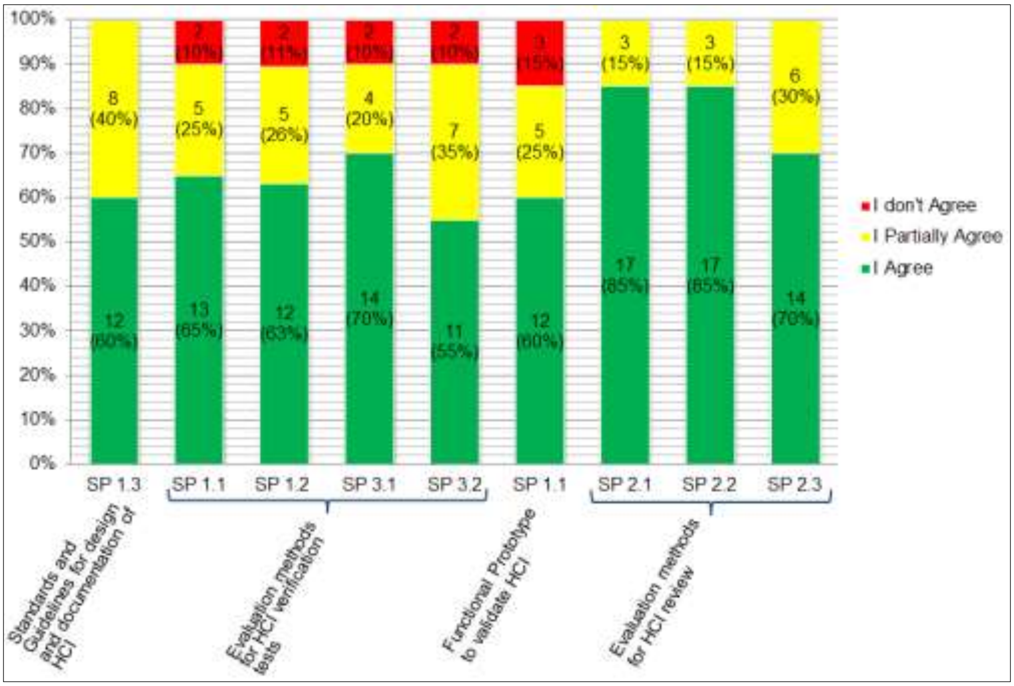


Figure 13. Quantitative results for Verification (category and practice)

Analyzing all justifications for the propositions of *Evaluation methods for HCI verification tests* for all practices, we observe that:

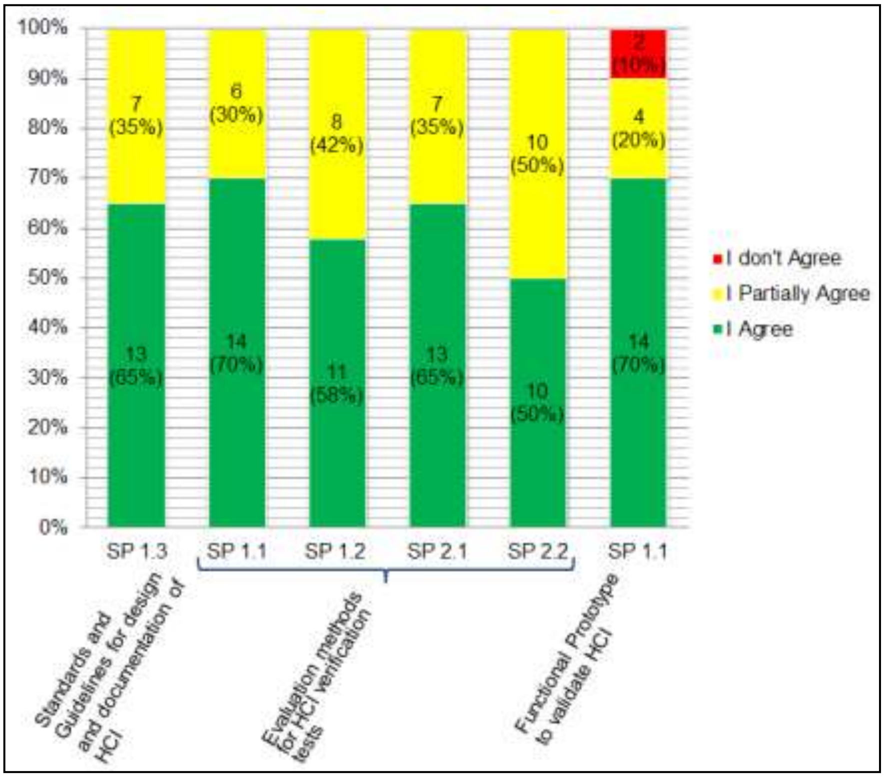
- two experts who disagreed and one who partially agreed suggested the inclusion of classical verification tests from software engineering;
- the experts who disagreed (E6, E16) and the one who partially agreed suggested the use of verification by HCI experts and not validation by experts (as proposed);
- four experts (E14, E17, and E10 for SP3.2) suggested considering statistics tools and methods for analysis;
- one expert (E19) suggested including an accessibility test. Although accessibility is an important non-functional requirement nowadays, we decided not to emphasize this quality attribute over others, given that each application has its specific requirements.

Based on these justifications we recognize that the *Evaluation methods for HCI verification tests* category was really varied in verification and validation approaches. We decided, therefore, to split it into two categories named *Evaluation methods for HCI verification* and *Evaluation methods for HCI validation* with adequate examples. In this way, the *Evaluation methods for HCI verification* replaced the previous one for the four practices (SP1.1, SP1.2, SP3.1, and SP3.2).

6.5. Quantitative and qualitative analysis for Validation (VAL)

Figure 14 presents the results for VAL process area. Four of the five propositions were accepted (no disagreements) and the last proposition only had 2 disagreements.

Figure 14. Quantitative results for Validation (category and practice)



The experts' justifications for the partial agreements and disagreements were quite similar to the observations made for VER process area and therefore implied similar decisions. In general, the experts indicated that:

- they do not agree with validation by experts (E8 and E16) in the *Evaluation methods for HCI verification tests.* They argued that the validation should be done with end users and the verification should be done earlier with HCI experts (as described in the previous section);
- similarly to VER process area, E3 and E10 disagree with the proposition of *Functional Prototype to validate HCI* justifying that the assessment should be performed with the final product and not the prototype;
- two experts (E15 and E18) suggest including communicability and user experience evaluation in the category that supports SP1.2.

Based on this analysis and considering the decisions previously described for VER process area we decided:

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

7. Synthesis of results

To summarize, at the beginning we had 10 HCI categories (Table 5). composed of 33 propositions that support 27 engineering practices of the CMMI-DEV (see Table 6). After the analysis of all interviews, we obtained 14 HCI categories (Table 12) composed of 39 propositions that support 26 engineering practices of the CMMI-DEV, as presented in Table 13. The interview with HCI experts resulted in:

- the exclusion of one practice (TS SP1.2), because the proposed category for this practice did not respond to the same goal and the category associated to this practice was excluded;
- the inclusion of two categories (*techniques to identify user needs* and *techniques to identify user and organizational requirements*) that imported some approaches from the preliminary category *operational concepts and scenarios specification for HCI*;
- the inclusion of two new categories (*techniques for interaction modeling* and *techniques for HCI documentation*);
- the inclusion of one category for *evaluation methods for HCI verification*;
- the modification of the name of the *evaluation methods for HCI verification tests* to *evaluation methods for HCI validation*;
- the inclusion of new six propositions according to the new categories and the suggestions of the experts.

By analyzing the suggestions of the experts, we also improved our propositions indicating when the outcome of the application of one HCI category in one practice should be used to support another specific practice. We represented this with the symbol ◊ in Table 13. For example, in RD (Requirements development) the outcomes produced by the *techniques to identify user needs* and *techniques to identify user and organizational requirements* categories that are applied in practices SP1.1 and SP1.2, should be used to support SP2.1 when applying the *task modeling* category, which means when defining task models.

Table 12. HCI Engineering to support CMMI-DEV after interviews with Experts

HCI Category	Examples
Techniques to identify user needs	Brainstorming [21]; Interviews [21]; Surveys/Questionnaires [21]; Card Sorting [21]; Focus Groups [21]; Field Studies [21].
Techniques to identify user and organizational requirements	Persona [21] [74]; Scenario [21] [74]; User stories [21] [74], User profile (detailed) [21] [74]; Task analysis [21] [74]; Context-of-use analysis [21] [74]; Storyboards [21] [74]; Requirements specification templates (e.g. VOLERE[6], IEEE[7], RESCUE[8]).
Task Modeling	CTT (Concur Task Tree) [93]; K-MAD (Kernel of Model for Activity Description) or MAD (Model for Activity Description) [13]; HTA (Hierarchical Task Analysis) [72]; SADT (Structured Analysis and Design Technique) or SADT coupled with Petri Nets [100]; GTA (Groupware Task Analysis) [126]; Task Model Standard (W3C) [132]; HAMSTERS notation [77].
Standards and Guidelines for HCI design	Ergonomic criteria [106] [122]; ISO/IEC 9126-1 [58]; ISO 9241-11 [54]; ISO/IEC 25000 [60]; Accessibility standards and guidelines (WAI - W3C) [131]; Nielsen's Heuristics [86]; Golden Rules of Interface Design [110].
Prototype for HCI requirements	Paper Prototyping/Sketches [5][74]; Storyboards [5][74]; Wireframes [5][74]; Mockups [5][74]; Wizard of Oz [5][74]; Video prototyping [5][74].
Techniques to validate HCI requirements	ProtoTask (K-MAD) [13]; Task Model Simulator (CTT) [92]; Focus group to validate HCI requirements [85]; Thinking aloud [110].
Architecture patterns for HCI	Arch Model [4]; Language Model [40]; Seeheim Model [91]; PAC (Presentation-Abstraction-Control) Model [23]; PAC-AMODEUS Model [88]; MVC (Model-View-Controller) Model [42]; AMF: A multi-agent and multi-faceted architecture model [90] [118]; IRVO (Interacting with Real and Virtual Objects) Architectural Model [16]; CAMELEON-RT [2]; Frameworks[9].
Design patterns for HCI	A Pattern Language for Human-Computer Interface Design [120]; A Pattern Approach to Interaction Design [11]; Pattern Languages in Interaction Design: Structure and Organization [124]; Designing interfaces: Patterns for Effective Interaction Design [119]; Patterns of HCI Design and HCI Design of Patterns [108].
Techniques for interaction modeling	MoLIC (Modeling Language for Interaction as Conversation) [3]; UAN (User Action Notation) [50]; TAG (Task-Action Grammar) [12].
Techniques for HCI documentation	Style guide [110]; Architecture for help [113]; Training Program [78].
Iterative and Evolutionary Prototypes (system versions)	User interface toolkits [5]; User interface builders [5]; User interface development environments [5].
Evaluation methods for HCI verification	Unit test; Integration test; System test; Acceptance test; Installation test [18].
Evaluation methods for HCI review	Semiotic inspection [114]; Heuristic evaluation [86]; Cognitive walkthrough [20][110]; Groupware Walkthrough [75]; Guidelines review [110]; Consistency inspection [110]; Metaphors of human thinking (MOT) [110]; Formal usability inspection [110].
Evaluation methods for HCI validation	Usability testing [25][110]; Communicability test [96]; Standardized usability questionnaires [105]; Post-experience interviews [105]; User experience evaluation [127].

[6] http://www.volere.co.uk/template.htm
[7] http://www.cse.msu.edu/~cse870/IEEEXplore-SRS-template.pdf
[8] https://www.city.ac.uk/__data/assets/pdf_file/0006/79881/RESCUE_Process_Doc_v4_1.pdf
[9] http://www.iso-architecture.org/42010/afs/frameworks-table.html **and** http://www.iso-architecture.org/42010/cm/

Table 13. Categories x CMMI-DEV practices after interviews with Experts

Categories	RD								TS				PI	VER								VAL				
	SP1.1	SP1.2	SP2.1	SP3.1	SP3.2	SP3.3	SP3.4	SP3.5	SP1.1	SP2.1	SP3.1	SP3.2	SP1.1	SP1.1	SP1.2	SP1.3	SP2.1	SP2.2	SP2.3	SP3.1	SP3.2	SP1.1	SP1.2	SP1.3	SP2.1	SP2.2
Techniques to identify user needs	●		◊																							
Techniques to identify user and organizational requirements	●	●	◊	●	◊	◊			◊													◊			◊	
Task Modeling	●	●	●	◊		●	◊	◊						◊	◊		◊	◊	◊	◊	◊	◊	◊		◊	◊
Prototype for HCI requirements		●	◊					●	◊																	
Standards and Guidelines for HCI design		●	●		●	◊	◊			●						●			◊		◊			●		◊
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 validation																						●	●		●	●

Categories	RD								TS				PI	VER								VAL				
	SP1.1	SP1.2	SP2.1	SP3.1	SP3.2	SP3.3	SP3.4	SP3.5	SP1.1	SP2.1	SP3.1	SP3.2	SP1.1	SP1.1	SP1.2	SP1.3	SP2.1	SP2.2	SP2.3	SP3.1	SP3.2	SP1.1	SP1.2	SP1.3	SP2.1	SP2.2
Evaluation methods for HCI review																	●	●	●							
Total of propositions (●) - 39	3	4	2	1	1	1	1	2	2	3	2	1	1	2	1	1	1	1	1	1	1	2	1	1	1	1
	15								**8**				**1**	**9**								**6**				

8. Long-term Validation in Practice

We argue that to evaluate the effectiveness of our proposal we should compare the results of studies using the HCI approaches (previouly presented) and those not using them. To start this validation, we decided to take advantage of the fact that HCI issues are taught in the Computer Science master's degree at the University of Valenciennes. In this master's degree program, there is a specific HCI course where the professor (one of the authors) has asked for a requirement specification of a typical interactive system as a final assessed project. The professor has been carrying out the same project since 2010. In this project, the students receive a detailed description of a real problem involving an industrial process with five interconnected mixing stations (that need a supervisory HCI). The real problem considers seven different profiles of potential users: supervisors, overseers, rounds men, production engineers, experts, maintenance technicians and fire department employees. The system must be displayed on a single or several screens in a control room occupied simultaneously by two human operators, called supervisors, working in rotating shifts (3 x 8 hours). The professor does not give any specific orientation to produce the requirement specification, i.e., the students are free to use whatever they have learned up to the moment of the study. As a consequence, all reports produced by the students since 2010 were available to us.

Looking at the HCI categories that may be applied in the phase of requirements specification, we identified Task Modeling as the most pertinent one to start our validation; since task modeling is taught in the courses, it is considered as essential for interactive system design, and is usually presented in the requirement specification.

We therefore considered that we could analyze the reports of all previous years and consider them as the results of doing requirement specification without an indication of how to perform it. After that, we would give some indication of how to perform the requirement specification providing a list of approaches for the Task Modeling category. Our assumption is that once the students had a list of suggestions related to HCI approaches, they would perform task modeling using them. Confirming this assumption, we can envisage that with the definition of how to perform requirement specification for interactive systems, we can have better results on the application of HCI issues in practice.

Our main research question in this study was: (Question 1) *To what extent has task modeling been applied in requirement specifications of typical interactive systems?* Moreover, we defined two complementary questions for our analysis: (Question 2) *Which are the methods used for task modeling?*, and (Question 3) *How detailed was the task modeling?*

We therefore performed this validation in two iterations. In the first iteration (hereafter named iteration 1) we performed an analysis of all reports produced by the students over the five years (2010 to 2014). For this iteration we cannot change any variable, because the data concerns the past (2010 to 2014). Given that the students were free to use whatever they wanted, we considered this iteration as a first analysis of the study without using our approach, which presents what should be used in each moment. For this iteration 43 project reports were analysed for the purposes of this validation. The complete result from this iteration is detailed and discussed in [44][10].

In iteration 2 (2015 to 2016, 2 years), we decide to change one variable of this study to confirm our assumption. We presented the students with a list of HCI approaches (including Task Modeling) that could be used for requirement specification activities. Then, we also performed a descriptive analysis of the requirements specification produced, and we compared the results with the results of the first iteration. In the second interation, we analyzed 22 project reports.

In the following paragraphs, we present the results for each research question of the study by comparing the two iterations to show the effective use of task modeling in requirement specification.

8.1. Question 1 - To what extent has task modeling been applied in requirement specifications of typical interactive systems?

To decide which task modeling was relevant we evaluate the projects using the following 7-point Likert scale:

(-1) not considered – "task modeling not considered";
(0) bad - "bad task modeling";
(1) just mentioned but not detailed - "task modeling mentioned but not detailed";
(2) not enough - "task modeling is not enough";
(3) average - "average task modeling";
(4) good - "good task modeling"; and,
(5) very good - "task modeling very well defined".

We considered as satisfactory the task modeling evaluated as greater or equal to 3 (average).

The results for the first iteration indicate that the level of importance given to task modeling for system specification was not so high. Only 49% (22/43 - see Figure 15) of the reports performed task modeling. A small part (21%, i.e., 9 out of 43 - see "satisfactory results" in Figure 15) of the reports presented a good result of task modeling. Considering that the content of task modeling was taught in theoretical and practical classes, we expected to find more meaningful results. We are concerned about the fact that a significant proportion of students did not consider the modeling of user tasks in the specification phase, which is the phase where the modeling of tasks has more emphasis on the system development [22] [49]. For the second iteration we have better results.

[10] In this publication, we analyzed 63 project reports from 2010-2014 (five years) where 43 are from Computer Science and 20 from Automation and Human-Machine Systems programs.

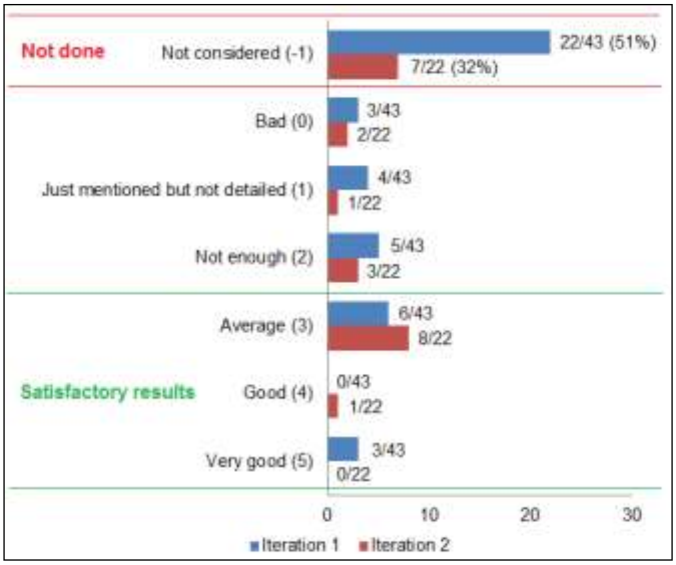


Figure 15. Comparison of the two iterations

Regarding the second iteration, 68% (15/22 - see Figure 15) of the reports presented task modeling. In addition, 9/22 (40%) reports presented task modeling that we consider satisfactory for this study (see “satisfactory results” in Figure 15). We can note **an improvement** (from 49% to 68%) regarding the percentage of reports (see Figure 15) that performed task modeling between the first and the second iterations. We also got **an improvement** (from 21% to 40%) regarding the percentage of reports that presented satisfactory task modeling. We can say that the suggestions of HCI approaches improved the results.

8.2. Question 2 - Which are the methods used for task modeling?

For this question, we expected to find the use of the modeling methods taught: CTT, HTA, MAD, and SADT & Petri Nets. However, after analyzing the first question we identified that the methods used were not only those taught in class. For the first iteration we found that only 18% (8/43) of the reports really used the taught task modeling formalisms (see Figure 16). We could suppose that not all the students knew what exactly to apply. However, this does not explain the fact that some project reports did not present any task modeling. In addition, the students had three sessions with the presence of the professor (supervised work classes) during which they could ask questions related to the project. For example, some reports used an activity diagram to model the user tasks. Even if these reports did not use task model formalisms, we appreciated the students who made an effort to model the tasks of the users.

In contrast, for the second iteration we found that all the reports (59% - 13/22, see Figure 17) that presented task modeling, really used the taught task modeling formalisms. When we compare the two iterations, we can note an improvement (from 18% to 59%) regarding the percentage of reports that used the task modeling formalisms taught in class. The improvement regarding the approaches used to perform task modeling was considerable; and in iteration 2 the majority (13/15) of the reports that presented task modeling used formal methods.

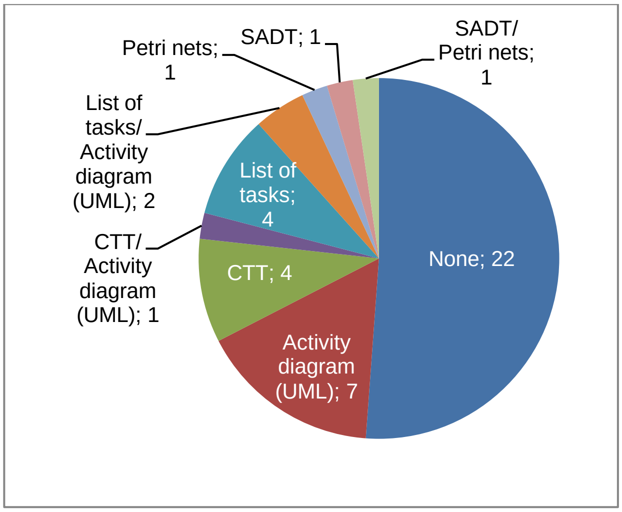


Figure 16. Task modeling formalisms – Iteration 1

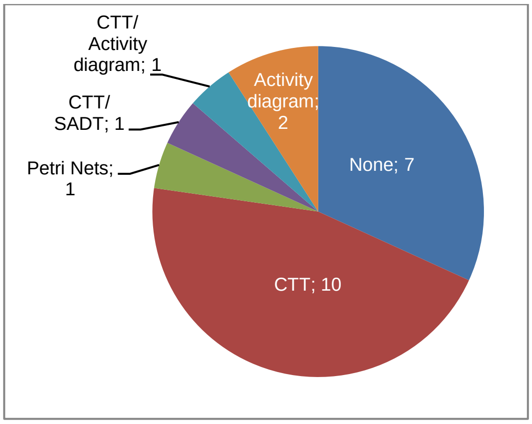


Figure 17. Task modeling formalisms – Iteration 2

8.3. Question 3 - How detailed was the task modeling?

For this question, we classified the task modeling as "global modeling" or "detailed modeling". For instance, when using CTT, we considered it to be "global modeling" when the report presents just a high level of task tree (abstract tasks) without defining the primitive tasks. On the other hand, "detailed modeling" considers several levels of abstraction in the task tree.

In the iteration 1, we found that only 14% (3/21 – see Figure 18) of the reports presented "detailed task modeling". For the iteration 2, 40% (6/15 - see Figure 18) of reports presented "detailed task modeling". We can note an improvement (from 14% to 40%) regarding the percentage of reports that presented detailed task modeling.

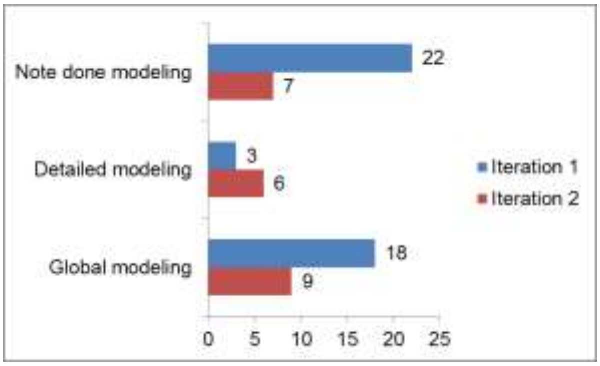


Figure 18. Task modeling detail

8.4. Complementary analysis

In general, for the first iteration we considered that the specification reports were good, but that the task models were not developed as expected. This leads us to believe that the applied methodology has been generally satisfactory, but that students do not see much interest in using task models. To verify if by using an HCI approach, we would obtain a better product, we also decided to consider the final grade of the requirement specification produced by the students. It is important to mention that task modeling is only one of the elements of the specification.

We considered the grade defined by the professor as representing the quality of the requirement specification. In Table 14 we observe that the worst grade, the best and the average for the requirement specification with task modeling were better than those that did not use it (see Table 14), for the first iteration. We can note that the average difference is small (5%) and the average value is low considering the minimum value (10) needed to validate the course. Moreover, one single project got the highest grade.

For the second iteration, only the worst grade for the requirement specification with task modeling was better than those that did not use it (see Table 14). Even if the best grade is lower for the student that did the task modeling, the worst grade is much better. Moreover, the average for the project with task modeling in iteration 2 is better than the projects with task modeling in 2010-2014. Moreover, considering that the grade is from 0 to 20, we note that the worst grade for specification with task modeling was much higher than 10 when applying task modeling in both cases. Ten is the minimum value needed to validate the course.

Table 14. Grade for the projects: (grade 0 – 20)

Iteration	Specification report	Number of projects	Worst grade	Best grade	Average
1	Without task modeling	22	6,0	15,50	10
	With task modeling	21	11,5	19	11
2	Without task modeling	7	8	19	15
	With task modeling	15	13	17	15

Although the task modeling is only one of the elements of the requirement specification, we could say that modeling user tasks probably helped the students to better understand the problem and, as a consequence, get better specifications.

In general, we can conclude that the results presented by the second iteration were better than the results presented by the first iteration. We believe that the reminder of the HCI approaches was the variable that contributed to the improvement of the results. We plan to continue our long-term validation in an academic environment with other HCI approaches from different categories.

9. Threats of validity

To analyze the results we considered the four threats of validity proposed by Wohlin *et al.* [129]: 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 and whether the result adequately reflects the effects. This threat is related to the building of our questionnaire. To minimize this threat we built the questionnaire using the original text extracted (specific goals and specific practices) from the official documentation of CMMI-DEV (in three languages: English, French and Portuguese). In addition, the proposition of HCI categories and examples were collected from literature and pre-validated by one of the authors, who has more than thirty years of experience in HCI. Moreover, we had the official documentation of CMMI-DEV during the interview to be consulted in case of doubts. Finally, the authors who conducted the interviews have a good knowledge of CMMI-DEV, having already participated in official CMMI-DEV implementations and appraisals.

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 first group of experts was selected from the professional network of one of the authors. After that, these experts suggested other names following the pre-defined fixed profile for the expert selection. We believe that maybe some experts did feel not comfortable in disagreeing with the propositions. To mitigate this bias, the author who knows some of the experts of the first group did not participate in any interview.

Another threat to the internal validity concerns the knowledge of the experts related to the HCI categories and examples proposed in our study. We assumed that the experts knew all the proposed approaches. To minimize this risk, we selected only professionals who have experience (academic and/or industrial, as showed in Table 7 they have 19 years of experience on average) in the HCI domain and have a Ph.D. degree. We decided that it was not necessary for the experts to be familiar with CMMI-DEV since the practices of engineering process areas are typically in the development of systems. In addition, the authors who conducted the interviews have academic and practical experience with CMMI-DEV, making it possible to clarify any doubts of the experts. However, we could not guarantee that even with these mitigation actions, the experts gave their real opinion. We believed that, since we were in an improvement approach, we could accept this risk.

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, this threat concerns the relation between the HCI categories associated with each specific practice. To reduce this risk we decided to perform interviews individually and not using a survey. In this way, we can clarify each doubt of the experts about the objective of the evaluation, the CMMI-DEV and the proposed HCI categories. In addition, when the experts did not agree or partially agreed with one or more propositions, they were asked to justify their opinion and include any other proposals they judged necessary. The final set of propositions resulted in the majority from the agreement or from partial agreements. All the modifications were made respecting the justifications of the experts.

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. The result could be biased if experts come only from one domain of expertise. For instance, experts working on real-time systems in the context of military or aerospace systems follow practices and standards that are very different from the ones working on information systems and web application design. Therefore, they could naturally inject a bias since they are more prone towards the approaches that are essential and frequently used in their working context. To minimize this risk we decided to perform the interview with experts not only with different expertise in HCI but also with wide experience recognized by the HCI community (e.g., being a program chair or member of program committee of HCI conferences, editor of journals and members of HCI associations).

Therefore we invited experts that are well-known for working on different technologies (e.g., web applications, information systems, critical systems, tabletop applications, and so on). In Table 7, we identified only their current interest but this has naturally evolved/modified throughout the years with new technologies, research and the new domains of application they have been working with. Therefore, even though we looked for experts with different backgrounds, we could not ensure that we covered all kinds of technologies and application domains, and we cannot ensure that they were well-balanced from this point of view.

To mitigate this issue, we conducted the interviews asking them to indicate approaches that could be used in general for any kind of interactive systems. Having wide experience (12/20 experts have more than 20 years of experience) and having worked in several domains throughout their career, we assumed their opinion to be reliable. Moreover, they were notified that all their suggestions for very specific kinds of application would be included as particular recommendations. In this way, the experts evaluated the examples one by one, including new ones that they considered relevant and eliminating some that they considered as not being used anymore. As a consequence of all these considerations, we accepted the risk of potential bias in their evaluation. Therefore, it is not possible to generalize this result.

10. Conclusion

This paper has presented a study of the Capability Maturity Model Integration for Development (CMMI-DEV) that aims to identify which HCI approaches could support engineering practices in the development of interactive systems. We have analyzed five process areas (Requirements Development, Technical Solution, Product Integration, Verification, and Validation) of the engineering category from this model. To answer our research question, *“what are the approaches that could concretely integrate CMMI-DEV process areas in interactive system development?”* we proposed a set of 14 HCI categories with a total of 77 HCI approaches. The HCI categories were associated with 26 engineering practices of the CMMI-DEV model, resulting in 39 propositions to support the users of this model.

Our on-going work is the definition of a methodological guide to support the users of CMMI-DEV to apply our proposition. This guide will present for each of the 26 specific practices how we should use the identified HCI approaches integrated to conventional approaches of software engineering in the development of interactive systems. These software engineering approaches are usually applied in a software process based on CMMI-DEV that indicates only what to do but not how to do it. As a consequence, the purpose of the methodological guide is to show how to do it for the development of interactive systems considering the HCI approaches investigated in this study.

This methodological guide will consider generic recommendations, defined by the experts and the literature, for any interactive system, such as the participation of the end user (participatory design) in the system design (Technical Solution process area) and the use of user-centered design approaches (Requirements Development and Technical Solution process area) that include the end user as part of the development team. In addition, it will also include:

- specific recommendations for specific types of interactive systems (for instance, in the case of a critical system, the use of techniques such as FRAM [130];
- specific suggestions for the integration of HCI with specific methodologies (e.g. agile methodologies [8] [9] [103];
- suggestions for identifying organizational context issues as suggested by an expert with 30 years of experience;
- the suggestion of more new techniques proposed by the research studies (for instance, the use of Worth Mapping [14] to design interactive systems with the finality to deliver worth in the real world and decrease the gap between the user and product, as suggested by an expert with 30 years of experience); and
- details of how to apply some of techniques or methods like the suggested HCI partners and standards.

Furthermore, we plan to perform a survey with users of the CMMI-DEV in industry using the HCI engineering approaches defined in this study. Our goal is to identify if the CMMI-DEV users know and use the HCI engineering approaches in practice, as they use software engineering approaches. The result of this survey could help to identify HCI engineering approaches that are not or are insufficiently used in industry. In this case, these approaches can be the object of future technology transfer from the academy to the industry.

Finally, considering that the CMMI-DEV is usually used for the definition of a software process for the development of an application, we plan to define a specific software process using our HCI categories (and approaches) for a specific kind of interactive system: interactive tabletop applications with tangible objects. Thus, we plan to use this software process in the development of an interactive tabletop application to collect evidence about the effective use of our HCI categories in practice.

Acknowledgements

We would like to acknowledge the financial support granted by CAPES - Science without Borders Program and the experts who participated in this study. The authors warmly thank the anonymous reviewers for their numerous constructive remarks.

References

[1] Ab Latif, R., Mohamed, R., Dahlan, A., Mat Nor, M.Z., 2016. Using Delphi technique: making sense of consensus in concept mapping structure and multiple choice questions (MCQ). Education in Medicine Journal, 8(3), 89-98.

[2] Balme, L., Demeure, A., Barralon, N., Coutaz, J., Calvary, G., 2004. CAMELEON-RT: A Software Architecture Reference Model for Distributed, Migratable, and Plastic User Interfaces. In: P. Markopoulos et al. (Eds.), EUSAI 2004, LNCS 3295, Springer-Verlag, pp. 291-302.

[3] Barbosa, S. D. J., Paula, M. G., 2003. Designing and Evaluating Interaction as Conversation: a Modeling Language based on Semiotic Engineering. In: Proceedings of DSV-IS 2003, pp. 16-33.

[4] Bass, L., Little, R., Pellegrino, R., Reed, S., Seacord, S., Sheppard, S., Szesur, M., 1991. The Arch model: Seeheim revisited. In: Proceedings of User Interface Developers Workshop. Seeheim.

[5] Beaudouin-Lafon, M., Mackay, W.E., 2009. Prototyping tools and techniques. In: Sears, A., Jacko, J.A. (Eds.), Human-Computer Interaction: Development Process. CRC Press, NW, pp. 121–143.

[6] Beecham, S., Hall, T., Britton, C., Cottee, M., Rainer, A., 2005. Using an expert panel to validate a requirements process improvement model. Journal of Systems and Software, 76(3), 251-275.

[7] Bell Canada, 1994. Trillium Model for Telecom Product Development & Support Process Capability (No. Release 3.0).

[8] Bertholdo, A.P.O., Silva, T.S., Melo, C.O., Kon, F., Silveira, M.S., 2014. Agile Usability Patterns for UCD Early Stages. In A. Marcus (Ed.), Design, User Experience, and Usability. Theories, Methods, and Tools for Designing the User Experience: Third International Conference, DUXU 2014, pp. 33-44.

[9] Bertholdo, A.P.O., Kon, F., Gerosa, M.A., 2016. Agile Usability Patterns for User-Centered Design Final Stages. In M. Kurosu (Ed.), Human-Computer Interaction. Theory, Design, Development and Practice : Proceedings of HCI International - Part I, pp. 433-444.

[10] Bevan, N., 2009. International Standards for Usability Should Be More Widely Used. Journal of Usability Studies, 4 (3), 106-113.

[11] Borchers, J., 2001. A Pattern Approach to Interaction Design. New York, NY, USA: John Wiley & Sons, Inc.

[12] Brown, J., 1996. Evaluation of the task-action grammar method for assessing learnability in user interface software. In: Sixth Australian Conference on Computer-Human Interaction, pp. 308-309.

[13] Caffiau, S., Girard, P., Scapin, D.L., Guittet, L., Sanou, L., 2008. Assessment of Object Use for Task Modeling. In: Forbrig, P., Paternò, F. (Eds.), Engineering Interactive Systems: Second Conference on Human-Centered Software Engineering, HCSE 2008, and 7th International Workshop on Task Models and Diagrams, TAMODIA 2008. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 14-28.

[14] Camara, F., Calvary, G., 2015. Worth-Centered Design in Practice: Lessons from Experience and Research Agenda. In: 15th IFIP TC 13 International Conference, pp. 123-139.

[15] Chagas, L. F., Carvalho, D. D., Lima, A. M., Reis, C. A. L., 2014. Systematic Literature Review on the Characteristics of Agile Project Management in the Context of Maturity Models. In: Mitasiunas, A., Rout, T., O'Connor, R.V., Dorling, A. (Eds.), Proceedings of 14th International Conference, SPICE 2014. Springer International Publishing, Cham, pp. 177-189.

[16] Chalon, R., David. B. 2005. IRVO: an Interaction Model for designing Collaborative Mixed Reality systems. In: HCI International Conference 2005, Las Vegas, United States. Lawrence Erlbaum Associates, Inc, pp.1-10.

[17] Chapman, L., Plewes, S., 2014. A UX maturity model: effective introduction of UX into organizations. In: International Conference of Design, User Experience,and Usability, Springer Int. Publishing, Cham, Switzerland, pp. 12–22 .

[18] CMMI Product Team, 2010. CMMI® for Development (CMMI-DEV), V1.3, (CMU/SEI-2010th-TR-033 ed.), Pittsburgh, PA, USA: Carnegie Mellon University.

[19] CMMI Product Team, 2015. Maturity Profile Report, July 2015. Retrieved from http://cmmiinstitute.com/resources/.

[20] Cockton, G., Woolrych, A., Lavery, D., 2009. Inspection-based evaluations. In: Sears, A., Jacko, J.A. (Eds.), Human-Computer Interaction: Development Process. CRC Press, NW, pp. 273–292.

[21] Courage, C., Baxter, K., 2005. Understanding your users: a practical guide to user requirements, methods, tools, and techniques. San Francisco, CA: Morgan Kaufmann Publishers.

[22] Courage, C., Redish, J. G., Wixon, D. 2009. Task Analysis. In: A. Sears & J. A. Jacko (Eds.), Human-Computer Interaction: Development Process. CRC Press, NW, pp. 33–53.

[23] Coutaz, J., 1987. PAC, an Object-Oriented Model for Dialog Design. In Proceedings of 2nd IFIP International Conference on Human-Computer Interaction (INTERACT 87), pp. 431-436, Stuttgart, Germany.

[24] Deutsche Akkreditierungsstelle Technik (DATECH), 2002. DATech-Prufbaustein, Usability-Engineering-Prozess, Version 1.2, Frankfurt/Main. (in German)

[25] Dumas, J.S., Fox, J.E., 2009. Usability testing: Current practice and future directions. In: Sears, A., Jacko, J.A. (Eds.), Human-Computer Interaction: Development Process. CRC Press, NW, pp. 231–250.

[26] Dyba, T., 2000. An Instrument for Measuring the Key Factors of Success in Software Process Improvement. Empirical Software Engineering, 5(4), 357-390.

[27] Earthy, J., 1998. Usability Maturity Model: Processes. INUSE Project deliverable D5.1.4. Version 1.1. (London: Lloyd’s Register, 71 Fenchurch St, London EC3M 4BS).

[28] Earthy, J., 1998. Usability Maturity Model: Human Centredness Scale. INUSE Project deliverable D5.1.4(s) Version 1.2. (London: Lloyd’s Register, 71 Fenchurch St, London, EC3M 4BS).

[29] Earthy, J., 2001. Process risk assessors version of HSL model, HFI CMM Study (No. Work package 3).

[30] Earthy, J., Sherwood-Jones, B., 2000, Human factors integration capability maturity model – assessment model. In: Human Interfaces in Control Rooms, pp. 320-236, IET, March.

[31] Eason, K., Harker, S.D., 1997. User Centred Design Maturity, Internal working document - Department of Human Sciences, Loughborough University.

[32] El Emam, K., Madhavji, N. H., 1996. An instrument for measuring the success of the requirements engineering process in information systems development. Empirical Software Engineering, 1(3), 201-240.

[33] Estorilio, C., Vaz, G. R. M., Lisboa, F. C., Bessa, L. O. F., 2015. The relationship between industrial process maturity and quality certification. Computer Standards & Interfaces, 39, 22-33.

[34] Fathy, S., Kornyshova, E., 2016. Approches de Design Rationale: Cadre de Référence. In : Congrès INFORSID, pp. 83-98.

[35] Fenton, N., Pfleeger, S. L., 1997. Software metrics: a rigorous and practical approach. PWS Publishing Co. Boston, MA, USA.

[36] Ferre, X., Juristo, N., Moreno, A.M., 2005. Framework for Integrating Usability Practices into the Software Process. In: International Conference on Product-Focused Software Process Improvement, pp. 202-215.

[37] Ferre, X., Juristo, N., Moreno, A.M., 2005. Which, when and how usability techniques and activities should be integrated. In: Seffah, A., Vanderdonckt, J., Desmarais, M.C. (Eds.), Human-Centered Software Engineering — Integrating Usability in the Software Development Lifecycle, pp. 173-200.

[38] Fischer, H., 2012. Integrating usability engineering in the software development lifecycle based on international standards. In: Symposium on Engineering Interactive Computing Systems, pp. 321-324.

[39] Fischer, H., Strenge, B., Nebe, K., 2013. 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.

[40] Foley, J.D., van DAM, A., 1982. Fundamentals of Interactive Computer Graphics. Addison-Wesley (IBM Systems Programing Series), Reading, MA.

[41] Garzás, J., Pino, F.J., Piattini, M., Fernández, C.M., 2013. A maturity model for the Spanish software industry based on ISO standards. Computer Standards & Interfaces, 35(6), 616-628.

[42] Goldberg, A., 1984. SMALLTALK-80: The Interactive Programming Environment. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

[43] Gonçalves, T.G., Oliveira, K.M., Kolski, C., 2015. HCI engineering integrated with capability and maturity models. In: 27th Conference on l'Interaction Homme-Machine (IHM 2015), Toulouse, France, pp. 36:1-36:6, ACM.

[44] Gonçalves, T.G., Oliveira, K.M., Kolski, C., 2016. HCI Engineering integrated with Capability Maturity Models: a study focused on Requirements Development. In: International Conference on Research Challenges in Information Science, pp. 633-644, IEEE.
[45] Gonçalves, T.G., Oliveira, K.M., Kolski, C., 2017. The use of Task Modeling in Interactive System Specification. Cognition Technology & Work, 19 (2), 493-515. doi:10.1007/s10111-017-0427-1
[46] Gross, T., 2016. UCProMo – Towards a User-Centred Process Model. In: C. Bogdan, J. Gulliksen, S. Sauer, P. Forbrig, M. Winckler, C. Johnson, F. Kis (Eds.), Human-Centered and Error-Resilient Systems Development: IFIP WG 13.2/13.5 Joint Working Conference, 6th HCSE 2016, and 8th HESSD 2016, pp. 301-313.
[47] Gupta, A., 1997. The Humanware Process Improvement Framework: Interfacing User Centred Design and the Product Creation Process at Philips. In: Human Centred Process Improvement Group (HCPIG) Meeting, Teddington, UK.
[48] Hao, H.M., Jaafar, A., 2011. Usability in Practice: Perception and Practicality of Management and Practitioners. In: International Conference on Pattern Analysis and Intelligent Robotics, pp. 211-216.
[49] Hackos, J. T., Redish, J. C. 1998. User and Task Analysis for Interface Design. John Wiley & Sons, Inc.
[50] Hartson, H. R., Siochi, A. C., Hix, D., 1990. The UAN: A User-oriented Representation for Direct Manipulation Interface Designs. ACM Trans. Inf. Syst., 8(3), 181–203.
[51] Helgesson, Y. Y. L., Höst, M., Weyns, K., 2012. A review of methods for evaluation of maturity models for process improvement. Journal of Software: Evolution and Process, 24(4), 436-454.
[52] Helms, J.W., Arthur, J.D., Hix, D., Hartson, H. R., 2006. A field study of the Wheel – a usability engineering process model. Journal of Systems and Software, 79, 841-858.
[53] Hoda, R., Salleh, N., Grundy, J., Tee, H. M., 2017. Systematic literature reviews in agile software development: A tertiary study. Information and Software Technology, 85, 60-70.
[54] ISO, 1998. ISO 9241-11:1998 Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11: Guidance on usability. Geneva, Switzerland.
[55] ISO, 1999. ISO 13407:1999 Human-Centred Design Process for Interactive System. Geneva, Switzerland.
[56] ISO, 2000. ISO/TR 18529:2000 Ergonomics of human-system interaction - Human-centred lifecycle process descriptions. Geneva, Switzerland.
[57] ISO, 2010. ISO/TS 18152:2010 Ergonomics of human-system interaction - Specification for the process assessment of human-system issues. Geneva, Switzerland.
[58] ISO/IEC, 2001. ISO/IEC 9126-1:2001 Software engineering -- Product quality -- Part 1: Quality model. Geneva, Switzerland.
[59] ISO/IEC, 2004. ISO/IEC 15504-4: Information Technology - Process Assessment – Part 4 - Guidance on use for Process Improvement and Process Capability Determination. Geneva, Switzerland.
[60] ISO/IEC, 2014. ISO/IEC 25000: Software Engineering - Software Product Quality Requirements and Evaluation (SQuaRE) – Guide to SQuaRE. Geneva, Switzerland.
[61] Jokela, T., 2010. Usability Maturity Models: Making your Company User-Centered. User Experience Magazine, 9 (1).
[62] Jokela, T., 2008. Characterizations, Requirements, and Activities of User-Centered Design—the KESSU 2.2 Model. In: Law, E.L.-C., Hvannberg, E.T. & Cockton, G. (Eds.), Maturing Usability: Quality in Software, Interaction and Value, pp. 168-196.
[63] Jokela, T., Lalli, T., 2003. Usability and CMMI: Does a higher maturity level in product development mean better usability? In: Proceedings of Extended Abstracts on CHI'03, pp. 1010-1011, ACM.
[64] Jokela, T., Siponen, M., Hirasawa, N., Earthy, J., 2006. A survey of usability capability maturity models: implications for practice and research. Behaviour & Information Technology, 25 (3), 263-282.
[65] Kalinowski, M., Weber, K., Franco, N., Barroso, E., Duarte, V., Zanetti, D., Santos, G., 2014. Results of 10 Years of Software Process Improvement in Brazil Based on the MPS-SW Model.

In: 9th International Conference on the Quality of Information and Communications Technology (QUATIC), pp. 28-37.
[66] Kieffer, S., Vanderdonckt, J., 2016. STRATUS: a questionnaire for strategic usability assessment. In: Annual ACM Symposium on Applied Computing, ACM, New York, NY, pp. 205-212.
[67] Kitchenham, B., Charters, S., 2007. Guidelines for performing systematic literature reviews in software engineering, Technical Report EBSE 2007-001, Keele University and Durham University Joint Report, 57p.
[68] Kitchenham, B., Pfleeger, S. L., McColl, B., Eagan, S., 2002. An empirical study of maintenance and development estimation accuracy. Journal of Systems and Software, 64(1), 57-77.
[69] Lacerda, T.C., Wangenheim, C.G., 2017. Systematic literature review of usability capability/maturity models. Computer Standards & Interfaces (in press).
[70] Lauesen, S., Vinter, O., 2001. Preventing Requirement Defects: An Experiment in Process Improvement. Requirements Engineering, 6(1), 37-50.
[71] Lazar, J., Feng, J.H., Hochheiser, H., 2010. Research Methods in Human-Computer Interaction. United Kingdom: John Wiley & Sons Ltd.
[72] Limbourg, Q., Vanderdonckt, J., 2004. Comparing Task Model for User Interface Design. In: Diaper, D., Stanton, N.A. (Eds.), The Handbook of Task Analysis for Human-Computer Interaction. Lawrence Erlbaum Associates, Mahwah, NJ, pp. 135–154.
[73] Linstone, H.A., Turoff, M., 2002. The Delphi Method: techniques and applications. New Jersey. Avaiable in http://is.njit.edu/pubs/delphibook.
[74] Maguire, M., 2001. Methods to support human-centred design. International Journal of Human-Computer Studies, 55(4), 587-634.
[75] Mahatody, T., Sagar, M., Kolski, C., 2010. State of the Art on the Cognitive Walkthrough Method, Its Variants and Evolutions. International Journal of Human-Computer Interaction, 26 (8), 741-785.
[76] Marcus, A., Gunther, R., Sieffert, R., 2009. Validating a standardized usability/user-experience maturity model: a progress report. In: International Conference on Human Centered Design, Heidelberg: Springer, pp. 104–109.
[77] Martinie, C., Palanque, P., Winckler, M., 2015. Designing and Assessing Interactive Systems Using Task Models. In: Monteiro, I.T, Silveira, M.S. (Eds.), Tutorials of the 14th Brazilian Symposium on Human Factors in Computing Systems. SBC, pp. 29-58.
[78] Martinie, C., Palanque, P., Navarre, D., Winckler, M., Poupart, E., 2011. Model-Based Training: An Approach Supporting Operability of Critical Interactive Systems: Application to Satellite Ground Segments. In: Proceedings of EICS'11, pp. 53-62.
[79] Mayhew, D., 1999. The Usability Engineering Lifecycle: a practicioner's handbook for user interface design. CA: Morgan Kaufmann.
[80] McClelland, I., van Gelderen, T., Taylor, B., Hefley, B., Gupta, A., 1998. Humanware Process Improvement - institutionalising the principles of user centred design. In: III Human-Computer Interaction Conference on People and Computers - Tutorial PM14 Human-centred processes and their impact.
[81] Mostafa, D., 2013. Maturity models in the context of integrating agile development processes and user centred design. Diss. University of York.
[82] Nebe, K., Zimmermann, D., 2007. Aspects of Integrating User Centered Design into Software Engineering Processes. In: J. A. Jacko (Ed.), Human-Computer Interaction. Interaction Design and Usability: 12th International Conference HCI Part I, pp. 194-203.
[83] Nebe, K., Zimmermann, D., 2008. Suitability of Software Engineering Models for the Production of Usable Software. In: J. Gulliksen, M. B. Harning, P. Palanque, G. C. van der Veer, & J. Wesson (Eds.), Engineering Interactive Systems: EIS 2007 Joint Working Conferences, EHCI 2007, DSV-IS 2007, HCSE 2007, pp. 123-139.
[84] Nebe, K., Zimmermann, D., Paelke, V., 2008. Integrating Software Engineering and Usability Engineering. In: Pinder, S. (Ed.), Advances in Human Computer Interaction, InTech, pp. 33-350.
[85] Nielsen, J., 1997. The Use and Misuse of Focus Groups. IEEE Softw. 14, 94–95.

[86] Nielsen, J., 1994. Heuristic evaluation. In: Nielsen, J., Mack, R.L. (Eds.), Usability inspection methods. New-York: John Wiley & Sons, pp. 25-62.
[87] Nogueira, N. P., Furtado, E. 2013. Integration of IHC techniques in software processes based on a model of process improvement. In: Proceedings of XXXIX Latin American Computing Conference (CLEI), pp. 1–8. (text in Portuguese)
[88] Nigay, L., Coutaz, J., 1995. A Generic Platform for Addressing the Multimodal Challenge. In: Proceedings of CHI'95, ACM Press, Denver.
[89] Oktaba, H., Esquivel, C.A., et al., 2005. Modelo de Procesos para la Industria del Software MoProSoft, Versión 1.3. (in Spanish)
[90] Ouadou, K. 1994. AMF : Un modèle d'architecture multi-agents multi-facettes pour Interfaces Homme-Machine et les outils associés. Thèse de doctorat, Ecole Centrale de Lyon. (in French)
[91] Pfaff, G. E., (1985). User interface management system. Springer-Verlag.
[92] Paternò, F., 2004. ConcurTaskTrees: An Engineered Notation for Task Models. In: Diaper, D., Stanton, N.A. (Eds.), The Handbook of Task Analysis for Human-Computer Interaction. Lawrence Erlbaum Associates, Mahwah, NJ, pp. 483–501.
[93] Paternò, F., Mancini, C., Meniconi, S., 1997. ConcurTaskTrees: A Diagrammatic Notation for Specifying Task Models. In: Howard, S., Hammond, J., Lindgaard, G. (Eds.), Human-Computer Interaction INTERACT'97: IFIP TC13 International Conference on Human-Computer Interaction. Springer US, Boston, MA, pp. 362–369.
[94] Paulk, M. C., Weber, C. V., Curtis, B., Chrissis, M. B., 1995. The Capability Maturity Model: Guidelines for Improving the Software Process. Reading, MA: Addison-Wesley.
[95] Peres, A.L., Da Silva, T.S., Silva, F.S., Soares, F.F., De Carvalho, C.R.M., Meira, S.R.D.L., 2014. AGILEUX model: towards a reference model on integrating UX in developing software using agile methodologies. In: Agile Conference, IEEE, New York, NY, pp. 61–63 .
[96] Prates, R.O., Souza, C.S., Barbosa, S.D.J., 2000. A method for Evaluating the Communicability of User Interfaces. Interactions, 7 (1) 31-38.
[97] Raza, A., Capretz, L.F., Ahmed, F., 2012. An open source usability maturity model (OS-UMM). Computers in Human Behavior, 28, pp. 1109-1121.
[98] Rauch, T.L., Flanagan, G.A., 1995. Usability Management Maturity, Part 2: Usability Techniques – What Can You Do?, Special Interest Group. In: Conference on Human Factors in Computing Systems CHI'95, ACM, Denver, Colorado, pp. 338.
[99] Rauch, T.L., Kahler, S., Flanagan, G.A., 1996. Usability Management Maturity, Part 2: Usability Techniques – What Can You Do?, SIGCHI Bulletin, Vol. 28, No 4, October.
[100] Ross, D.T., 1977. Structured Analysis (SA): A Language for Communicating Ideas. IEEE Transation Software Enginnering, 3, 16–34.
[101] Salah, D., 2011. A Framework for the Integration of User Centered Design and Agile Software Development Processes. In: Proceedings of the 33rd International Conference on Software Engineering, pp. 1132-1133.
[102] Salah, D., Paige, R., Cairns, P., 2016. A Maturity Model for Integrating Agile Processes and User Centred Design. In: Clarke, P.M., O'Connor, R.V., Rout, T., & Dorling, A. (Eds.), Software Process Improvement and Capability Determination: Proceedings of 16th International Conference, SPICE 2016, pp. 109-122.
[103] Salah, D., Paige, R., Cairns, P., 2015. Patterns for Integrating Agile Development Processes and User Centred Design. In Proceedings of the 20th European Conference on Pattern Languages of Programs, pp. 19:1-19:10.
[104] Salah, D., Paige, R., Cairns, P., 2014. An Evaluation Template for Expert Review of Maturity Models. In: A. Jedlitschka, P. Kuvaja, M. Kuhrmann, T. Männistö, J. Münch, & M. Raatikainen (Eds.), 15th International Conference Product-Focused Software Process Improvement, pp. 318-321.
[105] Sauro, J., Lewis, J.R., 2012. Quantifying the User Experience - Practical Statistics for User Research. Morgan Kaufmann, MA.
[106] Scapin, D.L., Bastien, J.M.C., 1997. Ergonomic criteria for evaluating the ergonomic quality of interactive systems. Behaviour & Information Technology, 16(4-5), 220-231.

[107] Scheiber, F., Wruk, D., et al., 2012. Software Usability in Small and Medium Sized Enterprises in Germany: An Empirical Study. In: A. Maedche, A. Botzenhardt, L. Neer (Eds.), Software for People - Fundamentals, Trends and Best Practices, pp. 39-52.
[108] Seffah, A. 2015. Patterns of HCI Design and HCI Design of Patterns - Bridging HCI Design and Model-Driven Software Engineering. Human-Computer Interaction Series, Springer International Publishing.
[109] Seffah, A., Desmarais, M. C., Metzker, E., 2005. HCI, Usability and Software Engineering Integration: Present and Future. In: Seffah, A., Gulliksen, J., & Desmarais, M.C. (Eds.), Human-Centered Software Engineering — Integrating Usability in the Software Development Lifecycle, pp. 37–57, Springer Netherlands.
[110] Shneiderman, B., Plaisant, C., 2010. Designing the User Interface: Strategies for Effective Human-Computer Interaction. MA: Pearson, Fifth Edition.
[111] Silva, F. S., Soares, F. S. F., Peres, A. L., Azevedo, I. M., Vasconcelos, A. P. L. F., Kamei, F. K., Meira, S. R. L., 2015. Using CMMI together with agile software development: A systematic review. Information and Software Technology, 58, 20–43.
[112] Silva, W., Valentim, N.M.C., Conte, T., 2015. Integrating the Usability into the Software Development Process - A Systematic Mapping Study. In: International Conference on Enterprise Information Systems, pp. 105-113.
[113] Silveira, M.S., Barbosa, S.D.J., Souza, C.S., 2004. Designing online help systems for reflective users. Journal of the Brazilian Computer Society, vol.9, no.3, 25-40.
[114] Souza, C.S., Leitão, C.F., Prates, R.O., Silva, E.J., 2006. The Semiotic Inspection Method. In: Proceedings of Brazilian Symposium on Human Factors in Computing Systems, pp. 148-157.
[115] SOFTEX, 2016. MPS.BR - Brazilian Software Process Improvement, MR-MPS-SW - General Guide, January 2016. Retrieved from http://www.softex.br/mpsbr. (in Portuguese)
[116] Staggers, N., Rodney, M., 2012. Promoting usability in organizations with a new health usability model: implications for nursing informatics. In: International Congress on Nursing Informatics, Montreal, American Medical Informatics Associa-tion, pp. 396–401.
[117] Sward, D., Macarthur, G., 2007. Making user experience a business strategy. In: Workshop on Towards a UX Manifesto, Lancaster, UK, pp. 35–40.
[118] Tarpin-Bernard, F., Samaan, K., David, B. 2009. Achieving Usability of Adaptable Software: The Amf-Based Approach. In: Seffah A., Vanderdonckt J., Desmarais M.C. (eds) Human-Centered Software Engineering. Human-Computer Interaction Series. Springer, London.
[119] Tidwell, J., 2010. Designing Interfaces: Patterns for Effective Interaction Design (2nd Edition). O'Reilly Media.
[120] Tidwell, J., 1999. A Pattern Language for Human-Computer Interface Design.
[121] Torrecilla-Salinas, C.J., Sedeño, J., Escalona, M.J., Mejías, M., 2014. Mapping Agile Practices to CMMI-DEV Level 3 in Web Development Environments. In: 23rd International Conference on Information Systems Development (ISD 2014), pp. 348-355.
[122] Vanderdonckt, J., 1995. Guide ergonomique des interfaces homme-machine. Belgique: Presses Universitaires de Namur. (in French)
[123] Van Tyne, S., 2009. Corporate user-experience maturity model. In: International Conference on Human Centered Design, Springer, Heidelberg, pp. 635–639.
[124] van Welie, M., van der Veer, G. C., 2003. Pattern Languages in Interaction Design: Structure and Organization. In: Human-Computer Interaction - INTERACT'03, pp. 527-534. IOS Press.
[125] Vasmatzidis, I., Ramakrishnan, A., Hanson, C., 2001. Introducing usability engineering into the Cmm model: an empirical approach. In: Human Factors and Ergonomics Society Annual Meeting, SAGE Publications, Los Angeles, CA, pp. 1748–1752.
[126] Veer, G.C. van der, Lenting, B.F., Bergevoet, B.A.J., 1996. GTA: Groupware task analysis — Modeling complexity. Acta Psychol. (Amst.) 91, 297–322.
[127] Vermeeren, A.P.O.S., Law, E.L., Roto, V., Obrist, M., Hoonhout, J., Väänänen-Vainio-Mattila, K., 2010. User Experience Evaluation Methods: Current State and Development Needs. In: Proceedings of NordiCHI'10, pp. 521-530.
[128] Wangenheim, C.G., Hauck, J.C.R., Salviano, C. F., von Wangenheim, A., 2010. Systematic Literature Review of Software Process Capability/Maturity Models. In: International SPICE Conference on Process Improvement and Capability dEtermination.

[129] Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., Wesslén, A., 2012. Experimentation in Software Engineering. Springer.
[130] Woltjer, R., Hollnagel, E., 2008. Modeling and evaluation of air traffic management automation using the functional resonance accident model (FRAM). In: 8th International Symposium of the Australian Aviation Psychology Association.
[131] World Wide Web Consortium, 2015. WAI Guidelines and Techniques. Web Accessibility Initiative (WAI). https://www.w3.org/WAI/guid-tech.html.
[132] World Wide Web Consortium, 2014. MBUI - Task Models. W3C Working Group Note. April 2014. https://www.w3.org/TR/task-models/.
[133] Yousuf, M. I., 2007. Using Experts' Opinions Through Delphi Technique. Practical Assessment, Research & Evaluation, Vol 12, No 4.

Biographies

Taisa Guidini Gonçalves is a member of the Computer Science Department in the LAMIH CNRS UMR 8201. She has a Ph.D. in Computer Science focused on Human-Computer Interaction at the University of Valenciennes (France), and she has a MSc. in Software Engineering at the Federal University of Rio de Janeiro. She specializes in Human-Computer Interaction and integration of HCI models and tools in Software Engineering models, particularly in the CMMI-DEV model.

Káthia Marçal de Oliveira is an associate professor at the University of Valenciennes (France) and a member of the Computer Science Department in the LAMIH. She has a Ph.D. in software engineering focused on quality assurance. She works on the integration of Human-Computer Interaction and software engineering issues.

Christophe Kolski is a Professor in computer science at the University of Valenciennes (France) and a member of the Computer Science Department in the LAMIH. He specializes in human–computer interaction, software engineering for interactive system design and evaluation, and adaptive and tangible user interface.

Author version, published in: Computer Standards & Interfaces, 58, pp. 53-86, 2018.

Appendix A. Questionnaire for interview

University of Valenciennes and Hainaut-Cambrésis (UVHC)
Laboratory of Industrial and Human Automation control, Mechanical engineering and Computer Science
(LAMIH UMR CNRS 8201)

Questionnaire for interview

Domaine: Methods, techniques, standards and patterns of Human-Computer Interaction Engineering

Taísa Guidini Gonçalves
Kathia Oliveira
Christophe Kolski

June 2015

Questionnaire of interview - Methods, techniques, standards, and patterns of Human-Computer Interaction Engineering

This interview aims to validate methods, techniques, standards, and patterns of HCI Engineering identified from an exploratory study. In this study an analysis was carried out of the Software Process Capability and Maturity Model (Capability Maturity Model Integration (CMMI-DEV) from the point of view of the issues of Human-Computer Interaction Engineering. Therefore, we analyzed five process areas/processes. Engineering process areas cover the development and maintenance activities that are shared across engineering disciplines. The five Engineering process areas in CMMI-DEV are as follows:

Requirements Development (RD)
Technical Solution (TS)
Product Integration (PI)
Validation (VAL)
Verification (VER)

From this analysis, we identified ten (10) groups of methods, techniques, standards, and patterns of HCI Engineering that were associated with the different processes areas analyzed. Each process area has different Specific Goals (SG) and these goals are associated with different Specific Practices (SP). ***Do you agree, partially agree or not agree with each proposition? If you partially agree or do not agree, justify your answer, please.***

Respondent information

Name:

Date:

Training and Profession:

The working period in the HCI area:

CMMI Model and Engineering Process Areas

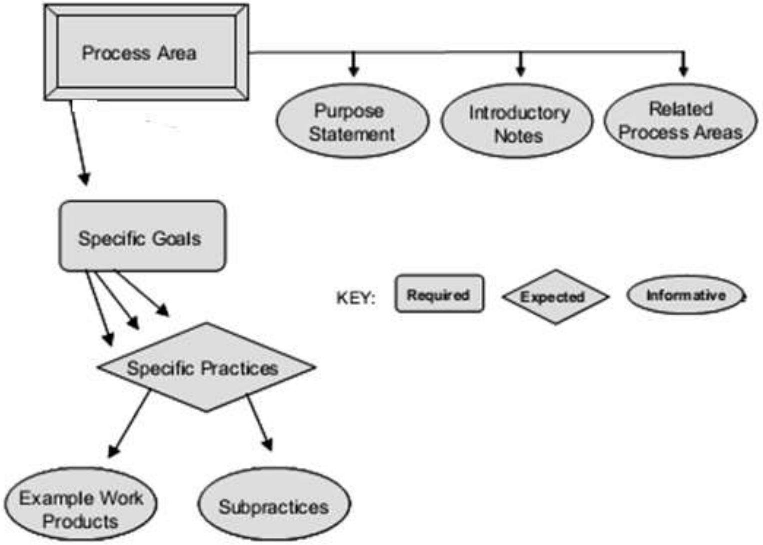


Figure 1. CMMI Model Components (CMMI Product Team, 2010)

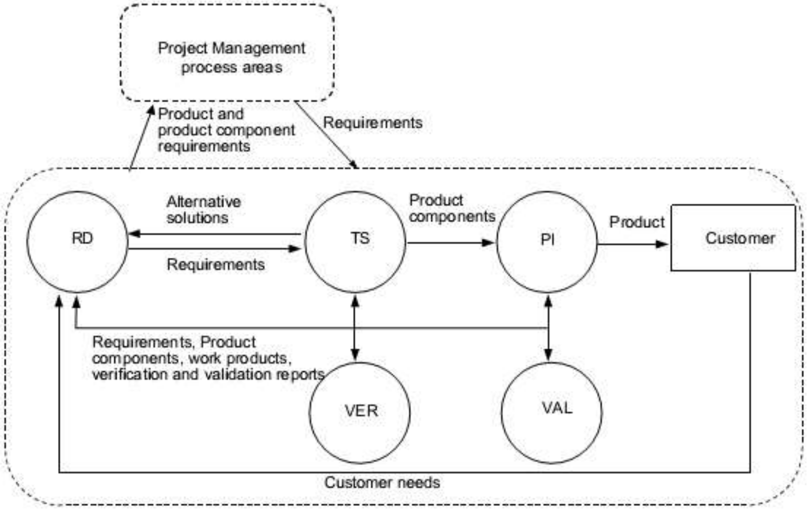


Figure 2. Engineering Process Areas (CMMI Product Team, 2010)

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 *SG 1 Develop Customer Requirements* 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.	**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) or SADT coupled with Petri Nets • GTA (Groupware Task Analysis)				
	SP 1.1 Elicit Needs Elicit stakeholder needs, expectations, constraints, and interfaces for all phases of the product lifecycle.	**Prototype for HCI requirements** Examples: • Rapid Prototyping Offline techniques: Paper and pencil (paper sketches, storyboards), Mockups, Wizard of Oz, Video prototyping Online techniques using software tools: No interactive simulations, Interactive simulations, Scripting languages				
Requirements Development	**SP 1.2 Transform**	**Task Analysis Methods for HCI**				

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	
		Examples: • CTT (Concur Task Tree) • K-MAD (Kernel of Model for Activity Description) • HTA (Hierarchical Task Analysis) • SADT (Structured Analysis and Design Technique) or SADT coupled with Petri Nets • GTA (Groupware Task Analysis)				
Requirements Development *SG 2 Develop Product Requirements* Customer requirements are refined and elaborated to develop product and product component requirements.	**SP 2.1 Establish Product and Product Component Requirement** Establish and maintain product and product component requirements, which are based on the customer requirements.	**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) or SADT coupled with Petri Nets • GTA (Groupware Task Analysis)				
Requirements Development *SG 3 Analyze and Validate Requirements* The requirements are analyzed and validated.	**SP 3.1 Establish Operational Concepts and Scenarios** Establish and maintain operational concepts and associated scenarios.	**Operational Concepts and Scenarios Specification for HCI** Examples: • Context awareness • Adapting to context • User profile • Persona • Use cases				
Requirements Development	**SP 3.2 Establish a Definition**	**Standards and Guidelines for design and documentation of HCI**				

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	
		Examples: • Ergonomic Criterion (Scapin and Bastien, 1993) • ISO/IEC 9126-1 (2001) • ISO 9241-11 (1998) • ISO/IEC 25000 (2014)				
Requirements Development *SG 3 Analyze and Validate Requirements* The requirements are analyzed and validated.	**SP 3.3 Analyze Requirements** Analyze requirements to ensure that they are necessary and sufficient.	**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) or SADT coupled with Petri Nets • GTA (Groupware Task Analysis)				
	SP 3.4 Analyze Requirements to Achieve Balance Analyze requirements to balance stakeholder needs and constraints.	**Techniques to validate HCI requirements** Examples: • Proto Task (K-MAD) • Task Model Simulator (CTT) • Focus Group to validate requirements				
	SP 3.5 Validate Requirements Validate requirements to ensure the resulting product will perform as intended in the end user's environment.	**Prototype for HCI requirements** Examples: • Rapid Prototyping Offline techniques: Paper and pencil (paper sketches, storyboards), Mockups, Wizard of Oz, Video prototyping Online techniques using software tools: No interactive simulations, Interactive simulations, Scripting languages				
Technical Solution *SG 1 Select Product Component*	**SP 1.1 Develop Alternative Solutions and Selection**	**Architecture Patterns for HCI**				

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	
		Examples: • MVC (Model-View-Controller) Model (Goldberg, 1983) • PAC (Presentation-Abstraction-Control) Model (Coutaz, 1987) • Arch Model (Bass et al., 1991)				
	SP 1.2 Select Product Component Solutions Select the product component solutions based on selection criteria.	**Operational Concepts and Scenarios Specification for HCI** Examples: • Context awareness • Adapting to context • User profile • Persona • Use cases				
Technical Solution *SG 2 Develop the Design* Product or product component designs are developed.	**SP 2.1 Design the Product or Product Component** Develop a design for the product or product component.	**Prototype for HCI requirements** Examples: • Rapid Prototyping Offline techniques: Paper and pencil (paper sketches, storyboards), Mockups, Wizard of Oz, Video prototyping Online techniques using software tools: No interactive simulations, Interactive simulations, Scripting languages				
	SP 2.1 Design the Product or Product Component Develop a design for the product or product component.	**Architecture Patterns for HCI** Examples: • MVC (Model-View-Controller) Model (Goldberg, 1983) • PAC (Presentation-Abstraction-Control) Model (Coutaz, 1987) • Arch Model (Bass et al., 1991)				
	SP 2.1 Design the Product or Product Component Develop a design for the product or product component.	**Standards and Guidelines for design and documentation of HCI** Examples: • Ergonomic Criterion (Scapin and Bastien, 1993) • ISO/IEC 9126-1 (2001) • ISO 9241-11 (1998) • ISO/IEC 25000 (2014)				
Technical Solution *SG 3 Implement the Product*	**SP 3.1 Implement the Design** Implement the designs of the	**Design patterns for HCI**				

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	
		Examples: • A Pattern Approach to Interaction Design (Borchers, 2001) • Pattern Languages in Interaction Design: Structure and Organization (van Welie and van der Veer, 2003) • Designing interfaces (Tidwell, 2010)				
	SP 3.2 Develop Product Support Documentation Develop and maintain the end-use documentation.	**Standards and Guidelines for design and documentation of HCI** Examples: • Ergonomic Criterion (Scapin and Bastien, 1993) • ISO/IEC 9126-1 (2001) • ISO 9241-11 (1998) • ISO/IEC 25000 (2014)				
Product Integration *SG 1 Prepare for Product Integration* Preparation for product integration is conducted.	**SP 1.1 Establish an Integration Strategy** Establish and maintain a product integration strategy.	**Prototype for HCI requirements** Examples: • Rapid Prototyping Offline techniques: Paper and pencil (paper sketches, storyboards), Mockups, Wizard of Oz, Video prototyping Online techniques using software tools: No interactive simulations, Interactive simulations, Scripting languages				
	SP 1.1 Establish an Integration Strategy Establish and maintain a product integration strategy.	**Functional Prototype to validate HCI** Examples: • Iterative and Evolutionary Prototypes User interface toolkits User interface builders User interface development environments				
Validation	**SP 1.1 Select Products for**	**Evaluation methods for HCI verification tests**				

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	
		Examples: • Usability tests Exploratory tests Assessment tests Validation or verification tests Comparison tests • Validation by HCI expert(s)				
	SP 1.1 Select Products for Validation Select products and product components to be validated and validation methods to be used.	**Functional Prototype to validate HCI** Examples: • Iterative and Evolutionary Prototypes User interface toolkits User interface builders User interface development environments				
	SP 1.2 Establish the Validation Environment Establish and maintain the environment needed to support validation.	**Evaluation methods for HCI verification tests** Examples: • Usability tests Exploratory tests Assessment tests Validation or verification tests Comparison tests • Validation by HCI expert(s)				
	SP 1.3 Establish Validation Procedures and Criteria Establish and maintain procedures and criteria for validation.	**Standards and Guidelines for design and documentation of HCI** Examples: • Ergonomic Criterion (Scapin and Bastien, 1993) • ISO/IEC 9126-1 (2001) • ISO 9241-11 (1998) • ISO/IEC 25000 (2014)				
Validation *SG 2 Validate Product or Product*	**SP 2.1 Perform Validation** Perform validation on selected	**Evaluation methods for HCI verification tests**				

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	
		Examples: • Usability tests ☐ Exploratory tests ☐ Assessment tests ☐ Validation or verification tests ☐ Comparison tests • Validation by HCI expert(s)				
	SP 2.2 Analyze Validation Results Analyze results of validation activities.	**Evaluation methods for HCI verification tests** Examples: • Usability tests ☐ Exploratory tests ☐ Assessment tests ☐ Validation or verification tests ☐ Comparison tests • Validation by HCI expert(s)				
Verification	**SP 1.1 Select Work Products**	**Evaluation methods for HCI verification tests**				

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	
		Examples: • Usability tests □ Exploratory tests □ Assessment tests □ Validation or verification tests □ Comparison tests • Validation by HCI expert(s)				
	SP 1.1 Select Work Products for Verification Select work products to be verified and verification methods to be used.	**Functional Prototype to validate HCI** Examples: • Iterative and Evolutionary Prototypes □ User interface toolkits □ User interface builders □ User interface development environments				
	SP 1.2 Establish the Verification Environment Establish and maintain the environment needed to support verification.	**Evaluation methods for HCI verification tests** Examples: • Usability tests □ Exploratory tests □ Assessment tests □ Validation or verification tests □ Comparison tests • Validation by HCI expert(s)				
	SP 1.3 Establish Verification Procedures and Criteria Establish and maintain verification procedures and criteria for the selected work products.	**Standards and Guidelines for design and documentation of HCI** Examples: • Ergonomic Criterion (Scapin and Bastien, 1993) • ISO/IEC 9126-1 (2001) • ISO 9241-11 (1998) • ISO/IEC 25000 (2014)				
Verification *SG 2 Perform Peer Reviews* Peer reviews are performed on	**SP 2.1 Prepare for Peer Reviews** Prepare for peer reviews of	**Evaluation methods for HCI review**				

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	
		Examples: • Heuristic evaluation • Cognitive walkthrough • Groupware walkthrough				
	SP 2.2 Conduct Peer Reviews Conduct peer reviews of selected work products and identify issues resulting from these reviews.	**Evaluation methods for HCI review** Examples: • Heuristic evaluation • Cognitive walkthrough • Groupware walkthrough				
	SP 2.3 Analyze Peer Review Data Analyze data about the preparation, conduct, and results of the peer reviews.	**Evaluation methods for HCI review** Examples: • Heuristic evaluation • Cognitive walkthrough • Groupware walkthrough				
Verification *SG 3 Verify Selected Work Products* Selected work products are verified against their specified requirements.	**SP 3.1 Perform Verification** Perform verification on selected work products.	**Evaluation methods for HCI verification tests** Examples: • Usability tests ☐ Exploratory tests ☐ Assessment tests ☐ Validation or verification tests ☐ Comparison tests • Validation by HCI expert(s)				
	SP 3.2 Analyze Verification Results Analyze results of all verification activities.	**Evaluation methods for HCI verification tests** Examples: • Usability tests ☐ Exploratory tests ☐ Assessment tests ☐ Validation or verification tests ☐ Comparison tests • Validation by HCI expert(s)				

Other suggestions: