HCI Engineering Integrated with Capability and Maturity Models

Taisa Guidini Gonçalves, Káthia Marçal de Oliveira, Christophe Kolski

HCI Engineering Integrated with Capability and Maturity Models

Taisa Guidini Gonçalves
UVHC, LAMIH CNRS UMR 8201
59313 Valenciennes, France
taisa.guidinigoncalves@etu.univ-valenciennes.fr

Káthia Marçal de Oliveira
UVHC, LAMIH CNRS UMR 8201
Le Mont Houy cedex 9
59313 Valenciennes, France
kathia.oliveira@univ-valenciennes.fr

Christophe Kolski
UVHC, LAMIH CNRS UMR 8201
Le Mont Houy cedex 9
59313 Valenciennes, France
christophe.kolski@univ-valenciennes.fr

Abstract
It is well known that despite the large advancement of Human-Computer Interaction (HCI) engineering as regards the definition of methods, techniques and standards, the majority of enterprises do not use them in practice. Contrary to this scenario, software engineering has been widely applied in industry through the implementation of Software Process Capability and Maturity (SPCM) models. We argue that the definition of a workbench of approaches from HCI that could really support SPCM models can promote a largely use of HCI engineering in industry. Based on this belief we conducted an exploratory study to identify what SPCM models refer in terms of HCI Engineering. We identified explicit and implicit citations that could be interpreted in benefit of HCI engineering. Based on this analysis, we identified a set of methods, techniques and standards that should be used for the definition of guidelines to apply HCI engineering in the implementation of SPCM models.

http://dx.doi.org/10.1145/2820619.2825017

Author Keywords
Human-computer interaction engineering, Software process improvement, Software Process Capability and Maturity Model.

ACM classification Keywords
H.5.2. User Interfaces (Evaluation/Methodology).

Introduction
Software process capability and maturity (SPCM) models are nowadays well established in industry [18]. These models are collection of software engineering best practices that help organization to improve their software process. The large number of official appraisals using these models indicates therefore that software engineering practices are actually used in industry. For instance, more than 10,000 official appraisals [2] using CMMI (Capability Maturity Model Integration) [1], an international SPCM model for software engineering domain most known in the world,

SG 1 Develop Customer Requirements
SP 1.1 Elicit Needs
SP 1.2 Transform Stakeholder Needs into Customer Requirements
SG 2 Develop Product Requirements
SP 2.1 Establish Product and Product Component Requirements
SP 2.2 Allocate Product Component Requirements
SP 2.3 Identify Interface Requirements
SG 3 Analyze and Validate Requirements
SP 3.1 Establish Operational Concepts and Scenarios
SP 3.2 Establish a Definition of Required Functionality and Quality Attributes
SP 3.3 Analyze Requirements
SP 3.4 Analyze Requirements to Achieve Balance
SP 3.5 Validate Requirements

Figure 1. Requirements Development process area [1]

are reported from over 40 countries. Other national SPCM models (such as the MR-MPS-SW Brazilian model [17] and the MoProSoft Mexican model [14]) are also being largely used in industry; e.g., there are more than 600 officially appraisals on the national standard (MR-MPS-SW) created in 2005. One could suppose that with this large use of SPCM, the practices strongly defended of the community of Human-Computer Interaction (HCI) were also used while using SPCM in the development of interactive systems. Nevertheless, it is well-known that Human-Computer Interaction (HCI) approaches are not or insufficiently used in a large number of enterprises.

We argued that HCI engineering is inherently related to the software engineering while applied for the interactive systems projects. Jokela and Lalli [12] point out, for example, that several process areas from CMMI have a direct relationship with usability practices, and, therefore, HCI engineering. Based on this belief we started a research about how to support the use of SPCM for the development of interactive systems by clearly defining which approach from HCI could be used in each moment.

To address this goal, it is essential (1) to identify a could support the implementation of SPCM for the development of interactive systems and, then, (2) to define guidelines to support their use integrated to SPCM models. This paper presents the research we performed to answer the goal (#1) that it will be the basis to support the goal (#2). Next section presents a brief overview of the SPCM models. Then, we describe our study of SPCM models documentation looking for references to HCI engineering and some preliminary results. Some related works relevant for our study are then presented. We finish this paper by describing our on-going works.

Software Process Capability and Maturity Models

Software process capability and maturity models aim to support organizations to define and continually improve their process using software engineering best practices. In the last two decades several SPCM models have been developed. Wangenheim *et al.* [18] identified 52 models that cover different domains (such as software engineering, e-commerce, security). In their study Wangenheim *et al.* [18] concluded that 50 from the 52 identified SPCM models are defined based on CMMI.

The core element of CMMI is the process area. A process area is a cluster of related practices in an area that, when implemented collectively, satisfies a set of goals considered important for making significant improvement in that area. CMMI for Development (CMMI-DEV) [1] version 1.3 consists of 22 process areas organized in four categories: engineering, support, project and process management. A process area has 1 to 4 specific goals (SG). Each specific goal (required component) is composed of specific practices (SP). They describe software engineering best practices that are specific to a single process area. Each specific practice (expected component) describes the activities that are important in achieving a required CMMI component, and is described with informative components: subpractices, notes, references, goal titles, practice titles, sources, example box, and work products. CMMI argues that these elements are not required but it is suggested to use them when it is relevant. Figure 1 presents specific goals (SG) and practices (SP) for the process area Requirements Development (RD).

Several national SPCM models have also been defined such as the Brazilian (MR-MPS-SW [17]) and the Mexican (MoproSoft [14]) models. The MR-MPS-SW is organized in 20 software processes and a set of outcomes for each process. It is completely compatible with CMMI [17]. Considering this panorama, we choose to work with CMMI-DEV since it is the most known SPCM model. As consequence, we also analyzed the Brazilian model thanks to its compatibility to CMMI-DEV. For reason of space we will describe in the next section only the study of CMMI-DEV.

CMMI-DEV According to HCI Point of View

Analyzing the categories of CMMI-DEV presented in the previous section, we focus our study only to engineering category since the process areas from other categories are more generic and serve as support or management activities for any process or software. The engineering category is composed of five process areas: Requirement Development (RD), Technical Solution (TS), Product Integration (PI), Verification (VER) and Validation (VAL).

We analyze all documentation considering the complete description of each practice (expected and informative components) since our goal was to investigate all indication of any citation that should considered HCI while being applied in the development of interactive systems. To analyze the documentation, we initially seek explicit citations of HCI engineering by looking for: (i) HCI keywords (for example, external interface, end user, prototype); (ii) examples of techniques or methods of HCI (e.g. end-user task analysis, HCI models); and (iii) examples of work products (e.g. interface design specifications, user manual). Then, we looked for citations that were not directly related to HCI Engineering but that we could interpret in benefit of the use of it while in the development of an interactive system. We classify this information as implicit citations. Explicit and implicit citations were highlighted in the text and reviewed together in-group in a final reading.

SP 1.1 Elicit Needs

Elicit stakeholder needs, expectations, constraints, and interfaces for all phases of the product lifecycle.

Eliciting goes beyond collecting requirements by proactively identifying additional requirements not explicitly provided by customers. Additional requirements should address the various product lifecycle activities and their impact on the product.

Examples of techniques to elicit needs include the following:

- Technology demonstrations
- Interface control working groups
- Technical control working groups
- Interim project reviews
- Questionnaires, interviews, and scenarios (operational, sustainment, and development) obtained from end users
- Operational, sustainment, and development walkthroughs and end-user task analysis
- Quality attribute elicitation workshops with stakeholders
- Prototypes and models

Figure 2. Analysis of Requirements Development process area (extract from [1])

Figure 2 shows an example of explicit citation identified for the process area Requirements Development (RD). We note the importance of involving the end-user in the requirements elicitation and the use of techniques that are related to HCI for this purpose. The citations are highlighted as explicit since it mentions the words "end user", "end-user tasks analysis" and "prototypes and models" which are methods used in HCI.

Several implicit citations can also be found. For instance, the need of quality attributes requirements in the context of the development of interactive system (in requirement development SP1.2) naturally implies to consider usability (ISO 9241-11 [7]), the quality attribute most explored for HCI. Moreover, validation and verification methods suggest the use of specific methods for evaluation of interactive systems (such as Usability tests, Validation by HCI expert, Heuristic evaluation, Cognitive walkthrough, Groupware walkthrough [4],[5],[13]).

We found some references to HCI engineering for all process areas in almost all specific practices. Some practices that we did not found concerned specific issues of software engineering, such as: the allocation of functional requirements to components showing no relationship with usability requirements; the interface and integration between two functional components.

Identification of HCI Approaches to Support CMMI Implementation

After identifying all citations, we organized them separately to identify the main approaches related to HCI engineering. Figure 3 presents a summary of this analysis, where we show the main citations (implicit and explicit) for each process area, such as cited in second section. The approaches findings are connected by arrows to indicate which citations were considered in the identification of each approach.

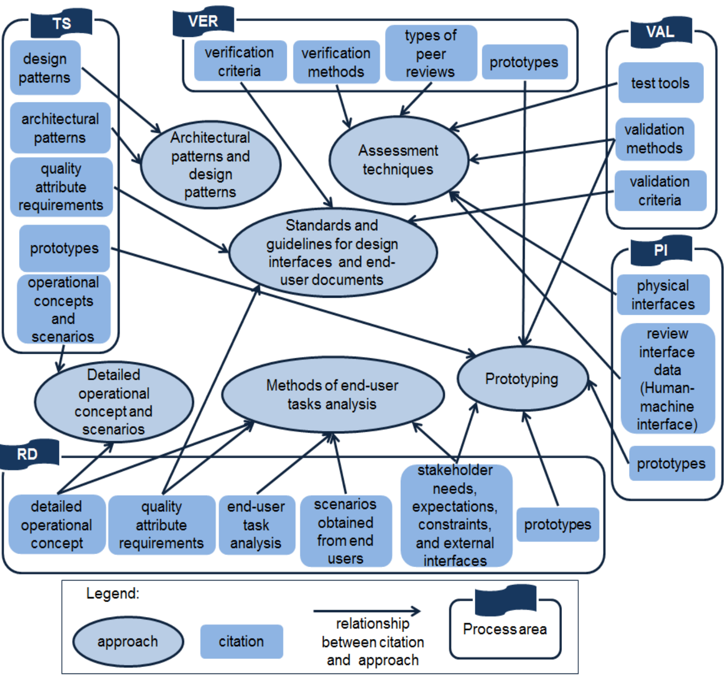

Figure 3. Analysis of the citations (implicit and explicit)

We note that we identified six groups of approaches: prototyping, assessment techniques, methods for end-user tasks analysis, detailed operational concept and scenarios, architectural patterns guidelines for interfaces design and end-user documents. For example, if we analyze Figure 3, prototyping is suggested for three process areas as follows: (a) Requirement Development (identify needs, expectations, constraints, and customer interfaces (RD SP 1.1); validate requirements (RD SP 3.5)); (b) Verification and Validation (verify and validate the HCI (VAL SP 1.1, VER SP 1.1)).

After this first analysis, we reviewed these approaches by looking in the literature and based in our own experience in HCI. As result, we refined two categories. The first one was prototyping, that we preferred to separate in prototypes used for requirements elicitation (that can be, for instance, mock-ups) and functional prototypes for validation (considered as prototypes since there are not ready to be used). The second one was the assessment methods. Following the software engineering classical classifications, we refine these methods in two groups: one related to verification and the other one with validation.

Each one of this final set of HCI approaches was described considering its purpose, examples of technique/methods/standards and to which specific practices should be used when implementing CMMI-DEV for the design of interactive systems as exemplified in Table 1. We had at the end ten categories of methods, techniques or standards as follows: Task Analysis Methods, Architecture Patterns, Design Patterns, Operational Concept and Scenario Specification, Prototype for HCI requirements, Techniques to validate HCI requirements, Standards and Guidelines for design and documentation of HCI, Evaluation methods (tests) for HCI, Functional Prototype to validate HCI, and Evaluation methods (review) for HCI.

Experts Review

This initial set was submitted for evaluation to 7 experts in HCI. Our goal was to confirm that the identified approaches can really be used to support the practices of the SPCM while one develops an expert system. They experts had different background and experience from eight to thirty years in the domain. They were interviewed using a form where they should answer if they agree that the proposed approach could support each practice of the SPCM. They should answer in a three-point scale ("I Agree", "I partially agree", "I do not agree") and justify their answer. They could also suggest new approaches to be used. Every meeting took in average 1hour. We start by present the

Process Area - Practice	Methods, techniques and standards of HCI Engineering Support
RD - SP 1.1 RD - SP 1.2 RD - SP 2.1 RD - SP 3.3	**Task Analysis Methods for HCI** **Purpose**: To use task analysis methods with the goal to identify stakeholder needs, expectations, constraints and interfaces. **Examples:** CTT (Concur Task Tree), MAD (Model for Activity Description), HTA (Hierarchical Task Analysis), GTA (Groupware Task Analysis)

Table 1. Example of HCI Engineering to support CMMI

structure of SPCM and our goal. Then they should read and answer his/her opinion. At the same time they could ask clear out doubts and explain their opinion.

As results we found: (a) 2 practices with 100% of agreement ("I agree"); (b) 26 practices with at least one "I partially agree"; (c) 16 practices with at least one to three maximum of "I do not agree". Based on these results, some modifications were performed such as: (a) the inclusion of new examples for some approaches; (b) the identification of new specific practices (among the 27 practices identified) where the approaches could be used; (c) the identification a new category for the elicitation of UI requirements using techniques likes brainstorming, focus group, and questionnaires.

Related Work

To our knowledge there is no work that integrates HCI practices to SPCM models. Nevertheless, we can find several works that relate software engineering and HCI and also specific maturity models for usability. Concerning the integration of software engineering and HCI, Ferre *et al.* [6] defined a framework that integrates practices of usability in the software process. To that end they identified the main activities of a user-centered software process (such as: specification of the context of use, usability specifications, prototyping, and usability evaluation) and a set of 35 techniques to support these activities. This work is very relevant and it is similar to ours while considering the integration of usability techniques in the software process but in our case, we intend to be more generic and integrate HCI engineering techniques in the practices of SPCM.

Raza *et al.* [16] proposed a usability maturity model for open-source projects. They defined a model composed of five maturity levels with the goal of identify the usability level of an enterprise while developing an open source project. This work focused on the usability activities but not integrated explicit guidelines for improving the development of interactive systems as we propose.

Jokela [10] and Jokela *et al.* [11] analyzed different usability capability maturity models existing in the literature. In general these models are based and use the structure of the CMM (Capability Maturity Model) [15], CMMI [1], ISO 13407 [8] and ISO/IEC 5504 [9]. These models can be classified into four categories [10]: standard process assessment models, non-standard models, generic models, and specific models. In their analysis, the authors [10] identified that the maturity models cover different organizational areas such as: performance of usability processes (user analysis, task analysis, usability evaluations, …), management of usability processes in development projects. The difference between our proposal and existing models is that our proposal will integrate explicitly HCI engineering practices in a specific SPCM, for the development of interactive systems. Our proposal will take into account the different HCI quality characteristics to integrate the HCI engineering practices.

Conclusion

This article has presented a study of software process capability and maturity models that aims to identify how the HCI engineering can support engineering practices advocated by these models. As a result a set of groups of methods, techniques and standards of HCI engineering were identified as support for the implementation of these models for the design of interactive systems. Our next step includes: (1)

investigate with professionals who work with the implementation of theses models in industry if they know and/or use these approaches, (2) define guidelines of using the approaches identified to support each practice from the SPCM. To address this last point we intend to consider the related works presented previously that considers usability and user-centered design processes.

Acknowledgements

The authors would like to thank CAPES - Science without Borders Program for the financial support to this work.

References

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

[2] CMMI Product Team. 2015. Maturity Profile Report, August 2015. Retrieved September, 2015 from http://cmmiinstitute.com/resources/.

[3] A. Sears and J.A. Jacko (Eds.). Human-Computer Interaction: Development Process, CRC Press, (2009).

[4] Cockton, G., Woolrych, A. and Lavery, D. Inspection-based evaluations. In [3], 273-292.

[5] Dumas, J. S. and Fox, J. E. Usability testing: Current practice and future directions. In [3], 231-250.

[6] Ferre, X., Juristo, N. and Moreno, A. M. Framework for Integrating Usability Practices into the Software Process. In Proc. of PROFES'05, (2005), 202-215.

[7] ISO 9241-11. Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11: Guidance on usability (1998).

[8] ISO 13407. Human-Centred Design Process for Interactive System (1999).

[9] ISO/IEC 15504. Software Engineering – Process Assessment – Part 2: Performing an Assessment (2003).

[10] Jokela, T. Usability Maturity Models: Making your Company User-Centered. User Experience Magazine, 9 (1) (2010).

[11] Jokela, T., Siponen, M., Hirasawa, N. and Earthy, J. 2006. A survey of usability capability maturity models: implications for practice and research. Behaviour & Information Technology, 25 (3), (2006), 263-282.

[12] Jokela, T. and Lalli, T. Usability and CMMI: Does a higher maturity level in product development mean better usability? In Proc. ACM CHI '03 (2003), 1010-1011.

[13] Mahatody, T., Sagar, M. and Kolski, C. State of the Art on the Cognitive Walkthrough Method, Its Variants and Evolutions. International Journal of Human-Computer Interaction, 26(8), (2010), 741-785.

[14] Oktaba, H., Esquivel, C. A., Ramos, A. S., Martínez, A. M., Osorio, G. Q., López, M. R., Hinojo, F. L. L., López, M. E. R., Mendoza, M. J. O., Ordóñez, Y. F., Lemus, M. A F. Modelo de Procesos para la Industria del Software MoProSoft, Versión 1.3 (2005).

[15] Paulk, M., Weber, C., Garcia, S., Chrissis, M. and Bush, M. The Capability Maturity Model: Guidelines for Improving the Software Process. Reading, MA: Addison-Wesley (1995).

[16] Raza, A., Capretz, L. F. and Ahmed, F. An open source usability maturity model (OS-UMM). Computers in Human Behavior, 28 (2012), 1109–1121.

[17] SOFTEX. MPS.BR - Brazilian Software Process Improvement - Implementation Guide - Part 11: Implementation and Evaluation of MR-MPS-SW: 2012 together with the CMMI-DEV v1.3. (2012). Retrieved April, 2015 from http://www.softex.br, in Portuguese.

[18] Von Wangenheim, C. G., Hauck J. C. R., Salviano, C. F. and Von Wangenheim, A. Systematic Literature Review of Software Process Capability/Maturity Models. In Proc. of SPICE (2010).