



Integration of human-computer interaction engineering issues into software process capability maturity models

Taisa Guidini Gonçalves

► To cite this version:

Taisa Guidini Gonçalves. Integration of human-computer interaction engineering issues into software process capability maturity models. Human-Computer Interaction [cs.HC]. Université de Valenciennes et du Hainaut-Cambresis, 2017. English. NNT : 2017VALE0040 . tel-01729393

HAL Id: tel-01729393

<https://theses.hal.science/tel-01729393>

Submitted on 12 Mar 2018

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

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

Thèse de doctorat

Pour obtenir le grade de Docteur de l'Université de VALENCIENNES ET DU HAINAUT-CAMBRESIS

Sciences et Technologie, Mention : Informatique

Présentée et soutenue par Taisa, GUIDINI GONÇALVES.

Le 27/11/2017, à Valenciennes

Ecole doctorale :

Sciences Pour l'Ingénieur (SPI)

Equipe de recherche, Laboratoire :

Département d'Informatique

Laboratoire d'Automatique, de Mécanique et d'Informatique Industrielles et Humaines
(LAMIH UMR CNRS 8201)

Integration of Human-Computer Interaction Engineering issues into Software Process Capability Maturity Models

Intégration des questions d'Ingénierie de l'Interaction Homme-Machine dans les Modèles d'Aptitude et Maturité de Processus Logiciel

JURY

Président du jury

- Winckler, Marco. Professeur des Universités. Université de Nice Sophia Antipolis, France.

Rapporteurs

- Dupuy-Chessa, Sophie. Professeur des Universités. Université de Grenoble, France.
- Vanderdonckt, Jean. Professeur des Universités. Université Catholique de Louvain, Belgique.

Examineurs

- Cavalcanti da Rocha, Ana Regina. Professeur des Universités. Université Fédérale de Rio de Janeiro, Brésil.

Co-directeurs de thèse

- Kolski, Christophe. Professeur des Universités. Université de Valenciennes et du Hainaut-Cambrésis, France.
- Marçal de Oliveira, Kathia. Maître de Conférences (HDR). Université de Valenciennes et du Hainaut-Cambrésis, France.

Table of Contents

ACKNOWLEDGMENTS	VII
ABSTRACT	X
RESUME.....	XI
LIST OF FIGURES.....	XII
LIST OF TABLES.....	XIV
<i>GENERAL INTRODUCTION.....</i>	1
Motivation.....	2
Research issues.....	3
Research Objectives.....	4
Research Methodology	4
Organization of the thesis.....	6
<i>CHAPTER 1 – SOFTWARE ENGINEERING AND HUMAN-COMPUTER INTERACTION: BASIC CONCEPTS FOR THIS THESIS.....</i>	7
1.1. Introduction	8
1.2. HCI: some basic concepts	9
1.3. Human-Computer Interaction Standards.....	10
1.3.1. ISO 13407 and ISO 9241-210	10
1.3.2. ISO/TR 18529	11
1.3.3. ISO/PAS 18152 and ISO/TS 18152	12
1.4. Standards for software development and process improvement	12
1.4.1. ISO/IEC 12207	13
1.4.2. ISO/IEC 15504 and ISO/IEC 330XX	14
1.5. Software Process Capability Maturity Models.....	15
1.5.1. Capability Maturity Model and Capability Maturity Model Integration	16
1.5.1.1. Capability Maturity Model Integration for Development	16
1.5.2. Brazilian Software Process Improvement Program	18
1.5.2.1. MPS for Software reference model	19
1.6. Equivalence and synthesis of the models	21

1.7. Synthesis and Conclusion	22
-------------------------------------	----

CHAPTER 2 – STATE-OF-THE-ART: THE ALIGNMENT OF THE HUMAN-COMPUTER INTERACTION ENGINEERING AND SOFTWARE ENGINEERING..... 25

2.1. Introduction	26
-------------------------	----

2.2. Usability Capability/Maturity (UCM) Models	26
---	----

2.2.1. The evolution of the Usability Capability/Maturity models	26
2.2.1.1. The studies of Jokela et al. (Jokela (2001), Jokela et al. (2006) and Jokela (2010))	26
2.2.1.2. The study of Salah et al. (2014).....	29
2.2.1.3. The recently study of Lacerda & Wangenheim (2018)	29
2.2.1.4. Synthesis about the literature of Usability Capability/Maturity models	30
2.2.2. Non-standard process models	34
2.2.3. Standard process models	36
2.2.4. Generic models	39
2.2.5. Synthesis about some Usability Capability/Maturity propositions	43

2.3. Integration of Human-Computer Interaction Engineering and Software Engineering.....	43
--	----

2.3.1. Integration of HCI approaches in software development processes or standards	44
2.3.1.1. Process Models	44
2.3.1.2. User/human-centered design issues	44
2.3.1.3. Software process.....	45
2.3.1.4. Synthesis about the integration of HCI and SE	46
2.3.2. Systematic literature reviews.....	49
2.3.3. Towards the integration of HCI approaches in SPCM models.....	49

2.4. Human-Computer Interaction in practice.....	50
--	----

2.4.1. Knowledge about HCI approaches	51
2.4.2. Use of HCI approaches	56
2.4.3. Other aspects related to HCI in practice	59
2.4.4. Synthesis about HCI in practice	59

2.5. Synthesis and Conclusion	59
-------------------------------------	----

CHAPTER 3 – INTEGRATING HCI APPROACHES INTO SPCM MODELS 61

3.1. Introduction	62
-------------------------	----

3.2. Engineering Process Areas/Processes to be studied	63
--	----

3.3. Phase 1 - Study of the models	64
--	----

3.4. Phase 2 - Identification of HCI approaches	68
---	----

3.5. Phase 3 – Evaluation and improvement with experts	70
--	----

3.5.1. Planning the evaluation.....	71
3.5.2. Performing the Interviews	74
3.5.3. Analysis and Synthesis of HCI approaches	75
3.5.3.1. General Analysis.....	77

3.5.3.2. Analysis for Requirements Development (RD)	78
3.5.3.3. Analysis for Technical Solution (TS)	82
3.5.3.4. Analysis for Product Integration (PI)	84
3.5.3.5. Analysis for Verification (VER).....	85
3.5.3.6. Analysis for Validation (VAL)	86
3.5.3.7. Synthesis of Analysis	87
3.5.4. Threats of validity	89
3.6. Using HCI approaches to support the development of interactive systems that follow SPCM models	92
3.6.1. Requirements Development (RD)	92
3.6.2. Technical Solution (TS)	97
3.6.3. Product Integration (PI).....	99
3.6.4. Verification (VER)	100
3.6.5. Validation (VAL).....	103
3.7. Synthesis and Conclusion	104
 CHAPTER 4 – LONG-TERM VALIDATION IN ACADEMIC ENVIRONMENT	 107
4.1. Introduction	108
4.2. Study Context	109
4.2.1. The objective and the questions of the study.....	109
4.2.2. The HCI course in the Master Program	110
4.2.3. The project of the study	113
4.3. First iteration: Descriptive analysis	114
4.3.1. Students' profile	114
4.3.2. Question 1: To what extent has task modeling been applied in requirement specifications of typical interactive systems?.....	116
4.3.3. Question 2: Which are the methods used for task modeling?	117
4.3.4. Question 3: How detailed was the task modeling?	118
4.3.5. Question 4: Does the task modeling consider all profiles defined in the problem?	119
4.3.6. Question 5: Are the user profiles described in task modeling also described in use case diagrams? .	120
4.3.7. Question 6: What are the students' feedback considering the project subject, evaluation and pedagogical issues?	121
4.4. Second iteration: Descriptive analysis	124
4.4.1. Students' profile	125
4.4.2. Question 1: To what extent has task modeling been applied in requirement specifications of typical interactive systems?.....	127
4.4.3. Question 2: Which are the methods used for task modeling?	127
4.4.4. Question 3: How detailed was the task modeling?	128
4.4.5. Question 4: Does the task modeling consider all profiles defined in the problem?	129
4.4.6. Question 5: Are the user profiles described in task modeling also described in use case diagrams? .	129
4.4.7. Question 6: What are the students' feedback considering the project subject, evaluation and pedagogical issues?	131
4.5. Discussion and comparison of the two iterations.....	134

4.5.1. Question 1: To what extent has task modeling been applied in requirement specifications of typical interactive systems?	134
4.5.2. Question 2: Which are the methods used for task modeling?	137
4.5.3. Question 3: How detailed was the task modeling?	137
4.5.4. Question 4: Does the task modeling consider all profiles defined in the problem?	138
4.5.5. Question 5: Are the user profiles described in task modeling also described in use case diagrams? ..	138
4.5.6. Conclusion of the iterations' comparison	139
4.6. Threats of validity	139
4.7. Synthesis and Conclusion	140
 <i>CHAPTER 5 – SURVEY ABOUT THE PERCEPTION OF KNOWLEDGE AND USE OF HCI APPROACHES WITH SPCM MODELS CONSULTANTS.....</i>	<i>143</i>
5.1. Introduction	144
5.2. Studies context	145
5.2.1. Objective	145
5.2.2. Hypothesis	145
5.3. Empirical study in the Brazilian context	147
5.3.1. Instrument: a web questionnaire	147
5.3.2. Subjects and planning	148
5.3.3. Study execution and analysis of the results	150
5.3.3.1. Descriptive Data	150
5.3.3.2. Answering hypotheses	152
5.3.4. Discussion related to the literature	155
5.4. Empirical study in the international context	161
5.4.1. Instrument: a web questionnaire	161
5.4.2. Subjects and planning	161
5.4.3. Study execution and analysis of the results	163
5.4.3.1. Descriptive data	163
5.4.3.2. Analysis of the results	165
5.5. Threats of Validity Analysis	169
5.6. Synthesis and Conclusion	170
 <i>GENERAL CONCLUSION</i>	<i>173</i>
Contributions of this thesis	174
Limitations	175
Future works	176
 REFERENCES.....	179

ANNEXES.....	191
Annex A. Analysis of CMMI-DEV	191
A.1. Specific Practices of Requirements Development process area	191
A.2. Specific Practices of Technical Solution process area	193
A.3. Specific Practices of Product Integration process area	194
A.4. Specific Practices of Verification process area	195
A.5. Specific Practices of Validation process area	196
Annex B. Questionnaire for interview	197
Annex C. Evaluation questionnaire	208
Annex D. Questionnaire for Peer review	211
Annex E. Web Questionnaire	218
Annex F. Form of evaluation	226

“This is a love song
See where your heart is
Put it in the palm of your hand
You must offer
The most sincere love
The purest smile and the most fraternal look
The world needs
Know the truth
The past does not come back; we do not have a future and today is not over
So love more, hug more
Because we do not know how much time we have to breathe
Talk more, listen more
It is worth remembering that life is too short”
(Truths of the time - Thiago Brado)

From the original “Essa é uma canção de amor
Veja onde está o seu coração
Coloque-o na palma da mão
É preciso ofertar
O amor mais sincero
O sorriso mais puro e o olhar mais fraterno
O mundo precisa
Saber a verdade
Passado não volta; futuro não temos e o hoje não acabou
Por isso ame mais, abrace mais
Pois não sabemos quanto tempo temos pra respirar
Fale mais, ouça mais
Vale a pena lembrar que a vida é curta demais”
(Verdades do Tempo - Thiago Brado)

Acknowledgments

First of all, I would like to thank God for the gift of life and for the wisdom that has been given to me.

I would like also to thank my parents, Paulo and Maria Rita, for the support and participation that they always have in my personal and professional life. Thank you very much for everything. I love you. *Eu amo vocês.*

My thanks to my brothers Tiago and Tomás, and to my sister Talita. The best friends of a lifetime. Also, my thanks to Carolina and Mayara for the care and support.

Thanks to Emanuele, Yago, and Augusto for making my life more joyful with their smiles.

My thanks to my grandmothers Brasilina and Virginie for her love and affection.

I would like to thank the love of my life, Salah-Eddine. Thank you very much for your love, affection, dedication, support, encouragement, good teachings and good times. I love you to infinity. Together we are more.

My thanks to all my family and my sweetheart's family for the support, love and affection.

I would like also to thank my advisors, Kathia Oliveira and Christophe Kolski. With them I learned how to do research in an effective and passionate way. Thank you for being always available, for your help and valuable advice. Thank you also for all that you could bring me during these 3 years of doctorate. Thank you very much for everything that you learned me and for the good times.

Thanks to Kathia Oliveira for your friendship, affection, dedication and good times. I hope that the life holds us long years of friendship.

I particularly thank my “rapporteurs” Sophie Dupuy-Chessa and Jean Vanderdonckt for the interest that they have shown in my thesis and for the valuable and rigorous evaluation that they have done. I would like also to thank my examiners Ana Regina Cavalcanti da Rocha and Marco Winckler. Thank you very much for your contributions. I particularly thank Ana Regina Cavalcanti da Rocha for her support and encouragement throughout this journey.

My thanks to my Valenciennes' family (Adam, Ahlem, Ahmed, Ali, Amira, Aymen, Catalina, Elise, Fadoua, Jocelene, Jose, Kathia, Nadya, Nour, Patrick, Ryma, Saif, Salah-Eddine, Samy, Sandro, Shadab, Wassim, Yasmine) and to my friends of the University of Valenciennes. Thanks also to all my Brazilian friends. Thank you very much for the good times and for the friendship.

I would like to thank Guilherme Travassos for his support and contribution with one part of the experimental concepts.

Thanks to LAMIH colleagues and mainly to the InterA team (particularly Bako Rajaonah, Emmanuelle Grislin, Houcine Ezzedine, Rene Mandiau, Said Hanafi, Sophie Lepreux, Sylvain Piechowiak, Veronique Delcroix) for their hospitality and friendship.

My thanks to LAMIH for all financial support to participate in conferences and workshops.

Thanks to Corinne Aureggi, Isabelle Oliveira-Machado, Marlene Genevieve, Melanie Lecq, and Sylvie Rubens for their collaboration in administrative procedures.

My thanks to CAPES - Science without Borders Program, for the financial support to carry out this thesis.

Finally, my thanks to University of Valenciennes for the opportunity to learn more about research.

This thesis is dedicated to my parents.
They are and always will be the greatest influence in my life.

Abstract

Software process capability maturity (SPCM) models are currently widely used in industry. To perform the practices defined in these models, software engineering approaches are applied. We also have experienced a large definition of methods, techniques, patterns, and standards for the analysis, 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 SPCM models, this thesis proposes to integrate HCI issues (concepts of design, implementation, and evaluation of interactive systems) in the most known international SPCM model (CMMI-DEV – Capability Maturity Model Integration for Development) and in the Brazilian SPCM model (MR-MPS-SW – MPS for Software reference model). To that end, we have worked on (i) the identification of appropriate HCI approaches for each practice of the engineering advocated by these models, (ii) the evaluation and improvement of the identified HCI approaches with HCI experts, (iii) the validation of the proposition in an academic environment, and (iv) the conduction of two empirical studies about the perception of knowledge and use of HCI approaches in the industry. As a result, we got 14 categories of HCI approaches with examples of methods, techniques, patterns, and standards adequate to perform each practice of engineering activities of the both models when developing interactive systems. Moreover, the empirical study, in Brazilian industry, confirmed statistically that consultants of those SPCM models do not know and do not use HCI approaches as well as they know and use software engineering approaches.

Keywords: Software Process Capability Maturity model; Human-Computer Interaction (HCI); CMMI-DEV, MR-MPS-SW; HCI methods, patterns, techniques and standards.

Résumé

Les modèles d'aptitude et maturité de processus logiciel (AMPL) sont actuellement largement utilisés dans l'industrie. Pour exécuter les pratiques définies dans ces modèles, des approches d'ingénierie logicielle sont appliquées. On constate également une grande définition en termes de méthodes, techniques, patrons et normes pour l'analyse, la conception, la mise en œuvre et l'évaluation de systèmes interactifs, axés sur les questions d'Interaction Homme-Machine (IHM). Néanmoins, il est bien connu que les approches d'IHM ne sont pas largement utilisées dans l'industrie. Afin de profiter de l'utilisation des modèles AMPL, cette thèse propose d'intégrer les questions d'IHM (concepts de conception, mise en œuvre et évaluation de systèmes interactifs) dans le modèle international le plus connu (CMMI-DEV – *Capability Maturity Model Integration for Development*) et dans le modèle brésilien (MR-MPS-SW – *MPS for Software reference model*). À cette fin, nous avons travaillé sur (i) l'identification des approches de l'IHM appropriées pour chaque pratique de l'ingénierie préconisée par ces modèles, (ii) l'évaluation et l'amélioration des approches de l'IHM identifiées avec des experts en IHM, (iii) la validation de la proposition dans un environnement académique, et (iv) la réalisation de deux études empiriques sur la perception de la connaissance et l'utilisation des approches de l'IHM dans l'industrie. En conséquence, nous avons obtenu 14 catégories d'approches de l'IHM avec des exemples de méthodes, techniques, patrons et normes propres à réaliser chaque pratique des activités d'ingénierie des deux modèles lors du développement de systèmes interactifs. De plus, l'étude empirique avec l'industrie brésilienne a confirmé statistiquement que les consultants de ces modèles AMPL ne connaissent et n'utilisent pas ou peu les approches de l'IHM, comme ils connaissent et utilisent des approches d'ingénierie logicielle.

Mots-clés : Modèle d'Aptitude et Maturité de Processus Logiciel ; Interaction Homme-Machine (IHM) ; CMMI-DEV, MR-MPS-SW ; méthodes, patterns, techniques et standards de l'IHM.

List of Figures

Figure 1. Overview of the CMMI official appraisals (in orange) around the world.....	3
Figure 2. Research Methodology	5
Figure 3. Human-centered design activities (adapted from (International Organization for Standardization, 2010a))	11
Figure 4. Entity relationship diagram (adapted from (International Organization for Standardization, 2000))	12
Figure 5. Process groups of the ISO 12207 (adapted from (International Organization for Standardization/International Electrotechnical Commission, 2008b))	14
Figure 6. Software Process Capability Maturity Models	16
Figure 7. CMMI-DEV structure (adapted from (CMMI Product Team, 2010)).....	17
Figure 8. MPS model components (adapted from (Softex, 2016c))	19
Figure 9. MR-MPS-SW structure (adapted from (Softex, 2016b))	20
Figure 10. Equivalence between the models.....	22
Figure 11. The evolution of the studies performed by Jokela <i>et al.</i>	28
Figure 12. KESSU model (adapted from (Jokela, 2008)).....	34
Figure 13. Structure of OS-UMM.....	35
Figure 14. UMM-P Components (adapted from (Earthy, 1999))	36
Figure 15. Components of the ISO/TR 18529 (adapted from (International Organization for Standardization, 2000)).....	38
Figure 16. Components of the HFIPRA model (adapted from (Earthy, 2001)).....	38
Figure 17. Corporate User Experience Maturity Model (adapted from (Van Tyne, 2009))	42
Figure 18. UX maturity levels (adapted from (Chapman & Plewes, 2014))	43
Figure 19. Research Methodology	62
Figure 20. Relationship among engineering process areas (CMMI Product Team, 2010)	63
Figure 21. Process areas of the engineering category	64
Figure 22. Example of the CMMI model components (extract from (CMMI Product Team, 2010))..	65
Figure 23. Examples of citations for Requirements Development (extract from (CMMI Product Team, 2010)).....	66
Figure 24. Analysis of the citations (implicit and explicit) to all process area	69
Figure 25. Extract of a filled questionnaire.....	74
Figure 26. Interview sessions (adapted from (Gonçalves et al., 2016a))	75
Figure 27. Detailed results by process area.....	78
Figure 28. Quantitative results for Requirement Development (category and practice).....	79
Figure 29. Quantitative results for Technical Solution (category and practice)	83
Figure 30. Quantitative results for Product Integration (category and practice).....	84
Figure 31. Quantitative results for Verification (category and practice)	86
Figure 32. Quantitative results for Validation (category and practice).....	87
Figure 33. Research Methodology	108
Figure 34. Likert scale	110
Figure 35. Content of HCI course (adapted from (Gonçalves, Oliveira, & Kolski, 2017c))	111
Figure 36. General steps in the development and evaluation of the requirement specification.....	112
Figure 37. Five interconnected mixing stations (published in (Gonçalves et al., 2017c) and adapted from (Kolski, Sagar, & Loslever, 2004)	113

Figure 38. General students' profile	115
Figure 39. Results of Task modeling	116
Figure 40. Overall result of task modeling methods	117
Figure 41. Detailed result of task modeling methods	118
Figure 42. Task modeling details	118
Figure 43. User profiles found in task models	119
Figure 44. User profiles found in use case diagrams	120
Figure 45. General results - user profiles found in Use Case diagrams and in Task models	121
Figure 46. Students' opinion about the studied project.....	122
Figure 47. Students' opinion about the pedagogy.....	123
Figure 48. Students' opinion about the evaluation.....	124
Figure 49. Students' profile	126
Figure 50. Task modeling	127
Figure 51. Task modeling methods.....	128
Figure 52. Details of the task modeling methods.....	128
Figure 53. Details of the task modeling	129
Figure 54. User profiles in task models	129
Figure 55. User profiles in use case diagrams	130
Figure 56. User profiles found in Use Case diagrams and in Task models	131
Figure 57. Opinion of the students about the studied project	132
Figure 58. Opinion of the students about the pedagogy.....	133
Figure 59. Students' opinion about the evaluation.....	133
Figure 60. Comparison of the two iterations.....	135
Figure 61. Research methodology	144
Figure 62. Web Survey	149
Figure 63. Consultant profiles.....	151
Figure 64. Formation degree	151
Figure 65. SPCM models consultants' formation area	151
Figure 66. Distribution of work time	152
Figure 67. Web questionnaire	162
Figure 68. Profile of the consultants	163
Figure 69. Formation of the consultants	163
Figure 70. Formation area of the consultants.....	164
Figure 71. Consultants per country	164
Figure 72. Distribution of work time with implementations.....	164
Figure 73. Maturity levels.....	165
Figure 74. Results by category found for "perception of knowledge".....	166
Figure 75. Results to highest levels for "perception of knowledge".....	167
Figure 76. Results to lowest levels for "perception of knowledge".....	167
Figure 77. Results by category found for "Use"	168
Figure 78. Results to highest levels for "Use"	168
Figure 79. Results to lowest levels for "Use"	169

List of Tables

Table 1. Categories and Process areas from CMMI-DEV	18
Table 2. CMMI capability and maturity levels (CMMI Product Team, 2010).....	18
Table 3. MR-MPS-SW maturity levels (Softex, 2016c).....	20
Table 4. Equivalence between CMMI-DEV and MR-MPS-SW maturity levels	21
Table 5. Example of the mapping between MR-MPS-SW and CMMI-DEV (Softex, 2016b).....	23
Table 6. Usability Capability/Maturity models.....	32
Table 7. Dimensions and key usability factors of OS-UMM.....	35
Table 8. UMM-P processes.....	37
Table 9. Capability Areas and Foci of assessment (Jokela et al., 2006)	40
Table 10. Process attributes and management practices	40
Table 11. User experience capability maturity model (Sward & Macarthur, 2007).....	41
Table 12. Practices and maturity levels (Marcus et al., 2009)	42
Table 13. Synthesis of the works about integration	47
Table 14. Studies about the state of HCI in practice.....	53
Table 15. HCI techniques/methods used in practice	58
Table 16. Process areas/Processes	63
Table 17. Examples of the CMMI-DEV analysis	67
Table 18. HCI approaches to support CMMI-DEV	70
Table 19. HCI approaches x CMMI-DEV Practices.....	71
Table 20. List of Experts (adapted from (Gonçalves et al., 2016a))	73
Table 21. Mode and Median of the results (Agree - A, Partially agree - PA, Don't agree - DA)	76
Table 22. General results about experts' level of agreement by process area	77
Table 23. Example of analysis to RD SP1.1 and RD SP1.2	80
Table 24. Analysis of the suggested techniques (adapted from (Gonçalves et al., 2016a)).....	81
Table 25. HCI approaches to support CMMI-DEV after interviews with experts	88
Table 26. Categories x CMMI-DEV practices after interviews with Experts	90
Table 27. Information on Master's degree program	111
Table 28. Profiles and tasks	114
Table 29. Mode of each item of the group profile	116
Table 30. Results of Use Case Diagrams and profiles.....	120
Table 31. Recall: Suggestions of Approaches to Designing Interactive Systems.....	124
Table 32. Mode of the group profile	126
Table 33. Use Case Diagrams and profiles	130
Table 34. Mode of the groups – time for performance	136
Table 35. Mode of the groups – work investment	136
Table 36. Mode of the groups – freedom of action.....	136
Table 37. Mode of the groups – difficulty of the work.....	137
Table 38. Profiles found in Task models	138
Table 39. Profiles found in Use case diagrams	138
Table 40. Grade for projects with and without task modeling (grade 0 – 20)	139
Table 41. HCI categories x SE categories.....	146
Table 42. Partner Organizations.....	150
Table 43. SPCM Models and maturity levels	152

Table 44. Mean of the demographic data per variable.....	153
Table 45. Mean of the questions per variable	154
Table 46. Results of paired t-test	154
Table 47. Results of paired t-test for each item	155
Table 48. HCI and SE rank for “knowledge”	156
Table 49. HCI and SE rank for “use”	157
Table 50. Rank of the HCI techniques/methods used in the practice	160
Table 51. Partner enterprises to CMMI Institute	162
Table 52. Means for the data.....	165
Table 53. Suitable UML diagrams	217

General Introduction

Motivation

Human-Computer Interaction (HCI) and Software Engineering (SE) are important domains for the development of interactive systems. The quality of an interactive system is usually considered dependent of the user interface design and evaluation. That implies the use of adequate HCI and SE approaches, and adequate software development process that includes end-users throughout all the process.

Human-Computer Interaction Engineering has made great progress since the eighties defining engineering approaches (methods, techniques, standards and patterns) to support the design, implementation, and evaluation of interactive systems. We can cite methods for task analysis (e.g. MAD, DIANE+ or CTT (Diaper & Stanton, 2004)), architectures patterns for HCI (e.g. Arch Model (Bass et al., 1991); PAC (Presentation-Abstraction-Control) Model (Coutaz, 1987); MVC (Model-View-Controller) Model (Goldberg, 1984)), and techniques for the user interface evaluation (e.g. usability tests, cognitive walkthrough, automated evaluation systems (Nielsen, 1993); (Shneiderman, Plaisant, Cohen, & Jacobs, 2009) and (Ivory & Hearst, 2001)). ISO standards have also been defined to support the software development, such as ISO 13407 (International Organization for Standardization, 1999) and ISO/TR 18529 (International Organization for Standardization, 2000). From hereafter in this document we will refer as **HCI approach** any method, technique, standard or pattern from HCI domain used to support engineering activities in the development of interactive systems.

Several usability capability/maturity (UCM) models (e.g., Usability Maturity Model: Human Centeredness Scale (Earthy, 1998); KESSU Model (Jokela, 2008) and Open source usability maturity model (Raza, Capretz, & Ahmed, 2012)) have also been developed to propose practices to introduce usability or HCI concerns in industry. However, they not provide sufficient support to be applied in practice (Lacerda & Wangenheim, 2018). Similarly, it is well known that HCI approaches are not or are insufficiently used in industry ((Venturi, Troost, & Jokela, 2006); (Bevan, 2009); (Hao & Jaafar, 2011); (Scheiber et al., 2012) and (Ogunyemi, Lamas, Adagunodo, Loizides, & Rosa, 2016)).

On the contrary, software process capability maturity (SPCM) models are nowadays well established in the industry ((Estorilio, Vaz, Lisboa, & Bessa, 2015) and (Wangenheim, Hauck, Salviano, & von Wangenheim, 2010)). Some of the most known are the CMMI-DEV – Capability Maturity Model Integration for Development (CMMI Product Team, 2010), an international model, and other national SPCM models such as the MR-MPS-SW (Brazilian model (Softex, 2016c)); the MoProSoft (Mexican model (Oktaba et al., 2005) (Oktaba & Vázquez, 2008); and the maturity model for the Spanish software industry (Garzás, Pino, Piattini, & Fernández, 2013). These models are a collection of software engineering best practices, organized in process areas or process, which help companies to improve their software process. These best practices present “what” to do in the development/maintenance of software systems but they do not specify “how” to do it.

A large number of official appraisals using these models indicate that software engineering practices are actually used in industry. For instance, more than 10,000 official appraisals (CMMI Product Team, 2015) using CMMI – Capability Maturity Model Integration (CMMI Product Team, 2010) are reported covering more than 80 countries (see Figure 1). Other SPCM models created later go in the same direction: there are more than 600 officially appraisals on the MR-MPS-SW Brazilian model created in 2005 (Kalinowski et al., 2014); more than 300 organizations certified on the MoProSoft Mexican model (Trujillo, Oktaba, Ventura, & Torres, 2013); and 38 enterprises certified on the Spanish model (Garzás et al., 2013). To perform what is proposed in those models, well-known

software engineering approaches (methods, procedures, standards, tools, techniques, and so on) are chosen and applied.



Figure 1. Overview of the CMMI official appraisals (in orange) around the world¹

We believe that probably UCM models are not used in the industry because they are not known as the SPCM models. As consequence, we argue that one way to make HCI concerns reach the industry is to explicitly integrate HCI approaches to support the SPCM models that are already widely used in practice.

Research issues

The HCI Engineering is intrinsically linked to Software Engineering while applying for the interactive systems projects. Jokela and Lalli (Jokela & Lalli, 2003) point out that several process areas from CMMI-DEV – Capability Maturity Model Integration for Development (CMMI Product Team, 2010) have a direct relationship with usability practices, which imply HCI engineering. The usability engineering and software engineering shared common goals, such as (Helms, Arthur, Hix, & Hartson, 2006): understanding customer and user needs; transforming needs into system requirements; designing to satisfy those requirements and testing to assure their realization in the final product.

Moreover, several works have discussed Human-Computer Interaction life cycles ((Hix & Hartson, 1993); (Nielsen, 1993); (Mayhew, 1999); (Kolski, Ezzedine, & Abed, 2001) and (Lepreux, Abed, & Kolski, 2003)), and the integration of HCI and Software Engineering domains ((Gross, 2016); (Salah, Paige, & Cairns, 2016); (Nogueira & Furtado, 2013); (Jokela, 2008); (Nebe & Zimmermann, 2007) and (Seffah, Desmarais, & Metzker, 2005)) for performing usable and useful systems.

One can suppose that if the software engineering practices are used, HCI approaches will also be used, when necessary, in the development of interactive systems. However, as previously mentioned several studies show the HCI approaches are not or are insufficiently used in practice.

Considering this scenery and that SPCM models are widely used in the industry we raised the following research question for this thesis:

¹ Figure font: <http://cmmiinstitute.com/>

How to support the users of software process capability maturity models in the development of interactive systems with the use of HCI approaches?

We believe that the indication of HCI approaches that can support the application of SPCM models in the interactive systems development can facilitate the concrete application of HCI approaches in industry.

Research Objectives

The main objective of this thesis is to integrate HCI approaches to support the analysis, design, implementation, and evaluation of interactive systems when using software process capability maturity models. In this context, we chose to work with two SPCM models: the CMMI-DEV since it is the most known and used in several countries, and the MR-MPS-SW, the Brazilian model, since this thesis is financed by the Brazilian research council.

To achieve the main objective, specific research objectives were defined:

- To identify HCI approaches to support interactive system development that follows CMMI-DEV and/or MR-MPS-SW models;
- To evaluate and improve the identified HCI approaches with domain experts;
- To conduct empirical studies in academic and industrial contexts with the proposal.

Research Methodology

To address research objectives we developed a methodology composed of five phases presented in Figure 2.

In the first phase *Study of the models*, an in-depth analysis of the CMMI-DEV and MR-MPS-SW documentation was performed to identify where HCI approaches should be used to implement the practices of engineering process areas/ process since our goal is to focus on activities related to analysis, design, implementation, and evaluation of interactive systems. We also studied the HCI literature (second phase *Identification of HCI approaches*) to identify HCI approaches to support the implementation of the practices. The results of these phases were a set of HCI issues, a list of HCI approaches (methods, techniques, standards, and patterns) that were organized into HCI categories to support the practices of engineering. A first proposition of which HCI category should support the SPCM models engineering practices was also generated.

Once we have the proposition, we should validate it using some evaluation approach. According to (Helgesson, Höst, & Weyns, 2012) and (Salah, Paige, & Cairns, 2014a) 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 responsible is an expert in the domain that 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.

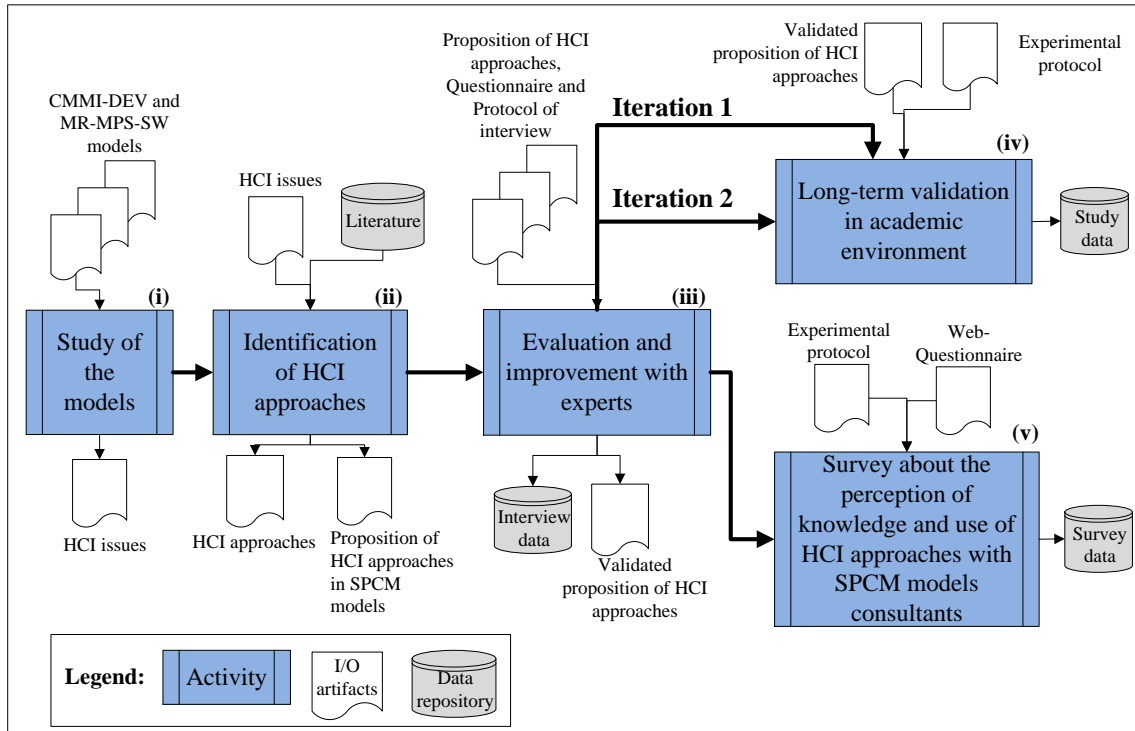


Figure 2. Research Methodology

Considering that the result of the second phase is an author proposition according to her knowledge and study (author evaluation), we decided to start by doing an evaluation with HCI experts. We were not expecting that all items in the proposition were perfectly correct, but that they could be used as start point for an evaluation by the experts and improved with other experts' suggestions. We planned, therefore, the third phase (*Evaluation and Improvement with experts*). To that end, we prepared a questionnaire where the experts should answer about their level of agreement with the proposition justifying their answer. The justifications would be used for the improvement of the proposition. Considering that we could have a lot of items to validate (since we associate HCI approaches with SPCM 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 HCI community (e.g., be program chair or member of program committee of HCI conferences). The results of the interviews were analyzed quantitatively and qualitatively in a way that we could integrate as much as possible all suggestions respecting the opinion of the experts. The detailed procedure and results of the first three phases will be presented in Chapter 3.

We argue that the experts' opinion is quite reliable to accept that the HCI approaches may support the practices of the SPCM models. However, we would like also to perform a practical setting evaluation. Aware that we could have a lot of propositions to validate since we were analyzing all practices of the engineering process areas we accepted that applying all the approaches would be a long-term validation. To confirm the effectiveness of each one of the propositions we should probably compare results of the application using the approaches and not using them. The practical setting evaluation requires therefore several projects requiring long-term studies what is not trivial in industry. As consequence, we planned to conduct two empirical studies.

The first one is an observational study represented in the fourth phase of our methodology named *Long-term validation in academic environment*. In this study we aim at validating the effective use of

approaches of one HCI category (Task modeling) in an academic environment (iterations 1 and 2). The details of this phase will be presented in Chapter 4.

The second study is a survey that aims to investigate the perception of knowledge and use of HCI approaches by SPCM models users related to their perception of knowledge and use of software engineering approaches. Considering that SPCM models are largely used in industry, which implies a large number of developers that use it; we decided to focus our investigation in a particular kind of users: the official consultants of both SPCM models. From our experience, the consultants usually introduce the use of approaches in the organizations when assisting the developers in the implementation of the SPCM models. Our goal with this study is to support our assumption that integrating HCI approaches into SPCM may help HCI concerns reach industry. This study is represented as the fifth phase of the methodology: *Survey about the perception of knowledge and use of HCI approaches with SPCM models consultants*. The procedure and the results of this phase will be presented in Chapter 5.

All these phases were supported by a continuous study of the literature that will be presented in chapters 1 and 2.

Organization of the thesis

This general introduction provided an overview of the thesis. It presented the motivations, research issues, objectives and research methodology. The structure of the chapters has also been delineated.

Chapter 1 will introduce the important fundamental concepts relevant to this thesis. These concepts are related to Human-Computer Interaction (HCI) Engineering, some ISO Standards, and Software Process Capability Maturity (SPCM) models.

Chapter 2 will present the state-of-the-art about Usability Capability/Maturity models, the integration of HCI and Software Engineering domains, and the use of HCI approaches in practice in the industry for the interactive systems development.

Chapter 3 will state the study related to the engineering practices present in the SPCM models and the HCI literature. It will also present the identification, validation and improvement of which HCI approaches support the development of interactive systems following SPCM models.

Chapter 4 will show how we have performed the long-term validation by presenting an observational study, performed in two iterations, about the use of approaches of one HCI category (Task modeling) in academic environment.

Chapter 5 will expose the results of two empirical studies about the perception of knowledge and use of HCI approaches by SPCM models consultants.

A **General Conclusion** summarizes the work carried out by identifying our main contributions and limitations in relation to the existing works and also present future research perspectives.

***Chapter 1 – Software Engineering and Human-Computer Interaction:
basic concepts for this thesis***

1.1. Introduction

Human-Computer Interaction (HCI) Engineering has been progressed over the last 30 years in the definition of models, methods, techniques and standards to support the analysis, design, implementation and evaluation of interactive systems. For instance, different software development life cycles as the star model (Hix & Hartson, 1993), the Nielsen's usability engineering life cycle (Nielsen, 1993) and the Mayhew's usability engineering life cycle (Mayhew, 1999) have been proposed. In addition, the classical life cycles of software engineering (such as, V-model (Thayer & McGettrick, 1993), spiral model (Boehm, 1988), etc.) have also been enhanced in terms of HCI (see for example (Kolski & Loslever, 1998); (Valentin, Valléry, & Lugongsang, 1993); (Kolski et al., 2001) and (Lepreux et al., 2003)).

Besides these life cycles, the HCI literature ((Courage & Baxter, 2005); (Sears & Jacko, 2009a) and (Sears & Jacko, 2009b)) offers methods, techniques and standards to support the implementation of all steps of these development life cycles. The ISO standards propose human-centered design (HCD) processes (e.g. ISO 13407 (1999), ISO/TR 18529 (2000) and ISO/TS 18152 (2010b)) that include the end-users in all development life cycle. Despite of this, we can cite methods and techniques for:

- Requirement analysis phase of process life cycles (e.g. task analysis methods (Diaper & Stanton, 2004); techniques to identify user needs and requirements (Courage & Baxter, 2005); prototyping techniques (Beaudouin-Lafon & Mackay, 2009));
- design and implementation phases (e.g. architecture patterns (Goldberg, 1984), (Coutaz, 1987) and (Bass et al., 1991); design patterns (Borchers, 2001), (van Welie & van der Veer, 2003), (Tidwell, 2010) and (Seffah, 2015));
- evaluation phase (e.g. techniques for validation and verification: usability tests (Shneiderman et al., 2009); standardized usability questionnaires (Sauro & Lewis, 2012) and (Assila, Oliveira, & Ezzedine, 2016); heuristic evaluation (Nielsen, 1994); cognitive walkthrough (Mahatody, Sagar, & Kolski, 2010) and, automated evaluation (Ivory & Hearst, 2001).

In parallel the Software Engineering (SE) community has used concretely in industry the models, methods, techniques and standards that have been developed for a long time. Software development life cycles or process models (e.g. the waterfall model (Royce, 1970), the V-model (Thayer & McGettrick, 1993), design methods (Céret, Dupuy-Chessa, Calvary, Front, & Rieu, 2013)) and ISO standards that provide processes for the development of a system or product have also been defined (e.g. ISO/IEC 12207 (2008b), ISO 25000 (2014)). We can also quote, for example, methods for analysis and design (e.g. UML diagrams), evaluation methods (e.g. formal review, product testing).

In particular, part of the diffusion of the SE domain in industry is due to the large dissemination of software process capability maturity (SPCM) models, for example: Capability Maturity Model Integration for Development – CMMI-DEV (CMMI Product Team, 2010) and MPS for Software reference model – MR-MPS-SW (Softex, 2016c), Brazilian model).

In this first chapter we propose an introduction about the basic concepts on top of which we developed this thesis. First, we present some basic concepts of HCI. Then, we briefly describe some ISO standards related to HCI and SE. Finally, we present the two SPCM models that are used in this thesis.

1.2. HCI: some basic concepts

Human-Computer Interaction (HCI) can be defined as the discipline responsible for the analysis, design, implementation and evaluation of interactive systems for human use (Preece, Sharp, & Rogers, 2015). This discipline has evolved since the 1980s and has built a rich literature that deals with different approaches (models, methods, techniques and standards).

In our work, we will use the **Human-Computer Interaction Engineering** terminology knowing that HCI Engineering is a branch of human knowledge that uses the approaches (models, methods, techniques and standards) of the Human-Computer Interaction discipline to build interactive systems².

Over time, the HCI discipline has evolved through various terminologies, such as: usability engineering (Nielsen, 1993), usability methods (International Organization for Standardization, 2002), process of human-system aspects (International Organization for Standardization, 2010b), human-centered design or user-centered design (International Organization for Standardization, 1999, International Organization for Standardization, 2010a), interaction design (Preece et al., 2015),

According to ISO 9241-210 (International Organization for Standardization, 2010a) **user-centered design** (UCD) is a way of designing interactive systems. The goal is to make the systems usable and useful having the users, their needs and requirements as central points of each phase of the design process.

Another important concept in HCI discipline is usability. **Usability** is “the degree to which a software can be used by specified consumers to achieve quantified objectives with effectiveness, efficiency, and satisfaction in a quantified context of use” (International Organization for Standardization, 2010a).

With the evolution of HCI domain, the **user experience** (UX) concept was created. The user experience can be translated as “a person’s perceptions and responses that result from the use or anticipated use of a product, system or service” (International Organization for Standardization, 2010a).

A new concept (**human-centered quality**) is under development. Its objective is to implement the processes of an organization so that the systems produced, acquired and operated have appropriate levels of accessibility, usability, user experience, and mitigation of risks that could arise from use (Bevan, Carter, Earthy, Geis, & Harker, 2016).

The development of an interactive system can be guided by processes that are normally focused on the user. In this way, the development processes should focus on activities related to user-centered design as suggested in ISO 9241-210 (International Organization for Standardization, 2010a).

For each step (or activity) of a user-centered development process, we need approaches (models, methods, techniques and standards) that focus on the user and allow us to have a usable interactive system. The HCI literature is rich in these approaches. For instance, Bevan (2003) presents a set of methods that can be used to support the user-centered design as described in ISO 13407 (International Organization for Standardization, 1999). Maguire (2001) also provides a set of methods to support human-centered design for each activity of the ISO 13407 (International Organization for Standardization, 1999).

² An interactive system is “the combination of hardware and software that exchanges data from and in the direction of a user, in order to help the user to perform his/her task” (International Organization for Standardization, 1999).

In addition, the ISO/TR 16982 (International Organization for Standardization, 2002) describes existing usability methods that can be used independently or in combination to ensure design and evaluation of a system. Guidance related to selection and use is provided, as well as guidance related to the life cycle phase (International Organization for Standardization, 2002). The main goal is to help project managers to make decisions about the choice of usability methods that support human-centered design (defined by ISO 13407).

1.3. Human-Computer Interaction Standards

The ISO standards of the Human-Computer Interaction (HCI) described in this section, propose processes and general frameworks to ensure the coherence, compatibility and quality of the development of human-centered systems. These standards focus on users and in the construction of usable solutions during the development.

1.3.1. ISO 13407 and ISO 9241-210

The ISO 13407 (International Organization for Standardization, 1999) provides a general framework for human-centered design activities that can be integrated into different processes throughout the life cycle of interactive systems. These activities are:

- plan the human-centered design process – a plan should be developed to specify how the human-centered activities can be placed in the global system development process;
- understand and specify the context of use – the characteristics of users, tasks, and organizational and physical environments define the context in which the system is used;
- specify the user and organizational requirements – the user and organizational requirements in relation to the description of the context of use are defined;
- produce design solutions – the design solutions are produced using the experience of the participants and the knowledge found in the literature, as well as the results of the context of use analysis; and
- evaluate designs against requirements – the evaluation must be performed at all stages of the system's life cycle.

This standard offers a description of each activity and its tasks, presenting a guide to select methods and techniques of human-centered design (International Organization for Standardization, 1999).

The need of a human-centered design approach is identified from the operational objectives of the system (e.g., the satisfaction of user requirements in terms of usability (International Organization for Standardization, 1999)). This standard has been developed as a set of processes that can be added to ISO/IEC 12207 to constitute a complete set of processes necessary for the development of interactive systems centered on the human (International Organization for Standardization, 2000).

The ISO 9241-210:2010 (International Organization for Standardization, 2010a) cancels and replaces the ISO 13407. This new version does not bring any changes related to the process level but just a technical revision. Requirements and recommendations related to the principles and human-centered design activities are provided in this standard. Its activities (see Figure 3) occur throughout the life cycle of the interactive systems.

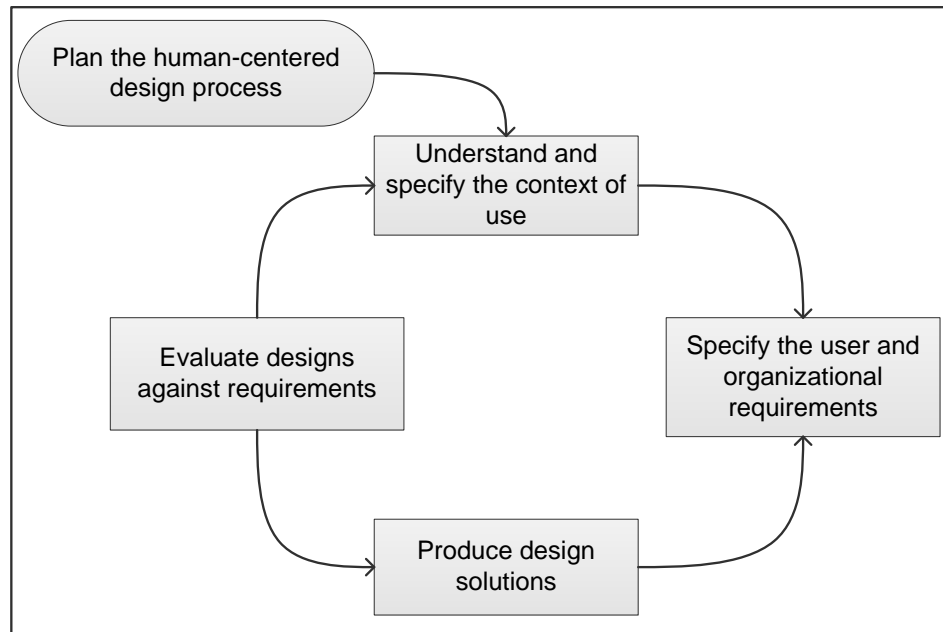


Figure 3. Human-centered design activities (adapted from (International Organization for Standardization, 2010a))

1.3.2. ISO/TR 18529

The ISO/TR³ 18529 (International Organization for Standardization, 2000) provides a model for the improvement and evaluation of human-centered processes, i.e., it extends and formalizes the human-centered processes defined in the ISO 13407 (International Organization for Standardization, 1999).

Seven processes for the development of human-centered systems are defined in this standard, where each process contains practices and uses/generates work products (see Figure 4). These practices describe what needs to be done to represent and include users of a system over the life cycle (International Organization for Standardization, 2000). In addition, the model has been developed in accordance with the ISO/IEC 15504 and an approach of evaluation can be carried out to determine the process capability of an organization.

Nowadays, a draft of international standard⁴ (ISO/DIS 9241-220.2 Ergonomics of human-system interaction – Part 220: Processes for enabling, executing and assessing human-centered design within organizations) is under development to replace the ISO/TR 18529.

According to Bevan et al. (2016) this draft of international standard (ISO/DIS 9241-220.2) is intended to provide a comprehensive description of the processes that support the activities required in the human-centered design. In this new version, the processes will be placed in four different areas (levels) of an organization. The process groups linked to each level are called Human-Centered Processes (HCP) Categories.

The implementation of these process categories can ensure that systems produced, acquired and operated by an organization have appropriate levels of accessibility, usability, user experience and risk mitigation that could result from its use (Bevan et al., 2016).

³ A Technical Report (TR) “is entirely informative in nature and does not have to be reviewed until the data it provides are considered to be no longer valid or useful” (International Organization for Standardization, 2000).

⁴ <https://www.iso.org/standard/63462.html>

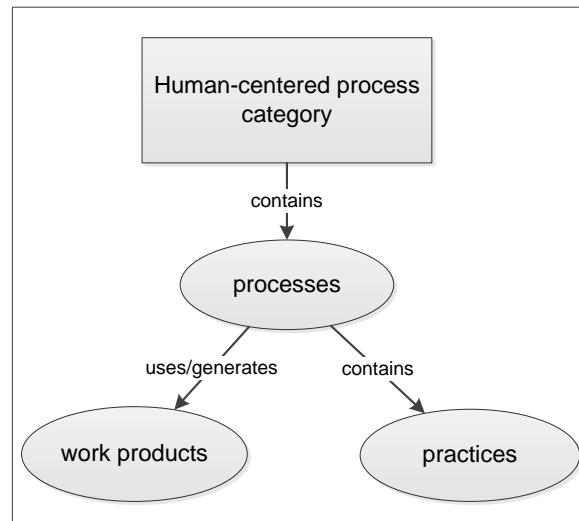


Figure 4. Entity relationship diagram (adapted from (International Organization for Standardization, 2000))

1.3.3. ISO/PAS⁵ 18152 and ISO/TS⁶ 18152

The standard ISO/PAS 18152 (International Organization for Standardization, 2003) describes the processes that deal with human-system (HS) problems and the results of these processes. It details the practices and work products that are associated with the results of each process. Its development has been done in accordance with ISO/IEC 15504 and an approach of evaluation can be carried out to determine the maturity of an organization in the execution of the processes.

Its more recent version, ISO/TS 18152 (International Organization for Standardization, 2010b) does not present many changes. The processes of this standard (the human-system process model or HS model) present a compilation of good practices in ergonomics/human factors, human/user-centered design, and integration of human factors from a range of industries into the whole world. In addition, the standard extends and formalizes the human-centered processes defined by ISO 13407 (International Organization for Standardization, 1999). Particularly, the “Human-Centered Design” process of this ISO, defines basic practices for the four activities of the ISO 13407 general framework.

An ISO/PAS or ISO/TS is reviewed after each three years in order to decide whether it will be confirmed for a further three years, revised to become an International Standard, or withdrawn (International Organization for Standardization, 2010b). If an ISO/PAS or ISO/TS is confirmed, it is reviewed again after a further three years, at which time it must either be transformed into an International Standard or be withdrawn (International Organization for Standardization, 2010b).

1.4. Standards for software development and process improvement

The software engineering standards⁷ described here are concerned with process to guide the development and management of the software, and process improvement. ISO/IEC 12207 proposes a

⁵ An ISO/Publicly Available Specification (ISO/PAS) “represents an agreement between technical experts in an ISO working group and is accepted for publication if it is approved by more than 50% of the members of the parent committee casting a vote” (International Organization for Standardization, 2010b).

⁶ An ISO/Technical Specification (ISO/TS) “represents an agreement between the members of a technical committee and is accepted for publication if it is approved by 2/3 of the members of the committee casting a vote” (International Organization for Standardization, 2010b).

set of processes for the software life cycle. ISO/IEC 15504 proposes the evaluation of these processes by looking at the capability of the process and the maturity of the organization.

1.4.1. ISO/IEC 12207

The standard ISO/IEC 12207 (International Organization for Standardization/International Electrotechnical Commission, 1995), which had its first version in 1995, provides a general framework of processes for the software development and software management. Following this version, in 2002 and 2004, improvements were made on the form of amendments (referred as Amendments 1 and 2, respectively). These improvements have brought many advantages over certain processes and their structure, as well as the representation of software engineering, the needs met by users of the standard and the harmonization with the ISO/IEC 15504 family.

Its current version, ISO/IEC 12207:2008 (International Organization for Standardization/International Electrotechnical Commission, 2008b), aims to establish a general framework for software life cycle processes. This framework consists of: processes, activities, tasks, goals and results, which have been proposed to be used throughout the software lifecycle (acquisition, provision, development, operation and maintenance of software products).

Figure 5 presents the process groups of this standard focusing on the system context – considering processes to support the agreement, project management and technical activities; and, on software context – that considers process for the implementation and reuse of software and process to support activities of the software process.

The ISO/IEC 12207:2008 does not define usability or a usability engineering process. However, the Appendix E of this standard describes how to create a process view for usability. A **process view** can be developed to organize the processes, activities and tasks selected from ISO/IEC 12207:2008 or ISO/IEC 15288:2008 (International Organization for Standardization/International Electrotechnical Commission, 2008c) in order to support a particular area so as to cover all or part of the life cycle (International Organization for Standardization/International Electrotechnical Commission, 2008b).

At the moment, a new version⁸ (ISO/IEC/IEEE FDIS 12207 Systems and software engineering – Software life cycle processes) of this standard is being developed.

⁷ Draft International Standards “adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote” (International Organization for Standardization, 2003).

⁸ <https://www.iso.org/standard/63712.html>

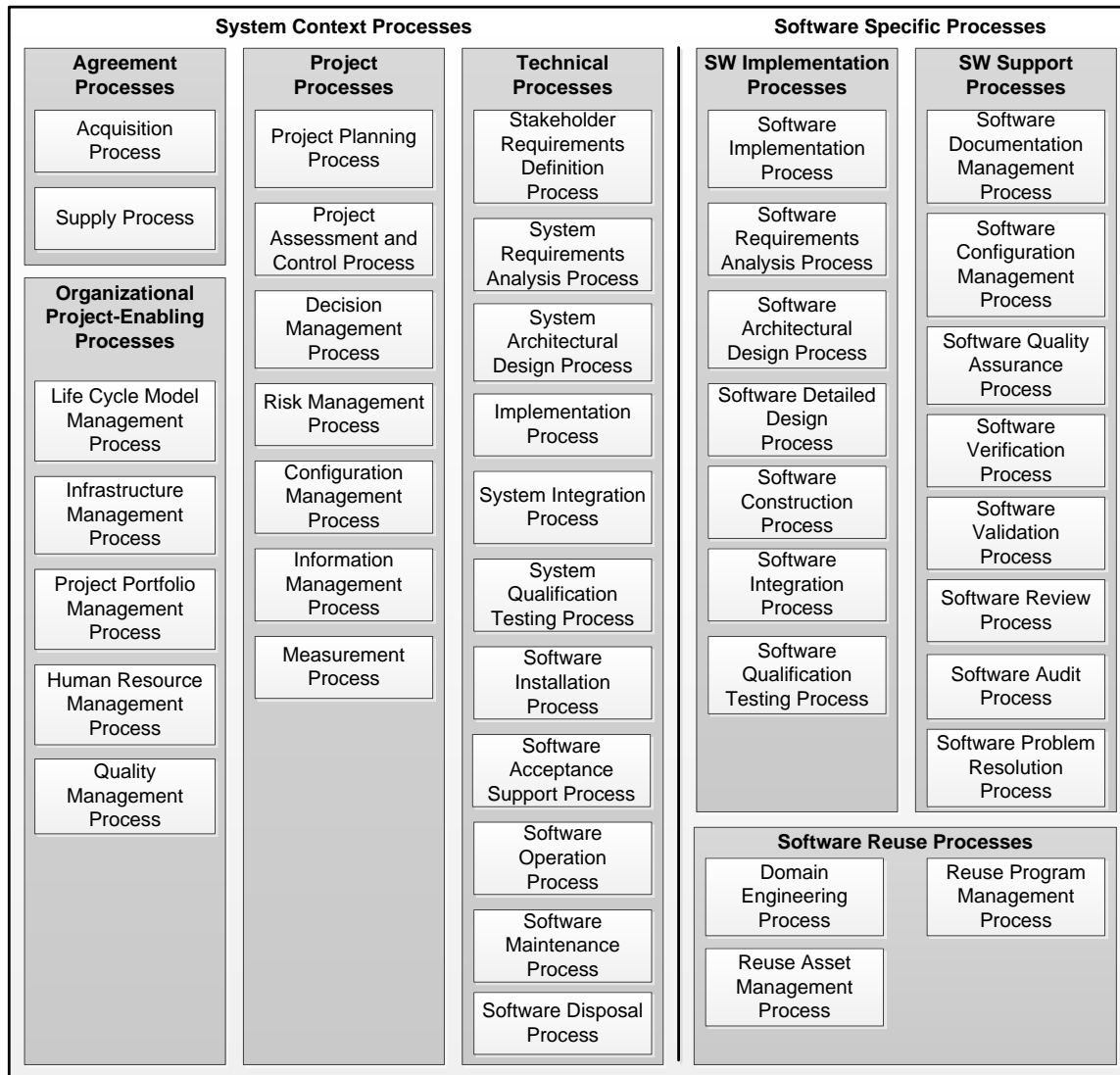


Figure 5. Process groups of the ISO 12207 (adapted from (International Organization for Standardization/International Electrotechnical Commission, 2008b))

1.4.2. ISO/IEC 15504 and ISO/IEC 330XX

The ISO/IEC 15504 (initially called SPICE project - Software Process Improvement and Capability dEtermination, in 1993) provides a framework for process evaluation. This standard is based on the standard for software lifecycle processes (ISO/IEC 12207 (International Organization for Standardization/International Electrotechnical Commission, 1995)) and concepts inherited from maturity models such as Trillium (Bell Canada, 1994) and Capability Maturity Model (CMM) (Paulk, Weber, Curtis, & Chrissis, 1995). The framework can be used by organizations that are involved in planning, management, monitoring, controlling, and improvement of the entire lifecycle of products and services.

On one side, the ISO/IEC 15504-2:2003 (International Organization for Standardization/International Electrotechnical Commission, 2003) defines the **process capability** as a six-point ordinal scale. This capability can be evaluated at the beginning of the scale (incomplete level) at the end (optimized level). This type of capability representation is called **continuous representation** and the process capability measurement is based on a set of process attributes defined by this standard.

On the other side, ISO/IEC TR 15504-7:2008 (International Organization for Standardization/International Electrotechnical Commission, 2008a) defines organizational maturity as an ordinal scale of six points. This scale is used to assess the maturity of the lower end (level 0 - immature organization) to the upper end of the scale (level 5 - Innovative organization). The maturity is the extent to which the organization has executed, managed, and explicitly and consistently established its processes with predictable performance. In addition, it demonstrates the ability of the organization to modify and adapt the performance of the fundamental processes to achieve its business objectives (International Organization for Standardization/International Electrotechnical Commission, 2008a). In this case the representation is called **staged representation**.

The ISO/IEC 330XX family provides a framework for evaluating characteristics of the process quality. A set of requirements for the evaluation process and the resources needed to implement it effectively are provided by these standards (International Organization for Standardization/International Electrotechnical Commission, 2015). This family of standards replaces the ISO/IEC 15504 family and retains the goal of assessing **capability process** and **organizational maturity**.

1.5. Software Process Capability Maturity Models

Software process capability and maturity (SPCM) models can be defined as a collection of software engineering best practices, organized in process areas or process, which help companies to improve their software process.

The concept of **maturity** is addressed in standards, models, methodologies and guides, and it can help an organization to improve its operations. However, most of the available approaches are related to a specific part of their activity. They do not have a systemic view of the problems of the organizations. The improvement of a single sector contributes to perpetuating the barriers and divisions that exist in the organizations (CMMI Product Team, 2010).

Some models (see Figure 6 where CMMI is one of the first models that served as basis for others) offer the opportunity to avoid these obstacles and divisions transcending all disciplines. We can cite: the Capability Maturity Model Integration for Development – CMMI-DEV (CMMI Product Team, 2010); the MPS for Software Reference Model – MR-MPS-SW (Softex, 2016c), Brazilian model; the Processes Reference Model MoProSoft (Oktaba et al., 2005); (Oktaba & Vázquez, 2008), Mexican model; and the Spanish maturity model (Garzías et al., 2013). These models are largely known and used in the industry.

In the following sections, we present the general concepts of two of these models, CMMI-DEV and MR-MPS-SW, which describe an evolutionary approach of process improvement.

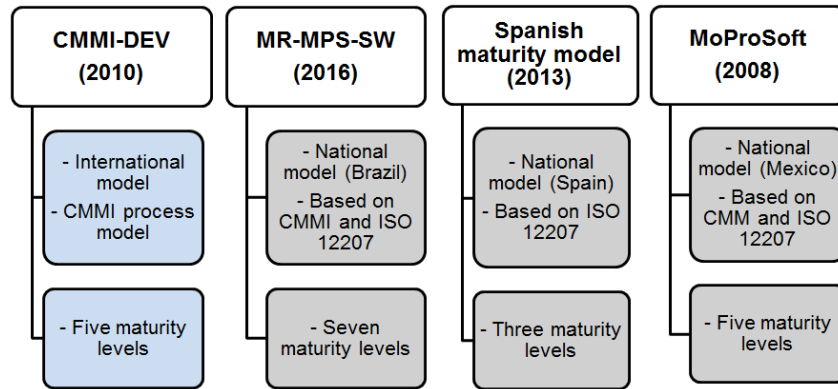


Figure 6. Software Process Capability Maturity Models

1.5.1. Capability Maturity Model and Capability Maturity Model Integration

Capability Maturity Model – CMM (Paulk et al., 1995) is a process improvement model defined by the Software Engineering Institute (SEI) during the 1990s by request of 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 organizations to move from immature processes to mature and better processes (CMMI Product Team, 2010).

The Capability Maturity Model Integration – CMMI (CMMI Product Team, 2010) was an initiative of members working in the industry, the US government and the SEI, that represents an evolution of CMM models. The CMMI is composed of a constellation, in other words, a set of CMMI components used to create models, training materials and evaluation documents for a given domain (such as development, acquisition, services, etc.). CMMI models, training materials and assessment components are provided through the CMMI framework (CMMI Product Team, 2010).

All CMMI models are based on the CMMI Model Foundation and provide good practices to help organizations to improve their processes. These CMMI models are not software development processes or process descriptions. They are used for the realization 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.

In this work we are interested in the CMMI-DEV model (CMMI Product Team, 2010). In the next section we discuss the main concepts defined in this model.

1.5.1.1. Capability Maturity Model Integration for Development

The Capability Maturity Model Integration for Development - CMMI-DEV (CMMI Product Team, 2010) provides a set of guidelines for applying best practices to the development of products and services. It is structured in a set of components (see Figure 7) grouped into three categories (CMMI Product Team, 2010): (i) required – the components (generic and specific goals) from this category are essential to achieve 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 (subpractices, example boxes, notes, references, sources, typical work products, etc.) help users of the model to understand the required and expected components and give suggestions to apply the activities.

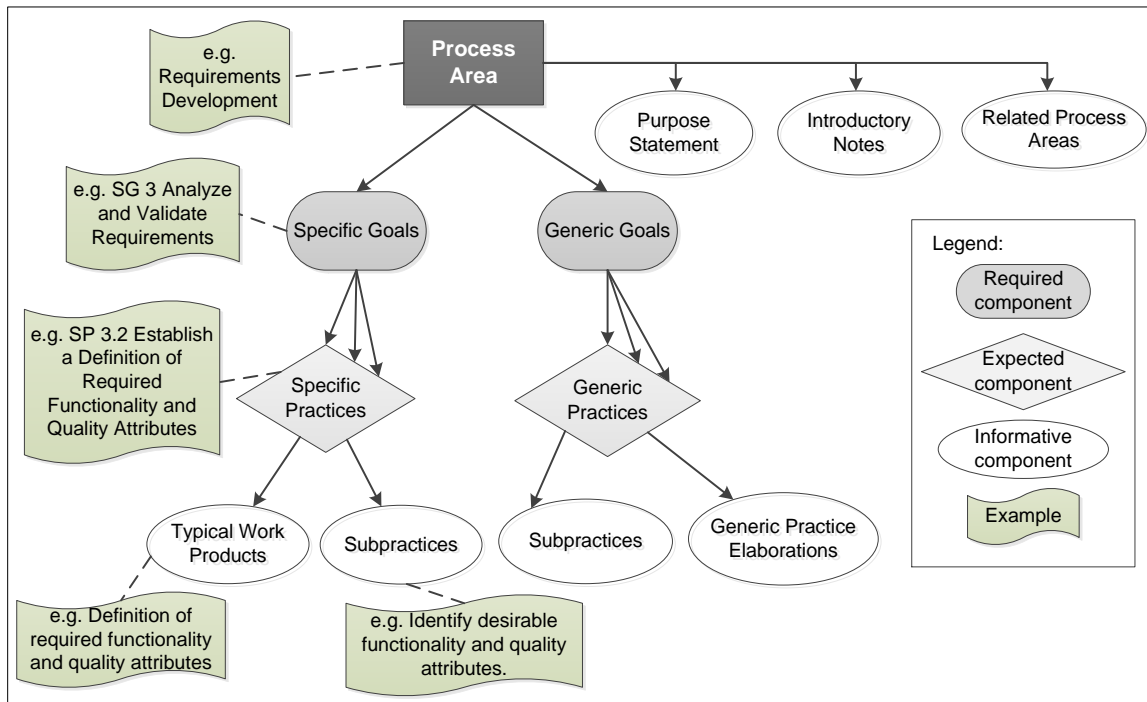


Figure 7. CMMI-DEV structure (adapted from (CMMI Product Team, 2010))

The core element of CMMI-DEV is the **process area** (see Figure 7 – e.g. *Requirements development*) that is a cluster of related practices in an area that, when implemented collectively, satisfies a set of goals considered very important for making a significant improvement in that area. The version 1.3 is composed of 22 process areas (the core element of the model) and brings together good practices of development from the industry and government (CMMI Product Team, 2010). These process areas are organized into four categories: project management, process management, engineering, and support (see Table 1).

A process area has 1 to 3 **Specific Goals** - SG (see Figure 7 – e.g. *SG3 Analyze and Validate Requirements - The requirements are analyzed and validated*). SG describes the unique characteristics that must be present to satisfy the process area. It is composed of **Specific Practices** - SP (see Figure 7 – *SP3.2 Establish a Definition of Required Functionality and Quality Attributes - Establish and maintain a definition of required functionality and quality attributes*) that describe the activities expected to result in achievement of the specific goals of a process area. 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 levels are defined: **capability level** and **maturity level**. The maturity level allows organizations to improve processes addressing a set of predefined process areas. This approach to improvement is called **staged representation**. The capability level allows the organization to improve processes in an individual (or group) process area, and this way of improvement is called **continuous representation**. Table 2 illustrates the capability and maturity levels of the CMMI-DEV model.

Table 1. Categories and Process areas from CMMI-DEV

Category	Process Areas
Project management	Quantitative Project Management (QPM)
	Integrated Project Management (IPM)
	Risk Management (RSKM)
	Project Monitoring and Control (PMC)
	Project Planning (PP)
	Requirements Management (REQM)
	Supplier Agreement Management (SAM)
Process management	Organizational Performance Management (OPM)
	Organizational Process Performance (OPP)
	Organizational Process Definition (OPD)
	Organizational Process Focus (OPF)
	Organizational Training (OT)
Engineering	Product Integration (PI)
	Requirements Development (RD)
	Technical Solution (TS)
	Validation (VAL)
	Verification (VER)
Support	Causal Analysis and Resolution (CAR)
	Decision Analysis and Resolution (DAR)
	Configuration Management (CM)
	Measurement and Analysis (MA)
	Process and Product Quality Assurance (PPQA)

Table 2. CMMI capability and maturity levels (CMMI Product Team, 2010)

Level	Continuous representation (capability levels)	Staged representation (maturity levels)
Level 0	Incomplete	-
Level 1	Performed	Initial
Level 2	Managed	Managed
Level 3	Defined	Defined
Level 4	-	Quantitatively Managed
Level 5	-	Optimizing

1.5.2. Brazilian Software Process Improvement Program⁹

The MPS.BR is the Brazilian Software Process Improvement Program (Softex, 2016c), coordinated by the Association for Promotion of the Brazilian Software Excellence (SOFTEX). The main objectives of MPS.BR are: definition of models for the improvement and evaluation of software and services processes, which the main focus is micro, small and medium-sized enterprises; the training of consultants and institutions of implementation and evaluation of the MPS models. The MPS Models is developed by MPS.BR.

Figure 8 illustrates the five components of the MPS Model (Softex, 2016c); (Kalinowski et al., 2015); (Kalinowski et al., 2014): MPS for Software reference model (MR-MPS-SW); MPS for Services reference model (MR-MPS-SV) which one of the technical basis is the CMMI for Services (CMMI-SVC); the MPS for People Management reference model (MR-MPS-RH) which the technical basis is composed of People Capability Maturity Model (P-CMM), PNQ – National Quality Award (Fundação

⁹ The abbreviations used for the components of the MPS model are those that were defined in the original model.

Nacional da Qualidade, 2011), among others; MPS Assessment Method (MA-MPS); and the MPS Business Model (MN-MPS). Each component is composed of guides and/or documents.

In this thesis we are interested in the MR-MPS-SW (Softex, 2016c). In the next section we discuss the main concepts defined in this model.

1.5.2.1. MPS for Software reference model

The MPS for Software reference model (MR-MPS-SW) discusses the concepts of maturity levels and process capability focusing in evaluation and improvement of the software/service quality and productivity (Softex, 2016c). This model was developed on the basis of ISO/IEC 12207 (International Organization for Standardization/International Electrotechnical Commission, 2008b) and CMMI-DEV (CMMI Product Team, 2010). It is composed of a MPS-SW general guide, MPS-SW implementation guides and an MPS acquisition guide as shown in Figure 8.

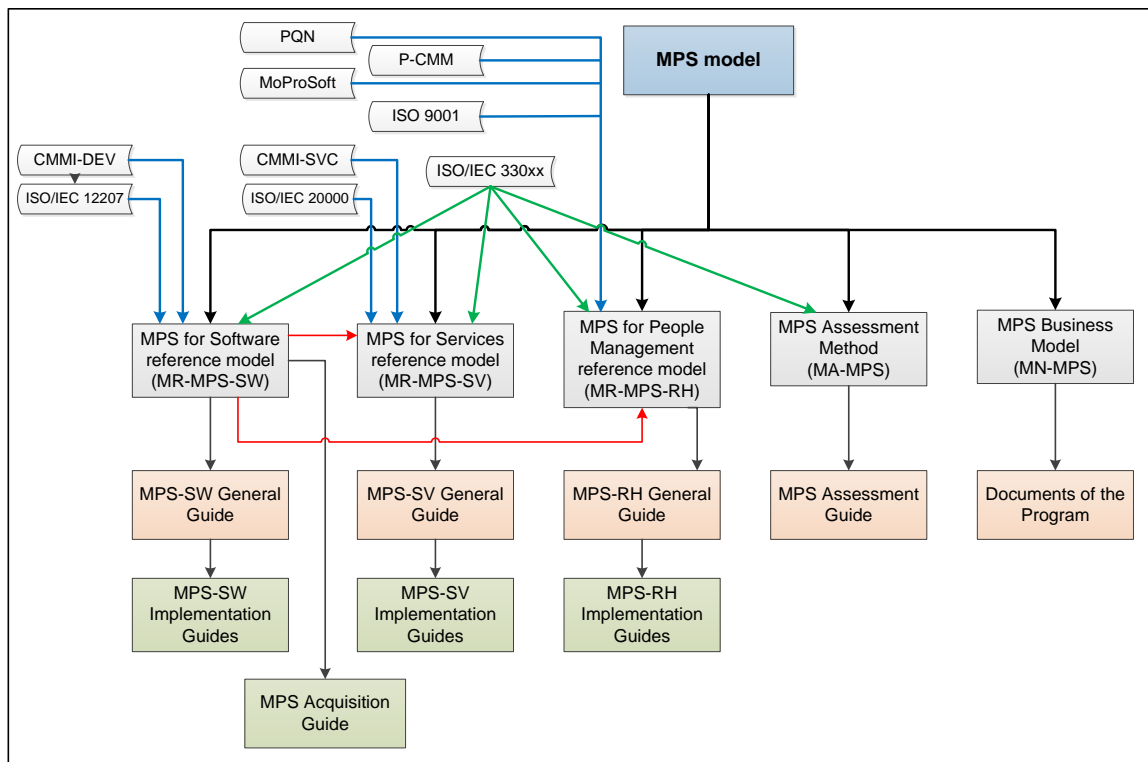


Figure 8. MPS model components (adapted from (Softex, 2016c))

The general guide describes the MR-MPS-SW model (Softex, 2016c) according to the following aspects: **MPS maturity levels** that are a combination of processes and their capability; **processes**, as well as its purposes and **expected results**; and **process attributes (PA)** that define the level of process capability for each maturity level. On the other hand, this guide does not define the activities and tasks required to meet the purpose and expected results (Softex, 2016c).

In addition, the MPS-SW implementation guides represent support for implementing the model. They describe theoretical foundations related to the processes and how to implement their expected outcomes. The acquisition guide offers good practices for organizations that desire to acquire software and services.

MR-MPS-SW model provides seven sequential and cumulative **maturity levels**, with a maturity scale starting at **level G** and progressing to **level A**, as well as nineteen processes that are shared between the seven maturity levels. Each maturity level represents a combination of processes and its capabilities. Table 3 shows each maturity level.

Table 3. MR-MPS-SW maturity levels (Softex, 2016c)

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

The **process** definition of the MR-MPS-SW model follows the requirements of a process reference model (purpose/goal and expected results) described in ISO/IEC 15504-2 (International Organization for Standardization/International Electrotechnical Commission, 2003). The objective expected with the execution of the process represents the **purpose/goal**. The **expected results** represent the objectives that must be achieved with the actual execution of the process. The process capability is represented by a set of attributes (see Figure 9) that measures it. To have a certain maturity level the organization must meet all the process attributes required for all processes related to the maturity level. Figure 9 shows the elements of the MR-MPS-SW model.

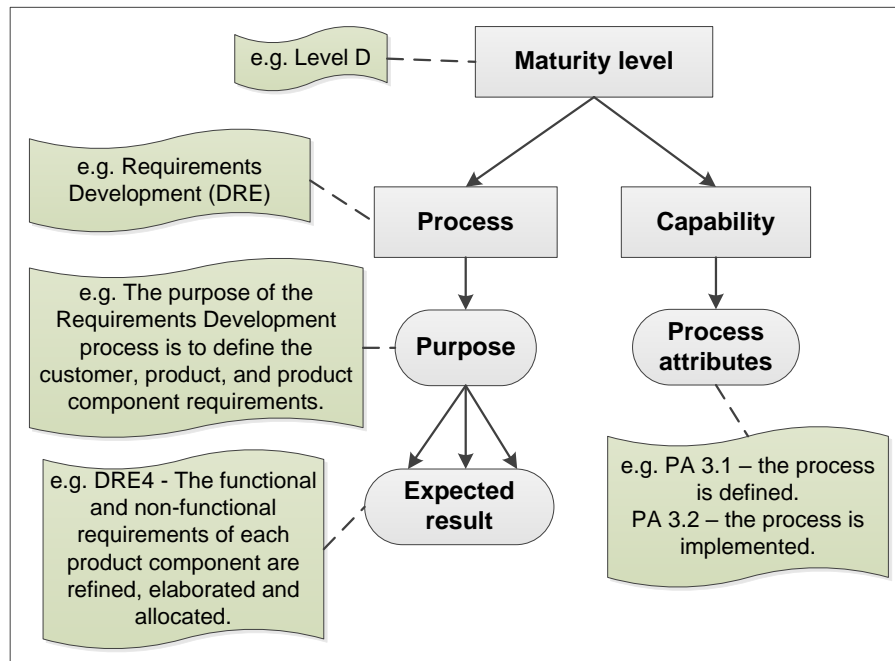


Figure 9. MR-MPS-SW structure (adapted from (Softex, 2016b))

1.6. Equivalence and synthesis of the models

The CMMI-DEV (CMMI Product Team, 2010) and MR-MPS-SW (Softex, 2016c) models can be used jointly by organizations wishing to improve their processes. This reality is possible thanks to the document published by SOFTEX (Softex, 2016b) that presents the equivalences between the models. CMMI-DEV is the main model studied in this thesis, as well as the MR-MPS-SW national model. CMMI-DEV is used worldwide, and both models are used by Brazilian organizations.

As described in the previous sections, its models support the process improvement of systems development organizations. In particular, the process areas (CMMI-DEV) or processes (MR-MPS-SW) normally do not match one-to-one to those used in organizations. Depending on the needs of the organizations, these process areas or process are integrated or adapted to the processes of the organizations (CMMI Product Team, 2010).

Table 4 presents the compatibility between the maturity levels of the CMMI-DEV and the MR-MPS-SW. The MR-MPS-SW model has three maturity levels (D, E, and G) more than the CMMI-DEV model, to meet the needs of micro, small and medium-sized enterprises (Softex, 2016b).

Table 4. Equivalence between CMMI-DEV and MR-MPS-SW maturity levels

CMMI-DEV levels	MR-MPS-SW levels
5 - Optimizing	A - Optimizing
4 - Quantitatively Managed	B - Quantitatively Managed
3 - Defined	C - Defined
<i>no equivalence</i>	D - Widely Defined
<i>no equivalence</i>	E - Partially Defined
2 - Managed	F - Managed
<i>no equivalence</i>	G - Partially Managed
1 - Initial	<i>no equivalence</i>

Besides the equivalence between the maturity levels, the implementation guide (Softex, 2016b) presents the compatibility between the main elements of both models. Figure 10 presents the equivalence between the elements of the models represented by the same colors as follows: a **process** in MR-MPS-SW is equivalent to a **process area** in CMMI-DEV; the **purpose** of the process is equivalent of the set of **specific goals** of the corresponding process area; and, an expected result MR-MPS-SW is equivalent to a **specific practice** in CMMI-DEV.

The guide defined by SOFTEX (Softex, 2016b) associates each **expected result** of MR-MPS-SW with each **specific practice** of CMMI-DEV, defining also if it is equivalent, jointly equivalent, not equivalent and inexistent. Table 5 presents an example of the technical mapping made for Requirements Development process of the MR-MPS-SW and the process area of the CMMI-DEV.

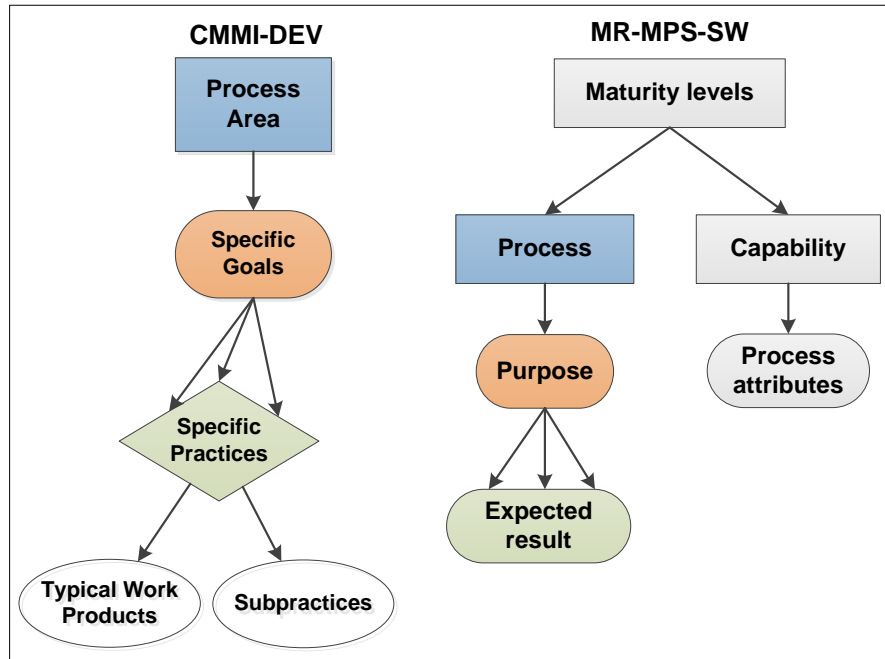


Figure 10. Equivalence between the models

1.7. Synthesis and Conclusion

Human-Computer Interaction and Software Engineering represent two disciplines very rich in terms of approaches (models, methods, techniques and standards) that are important for the development of interactive systems. In this first chapter, we have introduced a general view about HCI and SE concepts. We have presented HCI basic concepts and works that propose HCI approaches for human/user-centered design. In addition, we were also interested in presenting ISO standards for HCI and SE. Finally, we presented two Software Processes Capability Maturity (SPCM) models and the equivalence between these models.

Some concepts presented in this chapter will be very important for the next chapters. So the concepts to be retained are:

- HCI approaches in this thesis are methods, techniques, standards and patterns to support the analysis, design, implementation and evaluation of interactive systems;
- Process area (from CMMI-DEV) or process (from MR-MPS-SW) is composed of a set of practices related to a particular software engineering area (e.g., project planning);
- Specific practice (from CMMI-DEV) or expected result (from MR-MPS-SW) describes what should be addressed for the achievement of the goals of a process area or process;
- Maturity level allows organizations to improve their processes addressing a set of predefined process areas or processes.

In the next chapter, we will present a state-of-the-art about the alignment of the Human-Computer Interaction Engineering with the Software Engineering.

Table 5. Example of the mapping between MR-MPS-SW and CMMI-DEV (Softex, 2016b)

Requirements Development process MR-MPS-SW <i>Expected results of the process</i>		Requirements Development process area CMMI-DEV <i>Goal and specific practice</i>		Classification and Considerations	
		SG1	Stakeholder needs, expectations, constraints, and interfaces are collected and translated into customer requirements.		
DRE1	The needs, expectations and constraints of the customer, both the product and its interfaces, are identified.	SP1.1	Elicit stakeholder needs, expectations, constraints, and interfaces for all phases of the product lifecycle.	Not equivalent	Both MR-MPS-SW and CMMI-DEV require identification of the needs, expectations and constraints of the product and its interfaces. However, the MR-MPS-SW requires lifting only with the client, while the CMMI-DEV requires lifting with stakeholders, which may involve the customer, end users, suppliers, developers and testers, among others.
DRE2	A defined set of customer requirements is specified and prioritized from the identified needs, expectations, and constraints.	SP1.2	Transform stakeholder needs, expectations, constraints, and interfaces into prioritized customer requirements.	Equivalent	-
		SG2	Customer requirements are refined and elaborated to develop product and product component requirements.		
DRE3	A set of functional and non-functional requirements of the product and the product components, that describe the solution to the problem to be solved is defined and maintained from the customer’s requirements.	SP2.1	Establish and maintain product and product component requirements, which are based on the customer requirements.	Equivalent	Although the wording is not the same, DRE3 and SP2.1 has the same requirements associated with the definition and maintenance of the product and product components requirements, based on customer requirements.
DRE4	The functional and non-functional requirements of each product component are refined, elaborated and allocated.	SP2.2	Allocate the requirements for each product component.	Jointly equivalent	MR-MPS-SW requires the refinement, elaboration and allocation of the functional and non-functional requirements of each product component. CMMI-DEV only requires in this practice the allocation of requirements to the product components, which is only part of what is required in DRE4.

***Chapter 2 – State-of-the-Art: The alignment of the Human-Computer
Interaction Engineering and Software Engineering***

2.1. Introduction

As described in Chapter 1, Human-Computer Interaction (HCI) engineering has made significant progress in defining methods, techniques, patterns and standards to support the development of interactive systems. However, there are few published documents mentioning or showing the effective use of these in terms of practice in the industry. In another way, the Software Engineering (SE) community began to use in the industry methods, techniques, patterns and standards that have been developed since thirty years with the dissemination of software capability models (SPCM).

Although HCI engineering is not yet widely used in industry, in recent years several studies have proposed to integrate HCI and SE. The HCI community seeks to show how their set of knowledge (methods, techniques, patterns and standards) can be integrated into the classical SE processes. HCI community has also benefits from SE research to propose their own models, for instance usability capability/maturity (UCM) models.

Moreover, several studies have investigated the knowledge about HCI and/or the use of HCI approaches in practice. However, these studies were not developed in the context of software development with SPCM models implementations. They report the practice of HCI, usability and User Experience in the industry for different countries, showing the difficulties and benefits found.

In this chapter we will present a review of UCM models (section 2.1), an overview of the works that seeks to integrate HCI issues with SE (such as models, software development process, etc. – section 2.2), and also of the works that have investigated the practice of HCI in the industry (section 2.3).

2.2. Usability Capability/Maturity (UCM) Models

The literature presents some revision/survey/literature reviews that present and discuss the characteristics of various UCM models. Taking advantage of these reviews we identified the set of UCM models important for the context of this work. Thus, section 2.1.1 summarizes these reviews in order to identify the whole set of UCM models that are analyzed according to specific criteria. Section 2.1.2 presents the most relevant UCM models.

2.2.1. The evolution of the Usability Capability/Maturity models

Several authors have performed reviews (surveys, state of art or literature reviews) about usability capability/maturity models. In general, these studies identify any kind of software process, approach, and model that deals with usability issues. These studies have been performed for three main groups as presented in this section.

2.2.1.1. The studies of Jokela et al. (Jokela (2001), Jokela et al. (2006) and Jokela (2010))

Timo Jokela and other authors have performed reviews about UCM models since 2001. They carried out three studies being the last one published in 2010.

In 2001, Jokela (2001) presented a review of **six usability capability assessment (UCA) approaches**¹⁰ found in the literature, which are: (i) *Trillium* (Bell Canada, 1994); (ii) *Usability Leadership Assessment* (Flanagan & Rauch, 1995) and (Flanagan, 1996); (iii) *Human ware Process Assessment* (Gupta, 1997) and (Taylor, Gupta, McClelland, van Gelderen, & Hefley, 1998); (iv)

¹⁰ In this work, Jokela refers the models as approaches. A discussion about the use of this term will be presented in section 2.2.1.4.

Usability Maturity Model: Human-Centeredness Scale - UMM-HCS (Earthy, 1998); (v) *ISO/TR 18529* (International Organization for Standardization, 2000); and (vi) *Quality In Use Processes and Their Integration* - QIU (Earthy, 2000a) and (Earthy, 2000b).

In general, these approaches address characteristics that are relevant to planning and implementing user-centered design improvement actions (Jokela, 2001). According to the author, a common characteristic between the approaches is the **capability scale** that makes possible rating the usability capability of an organization (for example, 1 - poor capability, and 5 - excellent capability).

In another way, one difference between the approaches is the number of **dimensions** (organizational characteristics) that are related to usability capability. For instance, *Trillium* (Bell Canada, 1994) defines one dimension (“user-centered design”) that is placed in level 3; *Usability Leadership Assessment* (Flanagan & Rauch, 1995) and (Flanagan, 1996) presents nine dimensions; *Human ware Process Assessment* (Gupta, 1997) and (Taylor et al., 1998) presents ten dimensions; *UMM-HCS* (Earthy, 1998) presents one dimension (“human-centeredness”) that is placed in level E; *ISO/TR 18529* (International Organization for Standardization, 2000) presents seven dimensions called processes; and *Quality In Use Processes and Their Integration* - QIU (Earthy, 2000b) and (Earthy, 2000a) presents thirteen dimensions.

The author summarize that usability capability assessments represent a basis for planning and implementing improvement actions in development organizations about user-centered design. However, little research results exist about the effectiveness and theoretical basis of the approaches. ISO/TR 18529 gives the best basis as UCA. Finally, the implementation of an UCA in an organization implies to follow the good practices of engineering (Jokela, 2001).

In 2006, Jokela et al. (2006) presented a new survey about UCM models identifying **eleven models** of which five had not be presented in the study of 2001: *User Centered Design Maturity* - UCDM (by Jokela et al. (2006)); *KESSU Usability Design Process Model* (Jokela, 2004) and (Jokela, 2008); *Procedures for usability engineering process assessment* - DATech-UEPA (DATECH, 2002); *Human-centered design – Process Capability Model* - HCD-PCM design (by Jokela et al. (2006)); *Human-centered design – Process Capability Model* - HCD-PCM visioning (by Jokela et al. (2006)); and for the model QIU they considered the most recent version of the model (*Human Factors Integration Process Risk Assessors* – HFIPRA (Earthy, Bowler, Forster, & Taylor, 1999) and (Earthy, 2001); and *ISO/PAS 18152* (International Organization for Standardization, 2003). The eleven models (Jokela et al., 2006) were characterized according to different criteria, such as (see details in Table 6):

- Implementation of user-centered design in practice – this criterion examines the extent to which usability engineering activities are implemented and managed in development projects;
- Practical guidance (documentation) – it examines the extent to which there is guidance for practitioners to carry out an evaluation of the model;
- Empirical research – it examines whether a model is supported by research data.

In general, most of the models (8/11) did not report research results and six out of the eleven presented little concrete guidance or were not writing in English.

After few years, Jokela (2010) presents a new revision of these models. According to the author these models “are methods for the development of user-centered design processes in companies to facilitate the usability methods for the creation of usable products”. In this work the author takes up the **eleven** models analyzed in the previous work (Jokela et al., 2006) and adds a new model - *Standardized*

Usability/User-Experience Maturity Model – SU/UXM (Marcus, Gunther, & Sieffert, 2009). These models were categorized by the author into four types:

- Standard process models (standard process) – they are models that use the format of process evaluation models as in software engineering (ISO/IEC 15504);
- Non-standard processes models (non-standard processes) – they are models that examine processes with non-standard approaches;
- Generic models (generic) – they are models that include aspects of process, but also issues, such as management awareness, skills, and organizational position on usability;
- Specific models (specific) – they are models that have limited focus.

Most of the models (4/12) were categorized as generic and three models were classified as standard processes, including ISO/TR 18529 (International Organization for Standardization, 2000) and ISO/TS 18152 (International Organization for Standardization, 2010b).

Some models (ISO/TR 18529, ISO/TS 18152, UMM-P, HFIPRA) presented in Jokela et al. (2006) and Jokela (2010) use the format and requirements of the process assessment models used in software engineering (ISO/IEC 15504 (International Organization for Standardization/International Electrotechnical Commission, 2004b)). There are also models (UMM-HCS, UMM-P, ISO/TR 18529, DATech-UEPA, KESSU, HFIPRA, ISO/TS 18152) based on ISO 13407 (International Organization for Standardization, 1999) and models (Trillium, HPA) based on a previous version of CMMI (the CMM (Paulk et al., 1995)). Trillium (Bell Canada, 1994) and HPA (Gupta, 1997) and (Taylor et al., 1998) use the same structure of CMM/CMMI-DEV, that is process area, goals and practices, but they do not consider any process area or practice based on CMM.

Figure 11 summarizes the studies performed by Jokela and other authors (Jokela et al.).

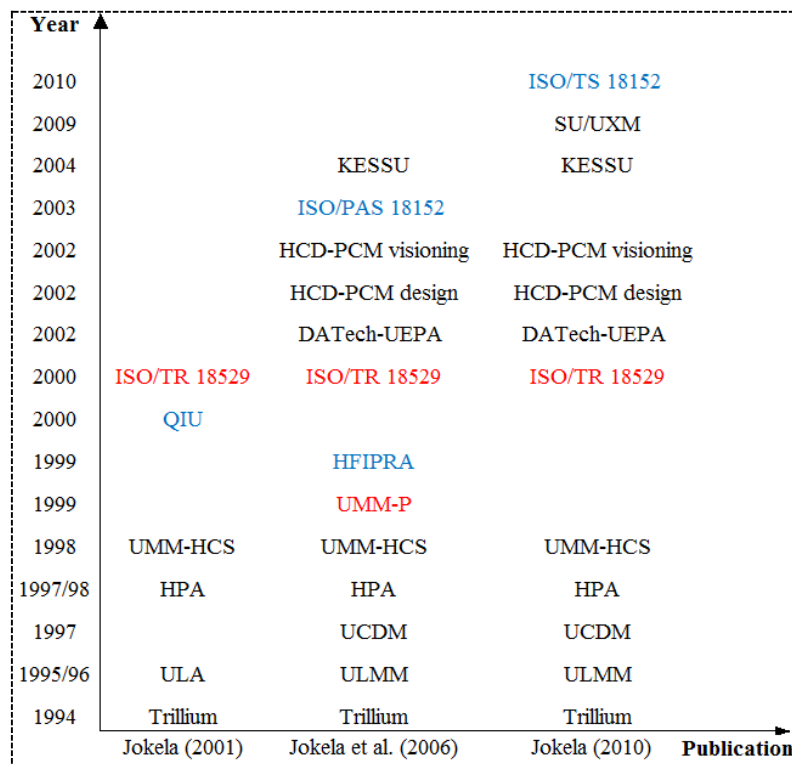


Figure 11. The evolution of the studies performed by Jokela *et al.*

Two models are evolution from the first version identified (in 2001 or 2006). They are highlighted in blue (e.g. QIU in Jokela (2001), HFIPRA and ISO/PAS 18152 in Jokela et al. (2006)) and red (ISO/TR 18529 in Jokela (2001), UMM-P in Jokela et al. (2006)). We noted that both studies generated two standards that were presented in previous chapter: ISO/TR 18529 and ISO/TS 18152.

2.2.1.2. The study of Salah et al. (2014)

With the goal to explore and evaluate the role that usability maturity models can play in the integration of agile processes and user-centered design, Salah et al. (2014) performed a review of some usability maturity models. The authors described in few lines **twelve models** of which nine were previously presented in Jokela et al. (2006) and/or Jokela (2010). The novelty of this work compared to the previous works is the presence of two new models: *Corporate Usability Maturity Model* (Nielsen, 2006a) and (Nielsen, 2006b), and *Open Source Usability Maturity Model - OS-UMM* (Raza et al., 2012). In addition, HCD-PCM Design and HCD-PCM Visioning models quoted in the previous study are considered as a single model by Salah et al. (2014).

The authors performed a comparative study of the models based on four criteria to compare the main characteristics of the different models. According to the first criteria (**lightweight**), the model should be lightweight, that means do not have overhead in the agile project schedule and the cost to carry out the evaluation of the model should be low in relation to time and human resources. The second criteria is about the **detailed English documentation**, defining that the model should provide detailed documentation that provides guidance for practitioners and also the model should be documented in English. The third criteria states that the model should be **domain independent**; i.e., it should be suitable for utilization in all organizations regardless of their domain of business. The last criteria is concerned with the **empirically evaluation**, establishing that the model should have been evaluated in empirical studies.

As a conclusion, they found that three models are not available in English, only four models presented detailed documentation, seven models were classified as a generic domain, only two models (**Nielsen's model** and **UMM-HCS**) were considered lightweight, and only three models were evaluated with empirical studies. As consequence, the authors found only two models (**Nielsen's model** and **UMM-HCS**) to be used in their case studies that propose the integration of agile processes and user-centered design.

2.2.1.3. The recently study of Lacerda & Wangenheim (2018)

Recently, (Lacerda & Wangenheim, 2018) performed a systematic literature review to identify capability/maturity models that focus in usability engineering and assist the model assessment process. They found **fifteen** usability capability/maturity models and they used the paper of Jokela et al. (2006) as a control paper in their systematic literature review.

The authors found in their research five models that were previously presented by Jokela et al. (2006): (i) *Usability Maturity Model: Human-Centeredness Scale* - UMM-HCS (Earthy, 1998); (ii) *Usability Maturity Model: Processes* - UMM-P (Earthy, 1999) and *ISO/TR 18529* (International Organization for Standardization, 2000); (iii) *Human Factors Integration Capability Maturity Model* - HFICMM (Earthy et al., 1999) or *Human Factors Integration Process Risk Assessors* - HFIPRA (Earthy, 2001); (iv) *Assessment of user-centered design processes basis for improvement action* or *KESSU Usability Design Process Model* (Jokela, 2004) and (Jokela, 2008); and (v) *ISO/TS 18152* (International

Organization for Standardization, 2010b). For this last one, Jokela et al. (2006) considered the version of 2003 and this ISO standard validates the HFICMM or HFIPRA model

They also considered one model previously presented in Jokela (2010): *Standardized Usability/User-Experience Maturity Model* (Marcus et al., 2009), and one model previously presented in Salah et al. (2014): *Open Source Usability Maturity Model - OS-UMM* (Raza et al., 2012).

As consequence, only eight new models out of the fifteen were identified in this study: (i) Introducing usability engineering into the CMM model: an empirical approach (Vasmatzidis, Ramakrishnan, & Hanson, 2001); (ii) Making User Experience a Business Strategy (Sward & Macarthur, 2007); (iii) Corporate User Experience Maturity Model (Van Tyne, 2009); (iv) New Health Usability Model: Implications for Nursing Informatics (Staggers & Rodney, 2012); (v) Maturity Models in the Context of Integrating Agile Development Processes and User-Centered Design (Mostafa, 2013); (vi) UX Maturity Model: Effective Introduction of UX into Organizations (Chapman & Plewes, 2014); (vii) AGILEUX Model – Towards a Reference Model on Integrating UX in Developing Software using Agile Methodologies (Peres et al., 2014); and (viii) STRATUS: a questionnaire for strategic usability assessment (Kieffer & Vanderdonckt, 2016).

They analyzed the models following different criteria, such as: type of model – classification of the model in “maturity” or “capability” model; validation – form of validation or evaluation of the model; and domain – the domain for which the model was designed. As a conclusion, they found that twelve models are maturity models; only six models presented a form of validation (expert evaluation, case study, author evaluation); and nine models were developed to usability/UX domain.

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. In this case, it is necessary to seek other sources or make arrangements of different models and methods.

2.2.1.4. Synthesis about the literature of Usability Capability/Maturity models

As a total, twenty-two models were identified in the reviews previously presented. Some models have evolved over time as the UMM-P and HFIPRA/HFICMM (see Table 6). These models were published, respectively, by ISO/TR 18529 and ISO/TS 18152 (Jokela et al., 2006). In October 2017, we found a new model – Assessment model for HCI practice maturity (Ogunyemi, Lamas, Stage, & Lárusdóttir, 2017) that is also included in Table 6. All works were deeply analyzed considering several criteria presented in Table 6.

The first criterion presented in Table 6 was only the identification of the models that were used as basis for the development of each UCM models. We note that some models used ISO 13407 and ISO 18529 as basis, and three models used the CMM. Other models are in conformance, or are based on ISO/IEC 15504.

The second criterion is the classification of the models by **category**. We used the same classification defined by Jokela et al. (2006) and Jokela (2010): standard process, non-standard process, generic, and specific. Twelve out of twenty-three models were previously classified by Jokela et al. (2006) and Jokela (2010). Thus, we classified the eleven other models following the description for this categories presented in section 2.1.1.1.

Based on the classification of Jokela et al. (2006) and Jokela (2010), we analyzed the models considering the level of detail of the documentation and language. These results are presented in the

Documentation column in Table 6. We note that nine models presented limited documentation and for three models English documentation was not available. In addition, only three models presented detailed guidance.

Recurrently the authors have named the models in different ways such as approaches, models or process. Therefore, we decided to come back to the definition of capability and maturity to analyze the models. We recall that the term **capability/maturity model** comes from the concept of **process improvement** – that the purpose is “to continually improve the organization’s effectiveness and efficiency through the processes used and maintained aligned with the business need” (International Organization for Standardization/International Electrotechnical Commission, 2004b), and it was created because the organizations needed to improve their software quality and process to become more mature (Paulk, Curtis, Chrissis, & Weber, 1993).

A **capability/maturity model** for any domain presents two concepts:

- (i) **process capability** – “a characterization of the ability of a process to meet current or projected business goals” (International Organization for Standardization/International Electrotechnical Commission, 2004a) that is measured as **process capability level** which represents the capability of the process; each level builds on the capability of the level below (International Organization for Standardization/International Electrotechnical Commission, 2004a). ISO 15504 proposes six capability levels; and,
- (ii) **(organizational) maturity level** – “point on the ordinal scale of organizational maturity that characterizes the maturity of the organization in the scope of the organizational maturity model used; each level builds on the maturity of the level below” (International Organization for Standardization/International Electrotechnical Commission, 2008a). The organizational maturity level rating is derived from the process profiles determined by the process capability levels.

Following this definition, we analyzed all models to classify if they are considered as capability and/or maturity models (see seventh and eighth columns of the Table 6). We note that three models are neither capability nor maturity models. Thus, these models are developed to evaluate the current state or the practice of usability of an organization, but not to improve the usability or UCD process.

The penultimate criterion is the classification of the models by **domain**. This criterion was used by (Lacerda & Wangenheim, 2018). We analysed all other models indentifying each domain. Our interest is only the models related to HCI issues in the context of software development. Only two models are not for this domain: **Trillium** that is specific for Telecom and **Health Usability Model** for Healthcare. Indeed, usability is considered as one of the aspects (such as skills, management practices) inside the models.

Finally, we analyzed the works to identify if they were validated. This analysis was previously performed by others authors (such as Jokela (2001) and Lacerda & Wangenheim (2018)). In this way, we updated this analysis with the models cited in Jokela et al. (2006), Jokela (2010) and Salah et al. (2014). We used our own classification based in the other works as follows: not validated, validated with case studies, and validated with experts. We note that fourteen models were not validated and seven performed case studies. Considering the scope of our study we will present in the next section more details about the UCM models classified as capability/maturity models and for the domain of usability. There are two models (#7 and #21) that will be presented in section 2.3 because these works discuss the integration of HCI approaches in SPCM models.

Table 6. Usability Capability/Maturity models

#	Model	Date	Basis models	Category*	Documentation*	Maturity level	Capability level	Domain	Validation
1	Trillium**	1994	CMM version 1.1	Non-standard process	Relatively detailed	No	Yes	Telecom	Not validated
2	Usability Leadership Assessment (ULA) or Usability Leadership Management Maturity (ULMM)**	1995 1996	-	Generic	Limited	Yes	No	Usability	Not validated
3	Human Ware Process Improvement (HPI) or Human Ware Process Assessment (HPA)**	1997 1998	CMM PDCA	Non-standard process	Limited	No	No	Human ware	Not validated
4	User Centered Design Maturity (UCDM)**	1997	-	Generic	Limited	Yes	No	User-Centered Design	Not validated
5	Usability Maturity Model: Human-Centeredness Scale (UMM-HCS)**	1998	ISO 13407	Generic	Rather detailed	Yes	No	Usability	Validated with case studies
6	Usability Maturity Model: Processes (UMM-P)** ISO/TR 18529 *	1999 2000	ISO 13407	Standard process	Detailed guidance and training available	No	Yes	Ergonomics of human-system interaction, Human-centered process	Validated with case studies and experts
7	Introducing usability engineering into the CMM model*	2001	CMM	-	Rather detailed	Yes	No	Usability engineering	Not validated
8	DATEch-UEPA**	2002	ISO 13407	Specific	In German	Yes	No	Usability engineering	Not validated
9	Human-centered design – Process Capability Model (HCD-PCM design)**	2002	ISO 18529	Standard process	In Japanese	No	Yes	Human-centered design	Not validated
10	Human-centered design – Process Capability Model (HCD-PCM visioning)**	2002	ISO 18529	Specific	In Japanese	No	Yes	Human-centered design	Not validated
11	KESSU Usability Design Process Model***	2004 2008	ISO 13407 ISO 18529	Non-standard process	Rather detailed	No	Yes	Usability	Validated with case studies
12	Corporate UX Maturity	2006	-	Generic	Limited	Yes	No	User experience	Not validated
13	Making User Experience a Business Strategy*	2007	-	Generic	Limited	Yes	No	User experience	Not validated
14	Standardized Usability/User-Experience Maturity Model*	2009	-	Generic	Limited	Yes	No	User experience	Not validated
15	Corporate User Experience Maturity Model*	2009	CMMI	Generic	Limited	Yes	No	User experience	Not validated

#	Model	Date	Basis models	Category*	Documentation*	Maturity level	Capability level	Domain	Validation
16	Quality In Use Processes and Their Integration (QIU)	2000	-	Standard process	Detailed guidance and training available	No	Yes	Ergonomics of human-system interaction	Validated with case studies and experts
	HFICMM or Human Factors Integration Process Risk Assessment (HFIPRA)*	2001	ISO 13407						
	ISO/PAS 18152, ISO/TS 18152**	2003 2010	ISO 18529						
17	Open source usability maturity model (OS-UMM)	2012	-	Non-standard process	Rather detailed	Yes	No	Open source usability	Validated with case studies
18	Health Usability Model: Implications for Nursing Informatics♥	2012	-	Generic	Limited	Yes	No	Healthcare	Not validated
19	A Maturity Model for Integrating Agile Development Processes and User Centered Design (AUCDI Maturity Model) ♥	2013	ISO 13407	Non-standard process	Detailed guidance	Yes	No	Agile development, User-centered design	Validated with experts
20	UX Maturity Model♥	2014	-	Generic	Limited	Yes	No	User experience	Not validated
21	AGILEUX Model♥	2014	CMMI MR-MPS ISO 18529	Non-standard process	Rather detailed	Yes	No	User experience, Agile methodologies	Validated with experts
22	STRATUS model: a questionnaire for strategic usability assessment♥	2016	-	Generic	Rather detailed	No	No	Usability	Validated with one case study
23	Assessment model for HCI practice maturity	2017	-	Generic	Rather detailed	No	No	Human-centered design	Validated with case studies

● Based in Jokela et al. (2006) and Jokela (2010); ♦ Models presented in Jokela et al. (2006) and also in Jokela (2010); ♣ Models presented in Jokela et al. (2006) and also Salah et al. (2014); ♠ Models presented in Jokela et al. (2006) and also in Lacerda & Wangenheim (2018); ♥ Models presented in Lacerda & Wangenheim (2018).

2.2.2. Non-standard process models

In this section we will present the models categorized as **Non-standard process**.

i) KESSU Usability Design Process Model

The KESSU model defines usability design through “**processes**” and “**outcomes**” (Jokela, 2004) and (Jokela, 2008). For this model, usability methods are the practical means to execute the processes and generate the outcomes. It examines the performance rather than the management aspects of user-centered processes. Figure 12 presents the general view of the model, which includes **seven processes** that are divided into two categories.

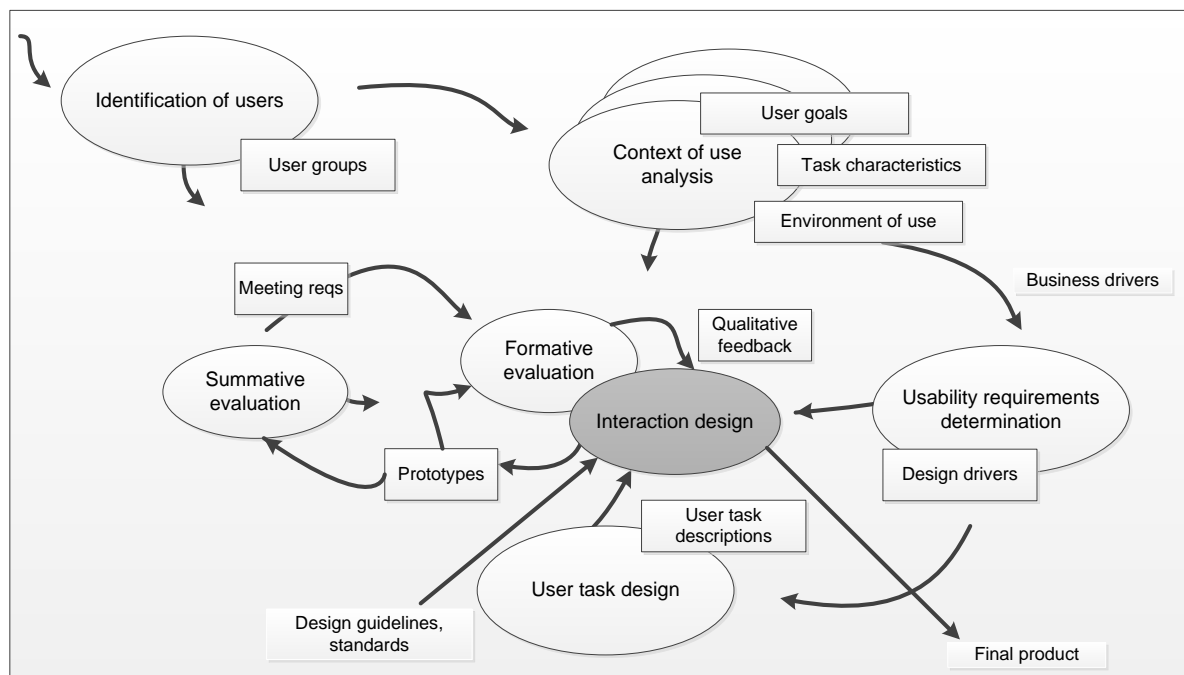


Figure 12. KESSU model (adapted from (Jokela, 2008))

The **usability engineering process** category (ellipses with white color) is composed for the following processes: (i) identification of users; (ii) context of use analysis; (iii) usability requirements determination; (iv) user task design; (v) summative evaluation; and (vi) formative evaluation. These processes feed user-driven input to the interaction design. The **user interaction design process** category (ellipse with gray color) is composed for one process: (vii) interaction design, which produces the product solutions.

ii) Open Source Usability Maturity Model (OS-UMM)

This usability maturity model is specific for open source projects and presented five maturity levels (1- preliminary, 2- recognized, 3- defined, 4- streamlined, and 5- institutionalized) to evaluate the usability maturity (Raza et al., 2012). Usability aspects are analyzed according to eleven “**key usability factors**” that is composed of different “**statements**”. The **key usability factors** are shared into four “**dimensions**” (usability methodology, design strategy, assessment and documentation). Figure 13 presents the structure of the model.

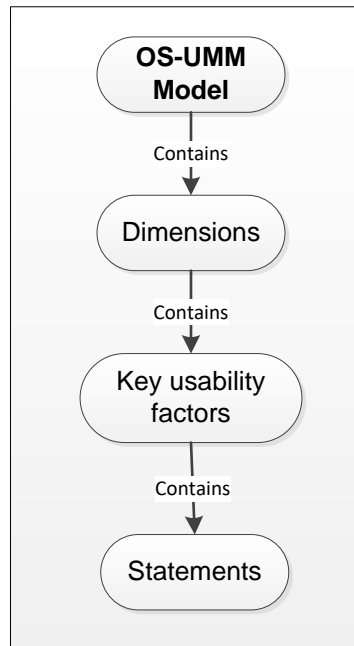


Figure 13. Structure of OS-UMM

The maturity assessment is done using questionnaires for each **maturity level** and each **key usability factor** is evaluated in each maturity level. Table 7 presents the dimensions and the key usability factors. The authors recognize that the model does not provide explicit guidelines (such as CMMI) for implementing the statements.

Table 7. Dimensions and key usability factors of OS-UMM

Dimension	Key usability factors
Usability methodology	1. Users' Requirements 2. Users' Feedback 3. Usability Learning
Design strategy	4. User-Centered Design (UCD) Methodology 5. Understandability 6. Learnability 7. Operability 8. Attractiveness
Assessment	9. Usability Bug Reporting 10. Usability Testing
Documentation	11. Documentation

iii) AUCDI maturity model

AUCDI maturity model integrates agile processes and user-centered design (Mostafa, 2013) and (Salah et al., 2016). It is composed of four **dimensions**:

- UCD infrastructure: composed of funds, staff, tools, methods, management support, training, utilization of standards, patterns and style guides and colocation of developers and UCD practitioners;
- AUCDI process: focuses on the planning and implementation of UCD activities and agile development principles to achieve the integration;
- people involved in the integration process: customers, users, developers, UCD practitioners and XP coach (in case of XP), scrum master and product owner (in case of Scrum); and

- UCD continuous improvement: practices such as, the UCD monitoring process across projects.

The **AUCDI processes** are: (i) planning the UCD process; (ii) user and task analysis; (iii) user and user interface design requirements; (iv) lightweight documentation; (v) synchronization efforts between UCD practitioners and developers; (vi) coordination and effective scheduling of UCD practitioners and developers activities; (vii) interaction and user task design; and (viii) usability evaluation.

Each process has a set of practices that utilize and produce associated work products that take the form of designs, documents, prototypes, working code, training courses, or individual awareness. The model provides also an assessment tool and presents six maturity levels (0- not possible, 1- possible, 2- encouraged, 3- enabled/practiced, 4- managed, and 5- continuous improvement).

2.2.3. Standard process models

In this section we will present the models categorized as **Standard process**.

i) Usability Maturity Model: Processes (UMM-P) and ISO/TR 18529:2000

The UMM-P model is composed of “**human-centered processes**” for the use in the assessment and improvement of the human-centered processes in system development (Earthy, 1999). The model is based and extends the ISO 13407. It presents **seven processes** that contain “**base practices**”. These base practices are sub-processes of a process. The processes use/generate “**work products**” and the base practices describe what is needed to be done in order to represent and include users of a system throughout the life cycle. Figure 14 illustrates the components of the model, previously described. The processes of this model are described in the format defined in ISO/IEC 15504.

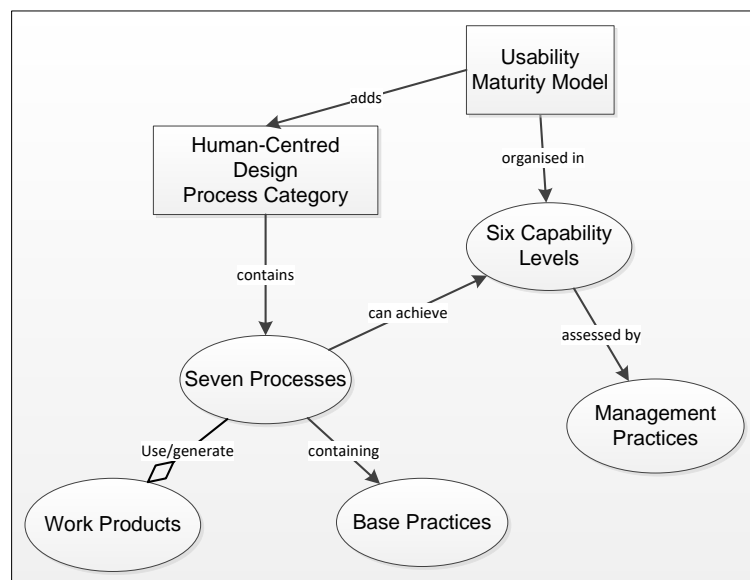


Figure 14. UMM-P Components (adapted from (Earthy, 1999))

In Table 8 we presented the processes HCD.4, HCD.5 and HCD.6 that are related to HCI. We can note that the process HCD.4 is related to the understanding and specification the context of use. The model presents capability maturity levels according to ISO/IEC 15504.

Table 8. UMM-P processes

Processes	
<i>HCD.4 Understand and specify the context of use</i>	
HCD.4.1 Identify and document user's tasks.	HCD.4.2 Identify and document significant user attributes.
HCD.4.3 Identify and document organizational environment.	HCD.4.4 Identify and document technical environment.
HCD.4.5 Identify and document physical environment.	
<i>HCD.5 Produce design solutions</i>	
HCD.5.1 Allocate functions.	HCD.5.2 Produce composite task model.
HCD.5.3 Explore system design.	HCD.5.4 Use existing knowledge to develop design solutions.
HCD.5.5 Specify system.	HCD.5.6 Develop prototypes.
HCD.5.7 Develop user training.	HCD.5.8 Develop user support.
<i>HCD.6 Evaluate designs against requirements</i>	
HCD.6.1 Specify and validate context of evaluation.	HCD.6.2 Evaluate early prototypes in order to define the requirements for the system.
HCD.6.3 Evaluate prototypes in order to improve the design.	HCD.6.4 Evaluate the system in order to check that the system requirements have been met.
HCD.6.5 Evaluate the system in order to check that the required practice has been followed.	HCD.6.6 Evaluate the system in use in order to ensure that it continues to meet organizational and user needs.

The ISO/TR 18529:2000 standard (International Organization for Standardization, 2000) presents a definition of processes to provide a basis for planning the human-centered activities on a project. Specifically, it involves the design, use and evaluation of life cycle processes of systems, hardware and software. It consists of seven “**processes**” that contain “**practices**” and use/generate “**work products**”. These practices describe what is needed to be done in order to represent and include users of a system throughout the life cycle. The processes are described in the format defined in ISO/IEC 15504. Figure 15 presents the formal components of the model.

This ISO standard validates the UMM-P model previously described, the processes of which are the same as those of the UMM-P model with some modifications. The practice HCD.5.5 in the ISO/TR 18529 is “Specify system and use”. As previously explained, the model presented capability levels according to ISO/IEC 15504.

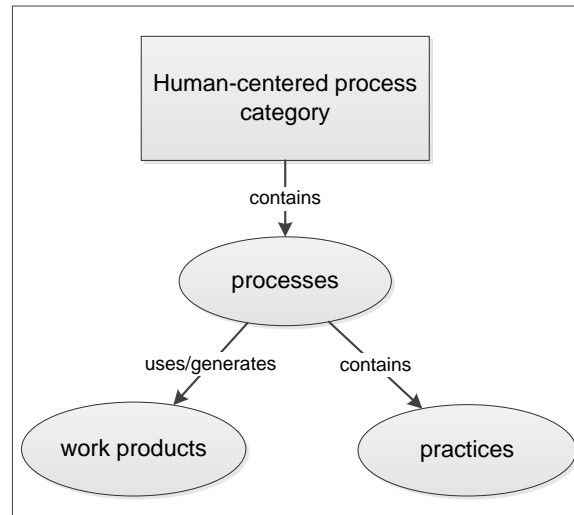


Figure 15. Components of the ISO/TR 18529 (adapted from (International Organization for Standardization, 2000))

ii) Human Factors Integration Process Risk Assessment (HFIPRA) and ISO/PAS 18152:2003 - ISO/TS 18152:2010

The HFIPRA model allows improving the quality of user-centered design of a system (Earthy et al., 1999) and (Earthy, 2001). This model is an evolution of QIU model proposed by (Earthy, 2000b) and (Earthy, 2000a). It identifies and defines twenty-one “**human system sub processes**” (HS) that are organized in four “**super-processes**” (e.g. HS.3 Usability engineering). These processes influence the usability of a product and detail their “**base practices**” and “**work products**”. Each HS sub process (e.g. HS.3.1 Context of use) contains various **base practices** (e.g. HS.3.1.BP1 Define the scope of the context of use for the product system). Figure 16 shows the components of the model.

The super-process related to HCI issues is HS.3 Usability engineering, which is composed of four processes: (i) HS.3.1 Context of use; (ii) HS.3.2 User requirements; (iii) HS.3.3 Produce design solutions; and (iv) HS.3.4 Human factors evaluation.

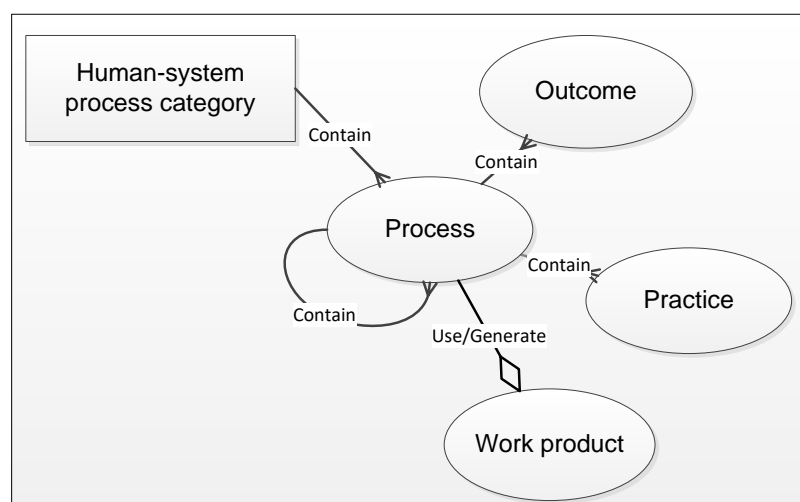


Figure 16. Components of the HFIPRA model (adapted from (Earthy, 2001))

ISO/TS 18152 presents a human-systems (HS) model for use with ISO/IEC 15504. It describes “**processes**” that address human-system issues and the “**outcomes**” of these processes. It also details the “**practices**” and “**work products**” associated with achieving the outcomes of each process. This ISO standard validates the HFIPRA model. Human-systems processes are divided into three categories: HS.1 Lifecycle involvement, HS.2 Integrate human factors, and HS.3 Human-centered design. The category HS.3 Human-centered design related to HCI issues is composed of four processes: (i) HS.3.1 Context of use; (ii) HS.3.2 User requirements; (iii) HS.3.3 Produce design solutions; and (iv) HS.3.4 Evaluation of use. This model presents capability levels such as ISO/TR 18529.

2.2.4. Generic models

In this section we will present eight models categorized as **Generic**.

i) Usability Leadership Assessment (ULA) or Usability Leadership Management Maturity (ULMM)

This model has been developed with the aim of improving the state of usability in software development projects (Flanagan & Rauch, 1995) and (Flanagan, 1996). It examines the organizations according to three “**Categories**”: organization, skills and process. In each category we find “**Attributes of usability management maturity**”, and as a total the model presents nine attributes:

- Organization category: 1- awareness, 2- activities, and 3- improvement actions;
- Skills category: 4- character, vitality, impact, and 5- resources;
- Process category: 6- early/continual user focus, 7- integrated design, 8- early/continual user tests, and 9- iterative design.

In addition, the model presents five maturity levels (1 to 5).

ii) User-centered design maturity (UCDM)

This model was developed as a tool for benchmarking information systems capability in the UK public sector (Jokela et al., 2006). According to Jokela et al. (2006) the model has five “**Capability Areas**” (such as formative evaluation) and fifteen “**Foci of assessment**” (such as early usability evaluation). Table 9 presents all capability areas and foci of assessment. It also presents five maturity levels (1- uncertainty, 2- awakening, 3- enlightenment, 4- wisdom, and 5- certainty).

i) Usability Maturity Model: Human-Centeredness Scale (UMM-HCS)

This model assesses the maturity level of an organization according to its ability to achieve human-centered design. Each maturity level is defined according to “**process attributes**” and “**management practices**” (Earthy, 1998). It is composed of six levels: X- unrecognized, A- recognized, B- considered, C- implemented, D- integrated, and E- institutionalized.

The **management practices** related to HCI are presented in Table 10. We can note “B.2 user focus attribute” from level B that presented as practices “B2.1 user consideration training” and “B2.2 context of use training”.

Table 9. Capability Areas and Foci of assessment (Jokela et al., 2006)

Capability Areas	Foci of assessment
Project structure and goals	Mission integration Stakeholder engagement Iterative development
Requirements management	Stakeholder requirements generation Integrated design
Systems design	User-Centered technical design User-Centered purchasing/contracting User-Centered social systems design
Formative evaluation	Early usability evaluation Early acceptability evaluation
Support and implementation	Implementation and change management User support Health and safety Customization and local design Summative evaluation

Table 10. Process attributes and management practices

Level	Process attributes	Management practices
A	A.2 Performed processes attribute	A2.1 Information collection. Information is collected which could be used to take account user requirements. A2.2 Performance of relevant practices. Practices are performed which could be used to include information about user requirements in the system or service.
B	B.2 User focus attribute	B2.1 User consideration training. Staff is made aware that the needs of the end users of the system should be considered when developing or supporting the system. B2.2 Context of use training. Staff is made aware that end users' skills, background and motivation may differ from developers or system support staff.
C	C.1 User involvement attribute	C.1.1 Active involvement of users. The development process ensures understanding of user needs through user involvement in all development phases. C.1.2 Elicitation of user experience. The design solution is shown to stakeholders and they are allowed to perform tasks (or simulated tasks). C.1.3 End users define quality-in-use. Systems are tested using measures of quality in use derived from end users. C.1.4 Continuous evaluation. Early and continual testing is an essential element of the development methodology. The process is based on the necessity for feedback from users.

ii) Corporate UX Maturity

This model was developed by Jakob Nielsen (Nielsen, 2006a) and (Nielsen, 2006b). It is composed of eight maturity levels (stage 1- hostility toward usability, stage 2- developer centered user experience, stage 3- skunk works user experience, stage 4- dedicated UX budget, stage 5- managed usability, stage 6- systematic usability process, stage 7- integrated user-centered design, and stage 8- user-driven corporation) where the organizations progressing through a sequence of steps (usability principles and practices) so that their user experience processes evolve and become mature.

According to the author companies can remain in stage 1 for decades (Nielsen, 2006b). It will only change of stage when a major software design problem is presented. Thus, this can be very fast or very slow. Once the company changes from stage 1 to stage 2, people begin to worry about usability (Nielsen, 2006b). After, for stages 2 to 4, they can spend from 2 to 3 years in each stage. The companies can take about 20 years to move from stage 2 (very immature related to user experience) to stage 7 (very mature related to user experience). The author also argues that companies probably need another 20 years to reach the last stage. According to the author the last four levels represent the longest part to get the lowest stage (stage 8).

iii) Making User Experience a Business Strategy

The User Experience Capability Maturity Framework (Sward & Macarthur, 2007) is a formal approach to assessing an organization's capability to engage in User Experience Design according to these dimensions: (i) user-centric processes; (ii) staffing and training; (iii) organizational alignment; (iv) management commitment; and (v) strategy and visioning.

This model presents five maturity levels (initial, repeatable, defined, managed, and optimized) and the major characteristics of each level are presented in Table 11. We can note, for example, that this model includes “process metrics for practices in user- experience design group” as characteristic in “repeatable level”.

Table 11. User experience capability maturity model (Sward & Macarthur, 2007)

Maternity level	Major Characteristics
Initial	Base UX practices, User Experience Design (UXD) professionals own the UX and are integrated with development teams.
Repeatable	Process metrics (PM) for practices in UXD group, PM is accountable for UX, and UXD lifecycle integration with input on product planning.
Defined	Process metrics to manage UXD and engagements, portfolio owner is accountable for UX, and business process integration with input on product portfolios.
Managed	Managed UXD process with UX recognized leadership, UX is owned by the organization, and UX architecture impacts strategic planning.
Optimized	Optimize processes, an executive drives UX to respond to business changes and sets firm strategies.

iv) Standardized Usability/User-Experience (UX) Maturity Model

The model is an ongoing proposal that seeks to develop a more general, valid and reliable model that would be supported by the experience and knowledge of usability/user experience developers (Marcus et al., 2009).

It is composed of “**management practices**” that are measured according to five maturity levels (1- initial, 2- repeatable, 3- defined, 4- managed, and 5 optimized). Table 12 presents the practices and the maturity levels defined in the model, where we can note that in level 2 (repeatable) the practice “UX development” requires qualitative and process metrics. Further (in level 4- managed) the same practice requires managed process.

Table 12. Practices and maturity levels (Marcus et al., 2009)

Management practice Level	UX Development	Staffing Resources	Management Commitment	Organizational Alignment	Vision & Strategy
5- Optimized	Continual process improvement	UX executive	Maintenance commitment	UX part of business strategy processes	Firm level vision and strategy
4- Managed	Managed process	UX leadership	Organizational ownership	UX architect	Strategic planning
3- Defined	User data provided to management	Managed engagement	Portfolio ownership management	Integration with broader business processes	Portfolio planning
2- Repeatable	Qualitative and process metrics	UX Operations	Project manager owns relationship	Product development include UX processes	Product planning
1- Initial	UX basic practices	Staff with UX professionals	UX professionals own relationship	Localized product development team integration	Localized product optimization

v) Corporate User Experience Maturity Model

The Corporate User Experience Maturity Model (Van Tyne, 2009) was inspired by CMMI (CMMI Product Team, 2010) and in Corporate UX Maturity (Nielsen, 2006a) and (Nielsen, 2006b). The model presents five maturity levels (0- initial, 1- professional discipline, 2- managed process, 3- integrated UX, and 4- customer-driven corporation) as presented in Figure 17. The objective of the model is to define the user experience maturity of the organizations.

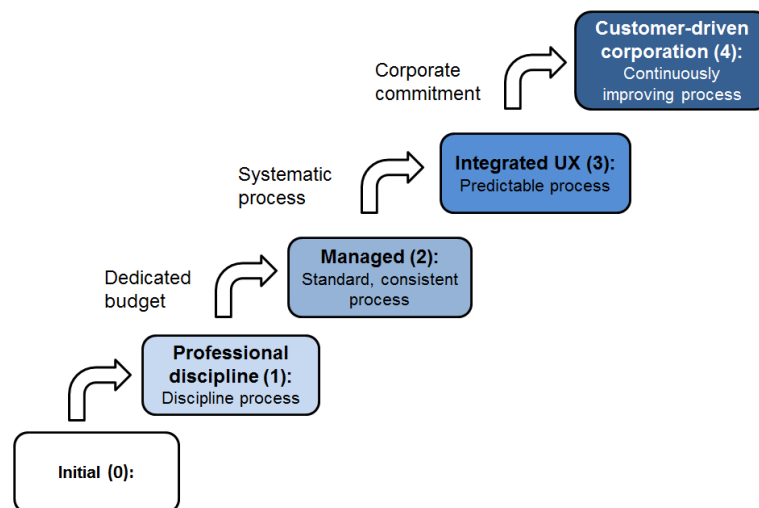


Figure 17. Corporate User Experience Maturity Model (adapted from (Van Tyne, 2009))

vi) UX maturity model

This model was created for assessing the level of UX maturity of the organizations (Chapman & Plewes, 2014). The goal is to identify what the organization already knows/use in terms of UX before adopting UX practices.

The model is composed of **key indicators of UX maturity** (e.g., timing of initial UX, availability of resources, leadership and culture) and it presents five UX maturity levels (1- beginning, 2- awareness,

3- adopting, 4- realizing, and 5- exceptional) as presented in Figure 18. We can note that the key indicators of UX maturity start in level 2, such as timing of initial UX that was performed after coding.

UX maturity stages					
	1 Beginning	2 Awareness	3 Adopting	4 Realizing	5 Exceptional
Timing of Initial UX	No UX	After coding	Mixed: after coding, before coding, and sometimes in parallel with coding	Prior to coding	As part of business & market requirements
Availability of Resources	None	Visual Design	Visual, Interaction, Research, Lower Level Management	Visual, Interaction, Research, Lower and Upper Level Management	Visual, Interaction, Research, Management Executive
Leadership and Culture	None	None	No clear owner, shared amongst functions	Clear owner in Upper Management	Clear owner and well understood, represented at executive levels

Figure 18. UX maturity levels (adapted from (Chapman & Plewes, 2014))

2.2.5. Synthesis about some Usability Capability/Maturity propositions

We highlight that the propositions from Table 6 focused on the definition of usability maturity/capability approaches and there is no approach that proposes the integration of HCI in SPCM models. We can observe that most of these approaches are in conformance, or are based on CMM/CMMI and ISO/IEC 15504, but they do not consider the engineering process areas of CMMI-DEV.

Most of the approaches do not provide adequate support to be applied in practice, and few of them were validated with case study or expert review. In addition, no results exist about the effective implementation of these approaches in organizations, as we have seen in SE domain for SPMC models.

For this reason we also investigate the literature to found works that propose the integration of HCI engineering and software engineering. We will present these works in the next section.

2.3. Integration of Human-Computer Interaction Engineering and Software Engineering

Several works have proposed the integration of Human-Computer Interaction and Software Engineering for performing usable and useful systems. We identify works that have proposed the integration of HCI approaches in software development processes or standards; and relevant

systematic literature reviews (SLR)¹¹ related to the integration of SE and HCI domains. In the next sections we will present and discuss these works.

2.3.1. Integration of HCI approaches in software development processes or standards

In this section we will present some works that have proposed the integration of HCI approaches in software development processes or standards. Some of them are proposed as process models (presented section 2.3.1.1), others focused on human-centered design issues (section 2.3.1.2), and others include HCI in the software development processes (section 2.3.1.3).

2.3.1.1. Process Models

i) UCProMo (User-Centered Process Model)

UCProMo – User-Centered Process Model (Gross, 2016) provides an integrated approach based on existing process models (such as the waterfall model, the spiral model, the Unified Process, the star model and the standard process model “Human-centered design for interactive systems” ISO 9241-210:2010), methods and tools of the Software engineering and Human-Computer Interaction domains.

The model consists of seven steps which cover all life cycle of system development: step 0- Plan the Human-Centered Design Process; step 1- Understand and Define Users, Tasks, and Contexts; step 2- Specify System Requirements; step 3- Design User Tasks, and User Interactions; step 4- Develop the System; step 5- Evaluate the System; and step 6- Deploy the System. For step 1 to 3 are proposed the construction of different models, for example: step 1 - user model, task model, context model; Step 2 - integration model; Step 3 - interaction space model. UCProMo provides clear steps that provide iteration and user participation throughout the process (Gross, 2016).

ii) Wheel: a usability engineering process model

Helms et al. (2006) present a field study and the development of a usability engineering process model. This process model provides a general framework which developers can adjust specific existing or new techniques, methods, or activities (such as: user/task model, usage scenarios, lo-fi prototype, hi-fi prototype) to apply best usability practices.

This framework was created from the examination, adaptation, and extension of usability engineering and software methodologies (LUCID framework of interaction design, Star life cycle of usability engineering, and waterfall and spiral models of software engineering) and presents the concept of cycles and activities (analyze, design, implement, and evaluate).

2.3.1.2. User/human-centered design issues

i) Usability method set supporting human-centered design

The integration of software engineering and usability engineering was proposed ((Fischer, 2012); (Fischer, Streng, & Nebe, 2013)) through the analysis of standards (ISO 12207 and ISO 9241-210). The result of analysis presents activities, artifacts and correlations of HCI and software engineering, which were validated by HCI experts.

¹¹ A systematic literature review (Kitchenham & Charters, 2007) “is a means of evaluating and interpreting all available research relevant to a particular research question, topic area, or phenomenon of interest”.

Moreover, they propose a tool (Fischer et al., 2013) that focuses on the selection of appropriate usability methods regarding their applicability in the various phases of system development. It presents also the interdependencies between multiple methods.

ii) Integrating user-centered design into software engineering process

One approach aims to identify integration points between software engineering and usability engineering (Nebe & Zimmermann, 2007), (Nebe & Zimmermann, 2008) and (Nebe, Zimmermann, & Paelke, 2008). The authors considered three levels (standards, process models and operational process) to promote a good alignment between the two disciplines (Nebe & Zimmermann, 2007).

The standards define the general framework, process models describe systematic and traceable approaches, and at the operational level (operational process) these models are adjusted and put in practice. In the level of standards, there are aspects (ISO 12207 and ISO 13407) that allow integration. The analysis of these two standards resulted in five common activities (Nebe & Zimmermann, 2008): requirement analysis, software specification, software design and implementation, software validation and evaluation, and from these activities a general framework common to both disciplines was defined.

Next, for the model process level, the authors analyzed four different models of software engineering processes (linear sequential model, evolutionary development model, spiral model and V model) to determine their ability to create usable products (Nebe & Zimmermann, 2008). A profile with the strengths and weaknesses of each model was outlined to guide organizations in selecting a specific process model based on these specific criteria. As a result, they found that the V model can be considered more essentially capable of producing usable products than other models. As well as that, the authors see the result as an indicator that there is a weak integration between usability engineering and software engineering (Nebe & Zimmermann, 2008).

2.3.1.3. Software process

i) Integrating usability practices into the software process

This framework proposes the integration of usability practices into development life cycle (Alyahyan, Alnafjan, & Aldabbas, 2016). The framework is characterized by 10 selected user-centered design (UCD) methods in relation to five relevant criteria based on ISO/TR16982. Also, the selection of the methods is based on literature (Discount usability engineering approach (Nielsen, 1993), UsabilityNet methods (Bevan, 2003b) and Cost-effective user-centered design (Bevan, 2005)).

This framework offers basic methods that are recommended, cost-effective, simple to plan and apply, and easy to learn by developers (Alyahyan et al., 2016). In addition, it indicates which can be applied when resources, skills, time, and expertise are limited. The selected methods are placed in the framework according to the steps of the development process where they can be applied. However, the development life cycle should be based on an iterative approach. The authors used the steps of the development process presented in (Bevan, 2003b), but the organizations can translate these steps for their specific activity processes.

ii) Framework for integrating usability practices into the software process

A framework that integrates usability practices into the software process model was defined by (Ferre, Juristo, & Moreno, 2005a). They identified the main activities of a user-centered software process: (i)

specification of the context of use – user analysis and task analysis; (ii) usability specifications; (iii) develop product concept; (iv) prototyping; (v) interaction design; and (vi) usability evaluation.

A set of 35 usability techniques were identified from literature to support these activities (Ferre, Juristo, & Moreno, 2005b). These techniques were organized according to the kind of activities where they may be applied and to the best moment of application (Ferre et al., 2005a). The software process could be based on an iterative life cycle. In this work the authors propose the integration of usability activities into a software process based on the SWEBOK (Ferre et al., 2005b).

iii) Integrating Usability Techniques into Software Development

An approach (Anderson, Fleek, Garrity, & Drake, 2001) proposes the integration of usability techniques into software development combining OO analysis, design practices and usability techniques. In addition, they propose that user-centered design and evaluation can be the core component of the development process.

The main challenge of this approach was to integrate usability and software engineering teams, and also integrate usability activities (derived from a user-centered design process that combined contextual-inquiry design techniques and usage-centered design processes) in the organization development process (based on Rational Unified Process).

2.3.1.4. Synthesis about the integration of HCI and SE

Analyzing the five works considering their goals, the software process activities that they support, the HCI approaches suggested, and the performed validation, we note some weaknesses (see Table 13).

As can be observed in Table 13, only one work (Ferre et al. (2005a) and Ferre et al. (2005b)) quoted HCI techniques to be applied in some phases of a software process (focusing on requirements and final evaluation).

Gross (2016) proposed an own technique for user centered-design and its application in some phases of the software process. Both works are propositions of the authors without 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.

Some proposals concern the integration of HCI techniques and activities in the software development process, through general frameworks. The approaches to HCI and SE have a lot to offer. Therefore, even after developing broad approaches to unite the strengths and benefits of both disciplines, they are not yet fully utilized in practice (Gross, 2016).

The theory and practice still have problems to incorporate usability engineering methods into development processes. One challenge is to identify the integration points between the disciplines of software engineering and usability engineering that allow the collaboration, and acceptable organizational and operational effort (Nebe & Zimmermann, 2007).

Table 13. Synthesis of the works about integration

Reference	Objective	Activities/Phases/Process areas	HCI techniques	Validation	Main weakness
Gross (2016)	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).	<ul style="list-style-type: none"> - 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 	<p>Defined a set of specific techniques based on Use-case 2.0.</p> <p>No technique proposed in literature was used.</p>	Not validated	Use of the techniques defined by the authors and the proposition was not validated.
Alyahyan et al. (2016)	Proposed a framework to integrate usability practices into development life cycle based in approaches found in literature (Discount usability engineering approach (Nielsen, 1993), UsabilityNet methods (Bevan, 2003) and Cost-effective user-centered design (Bevan, 2005)).	<ul style="list-style-type: none"> - Planning and Feasibility - Requirements - Design - Implementation - Test and Measure - Post-Release 	10 user-centered design (UCD) methods selected from literature.	Not validated	The main focus is small-sized software development organizations.
Fischer (2012) and Fischer et al. (2013)	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.	<ul style="list-style-type: none"> - 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 & Zimmermann (2007), Nebe & Zimmermann (2008) and Nebe et al. (2008)	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.	<ul style="list-style-type: none"> - 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.

Reference	Objective	Activities/Phases/Process areas	HCI techniques	Validation	Main weakness
Helms et al. (2006)	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.	<ul style="list-style-type: none"> - 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 associating to which phase they should be used, once the focus of the paper is the definition of a generic HCI software process.
Ferre et al. (2005a) and Ferre et al. (2005b)	Proposed a framework for integrating usability practices into the software process.	<ul style="list-style-type: none"> - Requirements elicitation, analysis, and negotiation - Requirement specification - Interaction design - Requirements validation - Usability evaluation 	34 HCI techniques.	Not validated	The chosen 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.
Anderson et al. (2001)	Proposed the integration of usability techniques into software development combining OO analysis, design practices and usability techniques.	<ul style="list-style-type: none"> - Inception - Elaboration - Construction - Transition - Evolution 	Techniques such as: user profiles, affinity diagram, vision storyboards, prototypes, and UI evaluation.	Not validated	The association of HCI techniques with process development phases is not explicit.

2.3.2. Systematic literature reviews

Two relevant systematic literature reviews (SLR) related to the integration of SE and HCI domains (W. Silva, Valentim, & Conte, 2015) and (Hoda, Salleh, Grundy, & Tee, 2017) were found in literature.

i) W. Silva et al. (2015)

The first SLR is presented by W. Silva et al. (2015). They executed an SLR where they identified, categorized and summarized technologies (methods, techniques, models, tools, approaches, and other proposals created by the areas of HCI and software engineering) that can be used to improve the usability within software development processes. The results show that several technologies supporting 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.

ii) Hoda et al. (2017)

The second SLR is presented by Hoda et al. (2017). They performed a tertiary study (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. For this research area the authors found two SLR (Chagas, de Carvalho, Lima, & Reis, 2014) and (F. S. Silva et al., 2015), which we consider relevant to our work, as follow:

- The first SLR presented by Chagas et al. (2014) is interested in the characteristics of agile project management and, therefore, focused on project planning process area from CMMI-DEV from 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” (Chagas et al., 2014). Moreover, they recognize the lack of approaches to support the process.
- The second SLR presented by F. S. Silva et al. (2015) 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 been 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.

2.3.3. Towards the integration of HCI approaches in SPCM models

In this section we will present some works that start to investigate how to integrate HCI with SPCM models.

i) Vasmatzidis et al. (2001)

Vasmatzidis et al. (2001) performed a study to introduce usability engineering into a software development process defined based on the CMM model. The organization that used this software process has the CMM level 2 and the objective was to reach CMM level 3. The proposal of

integration included the usability engineering processes in 3 of the 7 phases of the software process: requirements management; project planning, tracking and oversight; and product development. In a new version of the software process, to reach CMM level 3, the authors propose a user-centered product design methodology composed of five phases: conceptual phase; User Interface design; User Interface validation; implementation; and user feedback. Each part of this methodology was included in the 3 of the 7 phases of the software process, previously described, as a usability practices and activities. This work is a case study of a company that intends to reach CMM level 3 focusing only in the definition of a specific process and not in the integration of HCI approaches in CMMI.

ii) Nogueira & Furtado (2013)

From a literature review, Nogueira & Furtado (2013)¹², 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 (requirements development, design and construction of the product, verification, and validation) of the Brazilian model (MR-MPS-SW (Softex, 2016c)). 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 (that means the techniques were probably chosen for the specific kind of application); it limits the example of techniques (for example, for verification and validation only one technique is suggested) and it is targeted to 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.

iii) Peres et al. (2014)

Peres et al. (2014) 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 ISO/TR 18529. The model focuses on the 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; and, measurement and analysis). That means they deal with some management process but not with engineering process. Moreover, this work is in a very initial stage and had no validation being a simple proposition of the authors.

2.4. Human-Computer Interaction in practice

As previously mentioned, HCI was not yet sufficiently applied in practice. We found several studies related to it. Some of these studies investigate the knowledge and/or the use of HCI approaches in practice, but not in the context of software development with SPCM models implementations. These studies reports the practice of HCI, usability and User Experience in the industry for different countries (see Table 14).

Generally, as presented in Table 14, the studies used a survey or questionnaire to investigate the practice of HCI in industry. We analysed how the knowledge and use of HCI were discussed in these studies. Moreover, considering that our research is focused on levels of SPCM that considers the need of a defined process, we also investigated if these studies use a software process (or activities of a

¹² This work is published in Portuguese.

software process) to make a mapping with HCI approaches. In Table 14 (column 2) we show the references, which are discussed in our research and cited by other studies also discussed in our research.

We found one study (Venturi et al. (2006)) that used the phases of a software process to understand HCI in practice but it did not specify which HCI approaches could be used in each phase of a software process. Four (Venturi et al. (2006); Scheiber et al. (2012); UXPA Curitiba (2016); Salgado et al. (2016)) of out twelve studies did not discuss about the knowledge of HCI; and four (Hussein et al. (2009); Hussein et al. (2010); Hao & Jaafar (2011); Ardito et al. (2014)) of out twelve studies did not discuss about the use of HCI approaches. We will explain more about these studies in the next sections.

2.4.1. Knowledge about HCI approaches

In this section we will present some studies that discuss about the HCI knowledge in the industry context.

Ji & Yun (2006) and Hussein et al. (2012) presented the same way to investigate the level of knowledge of HCI techniques in the practice using questionnaires and likert scales as follows:

- In Ji & Yun (2006), the level of knowledge of usability and UCD was measured using a 7-point scale from 1 (very low) to 7 (very high). The mode (the value that appears most often in a set of data values) for the level of knowledge on usability is the point 5 of the scale.
- For Hussein et al. (2012) the level of knowledge on UX, interaction design, usability, information architecture and HCI was measured using a five-point Likert scale from 1 (strongly disagree) to 5 (strongly agree), and an option for “never heard” was provided. One respondent said never heard about HCI, interaction design and UX, and two respondents said never heard about information architecture. All respondents claimed to have heard about usability.

In both works, although the evaluation was done using likert scale the results were not presented using likert scale.

Vukelja et al. (2007) estimated the HCI knowledge based on free text answers about the knowledge in HCI domain, using two questions: (i) the first one concerns the sources (for instance, experience) about HCI knowledge and (ii) the other concerns the books known by the participants in this area. HCI knowledge was rating as “high” and “low”. For example, a professional who knew many books was rated as “high”. The authors concluded that only 8.3% (7/84) of respondents have a high knowledge in HCI.

Hussein et al. (2009) investigated the perception of participants of HCI terminology, where most of them had heard about HCI (75% - 6/8) and usability (87% - 7/8). However, five participants never heard about interaction design.

Hussein et al. (2010) is an extension of Hussein et al. (2009) studies' where the knowledge of terminologies used in the HCI field are investigated. A high percentage (63.1% - 53/84) of participants has never heard about HCI term; 64.3% (54/84) of the participants have heard of usability; 90.5% (76/84) of the participants never heard about usability standards of user interface; and 57.1% (48/84) of participants who have never heard about Interaction Design.

Hao & Jaafar (2011) investigated the knowledge of usability and its importance in the practice of design and development of the system using a likert scale from 1 (strongly disagree) to 5 (strongly agree) however the results were not presented using likert scale.

Ardito et al. (2014) did not use a measure to investigate the level of knowledge about HCI, but they concluded that many software developers do not know well what usability is, and they know even less about UX.

In Ogunyemi et al. (2016), 77% (17/22) of the organizations indicate to be aware of HCI and 23% (5/22) are not aware of HCI. Although 17 organizations claim to be aware of HCI, the responses about the HCI methods applied in their companies do not support this claim. Only three respondents described relevant methods related to HCI. In the interviews the authors concluded that the level of knowledge of HCI in the companies is inadequate.

Table 14. Studies about the state of HCI in practice

Authors	Similar related work references	Type of study	Sample	Country	Objective	Perception of knowledge of HCI	Use of HCI
Ji and Yun (2006)	-	Survey	184 information technology development practitioners and 90 user interface/ usability practitioners	Korea	To understand the usability and UCD adoption issues in development environments, overall assessment of UCD/usability, and the most widely used methods and techniques.	Level of knowledge of usability and UCD	Use of usability methods
Venturi et al. (2006)	-	Web survey	83 professionals	United States and European countries	To improve the understanding of UCD adoption and to learn what kind of organizational issues must be proposed.	-	Use of usability methods and techniques
Vukelja et al. (2007)	-	Survey	134 software developers	Switzerland	To regard the engineering practices of software developers with a special focus on the design and development of user interfaces.	Knowledge in the area of UCD	Use of software development methods
Hussein et al. (2009)	-	Semi-structured interviews and focus group	8 professionals	Malaysia	The aims of the study were: (1) to learn about the status of interaction design and HCI methodologies used in IT projects; and (2) identify the influencing factors that contribute to design decisions.	Awareness of common HCI terminologies	-

Authors	Similar related work references	Type of study	Sample	Country	Objective	Perception of knowledge of HCI	Use of HCI
Hussein et al. (2010)	Hussein et al. (2009)	Ethnography study (in-depth and semi-structured interview, questionnaires and observations)	84 professionals	Malaysia	The objective of the study was to investigating what is the ICT personnel's awareness of HCI in the different sectors and working levels.	Knowledge of terminologies used in the HCI field	-
Hao and Jaafar (2011)	Hussein et al. (2009) Hussein et al. (2010)	Structured questionnaire and semi-structured interview	14 companies	Malaysia	To understand and to evaluate if the practice of usability is in the ICT companies, specifically in the interactive computer-based system.	Knowledge on usability	-
Hussein et al. (2012)	Vukelja et al. (2007) Ji and Yun (2006)	Survey	59 professionals	Malaysia	To identify what is the product/system development process in practice and whether UXD is incorporated into that process.	Level of knowledge of HCI issues	Techniques used in development
Scheiber et al. (2012)	-	Empirical study (expert interview and survey)	27 semi-structured expert interviews, and 345 answers for the survey	Germany	To explore the status quo of the knowledge, the importance, and the actual use of usability concepts among enterprises in Germany.	-	Usability integration aspects
Ardito et al. (2014)	-	Questionnaire-based survey, interviews, a focus group, and an exploratory study	36 companies	Italy	To investigate how the companies address usability and UX when create products.	Know of usability and UX	-

Authors	Similar related work references	Type of study	Sample	Country	Objective	Perception of knowledge of HCI	Use of HCI
UXPA Curitiba (2016)	-	Survey	361 UX professionals	Brazil	To collect information about education, experience, demographics and organizations in Brazil those execute usability/UX practices.	-	Usability/ UX activities
Ogunyemi et al. (2016)	Ardito et al. (2014) Venturi et al. (2006)	Exploratory investigation (online survey, semi-structured interviews)	22 companies	Nigeria	To understand the state of HCI practices in Nigeria.	HCI awareness	HCI methods used in companies
Salgado et al. (2016)	Scheiber et al. (2012) UXPA Curitiba (2016)	Survey	26 companies	Brazil	The adoption of usability and UX practices with special focus on evaluation methods.	-	Usability/ UX activities

UCD = user-centered design; IT = information technology; ICT = information communication technology; UXD = user experience design; HCD = human-centered design; UX = user experience.

2.4.2. Use of HCI approaches

In this section we will present some studies that discuss about the use of HCI approaches in the industry context.

Ji & Yun (2006) and Hussein et al. (2012) presented the same way to investigate the level of use of HCI techniques in the practice as follows:

- In Ji & Yun (2006), the participants had the possibility to choose different HCI methods that they normally use, according to a defined list. The authors found that “Task analysis” is the technique most used by development practitioners and UI/usability practitioners. After that, the technique most used by the participants is “Evaluate existing system”.
- In Hussein et al. (2012) the participants chose, according to a list, which techniques they use in UI development. As a result, the authors found that the techniques most used by the participants are “Task analysis” and “User Acceptance Test”. When we compare the results of Ji & Yun (2006) and Hussein et al. (2012), we can note that “Task analysis” was the technique most used for both participants.

Venturi et al. (2006) asked the participants about the use of UCD methods and techniques (defined list) in the different phases of a development life cycle (business analysis, requirements, analysis, design, implementation, testing, and deployment). The most frequently used methods were: user interviews (66/83 practitioners); prototyping techniques (high-fidelity (62/83) and low-fidelity (60/83)); usability evaluation methods - expert and heuristic evaluation (58/83) and qualitative, quick and dirty usability test (57/83).

Vukelja et al. (2007) asked the participants (134) about the use of software engineering methods in the software development. Regarding tests from software engineering, they said that the modules of the system and the systems are tested. The modules are tested in 76.2% of cases, and the systems in 98.1% of cases. In 77.1% of cases the tests are conducted in parallel with the development and at the end; but in 20% of cases the tests are conducted at the end. Documentation for the end user is write in 34.2% of cases in parallel with the development and at the end; unfortunately in 65.8% of cases only at the end. Regarding usability tests, only in 37.9% of the cases this type of test is conducted.

Ogunyemi et al. (2016) focused their study on the use of usability testing, UX design, and prioritization of HCD. For usability testing 12/22 (55%) organizations indicated they always conduct usability testing and 10/22 (45%) organizations indicated they sometimes do. Seventeen organizations (77%) said that they address UX, and five organizations (23%) indicate that they do not address UX. Four organizations said that they apply ISO guidelines for HCD and usability, 11 organizations do not apply, and 7 organizations reported that they do not know about these ISO guidelines.

UXPA Curitiba (2016) presents an overview of the context of UX and Usability professionals (361) in Brazil focusing on the information about education, experience, demographics and organizations that these professionals work. The five main activities conducted by the respondents during their work are: Low Fidelity Prototypes (74.8%), High Fidelity Prototypes (67.3%), User Interviews (61.2%), Heuristic Evaluation (60.4%), and Usability Test (55.1%). We can see that the most frequent activities among respondents were related to development of prototypes.

Salgado et al. (2016) extended the UXPA Curitiba (2016) studies' focusing on 26 small businesses enterprises of interactive systems development. They explore how the main usability practices have

been conducted by these organizations. Among other questions, the authors asked the organizations about activities of Usability/UX, user characteristics based on evaluations, and characteristics of heuristic evaluations. As a result, they found that usability tests and heuristic evaluation were the evaluation methods most used. They also identified the need of training and awareness about the importance of usability and UX. In addition, the use of accessibility evaluation methods is low.

Venturi et al. (2006) investigated the use of HCI techniques/methods in a specific software process. For them, *business analysis phase* starts with an analysis of competing or existing products and user interviews; *requirements phase* is carried out through user interviews, early human factor analysis, and use case analysis; *analysis phase* used user interviews, use case analysis, and lo-fi prototyping; for the *design phase* both hi-fi and lo-fi prototyping and qualitative, quick and dirty usability tests are used. They said that lo-fi prototyping is as frequently applied as hi-fi prototyping, and this can be explained by the fact that computer-based prototyping has become more affordable and easy to perform than some years ago. In addition, the most frequently used evaluation methods are qualitative. But, according to the authors, in 1991 the scenery was not the same where the focus was on techniques for summative evaluation. For the authors the results show that the early involvement of UCD practitioners in the product life cycle is more frequent compared to 10 years ago, and UCD plays a particular role in the requirements, design, and analysis phases.

Although these studies were conducted in different ways, we analyzed the works that presented explicitly results about the use of HCI techniques. Table 15 presents the ten techniques and methods from each study. We can note that: (i) **Task analysis** is classified as the most used method for Ji & Yun (2006) and Hussein et al. (2012); (ii) **Lo-fi prototypes** and **Hi-fi prototypes** were placed between the three first places for Venturi et al. (2006), Salgado et al. (2016) and UXPA Curitiba (2016); (iii) **Usability tests** was placed in fifth position for Venturi et al. (2006), Salgado et al. (2016) and UXPA Curitiba (2016), and in ninth and tenth position for Hussein et al. (2012) and Ji & Yun (2006), respectively.

Table 15. HCI techniques/methods used in practice

Rank	Salgado et al. (2016)	UXPA Curitiba (2016)	Hussein et al. (2012)	Ji and Yun (2006)		Venturi et al. (2006)
	<i>Subject: software developer</i>	<i>Subject: usability/UX professionals</i>	<i>Subject: software developer</i>	<i>Subject: development practitioners</i>	<i>Subject: UI/usability practitioners</i>	<i>Subject: UCD practitioners</i>
1	User interviews	Lo-fi Prototypes	Task analysis	Task analysis	Task analysis	User interviews
2	Lo-fi prototypes	Hi-fi prototypes	User Acceptance Test	Evaluate existing system	Evaluate existing system	Hi-fi prototyping
3	Hi-fi prototypes	User interviews	User experience	User analysis/ profiling	User analysis/ profiling	Lo-fi prototyping
4	Contextual analysis	Heuristic evaluation	Evaluate existing system	Surveys	Surveys	Expert or heuristic evaluation
5	Usability tests	Usability test	Surveys	Scenarios of use	Scenarios of use	Qualitative, quick and dirty usability test
6	Heuristic evaluation	Personas	Heuristics evaluation, usability expert evaluation	Screen mock-up test	Heuristics evaluation, usability expert evaluation	Observation of real usage
7	Personas	Survey	Scenarios of use	Navigation design	Navigation design	Scenarios
8	Survey	Contextual analysis	User analysis/ profiling	Usability checklists	Usability checklists	Style guides
9	Remote usability tests	UX Training	Lab usability testing	Participatory design	Focus group interview	Early human factors analysis
10	Guidelines/ Checklist review	Card sorting	Navigation design	Lab usability testing	Lab usability testing	Competitive analysis

2.4.3. Other aspects related to HCI in practice

Some studies discussed about the obstacles found in the practice of usability/HCI/UX, as follows:

- Hussein et al. (2012) identified for instance that 62.7% (37/59) of respondents declared lack of understanding and/or knowledge about HCI/usability/UX; 54.2% (32/59) of respondents declared lack of engineers trained in HCI/usability/UX; and 49.2% (29/59) of respondents declared lack of practice in usability/HCI/UX methodologies;
- Ji & Yun (2006) also identified some obstacles related to the adoption of UCD: the lack of understanding/knowledge about usability/UCD (57.2% - 157/274); the lack of practice in usability/UCD methodologies (46.3% - 127/274); the lack of engineers trained in usability/HCI (37.9% - 104/274). In these studies, the respondents were invited to choose one or more options regarding the obstacles.

We can see that the results found by Ji & Yun (2006) were better than the results found by Hussein et al. (2012). Although these studies are not performed in the same context, we can say that six years later it still lacks the understanding and/or knowledge about HCI/usability/UX. Ogunyemi et al. (2016) point the lack of standard tools for integration (9% - 2/22) and the lack of knowledge of best practices (23% - 5/22) as challenges for HCD practice.

2.4.4. Synthesis about HCI in practice

We share the same interest as these authors regarding the investigation of HCI engineering practice in the industry. We understand HCI engineering as a branch of human knowledge that uses HCI approaches (models, methods, techniques and standards) to develop interactive systems. We are interested in to investigate if the SPCM models consultants in the industry (particularly in Brazil), know and use HCI approaches integrated in all phases of the system development. Also, our interest is to compare the results with SE approaches related to the same practices of SPCM models. For this reason, in Chapter 5 we will present two empirical studies where we used questionnaires (web surveys) to collect data about the perception of knowledge and use of HCI approaches in practice.

2.5. Synthesis and Conclusion

In this chapter we presented a state-of-the-art about some works that proposed the integration of HCI and SE domains. Among the usability capability/maturity (UCM) approaches that have been presented, several approaches are defined specifically for usability or user experience. They also consider the performance and management of usability processes in development projects, but no results exist about the implementation of these approaches in the industry. On the other hand, these approaches do not propose the integration of usability issues in/with software process capability maturity (SPCM) models.

According to Jokela (2001), Jokela et al. (2006), Jokela (2010), Salah et al. (2014) and Lacerda & Wangenheim (2018), the main limitations of the UCM approaches are: (i) the lack of how some models were developed; (ii) the lack of validation of some models; (iii) the lack of guidance for the use of some models in practice; and (iv) the unavailability of an evaluation process model.

We have also showed approaches and frameworks that propose the integration of usability activities/practices or usability techniques in different development process or standards. These frameworks and approaches can be used by organizations, but their focus of integration is not the SPCM models that are widely used in the industry.

Although these approaches of usability capability/maturity and integration exist, some authors show that HCI/usability approaches are not used or little used in industry. Bevan (2009) argues that usability standards are not used in industry because of the complexity of their documentation, being not easy for designers to use them.

In addition, although there are works (Vasmatzidis et al., 2001), (Nogueira & Furtado, 2013) and (Peres et al., 2014) that propose the integration of HCI with SPCM models, we concluded that these works present similar purpose than ours but not the same. The CMMI-DEV model states “what” to do for each practice, but it does not say “how” the aspects of software development activities can be addressed. In addition, the CMMI-DEV model does not states “what” to do for user interface development. For this reason, we believe that is important one approach that proposes “how” to integrate the HCI issues into the software process capability maturity (SPCM) models.

Towards the use in the industry, we argue that usability or HCI issues should support the models already in practice. The usability capability/maturity approaches previously presented are still research projects not applied in industry. Moreover, they deal only with HCI issues (focused on usability) with no integration with the software engineering activities as defined by the classical SPCM models. This scenario motivated us to work on the integration of HCI issues in two SPCM models largely used in industry: the CMMI-DEV (CMMI Product Team, 2010) and the MR-MPS-SW (Softex, 2016c).

In the next chapter we will present the first steps of our proposition of integration of HCI issues into the software process capability maturity (SPCM) models.

Chapter 3 – Integrating HCI approaches into SPCM models

3.1. Introduction

After presenting a state-of-the-art on software process capability maturity (SPCM) models, usability capability/maturity (UCM) models, software engineering (SE) and human-computer interaction (HCI) integration approaches, and the use of HCI approaches in practice, we can say that the integration of HCI approaches in SPCM models can be a good way to motivate the application of HCI issues in industry. Indeed, the use of SE and HCI approaches at the same time is important in the development of interactive systems when the companies implementing SPCM models. Based on this belief, we propose an approach that integrates HCI approaches (methods, techniques, standards and patterns) into SPCM models (Gonçalves, Oliveira, & Kolski, 2015), (Gonçalves, Oliveira, & Kolski, 2016a), (Gonçalves, Oliveira, & Kolski, 2016b) and (Gonçalves, Oliveira, & Kolski, 2017b).

This chapter presents the first three phases of our research methodology (see Figure 19) that aims to obtain a proposition of integration of HCI approaches into SPCM models. Since the goal of this thesis (see section Research Objectives in General Introduction chapter) is to support the analysis, design, implementation and evaluation of interactive systems, we will focus our study in the engineering process area category of CMMI-DEV and its correspondent process in MR-MPS-SW. Thus, this chapter starts (section 3.2) by presenting briefly these process areas/processes in order to facilitate the understanding of the study performed in the three phases of the research methodology. In section 3.3, we will describe in detail the first phase (*Study of the models*) of the research methodology. Then, in section 3.4, we will present the results of the second phase (*Identification of HCI approaches*). In section 3.5 we will describe the third phase (*Evaluation and Improvement with experts*). Then, in section 3.6 we will present a first proposition of recommendations to use HCI engineering approaches to support the development of interactive systems following SPCM models. Finally, we will finish the chapter with a synthesis and conclusion section (section 3.7).

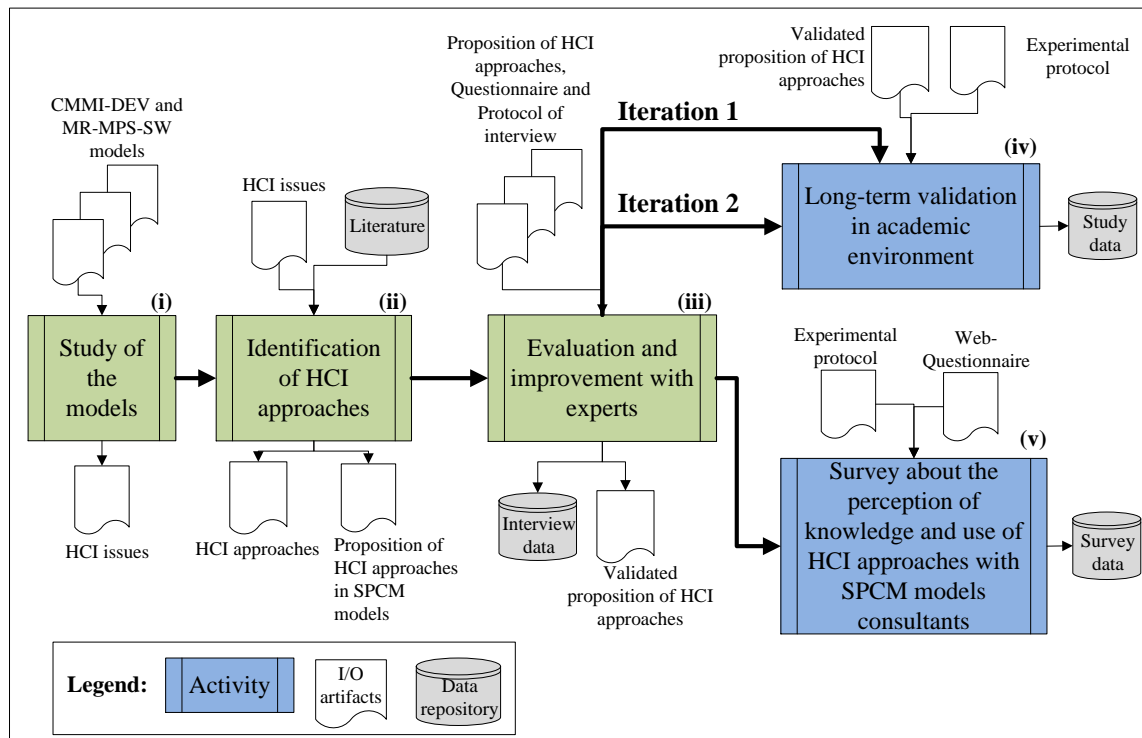


Figure 19. Research Methodology

3.2. Engineering Process Areas/Processes to be studied

The process areas/processes that are object of this thesis are presented in Table 16. For the national model MR-MPS-SW (Softex, 2016c) we keep the original acronym of the processes in Portuguese (CMMI Product Team, 2010). Regarding to the maturity level, these process areas are placed and evaluated in level 3. For the MR-MPS-SW model (Softex, 2016c) the same processes are evaluated in level D. Engineering process areas are responsible for the development and maintenance of engineering disciplines (CMMI Product Team, 2010).

Table 16. Process areas/Processes

Process areas (CMMI-DEV)	Processes (MR-MPS-SW)
Requirements Development (RD)	Requirements Development (DRE, in Portuguese)
Technical Solution (TS)	Design and Construction of the Product (PCP, in Portuguese)
Product Integration (PI)	Product Integration (ITP, in Portuguese)
Verification (VER)	Verification (VER)
Validation (VAL)	Validation (VAL)

Figure 20 shows the relationship among engineering process areas: Requirements Development (**RD** or **DRE**, in Portuguese), Technical Solution (**TS** or Design and Construction of the Product (**PCP**, in Portuguese), Product Integration (**PI** or **ITP**, in Portuguese), Verification (**VER**) and Validation (**VAL**).

The **RD** process area identifies customer needs, translates these needs into product requirements, and supplies these requirements to the **TS** process area. Then the **TS** process area analyzes the set of product requirements to produce a conceptual solution, and these requirements are used to establish an initial set of product component requirements. The **RD** process area also supplies requirements to the **PI** process area that combines the product components and verifies the interfaces.

Technical data packages for product components are developed by **TS** process area and used by **PI** process area. The **TS** process area relies in the **VER** process area that is responsible to perform design verification and peer reviews during design and prior to final build. It also verifies the interfaces and interface requirements of product components prior to product integration. The practices of the **VAL** process area are used during the product integration.

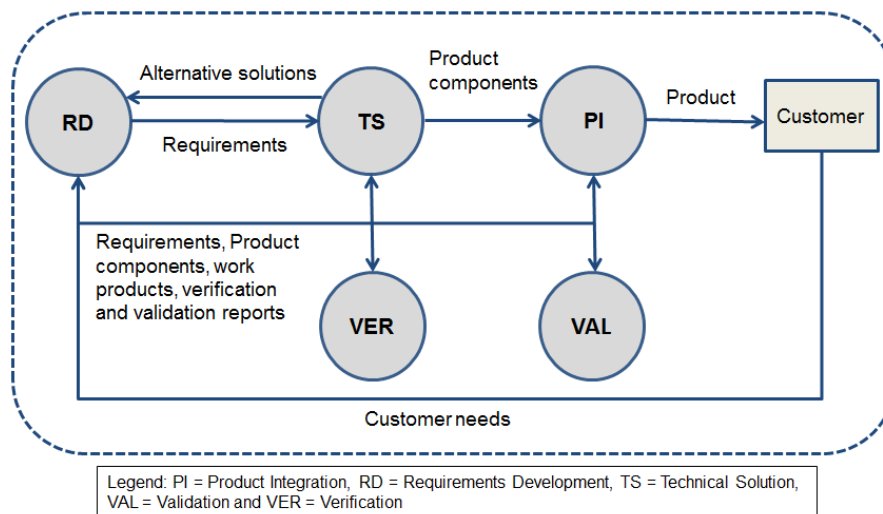


Figure 20. Relationship among engineering process areas (CMMI Product Team, 2010)

The **PI** process area uses the practices of both **VER** and **VAL** process areas in its implementation. Finally, the **VAL** process area validates products against the customer's needs, and the problems discovered during the validation are usually resolved in the **RD** or **TS** process area.

Figure 21 presents the specific goals (SG) and specific practices (SP) from all engineering process area.

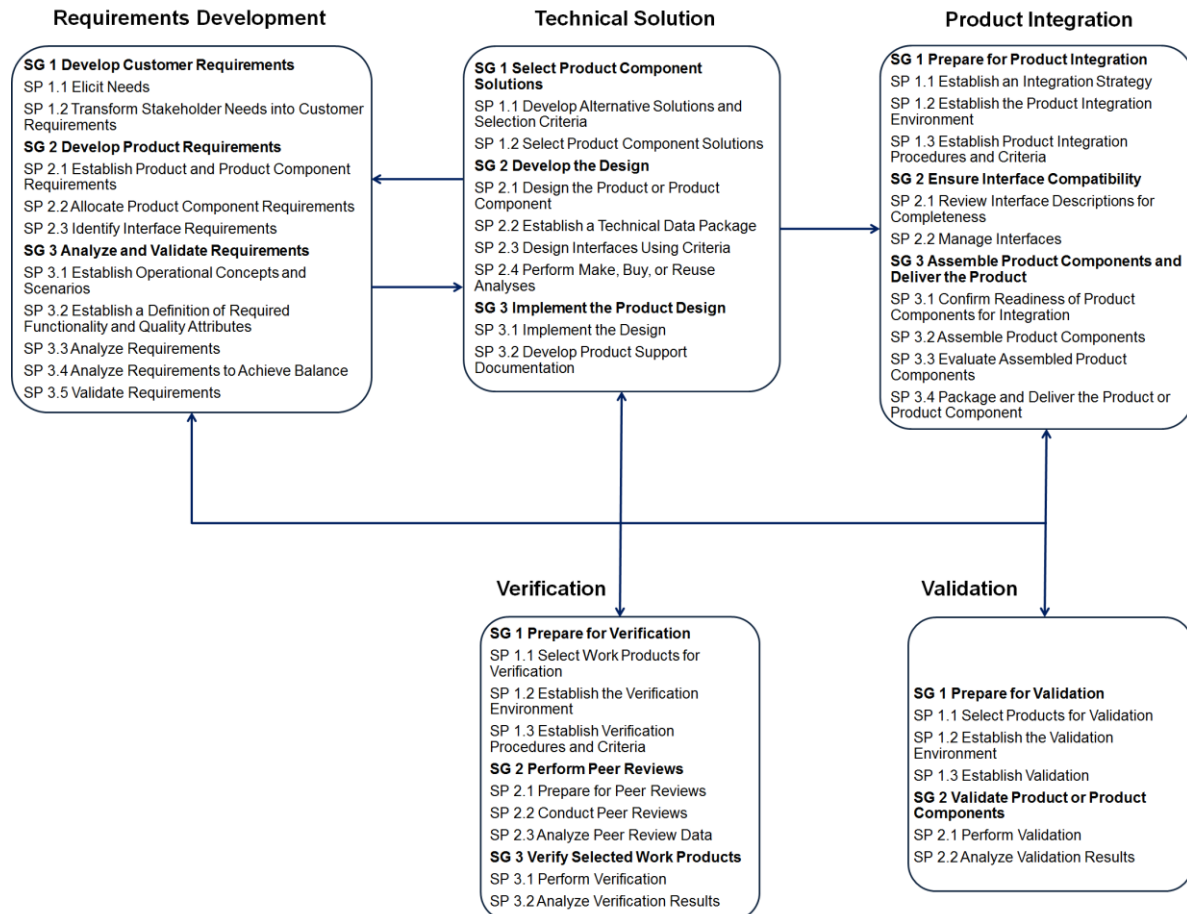


Figure 21. Process areas of the engineering category

3.3. Phase 1 - Study of the models

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.

The analysis of CMMI-DEV documentation consisted in reading the description of the model components (see Chapter 1), i.e. its required components (specific goals), expected components (specific practices) and informative components (sub-practices, example boxes, notes, references, sources, example work products; see Figure 22) for each engineering process area.

While reading CMMI-DEV documentation we looked for any citation of HCI issues. As stated by Jokela & Lalli (2003) the CMMI does not impose any requirements for usability, but however, it includes “hooks” where usability activities can be integrated. For the authors, the usability influence in the process areas but it is optional. Our goal is to indicate explicitly in these “hooks” which HCI approaches may be used while developing interactive systems. For instance, when we found citations

like *prototype* or *patterns* we analyzed them considering that we could use specific HCI approaches to produce them while developing interactive systems.

REQUIREMENTS DEVELOPMENT	
An Engineering Process Area at Maturity Level 3	
Purpose	Process area The purpose of Requirements Development (RD) is to elicit, analyze, and establish customer, product, and product component requirements.
SG 3	Analyze and Validate Requirements
	The requirements are analyzed and validated.
Specific Goal	The specific practices of the Analyze and Validate Requirements specific goal support the development of the requirements in both the Develop Customer Requirements specific goal and the Develop Product Requirements specific goal. The specific practices associated with this specific goal cover analyzing and validating the requirements with respect to the end user's intended environment.
	SP 3.5 Validate Requirements
Specific Practice	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.
Example box	Examples of techniques used for requirements validation include the following: <ul style="list-style-type: none"> • Analysis • Simulations • Prototyping • Demonstrations
	Example Work Products <ol style="list-style-type: none"> 1. Record of analysis methods and results
	Subpractices <ol style="list-style-type: none"> 1. Analyze the requirements to determine the risk that the resulting product will not perform appropriately in its intended use environment. 2. Explore the adequacy and completeness of requirements by developing product representations (e.g., prototypes, simulations, models, scenarios, storyboards) and by obtaining feedback about them from relevant stakeholders.
Reference	Refer to the Validation process area for more information about preparing for validation and validating product or product components. <ol style="list-style-type: none"> 3. Assess the design as it matures in the context of the requirements validation environment to identify validation issues and expose unstated needs and customer requirements.

Figure 22. Example of the CMMI model components (extract from (CMMI Product Team, 2010))

We started, therefore, seeking any explicit citation (see item (a) in Figure 23) that can be interpreted from HCI engineering point of view, by looking for: (i) HCI keywords (for example, *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 in benefit of the use of it (e.g. quality attributes, we can

interpret this as usability). We classify this information as implicit citations (see item (b) in Figure 23).

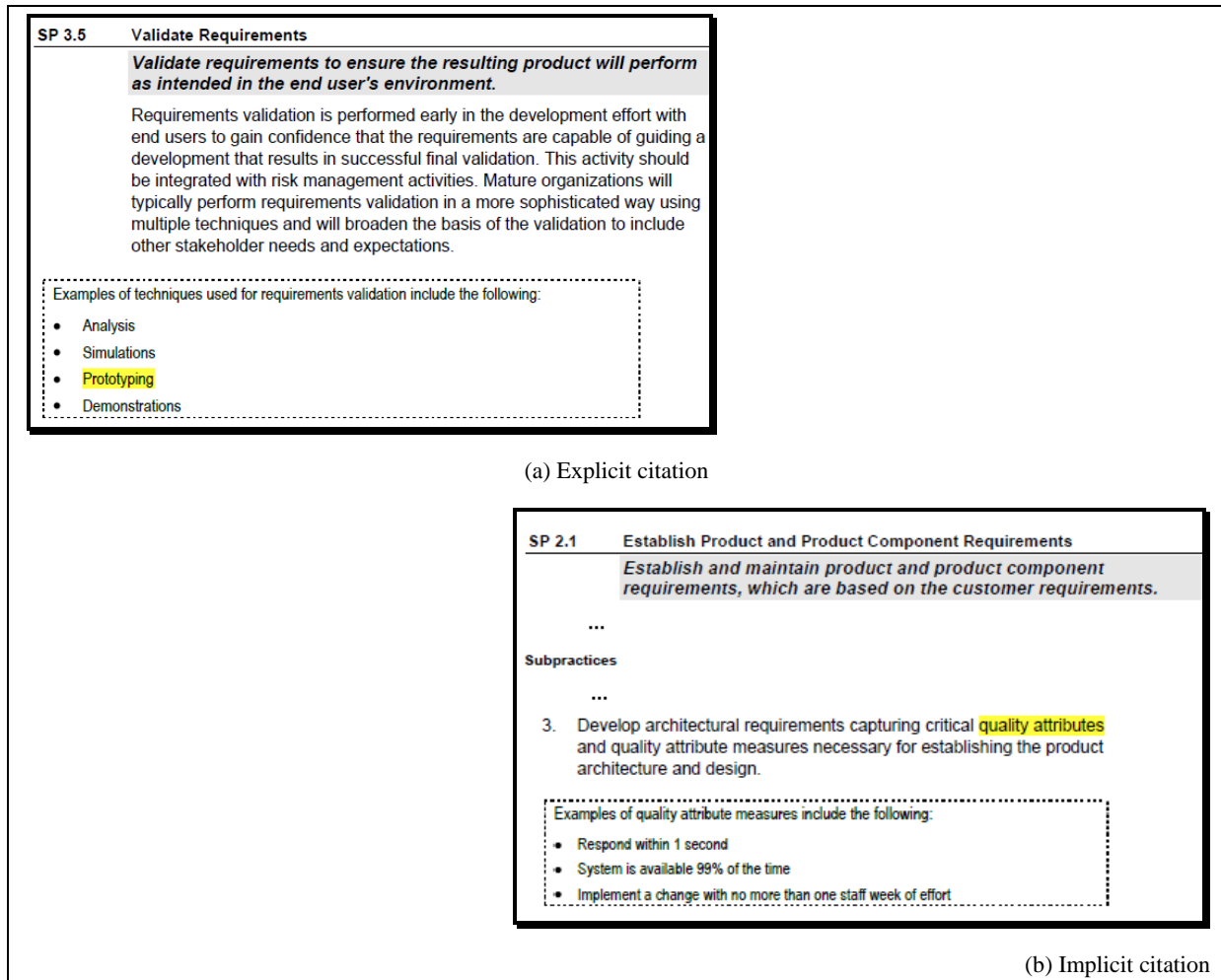


Figure 23. Examples of citations for Requirements Development (extract from (CMMI Product Team, 2010))

We present in Table 17 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), prototypes 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 to HCI issues. The analysis of all practices is presented in Annex A.

Table 17. 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.	<i>Subpractice 1:</i> “Engage relevant stakeholders using <u>methods for eliciting needs, expectations, constraints, and external interfaces.</u> ” <i>Examples of techniques:</i> “Questionnaires, interviews, and <u>scenarios obtained from end users</u> ”, “ <u>end-user task analysis</u> ” and “ <u>prototypes and models.</u> ”	Explicit
	SP1.2: Transform stakeholder needs, expectations, constraints, and Interfaces into prioritized customer requirements.	<i>Subpractice 1:</i> “Translate stakeholder needs, expectations, constraints, and interfaces into documented <u>customer requirements.</u> ” <i>Subpractice 2:</i> “Establish and maintain a prioritization of customer functional and <u>quality attribute requirements.</u> ” <i>Example Work Products:</i> “Prioritized <u>customer requirements</u> ”	Implicit
TS	SP1.1: Develop alternative solutions and selection criteria.	<i>Subpractice 4:</i> “Identify reusable solution components or applicable <u>architecture patterns.</u> ”	Implicit
	SP2.1: Develop a design for the product or product component.	<i>Subpractice 1:</i> “Establish and maintain <u>criteria</u> against which the design can be evaluated.” <i>An example of quality attribute:</i> “ <u>Usable</u> ”.	Explicit
PI	SP1.1: Establish and maintain a product integration strategy.	<i>Additional information:</i> “A product integration strategy addresses items such as: using models, <u>prototypes</u> , 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.	<i>Subpractice 4:</i> “Define <u>verification methods</u> to be used for each selected work product.” <i>Additional information:</i> “Verification for systems engineering typically includes <u>prototyping</u> , 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.	<i>Subpractice 2:</i> “Develop and refine <u>verification criteria</u> as necessary.” <i>An example of a source for verification criteria:</i> “ <u>Standards.</u> ”	Implicit
VAL	SP1.1: Select work products to be verified and verification methods to be used.	<i>Subpractice 4:</i> “Select the <u>evaluation methods</u> for product or product component validation.” <i>Examples of validation methods:</i> “Discussions with end users perhaps in the context of a formal review, <u>Prototype demonstrations.</u> ”	Explicit
	SP1.3: Establish and maintain procedures and criteria for validation.	<i>Subpractice 2:</i> “Document the environment, operational scenario, procedures, inputs, outputs, and <u>criteria for the validation</u> of the selected product or product component.” <i>An example of a source for validation criteria:</i> “ <u>Standards.</u> ”	Implicit

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 establish a technical data package for the project. SP2.3 refers to the interface between two functional components. SP2.4 is related to develop criteria for the reuse of product component designs and conduct 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 (only for one SP1.1 we found citations, the other eight practices we did not find) – 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.

3.4. Phase 2 - Identification of HCI approaches

The HCI literature was studied and from the analysis of CMMI-DEV documentation (presented in section 3.3), we proposed a set of HCI categories and examples (HCI approaches) identified from literature.

After identifying all citations, we organized them separately to identify the main approaches related to HCI and group them into HCI categories (Gonçalves et al., 2015), (Gonçalves et al., 2016a), (Gonçalves et al., 2016b) and (Gonçalves et al., 2017b). The categories' names were proposed based on the information collected from the literature.

Figure 24 presents the main keywords of the found explicit and implicit citations indicating which one helped in the identification of each defined HCI category.

From the analysis of all citations in the RD practices we identified five HCI categories (Gonçalves et al., 2016a): (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 eliciting needs*, *scenarios obtained from end-users*, etc.); (ii) *detailed operational concept and scenarios*, identified in the practice 3.1 that deals with “establish and maintain operational concepts and associated scenarios” (CMMI Product Team, 2010); (iii) *standards and guidelines for design interfaces*, for all citations that concern 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, analyzing citations for Verification and Validation (Gonçalves et al., 2016b), one new category was identified: *evaluation methods*, for all kind of evaluation techniques and methods used for verification, validation, and testing, such as peer review, inspection, and test. Since prototype in this analysis was related to the final validation, we refined the category Prototyping in two ones: *Prototype for HCI requirements*, which could include prototypes in papers, mockups, etc., and *Functional Prototype to validate HCI*, to represent the executable prototypes.

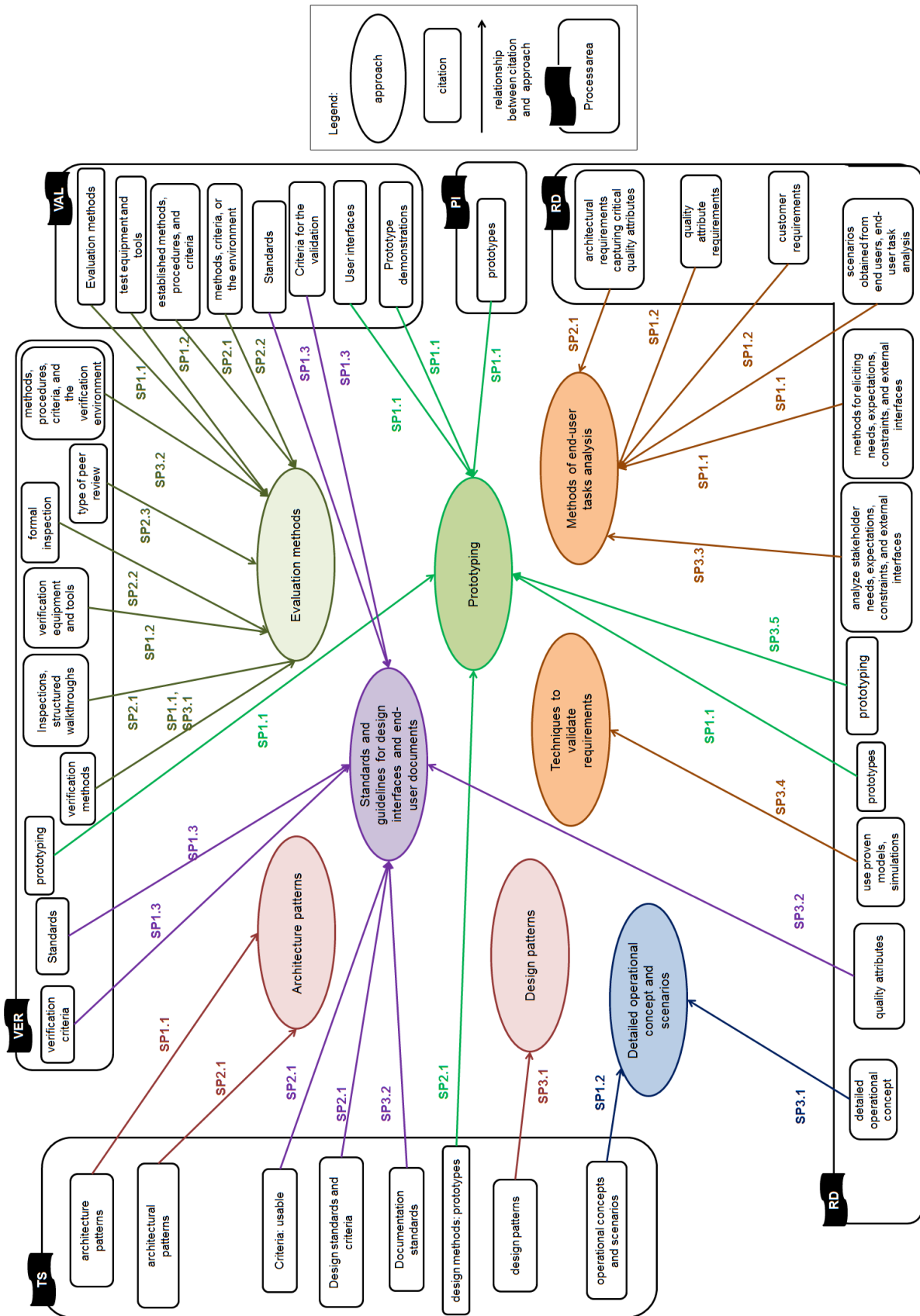


Figure 24. Analysis of the citations (implicit and explicit) to all process area

After this first analysis, we collected from the literature examples 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 as inspections, reviews, and so on; and, *Evaluation methods for HCI verification tests*, to include all kind of test.

With the identified categories we looked the literature to identify examples of HCI approaches that can be applied with any software development process and for several types of interactive systems. Table 18 presents all categories defined and the examples. Table 19 presents which categories could be applied when implementing CMMI-DEV practices for 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. We can note in Table 19, that we got a total of 33 propositions for 27 specific practices.

Table 18. HCI approaches to support CMMI-DEV

HCI Category	Examples
Task Analysis Methods for HCI	CTT (Concur Task Tree) (Paternò, Mancini, & Meniconi, 1997); K-MAD (Kernel of Model for Activity Description) (Limbourg & Vanderdonckt, 2004); HTA (Hierarchical Task Analysis) (Limbourg & Vanderdonckt, 2004); SADT (Structured Analysis and Design Technique) (Ross, 1977); GTA (Groupware Task Analysis) (Veer, Lenting, & Bergevoet, 1996).
Prototype for HCI requirements	Rapid Prototyping (Beaudouin-Lafon & Mackay, 2009): (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 (Courage & Baxter, 2005); (Maguire, 2001); Adapting to context (Courage & Baxter, 2005); (Maguire, 2001); User profile (Courage & Baxter, 2005); (Maguire, 2001); Persona (Courage & Baxter, 2005); (Maguire, 2001); Use cases (CMMI Product Team, 2010).
Standards and Guidelines for design and documentation of HCI	Ergonomic Criterion (Scapin & Bastien, 1997); ISO/IEC 9126-1(International Organization for Standardization/International Electrotechnical Commission, 2001); ISO 9241-11(International Organization for Standardization, 1998); ISO/IEC 25000 (International Organization for Standardization/International Electrotechnical Commission, 2014).
Techniques to validate HCI requirements	ProtoTask (K-MAD) (Caffiau, Girard, Scapin, Guittet, & Sanou, 2008); Task Model Simulator (CTT) (Paternò, 2004); Focus Group to validate requirements (Nielsen, 1997).
Architecture Patterns for HCI	Arch Model (Bass et al., 1991); PAC (Presentation-Abstraction-Control) Model (Coutaz, 1987); MVC (Model-View-Controller) Model (Goldberg, 1984).
Design patterns for HCI	Pattern to Interaction Design (Borchers, 2001); Pattern Languages in Interaction Design: Structure and Organization (van Welie & van der Veer, 2003); Designing interfaces: Patterns for Effective Interaction Design (Tidwell, 2010).
Functional Prototype to validate HCI	Iterative and Evolutionary Prototypes (Beaudouin-Lafon & Mackay, 2009): User interface toolkits, User interface builders, User interface development environments.
Evaluation methods for HCI verification tests	Usability tests (Dumas & Fox, 2009);(Shneiderman et al., 2009): Exploratory tests, Assessment tests, Validation or verification tests, Comparison tests; Validation by HCI expert(s) (Shneiderman et al., 2009).
Evaluation methods for HCI review	Heuristic evaluation (G Cockton, Woolrych, & Lavery, 2009); Cognitive walkthrough (G Cockton et al., 2009); Groupware walkthrough (G Cockton et al., 2009).

3.5. Phase 3 – Evaluation and improvement with experts

In the third 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 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 in case of the expert's judge necessary.

Table 19. HCI approaches x CMMI-DEV Practices

Categories	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					

3.5.1. Planning the evaluation

We had on mind that our initial proposal should be used as start point for an evaluation by the experts and, mainly, for its improvement with their suggestions. Considering that we could have a lot of items to evaluate (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 a questionnaire. To ensure the best quality of the results, we set as a profile that the experts should have a Ph.D., experience in HCI and should be well-known in HCI community (e.g., be program chair or member of program committee of HCI conferences).

At the beginning, in the first interviews with the experts, we were confronted with the following constraints: the experts imposed a limited time for an interview in a maximum of 2 hours and they would not feel comfortable with recorded interviews. With these constraints and the kind, we concluded we should do semi-structured interviews (Lazar, Feng, & Hochheiser, 2010) where the questionnaire was used to support the discussion in each interview section. One could think about the use of Delphi technique¹³ (Ab Latif, Mohamed, Dahlan, & Mat Nor, 2016), (Yousuf, 2007) and (Linstone & Turoff, 2002) usually used to make 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 (Linstone & Turoff, 2002) and (Yousuf, 2007); that was not our case since the proposed HCI approaches were available and consolidated in the literature;
- ii) usually with Delphi technique we have a series of questionnaires (Ab Latif et al., 2016), (Linstone & Turoff, 2002) and (Yousuf, 2007) with different kind of questions; in our case we had one questionnaire with 33 items to be validated (see Annex B);
- iii) the Delphi technique consumes a lot of time (Ab Latif et al., 2016), (Linstone & Turoff, 2002) and (Yousuf, 2007) and requires several sections for the result be effective; in our case we should plan only two-hour of interview with each expert.

Considering this plan (using structured interviews in a delimited frame of hour), we contacted 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 perform an interview. To assure that experts have the necessary background we selected only professionals that have experience (academic and/or industrial) in HCI domain and had a Ph.D. degree in HCI domain or Computer Science/HCI domain or Software Engineering/HCI domain. Most of the experts were selected from the research contacts of one of the advisors. This advisor has more than thirty years of experience in HCI, and that 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 20 presents the background and origin (country) from the 20 experts who participated in the interviews. We note that in average they have 19 years of experience (from 7 to 40 years). The experts are from 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 the industry, and 7 come from software engineering domain as well as HCI.

As can be noted in Table 20, in total twenty experts were interviewed. We have noted in the literature that studies with experts go from 11 experts (Dyba, 2000) to 30 experts to (El Emam & Madhavji, 1996) and samples of twenty experts are relevant to gain expert feedback (Beecham, Hall, Britton, Cottee, & Rainer, 2005). Based on these studies, we considered our sample of experts as acceptable to continue to the next phase of our research methodology.

¹³ Delphi “may be characterized as 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” (Linstone & Turoff, 2002).

Table 20. List of Experts (adapted from (Gonçalves et al., 2016a))

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.

To support the interviews, we elaborated a specific questionnaire (see Annex B). The questionnaire was composed of some introductory notes about the study, an overview of the CMMI process areas and about their main components (goals, specific goals, work products and sub-practices), personal information (name, education, profession, if they have 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 25).

Since the questionnaire was planned to be the thread of the interviews, it was elaborated in three languages possible to 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 used only as a start 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.

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 (Goldberg, 1984) • PAC (Presentation-Abstraction-Control) Model (Coutaz, 1987) • Arch Model (Bass et al., 1991)		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 25. Extract of a filled questionnaire

3.5.2. Performing the Interviews

The interviews were performed in 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), and 13 experts were interviewed in person and 7 by video conference (see Table 20). We present in Figure 26 an activity diagram that describes 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 agree” or “disagree”, they should justify with a description in the justification column.

In the same time, the interviewer took notes of verbal observations/explanations performed by experts during the interview and if is necessary, the interviewer has consulted the CMMI-DEV document for the specific practice in evaluation and presented the information to the expert. Figure 25 presents an extract of a form filled by one expert during an interview.

All the information collected was registered in the questionnaire by the expert, when in person, or by the interviewer and/or the expert when by video conference. Moreover, as previously mentioned, the

interviewer took informal notes of interesting comments made by the experts (references, sites do be later consulted, example of tools, etc.). During the interviews, we explained that the questionnaire was a support for the interview, which they should answer and justify each question, if necessary. We also explained that after all interviews we would analyze the results seeking to better integrate the results of all the interviews.

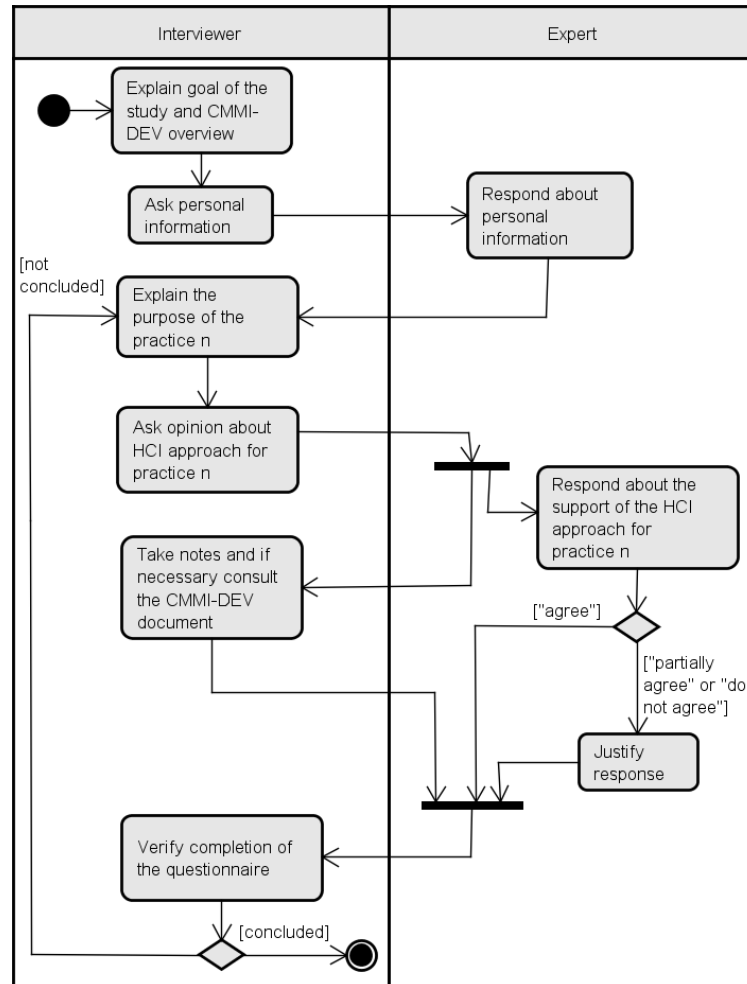


Figure 26. Interview sessions (adapted from (Gonçalves et al., 2016a))

3.5.3. Analysis and Synthesis of HCI approaches

We analyzed quantitatively and qualitatively all results of the interviews to make a synthesis of the proposition. Once the experts work with different kinds of interactive systems (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 analyses all comments of the experts. With this analysis we decided the improvements in a way that we could integrate as much as possible all propositions respecting the opinion of the experts.

To have an overview of the evaluation of the propositions, we calculated the mode¹⁴ and median value¹⁵ (see Table 21) for each practice of each process area. As the median is the value of the middle-

¹⁴ Mode is “the value of the most commonly occurring item” (Fenton & Pfleeger, 1998), p. 66.

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 based on the mode, since the mode is the value of the most commonly occurring answer. We note in Table 21 that we have 25 agreements, 7 partially agreements and only 1 disagreement (that probably means this proposition should be eliminated). As consequence, we concluded that in general our proposition was considered acceptable and improvements should be performed based on the comments of the experts.

Table 21. Mode and Median of the results (Agree - A, Partially agree - PA, Don't agree - DA)

HCI Categories		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
TS	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
	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
PI	Design patterns for HCI	SP3.2	PA	PA	PA
	Prototype for HCI requirements	SP3.1	A	A	A
	Functional Prototype to validate HCI	SP1.1	DA	PA	DA
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 analyze all comments in order to identify the propositions of improvements we considered the following:

¹⁵ Mode and median is usually recommended when we work with nominal and ordinal scales (see for instance, (Fenton & Pfleeger, 1998), p. 57).

- the full agreements (*I agree*) mean that the expert agrees with the proposition. When comments were included by the experts, they were also analyzed;
- the partial agreements (*I partially agree*) mean that the expert agrees but he or she suggests some modification; for instance, inclusion of other examples or, split the category into two new more specialized categories or rename 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 done; 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 carefully the suggestions and exclude only when we found enough confirmation that the proposition was not very adequate in the comments from other experts (who answered partial agreement).

We planned to analyze the comments in several steps. First, one of the researchers organized all responses in the same document (see an example in next section – Table 23), analyzed them and synthesized some improvements based on this analysis. After that, a second researcher reviewed all comments of the experts and the first synthesis of improvements, performing some modifications. Finally, all authors reviewed the comments and the synthesis of improvements in a three-hours meeting.

3.5.3.1. General Analysis

We started our data analysis by looking the general results for all practices for the five process areas (RD, TS, PI, VER, VAL). As presented in Table 22 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 propositions because they considered they did not know enough about the propositions. Expert E4 did not evaluate the proposition SP3.4 of RD; the expert E10 did not evaluate the propositions SP1.2 for VAL and VER, and the proposition SP3.1 of TS; and the expert E18 did not evaluate the proposition SP3.1 of TS. As consequence, we got 655 responses to analyze (see Table 22).

Table 22. 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 27 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 less (2/119 – 2%) and more (11/40 – 27%) disagreements respectively. We also note that we got only 8% of disagreement in the whole evaluation.

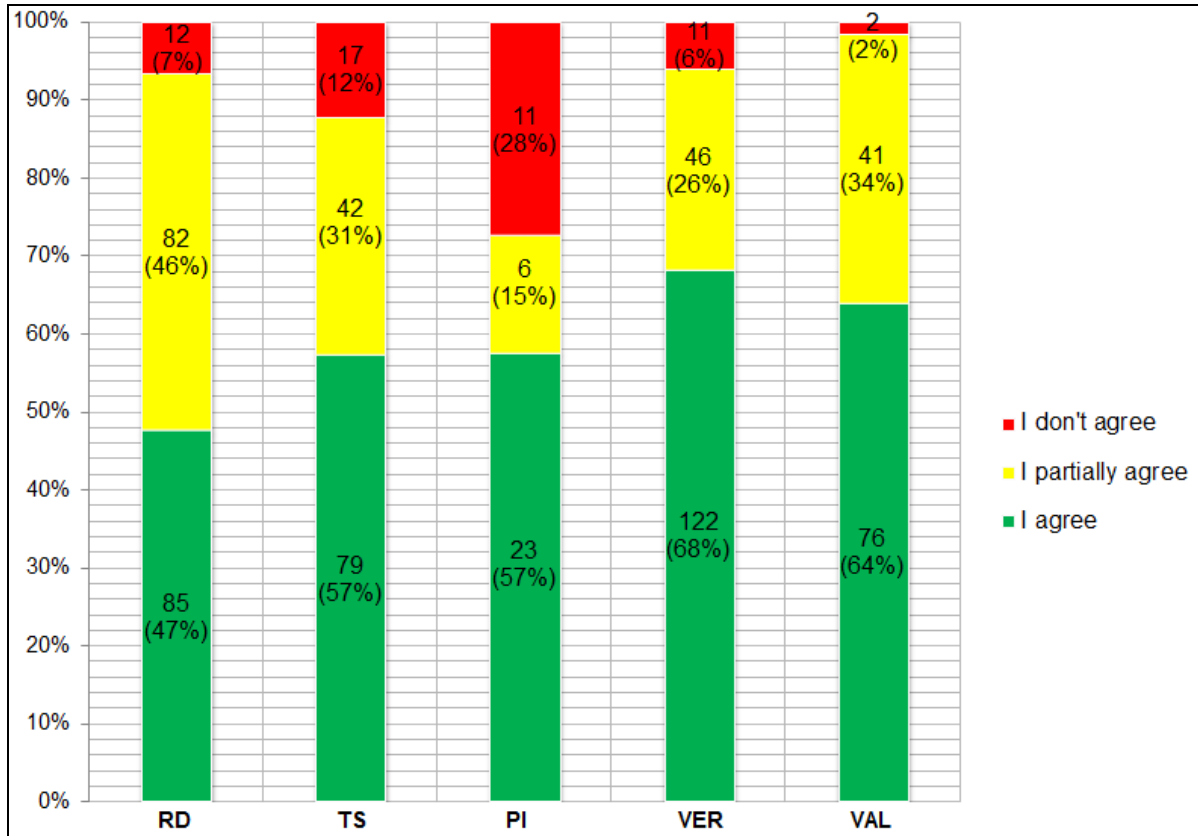


Figure 27. Detailed results by process area

3.5.3.2. Analysis for Requirements Development (RD)

In Figure 28 we can observe the details of the quantitative results for RD process area, considering the practices and the proposed HCI categories. Five HCI categories were proposed for RD (Gonçalves et al., 2016a) 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 of 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 the 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 experts who completely agree 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;

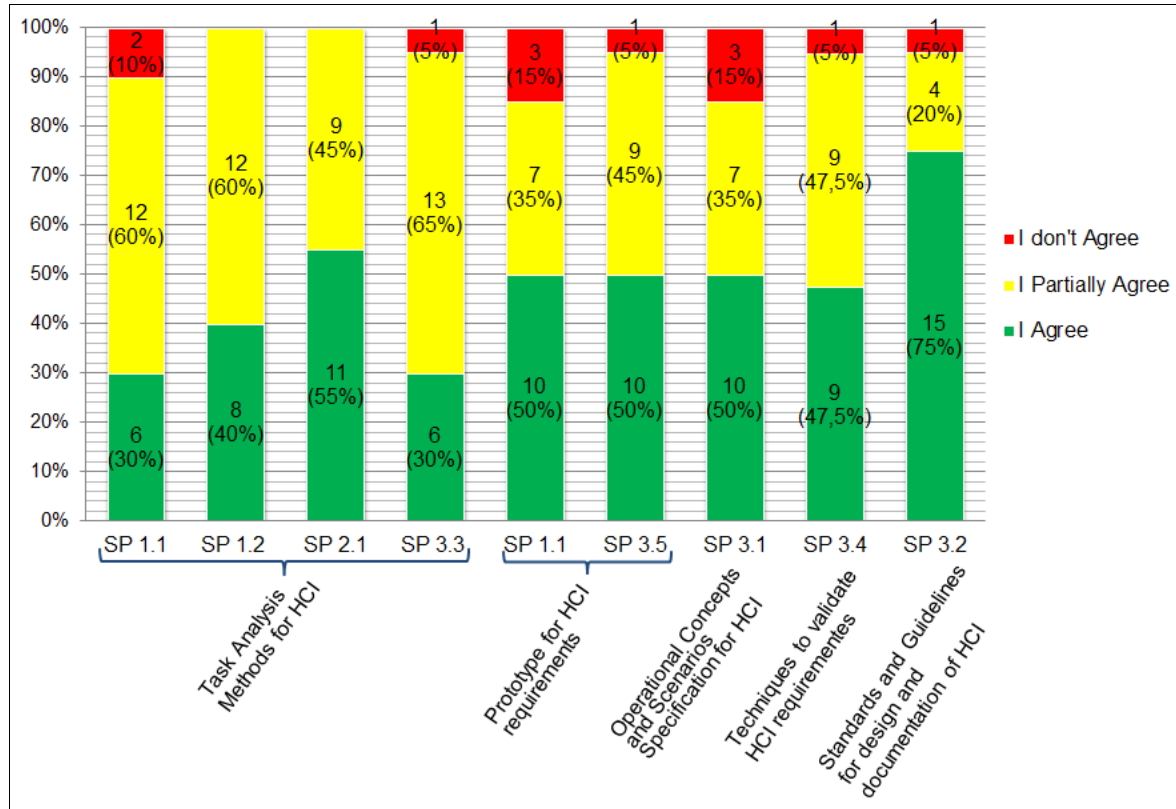


Figure 28. Quantitative results for Requirement Development (category and practice)

- *Techniques to validate HCI requirements* category can be considered adequate to SP3.4 since we had only one disagreement. However, 9 partial agreements indicate that improvements should be done; 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 equal from their questionnaire to the column in which it was given the answer on the ordinal scale. We used quotes when the comment was written by the experts in their forms, and brackets when the notes were introduced by us in the justification column during the interview. Table 23 presents an example of this form for the justifications of answers for the association of *Task analysis methods for HCI* with SP1.1 (Elicit Needs) and SP1.2 (Transform Stakeholder Needs into Customer Requirements).

With this form on hand, we 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 of 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.

Table 23. Example of analysis to RD SP1.1 and RD SP1.2

Category	Task Analysis Methods for HCI	Practice	SP 1.1
I agree	<i>E1, E2, E6, E13, E20, and E5- [To include a new category with elicitation techniques.]</i>		
I partially agree	<p><i>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.]</i></p> <p><i>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.]</i></p> <p><i>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.]</i></p> <p><i>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."</i></p> <p><i>E10- "Other steps and methods are required between obtaining needs and analysis tasks." [The expert suggests including personas and scenarios.]</i></p> <p><i>E11- "Possible inappropriate use of the model. I suggest the inclusion of Questionnaires, Focus Group, Scenarios, and Personas."</i></p> <p><i>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.]</i></p> <p><i>E15- "I suggest the inclusion of techniques to requirements elicitation: Brainstorming, Questionnaires, Interview, Observation, and Ethnography. Also, to include Scenarios."</i></p> <p><i>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."</i></p> <p><i>E17- "It does not integrate the contextual aspect. I suggest observation of the work environment."</i></p> <p><i>E18- "First of all makes scenarios to validate the understanding of needs with users. Also, use personas."</i></p>		
I don't agree	<p><i>E9- "Task Analysis is further modeling activity than an elicitation activity."</i></p> <p><i>E12- "Customer expressed through the prototype verification." [Use informal techniques.]</i></p>		
Conclusion	Replace the category with the two specialized categories: <i>Techniques to identify user needs</i> (proposed by E5, E8, E11, E3, E4, E14, E15, and E19); and <i>Techniques to identify user and organizational requirements</i> (proposed by E7, E8, E10, E11, E16, E17, E18, E14, and E15).		
Category	Task Analysis Methods for HCI	Practice	SP 1.2
I agree	<i>E1, E2, E3, E5, E8, E12, E19, and E6- "The prototyping is important in this practice"</i>		
I partially agree	<p><i>E4- "These methods allow focusing on priority tasks if used to communicate with the customer."</i></p> <p><i>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.]</i></p> <p><i>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."</i></p> <p><i>E10- "Translate the prioritized requirements in functionalities. Brainstorming is needed."</i></p> <p><i>E11- [It is not enough. The expert suggests the inclusion of quality attributes.]</i></p> <p><i>E13- "Other techniques to add to prioritization."</i></p> <p><i>E14- "The model includes critical tasks but does not prioritize the needs."</i></p> <p><i>E15- "To include Prototype."</i></p> <p><i>E16- "All aspects must be considered." [The experts suggest new methods for organizational context (FRAM and STAMP) - new category.]</i></p> <p><i>E17- "It does not integrate the contextual aspect. I suggest observation of the work environment."</i></p> <p><i>E18- "Only if it is a high level of abstraction. To add Scenarios and textual representations."</i></p> <p><i>E20- "I do not agree with SADT."</i></p>		
I don't agree	-		
Conclusion	Replace the category for the one specialized category <i>Techniques to identify user and organizational requirements</i> (proposed by E7, E10, E17, E18, and E4). The inclusion of two categories: <i>Prototype for HCI requirements</i> (proposed by E6, E9, and E15); <i>Standards and Guidelines for HCI design</i> (proposed by E11, E13, and E14).		

Thus, the suggested elicitation techniques were separately organized in a different document (see Table 24). Based on literature (Courage & Baxter, 2005); (Maguire, 2001), the techniques were grouped into two different categories: *Techniques to identify user needs* (techniques marked in gray in Table 24) and *Techniques to identify user and organizational requirements* (the one marked in black). 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 focus on the specification of requirements.

Table 24. Analysis of the suggested techniques (adapted from (Gonçalves et al., 2016a))

Technique \ Expert	E3	E4	E7	E8	E10	E11	E14	E15	E16	E17	E18	E19
Requirements specification templates (such as: VOLERE, RESCUE, IEEE)												
Interview												
Questionnaire												
Brainstorming												
Field study												
Scenarios												
Personas												
Storyboard												
Focus group												
Ethnography												
Task analysis												
Organizational Context												

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 the 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 consequence, we decided to rename the category *Task Analysis Methods for HCI* to *Task Modeling*.

In the final meeting, we re-analyzed the comments and confirmed the need for creation of the two categories for identification users' needs and requirements. Analyzing the comments from disagreement and partial agreement, we noted that they suggest the same idea: methods of elicitation should be performed to support both practices. For SP1.1 two experts disagree with the use of task modeling approaches. One suggests eliminating it, but another suggested only integrating other approaches. Analyzing the 12 partially 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 partially agreement that only suggests new approaches, and several agreements (6 for SP1.1 and 8 for SP1.2), we decided to keep this category to support both practices.

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

- For SP1.1 (Elicit Needs) we had 7 partially agreements and 3 disagreements for the category *Prototype for HCI requirements*. The three experts who disagree consider that when we elicit needs, it is too early to use prototypes. The partially agreement justifications were also in the same direction 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 have 10 agreements, we had noted that the experts who agree explained 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 (Establish Product and Product Component Requirement) we had 9 partially agreements and no disagreements. Four experts suggest including the use of the standards to

establish the non-functional requirements for user interfaces. Thus, the category *Standards and Guidelines for HCI design* was included for this practice.

- For SP3.1 (Establish Operational Concepts and Scenarios) we had 7 partially agreements and 3 disagreements. The category *Operational Concepts and Scenarios Specification for HCI* was eliminated since the experts justified their partially agreement and disagreements showing that this category was miscellaneous of approaches. Eight experts who evaluated as partial agreement and the three experts, who agreed, suggest that several mentioned approaches should be included in a category specifically to organizational requirements. These suggestions were considered in the analysis done in Table 24 and placed in the new category *Techniques to identify user and organizational requirements* that was therefore associated with this practice.
- For SP3.2 (Establish a Definition of Required Functionality and Quality Attributes), as mentioned we got 75% of agreements and the partial agreements were justified only with the inclusion of new examples, were all considered when proposed by more than two experts. The disagreement was from an expert of 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 was full agreement) we did not consider this disagreement.
- SP3.3 (Analyze Requirements) and SP3.4 (Analyze Requirements to Achieve Balance) 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 (Validate Requirements) we had 9 partially agreements and 1 disagreement. From them, 4 experts suggest including the category of *Techniques to validate requirements* to support this practice. The inclusion of the category was done. The disagreement just mentioned that prototypes should be simple combining scenarios and screens, not functional what do not justify the exclusion of this category.

3.5.3.3. Analysis for Technical Solution (TS)

Based on the quantitative results presented in Figure 29 and justifications in the questionnaires organized in the same way of the one presented in Table 23 for RD, we concluded that:

- *Architecture Patterns for HCI* should be kept for SP2.1 (Design the Product or Product Component) since the partial agreement included only some comments about the use of the practice observing that the patterns for HCI should be used in the beginning at a high level (for instance, three tiers architecture) what is confirmed by the only disagreement. However, for SP1.1 (Develop Alternative Solutions and Selection Criteria) partial agreements, two experts suggest the use of techniques for interaction dialog (such as UAN – User Action Notion (Hartson, Siochi, & Hix, 1990), MoLIC – Modeling Language for Interaction as Conversation (Barbosa & Paula, 2003), TAG – Task Action-Grammar (Brown, 1996)). 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 defend 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.

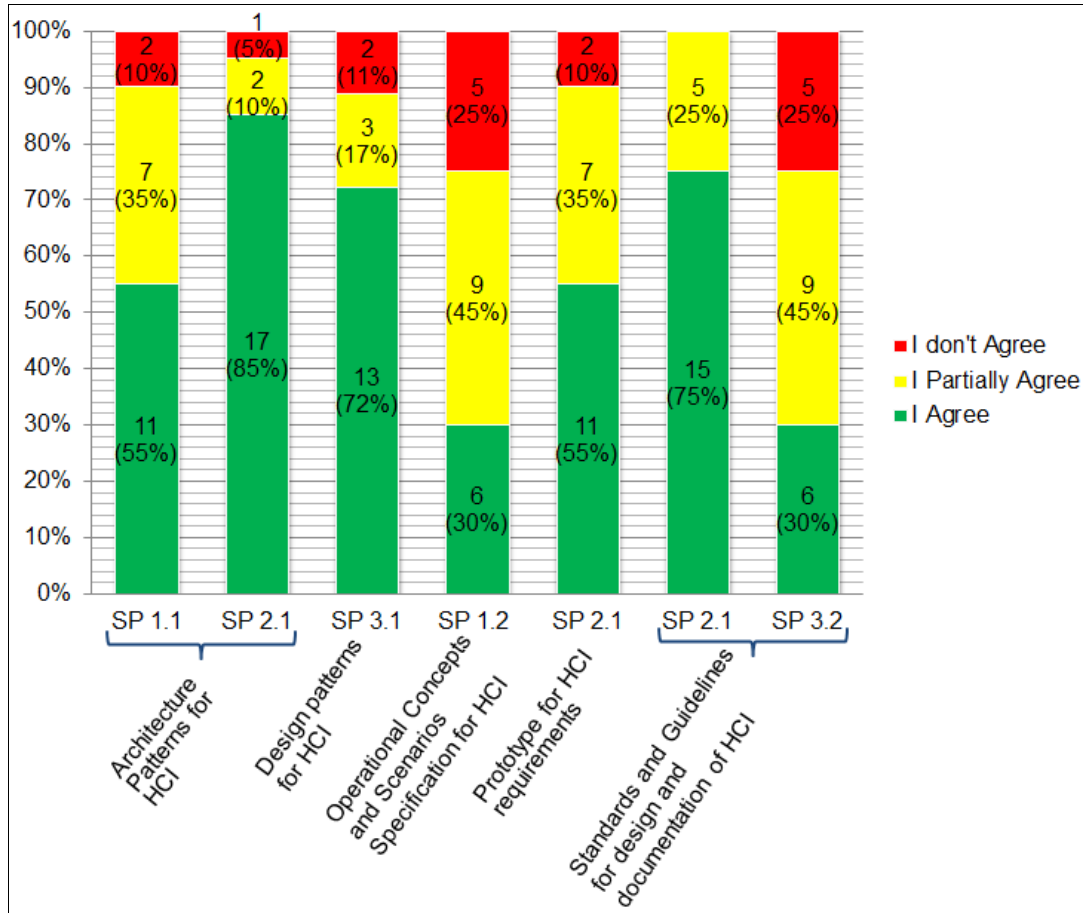


Figure 29. Quantitative results for Technical Solution (category and practice)

- *Design patterns for HCI* should be kept for SP3.1 (Implement the Design) thanks to the large agreement (13 full agreements and 3 partial ones that only give new suggestions of approaches); and the justification for the disagreements is that design patterns should be used only for the implementation, what is contrary to the common sense of using them since the design phase. The category *Architecture patterns for HCI* is also included as a set of approaches to support this practice by the suggestion of two experts.
- *Prototype for HCI requirements* should also be kept for SP2.1 (Design the Product or Product Component); 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 evolved. Other experts recommend also that the prototypes should be done by participatory design with the users in an interactive development. Those comments will be included as recommendations while using the prototypes.
- *Standards and Guidelines for Design and Documentation of HCI* should also be kept for SP2.1 (Design the Product or Product Component) since the partial agreements, in general, only comment that the standards are useful but do not reach the final goal. For SP3.2 (Develop Product Support Documentation), 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 respond 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 for *Standards and Guidelines for Design* being not associated anymore to this practice.

- *Operational concepts and scenarios specifications for HCI* category was a miscellaneous of approaches. The justifications for the disagreement of the proposition for SP1.2 (Select Product Component Solutions) 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 and E8) say 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 (Fathy & Kornysheva, 2016). Both suggestions are not specific for 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 consequence, this practice had no proposition specific for HCI being supported by the software engineering approaches currently used to select any product component solution.

3.5.3.4. Analysis for Product Integration (PI)

PI process area had only two propositions to be evaluated as presented in Figure 30:

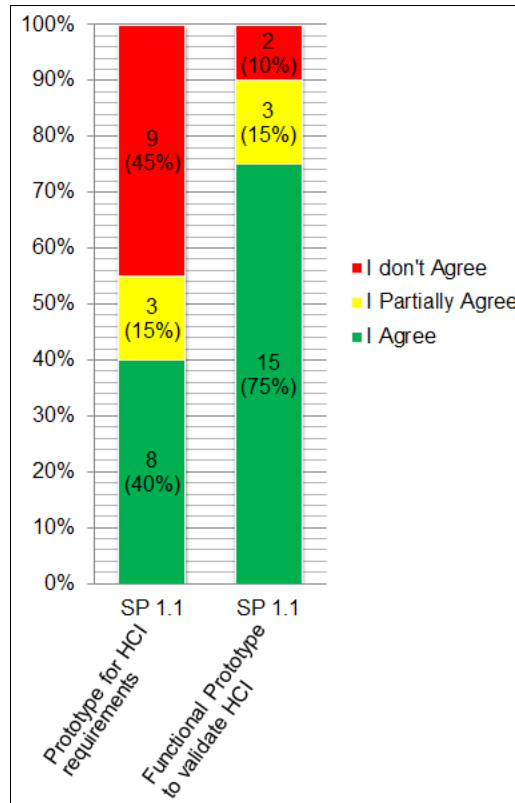


Figure 30. Quantitative results for Product Integration (category and practice)

- *Functional Prototype to validate HCI* should be kept (15 agreements) and the three partially agree only suggest some recommendations (such as the use of a good IDE – Integrated Development Environment). Therefore, it should be kept for the practice SP1.1 (Establish an Integration Strategy). Moreover, the disagreements were not valuable 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 (Establish an Integration Strategy) being the proposition that had more disagreement. Moreover, by calculating the median and mode of the experts' opinion we obtained as result "I don't agree" with this proposition. The disagreements were consensual saying that it is late to do prototypes for requirements and that it does not help in the integration, which was also confirmed by two partial agreements (E8 and E20). As consequence, despite six experts have agreed with this proposition, the analyzed justifications determine the exclusion of this proposition.

3.5.3.5. Analysis for Verification (VER)

In Figure 31 we can see the quantitative results for VER process area. In general, we note that:

- the propositions *Standards and guidelines for design and documentation of HCI* for SP1.3 (Establish Verification Procedures and Criteria) and *Evaluation methods for HCI verification tests* for SP2.1 (Prepare for Peer Reviews), SP2.2 (Conduct Peer Reviews) and SP2.3 (Analyze Peer Review Data) had no disagreement, which means they should be kept. The partial agreements were only suggestion of new examples in each category that were, therefore, included.
- *Functional prototype to validate HCI* for SP1.1 (Select Work Products for Verification) the 3 experts who disagree argue that the assessment should be carried out with the final product and not a prototype. However, the experts who partially agree consider 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 to *Iterative and Evolutionary prototype (system versions)*.

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

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

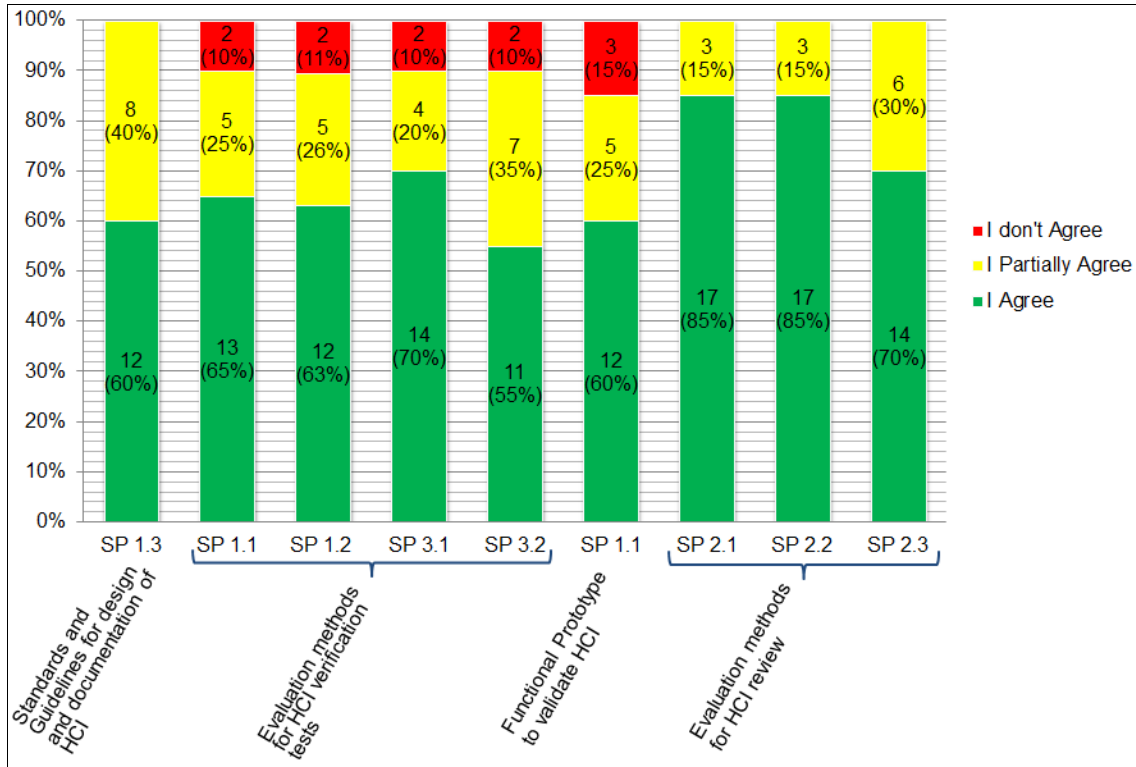


Figure 31. Quantitative results for Verification (category and practice)

Based on these justifications we recognize that the *Evaluation methods for HCI verification tests* category was really miscellaneous of 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 (Perform Verification), and SP3.2).

3.5.3.6. Analysis for Validation (VAL)

Figure 32 presents the results for VAL process area.

Four from the five propositions were accepted (no disagreements) and the last proposition had only 2 disagreements. The experts' justifications for the partial agreements and disagreements were quite similar to the observations made for VER process area and therefore implied in similar decisions. In general, the experts indicate 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 (Establish the Validation Environment).

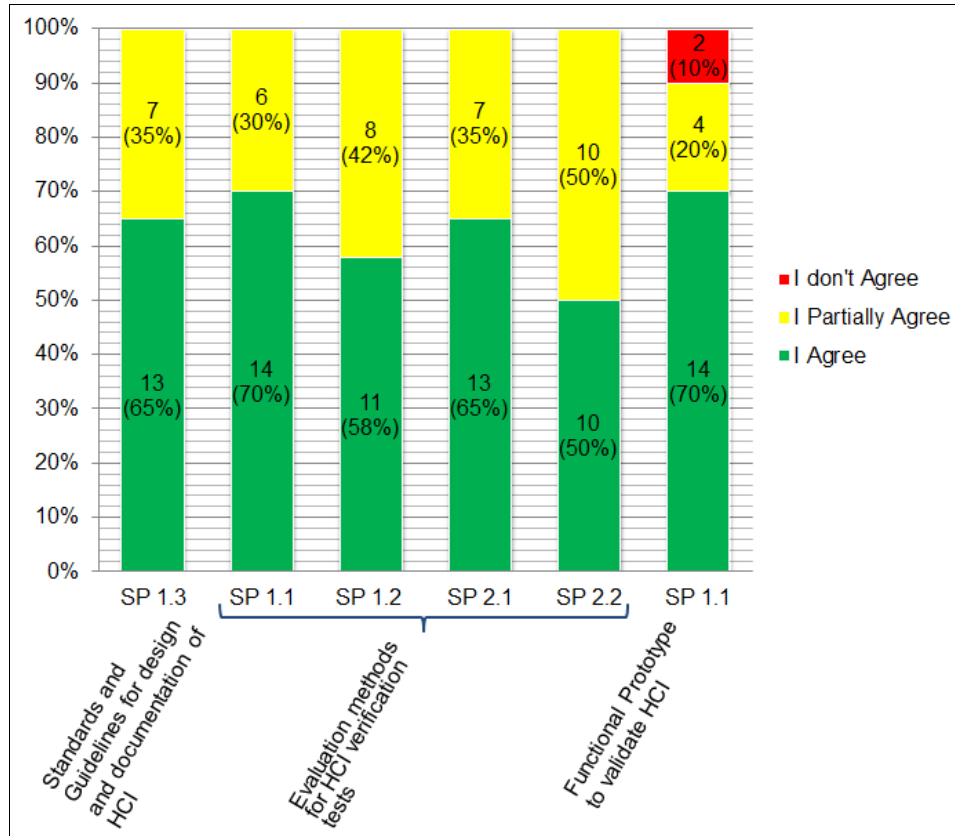


Figure 32. Quantitative results for Validation (category and practice)

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

3.5.3.7. Synthesis of Analysis

To summarize, at the beginning we had 10 HCI categories composed of 33 propositions that support 27 engineering practices of the CMMI-DEV (see Table 19). After the analysis of all interviews, we obtained 14 HCI categories (Table 25) composed of 39 propositions that support 26 engineering practices of the CMMI-DEV, as presented in Table 26. 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 with 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* by *evaluation methods for HCI validation*;
- the inclusion of new six propositions according to the new categories and the suggestions of the experts.

Table 25. HCI approaches to support CMMI-DEV after interviews with experts

HCI Category	Purpose	HCI approaches
Techniques to identify user needs	To use techniques with the goal to elicit user needs.	Brainstorming; Interviews; Surveys/Questionnaires; Card Sorting; Focus Groups; Field Studies (Courage & Baxter, 2005).
Techniques to identify user and organizational requirements	To use techniques with the goal to elicit user needs and to transform these needs in user requirements.	Persona; Scenario; User stories, User profile (detailed); Task analysis; Context-of-use analysis; Storyboards (Courage & Baxter, 2005) and (Maguire, 2001); Requirements specification templates (e.g. VOLERE ¹⁶ , IEEE ¹⁷ , RESCUE ¹⁸).
Task Modeling	To use task modeling methods with the goal to elicit user needs, to transform these needs in user requirements, to establish user interface requirements, and to analyze the user and user interface requirements.	CTT - Concur Task Tree (Paternò et al., 1997); K-MAD - Kernel of Model for Activity Description or MAD - Model for Activity Description (Limbourg & Vanderdonckt, 2004); HTA - Hierarchical Task Analysis (Limbourg & Vanderdonckt, 2004); SADT - Structured Analysis and Design Technique (Ross, 1977) or SADT coupled with Petri Nets (Abed, Bernard, & Angué, 1991); GTA - Groupware Task Analysis (Veer et al., 1996); Task Model Standard - W3C (World Wide Web Consortium, 2014); HAMSTERS notation (Martinie & Palanque, 2015).
Standards and Guidelines for HCI design	To use standards and guidelines to establish and maintain a prioritization of user quality attribute requirements, to design the user interface, and to establish verification and validation criteria.	Ergonomic criteria (Scapin & Bastien, 1997) and (Vanderdonckt, 1995); ISO/IEC 9126-1 (International Organization for Standardization/International Electrotechnical Commission, 2001); ISO 9241-11 (International Organization for Standardization, 1998); ISO/IEC 25000 (International Organization for Standardization/International Electrotechnical Commission, 2014); Accessibility standards and guidelines WAI/W3C (World Wide Web Consortium, 2015); Nielsen's Heuristics (Nielsen, 1994); Golden Rules of Interface Design (Shneiderman et al., 2009).
Prototype for HCI requirements	To use prototypes to transform user needs into user requirements and to validate user requirements.	Paper Prototyping/Sketches; Storyboards; Wireframes; Mockups; Wizard of Oz; Video prototyping (Beaudouin-Lafon & Mackay, 2009) and (Maguire, 2001).
Techniques to validate HCI requirements	To use techniques to analyze user requirements and to validate user requirements.	ProtoTask for K-MAD (Caffiau et al., 2008); Task Model Simulator for CTT (Paternò, 2004); Focus group to validate HCI requirements (Nielsen, 1997); Thinking aloud (Shneiderman et al., 2009).
Architecture patterns for HCI	To define architectural decisions and to select architecture patterns to design and implement the user interface.	Arch Model (Bass et al., 1991); Language Model (Foley & Van Dam, 1982); Seeheim Model (Pfaff, 1985); PAC (Presentation-Abstraction-Control) Model (Coutaz, 1987); PAC-AMODEUS Model (Nigay & Coutaz, 1995); MVC (Model-View-Controller) Model (Goldberg, 1984); CAMELEON-RT (Balme, Demeure, Barralon, Coutaz, & Calvary, 2004); Frameworks ¹⁹ .
Design patterns for HCI	To use design patterns to implement the design the user interface.	A Pattern Language for Human-Computer Interface Design (Tidwell, 1999); A Pattern Approach to Interaction Design (Borchers, 2001); Pattern Languages in Interaction Design: Structure and Organization (van Welie & van der Veer, 2003); Designing interfaces: Patterns for Effective Interaction Design (Tidwell, 2010).

¹⁶ <http://www.volere.co.uk/template.htm>

¹⁷ <http://www.cse.msu.edu/~cse870/IEEEExplore-SRS-template.pdf>

¹⁸ https://www.city.ac.uk/_data/assets/pdf_file/0006/79881/RESCUE_Process_Doc_v4_1.pdf

¹⁹ <http://www.iso-architecture.org/42010/afs/frameworks-table.html> and <http://www.iso-architecture.org/42010/cm/>

HCI Category	Purpose	HCI approaches
Techniques for interaction modeling	To build interactive models to help in the choice of the design solution.	MoLIC - Modeling Language for Interaction as Conversation (Barbosa & Paula, 2003); UAN - User Action Notation (Hartson et al., 1990); TAG - Task-Action Grammar (Brown, 1996).
Techniques for HCI documentation	To use techniques to produce the end user documentation.	Style guide (Shneiderman et al., 2009); Architecture for help (Silveira, Barbosa, & Souza, 2004); Training Program (Martinie, Palanque, Navarre, Winckler, & Poupard, 2011).
Iterative and Evolutionary Prototypes (system versions)	To use iterative and evolutionary prototypes to help in: product and/or product component design, integration, validation and verification.	User interface toolkits (Beaudouin-Lafon & Mackay, 2009); User interface builders (Beaudouin-Lafon & Mackay, 2009); User interface development environments (Beaudouin-Lafon & Mackay, 2009).
Evaluation methods for HCI verification	To select and use evaluation methods for products and product components verification.	Unit test; Integration test; System test; Acceptance test; Installation test (CMMI Product Team, 2010); (Shneiderman et al., 2009).
Evaluation methods for HCI review	To select and use review methods for products and product components review.	Semiotic inspection (Souza, Leitão, Prates, & Silva, 2006); Heuristic evaluation (Nielsen, 1994); Cognitive walkthrough (G Cockton et al., 2009); (Shneiderman et al., 2009); Groupware Walkthrough (Mahatody et al., 2010); Guidelines review (Shneiderman et al., 2009); Consistency inspection (Shneiderman et al., 2009); Metaphors of human thinking (MOT) (Shneiderman et al., 2009); Formal usability inspection (Shneiderman et al., 2009).
Evaluation methods for HCI validation	To select and use evaluation methods for products and product components validation.	Usability testing (Dumas & Fox, 2009); (Shneiderman et al., 2009); Communicability test (Prates, Souza, & Barbosa, 2000); Standardized usability questionnaires (Sauro & Lewis, 2012); Post-experience interviews (Sauro & Lewis, 2012); User experience evaluation (Vermeeren et al., 2010).

By analyzing the suggestions of the experts, we also improved our propositions indicating when to use the final artifact produced by one category to applying the approaches suggested for each practice (see \diamond in Table 26). For example, in the practice SP2.1 (Requirements development) we can use the final artifact produced by *techniques to identify user needs* and *techniques to identify user and organizational requirements* categories to produce a task model applying the *task modeling* category.

3.5.4. Threats of validity

To analyze the results we considered the four threats of validity proposed by Wohlin et al. (2012): construct validity, internal validity, conclusion validity, and external validity. We analyzed each one of them trying to define some mitigation as described below.

Threats to the **construct validity** illustrate from the relation between theory and observation and the questions of whether the treatment adequately reflects the cause; whether the result adequately reflects the effects. This threat is related to the build 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 advisors, 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 Ph.D. student and one of the advisors who made the interviews have a good knowledge of CMMI-DEV being already participated in officials' CMMI-DEV implementations and appraisals.

Table 26. Categories x CMMI-DEV practices after interviews with Experts

(● our proposition and ◇ when to use the final artifact produced by one category to applying the approaches suggested for each practice)

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																						●	●		●	●
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				

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 network professional of one of the advisors. After that, these experts suggested other names following the pre-defined fixed profile for the expert selection. We believe that maybe some experts could feel not comfortable to the discord of the propositions. To mitigate this bias, the advisor 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 that have experience (academic and/or industrial, as showed in Table 6 they have in average 19 years of experience) in 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 that made the interviews have academic and practical experience with CMMI-DEV, making possible to clarify doubts of the experts. However, we could not control that even with these mitigation actions, the experts did not give 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 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 agree with one or more propositions, they were asked to justify their opinion and include any other proposals they judge 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 large experience recognized by the HCI community (e.g., be program chair or member of program committee of HCI conferences, editor of journals and members of HCI associations).

Therefore we have 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 20, we identified only their current interest but this was naturally evolved/modified throughout the years with new technologies, research and new domain of application they have been working with. Therefore, even that we have looked for experts with different background we could not ensure cover all kind 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 large experience (12/20 experts have more than 20 years of experience) and having worked in several domains throughout their career, we assumed their opinion as reliable. Moreover, they were notified that all their suggestions for very specific kind of applications would be included as particular recommendations. In this way, the experts evaluated one by one of the examples, including new ones that they considered relevant and eliminating some that they considered not being used anymore. As consequence of all these considerations, we accepted the risk of potential bias in their evaluation. Therefore, it is not possible to generalize this result.

3.6. Using HCI approaches to support the development of interactive systems that follow SPCM models

This section describes how to use our proposal of integration of HCI approaches (methods, techniques, standards and patterns) into the engineering process areas of the international model CMMI-DEV (CMMI Product Team, 2010). Considering the equivalence mapping (Softex, 2016b) of this model with the CMMI-DEV (see section 1.6, Chapter 1), all suggestions described in this section can be also used for the processes of the level D of the national model MR-MPS-SW model (Softex, 2016c).

The HCI approaches (methods, techniques, standards and patterns) were integrated into the specific practices as **HCI categories** (see Table 25 that presents the HCI categories, the purpose of each category and a list of HCI approaches for each category).

We highlight that the communication between the members of Software Engineering and Human-Computer Interaction development teams is fundamental so that the proposition of integration produces a useful and usable iterative system.

In the following sections we will briefly describe the purpose of each engineering process area as well as its specific practices, and how HCI approaches should be used in these practices to support their implementation in the case of interactive system development. References of the application of the approaches are also quoted for further investigation. Finally, when possible, experts' suggestions collected during the process of validation and improvement of the HCI categories (see Chapter 3) regarding to each the specific practice are presented. To better distinguish them in the text, the experts' suggestions are presented in boxes.

3.6.1. Requirements Development (RD)

The **Requirements Development** process area is intended to define the requirements of the customer, product, and product components. According to the CMMI-DEV (CMMI Product Team, 2010) and MR-MPS-SW (Softex, 2016a) the activities related to this process area are:

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

MR-MPS-SW (Softex, 2016a) emphasizes that the specific practices of this process area are related to the specific practices of three other processes: Technical Solution (TS), Verification (VER) and Validation (VAL). The requirements produced by RD are the work product required for the beginning of the TS process area (Softex, 2016a). The VAL process area presents a direct intersection with RD, regarding to the validation of the requirements.

❖ Specific Practice 1.1 - Elicit Needs

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

The category *Techniques to identify user needs* was integrated in this practice with the objective to help in the elicitation of the needs of the user interfaces. The literature presents different techniques that can be used for this purpose, for example, **brainstorming, interviews, surveys/questionnaires, card sorting, focus groups, and field studies/observation** (Courage & Baxter, 2005). One example application of the technique **focus groups** for open source software development process can be found in (Llerena, Rodríguez, Castro, & Acuña, 2017a).

Other category that was integrated in this practice is *Techniques to identify user and organizational requirement*. Techniques as **scenario, user stories, storyboards, task analysis, persona, context-of-use analysis, user profile** (Courage & Baxter, 2005) and (Maguire, 2001), and **requirements specification templates** can be used to document and to refine the needs, expectations, constraints and user interfaces. One example application of the technique **user profile** for open source software development process can be found in (Llerena, Rodríguez, Castro, & Acuña, 2017b). This technique allows defining the different user profiles that a system can have.

The third category that was integrated in this practice is *Task Modeling*. The goal of this category is to produce **task models** that represent the needs of the different users that a system can have. Examples of methods that can be found in this category are: **CTT** - Concur Task Tree (Paternò et al., 1997); **K-MAD** - Kernel of Model for Activity Description or **MAD** - Model for Activity Description (Limbourg & Vanderdonckt, 2004); **HTA** - Hierarchical Task Analysis (Limbourg & Vanderdonckt, 2004); **SADT** - Structured Analysis and Design Technique or **SADT coupled with Petri Nets** (Ross, 1977); **GTA** - Groupware Task Analysis (Veer et al., 1996); **Task Model Standard** - W3C (World Wide Web Consortium, 2014); **HAMSTERS notation** (Martinie & Palanque, 2015). A Task Modeling application for reengineering processes for mobile learning user interfaces can be found in (Molina, Redondo, & Ortega, 2007), and a task modeling approach for safety-critical systems is presented (Giese, Mistrzyk, Pfau, Szwillus, & von Detten, 2008).

❖ Specific Practice 1.2 - Transform Stakeholder Needs into Customer Requirements

In this practice the needs, expectations, constraints and user interfaces identified in practice 1.1 are translated into user requirements. The prioritization of the requirements assists in determining the project scope, iteration or increment (Softex, 2016a). In addition, it ensures that critical requirements (both functional and non-functional) are handled quickly (CMMI Product Team, 2010). In the case of interactive systems, prioritization of critical (especially the non-functional) requirements is essential to ensure a usable and useful system.

The category *Techniques to identify user and organizational requirement* was integrated in this practice with the objective to prioritize the user requirements. In this level, the techniques of the category *Techniques to identify user and organizational requirement* previously used in practice 1.1 are revisited to evolve the records in order to define and prioritize user interface requirements. The literature presents many techniques that can be used to prioritize user interface requirements (Courage & Baxter, 2005) and (Maguire, 2001). **Persona** is an example of HCI technique that gathers information about users to understand their characteristics (Acuña, Castro, & Juristo, 2012). An application of this technique in the software development requirements phase can be found in (Acuña et al., 2012).

The second category integrated in this practice was *Task Modeling*. The objective is to evolve (initially produced in 1.1) or to product **task models** that represent the user requirements. Task model is a model-based approach to user interface design where the results are models that describe the activities that should be performed in order to reach users' goals (Paternò, 2001). Task models can be useful in different phases of the development of interactive applications: requirements analysis, design of the user interface, usability evaluation, documentation and others (Paternò, 2001). A joint application of UML diagrams and **Task Modeling** can be found in (Paternò, 2001). A tool for specifying task models (Responsive CTT) can be found in (Anzalone, Manca, Paternò, & Santoro, 2015) and this tool can be accessed through touch-based mobile devices, such as smartphones and tablets.

The category *Prototype for HCI requirements* integrated in this practice has as objective the construction of a prototype that can be used to discuss the prioritization of the requirements with the end user. Examples of prototype techniques that can be used for this purpose are: **Paper Prototyping/Sketches; Storyboards; Wireframes; Mockups; Wizard of Oz; Video prototyping** (Beaudouin-Lafon & Mackay, 2009) and (Maguire, 2001). The prototyping include different steps: (i) study of user requirements; prototype construction; and their validation by users (Softex, 2016a).

Experts suggestion:

In this level, the **prototypes** are used to support the discussion about the design with the team. They cannot use the prototype as a final system but they can build different versions of the prototype.

One expert (with 30 years of experience) suggests the use of techniques, such as **FRAM - Functional Resonance Accident Model** (Woltjer & Hollnagel, 2008) and **STAMP - System-Theoretic Accident Model and Processes** (Young & Leveson, 2014) to identify organizational context issues for critical interactive systems. Examples of the use of FRAM technique can be found in ((Ragosta, Martinie, Palanque, Navarre, & Sujan, 2015); (Carvalho, 2011) and (Bellini, Nesi, Pantaleo, & Venturi, 2016)).

The category *Standards and Guidelines for HCI design* also was integrated in this practice with the objective to establish and maintain a prioritization of user quality attribute requirements. Quality attributes such as **usability** could be considered since the identification of user needs, as well as during the elicitation and prioritization of the user requirements. Criteria, guidelines heuristics and rules are used in this moment according to the selected quality attribute requirements. We can cite as example, **ergonomic criteria** ((Scapin & Bastien, 1997) and (Vanderdonckt, 1995)); **ISO Standards** (ISO/IEC 9126-1 (International Organization for Standardization/International Electrotechnical Commission, 2001), ISO 9241-11 (International Organization for Standardization, 1998) and ISO/IEC 25000 (International Organization for Standardization/International Electrotechnical Commission, 2014)); **accessibility standards and guidelines** – WAI/W3C (World Wide Web Consortium, 2015); **Nielsen's heuristics** (Nielsen, 1994); and **Golden rules of interface design** (Shneiderman et al., 2009). In addition, these criteria, guidelines heuristics and rules are used in the evaluation phase.

❖ Specific Practice 2.1 Establish Product and Product Component Requirements

In this practice the user needs, expectations and constraints identified in 1.1 and 1.2 are translated in a set of functional and non-functional requirements of the product (user interface) and product components.

The *Task Modeling* category was integrated in this practice with the objective to produce or evolve **task models** that represent the user interface requirements. A tutorial about task model construction can be found in (Winckler & Pimenta, 2004). The use of Task Modeling and the construction of task models for context-sensitive user interfaces are discussed in (Pribeanu, 2007).

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

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

Experts suggestion:

One expert (with 30 years of experience) suggests the use of **Worth-Centered Design** (Gilbert Cockton, 2006) to design interactive systems with the finality to deliver worth in the real world and decrease the gap between the user and product. One example of the use of this approach in a mobile and context-aware application can be found in (Camara & Calvary, 2015)

❖ Specific Practice 3.1 - Establish Operational Concepts and Scenarios

In this practice the operational concepts and scenarios are developed for the product (user interface) and the product components.

The category *Techniques to identify user and organizational requirement* was included in this practice with the objective to construct scenarios that define the interaction of the user interface, the end user and the environment. In addition to this technique, the work products produced by the *Task Modeling* category are used to support the construction of the scenarios.

❖ Specific Practice 3.2 Establish a Definition of Required Functionality and Quality Attributes

In this practice a definition of quality attributes is established.

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

❖ Specific Practice 3.3 Analyze Requirements

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

The *Task Modeling* category was included in this practice with the aims to perform an analysis of user and user interface requirements, in order to remove conflicts encountered in relation to the user interface.

In addition to this category, the work products produced by the category *Techniques to identify user and organizational requirement*, as well as **scenarios** and detailed definitions of the requirements, can be used in this analysis (CMMI Product Team, 2010). The quality attributes defined above and supported by the *Standards and Guidelines for HCI design* category should also be considered in the requirements analysis.

❖ Specific Practice 3.4 Analyze Requirements to Achieve Balance

In this practice user and user interface requirements are analyzed in a way to balance stakeholder needs with design constraints.

The *Techniques to validate HCI requirements* category was integrated in this practice with the goal of analyzing user and user interface requirements in order to balance stakeholder needs with design constraints and minimize the risk of user interface development. The literature proposes models, simulators and techniques that can be used for this type of analysis. For example, **ProtoTask** for K-MAD (Caffiau et al., 2008); **Task Model Simulator** for CTT (Paternò, 2004); **Focus group** to validate HCI requirements (Nielsen, 1997); and **Thinking aloud** (Shneiderman et al., 2009).

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

❖ Specific Practice 3.5 Validate Requirements

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

The *Prototype for HCI requirements* category integrated in this practice aims to provide a first prototype or the evolution of the prototype built in 1.2, which helps in the validation of requirements with the end user. The prototypes are useful for evaluating critical or complex requirements (Softex, 2016a). We remember that the prototypes do not represent the final version of the system, but rather tools to discuss and evaluate requirements and design.

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

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

3.6.2. Technical Solution (TS)

The **Technical Solution** process area has the objective of selecting, designing, and implementing solutions to meet requirements. According to the CMMI-DEV (CMMI Product Team, 2010) the activities related to this process area are:

- Evaluating and selecting solutions that satisfy a set of allocated functional and quality attribute requirements;
- Developing detailed designs for the selected solutions;
- Implementing the designs as a product or product component.

According to MR-MPS-SW (Softex, 2016a) the practices of this process area are related to the practices of four other process areas: Requirements Development (RD), Product Integration (PI), Verification (VER) and Validation (VAL). The TS process area receives as input the requirements developed by RD to design and build the solution. The PI process area receives the requirements developed by RD and the product components designed and constructed by TS, in order to combine them and verify if the interfaces satisfy the interface requirements developed by RD (Softex, 2016a). The product components are verified by VER process area in relation to the requirements, and the final product is incrementally validated by the VAL process area (Softex, 2016a).

❖ Specific Practice 1.1 Develop Alternative Solutions and Selection Criteria

In this practice the solution alternatives and selection criteria are developed to meet the requirements of the product (user interface) and product components.

The *Architecture for HCI* category integrated in this practice aims to assist the choice of the architecture to be used in the system development (Softex, 2016a). The choice of the architecture will help in the selection of the best solution. In the case of interactive systems several architectures can be used for this purpose. For example, **Arch Model** (Bass et al., 1991); **Language Model** (Foley & Van Dam, 1982); **Seeheim Model** (Pfaff, 1985); **PAC (Presentation-Abstraction-Control) Model** (Coutaz, 1987); **PAC-AMODEUS Model** (Nigay & Coutaz, 1995); **MVC (Model-View-Controller) Model** (Goldberg, 1984); **CAMELEON-RT** (Balme et al., 2004); and **Frameworks**. We suggest also a reference literature that explains about software architecture in practice (Bass, Clements, & Kazman, 2012).

The *Techniques for interaction modeling* category was integrated in this practice with the objective to support the choice of the best development solution. Interaction modeling techniques or dialog modeling are used to produce **interactive models**, which describe the actions that users have to perform in the operation of a system (Winckler & Pimenta, 2004). Examples of these techniques are: **MoLIC** - Modeling Language for Interaction as Conversation (Barbosa & Paula, 2003); **UAN** - User Action Notation (Hartson et al., 1990); and **TAG** - Task-Action Grammar (Brown, 1996).

In addition to these categories, the work products produced by the category *Techniques to identify user and organizational requirement*, such as **scenarios**, **personas** and detailed definitions of requirements, can be used to assist the choice of the best solution. The prototypes produced by the *Prototype for HCI requirements* category must also be used with the same purpose.

Experts recommendation:

Two experts defend that at this moment the interactive components should be identified.

To that end, one expert (with 30 years of experience) suggests the use **Worth-Centered Design** (Gilbert Cockton, 2006) to design interactive systems. The finality of this approach is to deliver worth in the real world and decrease the gap between the user and product.

❖ **Specific Practice 2.1 Design the Product or Product Component**

In this practice the user interface is designed according to the requirements.

The category *Standards and Guidelines for HCI design* was included in this practice in order to help in the identification of the design criteria in relation to quality attributes. These criteria will be used to evaluate the interface design.

The category *Architecture patterns for HCI* integrated in this practice aims to aid and facilitate the design of the interface. The design of the interface depends of the type of architecture that will be developed for the product. The architecture patterns support functional requirements and quality attribute requirements, and are used to create the product architecture (CMMI Product Team, 2010). In this practice the architecture pattern is not used but chosen.

The *Iterative and Evolutionary Prototypes (system versions)* category was integrated in this practice in order to facilitate the design of the user interface. At this level a prototype can be produced as a zero version of the system. The literature presents different techniques that can be used to create this prototype. For example, **User interface toolkits** (Beaudouin-Lafon & Mackay, 2009); **User interface builders** (Beaudouin-Lafon & Mackay, 2009); and **User interface development environments** (Beaudouin-Lafon & Mackay, 2009).

In addition to these categories, the work products produced by the category *Techniques for interaction modeling*, that is, the *interactive task models* can be used to assist in the production of the new prototype (version zero of the system).

Experts recommendation:

Two experts affirm that a version zero of the final product can be used as a prototype, but the system may not be an extension of the prototype.

❖ **Specific Practice 3.1 - Implement the Design**

In this practice the design is implemented using a suitable method.

The *Architecture for HCI* category integrated in this practice aims to assist in the implementation of product design. The implementation of the design depends on the architecture pattern chosen. This will also help in choosing the design patterns needed to implement the product components.

The *Design patterns for HCI* category was integrated in this practice with the purpose of assisting the design of the user interface. The literature presents different **patterns** that can be used for this purpose, and these design patterns provide solutions to specific usability problems related to interface design and interaction (Folmer, Welie, & Bosch, 2006). For example, (A. C. da Silva, Silva, Pentead, & Silva, 2006).

& da Silva, 2004) and (A. C. da Silva, Silva, Penteado, & da Silva, 2005) use the patterns categories defined by (Alpert, 2003):

- (i) **Human-Computer Interaction Patterns** - these patterns are related to high level issues and assist in interaction design; examples of these patterns can be found in (Tidwell, 1999) and a practical application of some of these patterns can be found in (A. C. da Silva et al., 2005).
- (ii) **User Interface Patterns** - these patterns are related to specific interaction issues and are used to assist user interface design; examples of these patterns can be found in (Borchers, 2001), (Tidwell, 2010) and (van Welie & van der Veer, 2003), and a practical application of some of these patterns can be found in (A. C. da Silva et al., 2005).

The authors (da Silva et al. (2004) and da Silva et al. (2005)) discuss that these patterns can be used in different phases of the software development process (for example, in the interface evaluation), according to the development methodology adopted. Also, the patterns can be used to improve, by way of example, what is proposed by guidelines and HCI heuristics (which in our work are part of the *Standards and Guidelines for HCI design* category).

Other works that used patterns for user interface design are: (Juristo, López, Moreno, & Sánchez, 2003), (Folmer et al., 2006), (Seffah & Gaffar, 2007) and (Rodríguez, Acuña, & Juristo, 2015). In addition, in (Thanh-Diane, Vanderdonckt, & Seffah, 2016) we can find an User Interface Pattern Language Markup Language (UIPLML) that the objective is to define user interface patterns for multi-platform systems.

Experts recommendation:

One expert suggests specific design patterns for ubiquitous computing that should be considered in the development of this kind of application. For instance, one suggestion is the work presented by (Chung et al., 2004).

❖ Specific Practice 3.2 Develop Product Support Documentation

In this practice the end user documentation (such as, user manual, end user training material and online help) is developed.

The category *Techniques for HCI documentation* has as objective to define standards to be followed in the elaboration of the final user documentation. The literature presents techniques for this purpose, such as: **Style guide** (Shneiderman et al., 2009); **Architecture for help** (Silveira et al., 2004); and **Training Program** (Martinie et al., 2011).

In addition to these categories, the work products produced by the categories *Design patterns for HCI* and *Architecture patterns for HCI* can be used to assist in the preparation of the end user documentation. Several other work products, such as **persona**, **scenario**, **task model**, can also be used for this purpose.

3.6.3. Product Integration (PI)

The **Product Integration** process area aims to compose the product components, producing an integrated product consistent with its design, demonstrating that functional and non-functional requirements are satisfied for the user's environment.

A critical aspect of product integration is to ensure compatibility among the interfaces (CMMI Product Team, 2010). For this, the management of internal and external interfaces of the products and product components is necessary. These interfaces include the user interfaces.

According to MR-MPS-SW (Softex, 2016a) the practices of this process area are related to the practices of two other process areas: Technical Solution (TS) and Verification (VER). The practices that represent the intersection of this process area with TS process area are not discussed in this document since any HCI approach was related to them. The intersection of this process area with the VER process area is present in the practice regarding to the verification of the interfaces, integration environment, product components and the integrated product (Softex, 2016a). In addition, while performing unit, integration and regression tests, and peer reviews.

❖ **Specific Practice 1.1 Establish an Integration Strategy**

In this practice an integration strategy of the product components is established and maintained. The components to be integrated are determined and also the integration sequence. The chosen integration strategy must be consistent with the design, architecture, and product requirements (Softex, 2016a).

The *Iterative and Evolutionary Prototypes (system versions)* category was integrated in this practice with the objective of helping the integration of the product components. Especially in the verification and validation of the user interface relate to other interfaces. At this level, the prototype produced in TS can be evolved so that it presents the near real version of the system and the necessary conditions for the actual installation environment. The literature presents different techniques that can be used to create this prototype. For example, **User interface toolkits** (Beaudouin-Lafon & Mackay, 2009); **User interface builders** (Beaudouin-Lafon & Mackay, 2009); and **User interface development environments** (Beaudouin-Lafon & Mackay, 2009).

3.6.4. Verification (VER)

The **Verification** process area is intended to confirm that the selected work products properly meet the specified requirements. According to the CMMI-DEV (CMMI Product Team, 2010) the activities related to this process area are:

- The identification of work products to be verified, methods to be used to perform the verification, and the requirements to be satisfied by each selected work product;
- The determination of the environment to be used to carry out the verification;
- The development of verification procedures and criteria that are aligned with selected work products, requirements, methods, and characteristics of the verification environment;
- The performance of the verification according to available methods, procedures, and criteria.

We remember that the Verification and Validation process areas are similar, but they address different issues (CMMI Product Team, 2010). Validation demonstrates that that “you built the right thing”, whereas verification ensures that “you built it right” (CMMI Product Team, 2010).

According to the MR-MPS-SW (Softex, 2016a) the practices of this process area are related to the practices of three other process areas: Requirements Development (RD), Technical Solution (TS) and Product Integration (PI). The intersection of this process area with RD is presented in the practice RD SP 3.3 related to the analysis of the requirements developed, in order to guarantee that these are

necessary and sufficient (Softex, 2016a). At this time a verification of the requirements must be carried out. The relationship of this process area with TS is present in the practice that deals with the implementation and verification of the product components (Softex, 2016a). Product components can be verified by peer review and/or testing. The intersection with PI is presented in all integration steps. The integration of the components depends of the integration tests (verification techniques).

❖ **Specific Practice 1.1 Select Work Products for Verification**

In this practice the work products to be verified and the methods that will be used for verification are selected. For interactive systems it is important to select the **task models** that represent the user interface requirements, as well as the **prototypes** that present the implementation of these requirements.

The category *Evaluation methods for HCI Verification* integrated in this practice aims to assist in the selection of the verification methods necessary to verify the work products. In particular, this category presents verification testing methods. The methods must be selected according to the selected work products for verification. The literature offers several methods for this purpose. For example, **Unit test; Integration test; System test; Acceptance test; Installation test** (CMMI Product Team, 2010) and (Shneiderman et al., 2009).

The *Iterative and Evolutionary Prototypes (system versions)* category was integrated in this practice with the purpose of using the prototype to verify the suitability of the system design. As previously mentioned, the prototype is a work product to be checked.

❖ **Specific Practice 1.2 Establish the Verification Environment**

In this practice the environment required for the verification is established and maintained.

The category *Evaluation methods for HCI Verification* was integrated in this practice in order to help to establish the verification environment. The type of environment depends of the work products selected (such as, task models and prototype) and the verification methods used (CMMI Product Team, 2010).

❖ **Specific Practice 1.3 Establish Verification Procedures and Criteria**

In this practice the criteria and verification procedures for the selected work products are established and maintained.

The category *Standards and Guidelines for HCI design* was included in this practice in order to assist in the selection of the verification criteria. These criteria are defined to ensure that work products meet product requirements (CMMI Product Team, 2010). The criteria are selected according to the verification method that was chosen in 1.1.

❖ **Specific Practice 2.1 Prepare for Peer Reviews**

In this practice the preparation activities for the peer review of the selected products are carried out. For interactive systems it is important to select for peer review the **task models** that represent the requirements of the user interface.

The *Evaluation methods for HCI review* category was included in this practice in order to select the peer review method that will be used. In addition, it will assist in the identification of the team and of the reviewers who will participate in the review, in the preparation of the documents (checklists and

criteria) and in the preparation of the schedule (CMMI Product Team, 2010). The literature offers several peer review methods, such as: **Semiotic inspection** (Souza et al., 2006); **Heuristic evaluation** (Nielsen, 1994); **Cognitive walkthrough** (G Cockton et al., 2009); (Shneiderman et al., 2009); **Groupware Walkthrough** (Mahatody et al., 2010); **Guidelines review** (Shneiderman et al., 2009); **Consistency inspection** (Shneiderman et al., 2009); **Metaphors of human thinking** (MOT) (Shneiderman et al., 2009); **Formal usability inspection** (Shneiderman et al., 2009).

❖ **Specific Practice 2.2 Conduct Peer Reviews**

In this practice the peer review of the selected products is conducted with the objective of identifying the critical issues. That is, find and remove defects in advance (CMMI Product Team, 2010). Peer review is performed for the selected work products of the specification (such as, **scenarios** and **task models**), design (**prototypes**), test and implementation (**user interface**) activities (CMMI Product Team, 2010).

The *Evaluation methods for HCI review* category was included in this practice with the objective of assisting in the conduction of the peer review of the selected work products. The critical issues identified are communicated to the responsible for the work product who will make the corrections (CMMI Product Team, 2010).

❖ **Specific Practice 2.3 Analyze Peer Review Data**

In this practice the results obtained in each activity of the peer review are analyzed.

The category *Evaluation methods for HCI review* was included in this practice with the objective of assisting the analysis of the results found in the peer review. Data related to peer review activities should be recorded (CMMI Product Team, 2010). Among these data we can cite the type of peer review method that was used, the work product that was revised (such as, **task model**).

❖ **Specific Practice 3.1 Perform Verification**

In this practice the verification is performed for the selected work products. For interactive systems, we highlight the importance for the verification of the **task models** that represent the requirements of the user interface, as well as the **prototypes** that present the implementation of these requirements and the **user interface**.

The *Evaluation methods for integrated HCI Verification* category integrated in this practice aims to assist the verification of the selected work products. Verification of work products is done incrementally, so that problems can be identified early and corrected (CMMI Product Team, 2010). The method used in the verification should be documented as well as the deviations found during the verification (CMMI Product Team, 2010).

❖ **Specific Practice 3.2 Analyze Verification Results**

In this practice the results obtained in each verification activity are analyzed.

The *Evaluation methods for HCI Verification* category was included in this practice with the purpose of assisting the analysis of the results found in the verification. Data related to the verification activities should be recorded and compared to the verification criteria defined in 1.3 (CMMI Product Team, 2010). In this case, the result (the verification criteria) of the *Standards and Guidelines for HCI design* category is used in this practice.

3.6.5. Validation (VAL)

The **Validation** process area is intended to confirm that the product or product component will meet its intended use when placed in the target environment (user environment). According to the CMMI-DEV (CMMI Product Team, 2010) the activities related to this process area are:

- The identification of the product or product component to be validated and methods to be used to perform the validation;
- The determination of the environment to be used to carry out the validation;
- The development of validation procedures and criteria that are aligned with the characteristics of selected products, customer constraints on validation, methods, and the validation environment.
- The performance of the validation according to methods, procedures, and criteria.

According to the MR-MPS-SW (Softex, 2016a) the practices of this process area are related to the practices of three other process areas: Requirements Development (RD) and Product Integration (PI).

The intersection of this process area with RD is presented in practice RD SP 3.3 related to the validation of the developed requirements, to ensure the adequate performance of the product (Softex, 2016a).

The relationship of this process area with PI is presented in the evaluation of the integrated product components and in the evaluation of the final product delivered to the customer (Softex, 2016a).

Following we will present each specific practice of VAL. We also give some recommendations of how to implement these practices in the case of interactive system development.

❖ Specific Practice 1.1 Select Products for Validation

In this practice the products and product components to be validated and the methods that will be used for validation are selected. For interactive systems it is important to select validation for product components such as **user requirements, scenarios, task models, user interfaces** and **user manuals**.

The *Evaluation methods for integrated HCI validation* category integrated in this practice aims to assist in the selection of validation methods to validate products and product components. The methods should be selected according to the products and product components selected for validation. The literature offers several methods for this purpose. For example, **Usability testing** (Dumas & Fox, 2009); (Shneiderman et al., 2009); **Communicability test** (Prates et al., 2000); **Standardized usability questionnaires** (Sauro & Lewis, 2012); **Post-experience interviews** (Sauro & Lewis, 2012); **User experience evaluation** (Vermeeren et al., 2010).

The *Iterative and Evolutionary Prototypes (system versions)* category was integrated in this practice with the goal of to use the prototype (version zero of the system) to validate the product components. The prototype (version zero of the system) also needs to be validated by experts before making another validation object.

❖ Specific Practice 1.2 Establish the Validation Environment

In this practice the environment required for the validation is established and maintained.

The *Evaluation methods for HCI validation category* was integrated in this practice in order to help to establish the validation environment. The requirements for the validation environment are conducted by the product or product components (e.g. version zero of the system) selected and by the methods of validation (CMMI Product Team, 2010).

❖ **Specific Practice 1.3 Establish Validation Procedures and Criteria**

In this practice the validation criteria and procedures for the selected products and product components are established and maintained.

The category *Standards and Guidelines for HCI design* was included in this practice in order to assist in the selection of the validation criteria. These criteria are defined to ensure that the product will meet its intended use when placed in the target environment (CMMI Product Team, 2010). The criteria are selected according to the validation method that was chosen in 1.1.

❖ **Specific Practice 2.1 Perform Validation**

In this practice the validation is performed for the product and the product components. For interactive systems, we highlight the importance of considering the **task models** and **scenarios**, defined in the Requirement Development, during the validation procedures. Moreover, the validation of the task models by simulation is also recommended. Some tools to support this validation are: **ProtoTask** for K-MAD (Caffiau et al., 2008) and **Task Model Simulator** for CTT (Paternò, 2004)

The *Evaluation methods for integrated HCI validation category* integrated in this practice aims to assist in the effective validation of the selected product and product components. Validation is done incrementally, so that problems can be identified and corrected (CMMI Product Team, 2010). The method used in the validation should be documented, as well as the deviations found in the validation (CMMI Product Team, 2010).

❖ **Specific Practice 2.2 Analyze Validation Results**

In this practice the results obtained in each validation activity are analyzed.

The *Evaluation methods for HCI validation category* was included in this practice with the objective of assisting the analysis of the results found in the validation. Data related to validation activities should be recorded and compared with the validation criteria defined in 1.3 (CMMI Product Team, 2010). In this case the result (the validation criteria) of the *Standards and Guidelines category for HCI design* category is used in this practice.

3.7. Synthesis and Conclusion

In this third chapter we presented the results of the first three phases our research methodology. A study of CMMI-DEV and MR-MPS-SW models was performed to identify which HCI approaches (methods, techniques, standards and patterns) 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) composed of forty (40) specific practices related to engineering category from these models. As an initial result, we had defined thirty-three (33) propositions of integration of HCI approaches via ten (10) HCI categories for twenty-seven (27) specific practices. By evaluating this initial proposal with experts we got a final set of fourteen (14)

HCI categories associated with twenty-six (26) specific practices, resulting in thirty-nine (39) propositions. The categories suggest seventy-seven (77) HCI approaches as examples for use in practice. Finally, a discussion of how to use these HCI approaches to support the development of interactive systems following SPCM models was presented.

In the next chapter we will present the fourth phase that is a long-term validation in academic environment.

Chapter 4 – Long-term validation in academic environment

4.1. Introduction

This chapter will present the research and results related to the fourth phase (long-term validation in academic environment) of our research methodology (see item (vi) in Figure 33) presented in General Introduction.

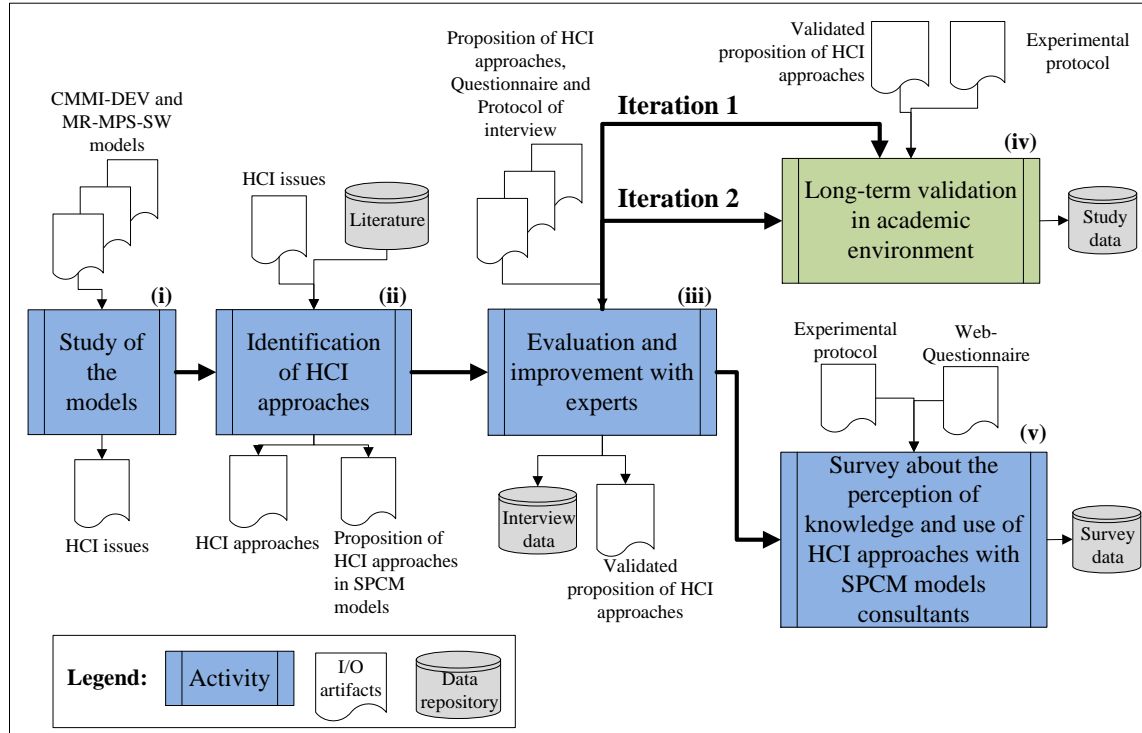


Figure 33. Research Methodology

We argue that to evaluate the effectiveness of our proposal we should probably to conduct some studies where the participants use the HCI approaches (presented in Chapter 3, Table 25) and not use the HCI approaches, and then we should compare the results. To start this validation, we decided to take advantage that HCI issues have been taught in Computer Science master of University of Valenciennes. In this master program there is a specific HCI course where the professor has asked as final project for means of evaluation, a requirement specification of a typical interactive system. To that end the teacher does not give any specific orientation to produce the requirement specification, i.e., the students are free to use whatever they had learned until the moment of the study. As consequence, we had available all reports of the students since 2010. Looking to the HCI categories that may be applied in the phase of requirements specification, we identified **Task Modeling** as the one more pertinent to start this validation; since task modeling is taught in the courses, it is claimed as essential for interactive system design, and usually is presented in the requirement specification.

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

To that end, we performed therefore one study (started in 2015) with two iterations regarding the fourth phase of our methodology (long-term validation in academic environment - see item (vi) in Figure 33).

In the first iteration we performed a descriptive analysis of all reports produced by the students during five years (2010 to 2014). For this iteration we cannot change any condition, because the data are regarding to the past (2010 to 2014). Considering that the students were free to use whatever they want, we considered this iteration as a first analysis of the study without using our approach, which presents what should be used in each moment.

In the second iteration (2015 to 2016), we decide to change one condition of this study to confirm our assumption. We presented to the students 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. For both iterations, the students had the same subject (project) to produce the requirements specification.

The results of first and second iterations are presented in section 4.3 and 4.4 respectively. In section 4.2 we present the details of the objective, the project given to the students and general steps followed in the course to produce the requirement specification. The section 4.5 presents a discussion and comparison about the two iterations and in section 4.6 we present the threats of validity of the study. Finally, we will finish the chapter with a synthesis and conclusion section (section 4.7).

4.2. Study Context

In this section we will present the objective of the study, the master program and the object used in this study, i.e., the project submitted to the students.

4.2.1. The objective and the questions of the study

The objective of this study is to investigate whether suggesting HCI approaches specific for one activity of the interactive system development (in this case, suggesting task modeling approaches for requirement specifications) can increase their use in practice. As previously described in the introduction, to that end we took advantage that in a HCI course the same project has been submitted to the students for the specification of an interactive system.

To address this goal and considering the project submitted to the students, we stated the following main research question:

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

Moreover, we defined the following secondary questions:

- **Question 2** - Which are the methods used for task modeling?
- **Question 3** - How detailed was the task modeling?
- **Question 4** - Does the task modeling consider all profiles²⁰ defined in the problem?

²⁰ We expected that each profile to be considered in different task models.

- **Question 5** - Are the user profiles described in task modeling also described in use case diagrams?

Based on the Evaluation questionnaire (Annex C) answered by students at the end of the project, we defined one more question:

- **Question 6** - What is the student's feedback considering the project subject, evaluation and pedagogical issues?

To answer questions 1, 2, 3, 4 and 5, the specification reports were analyzed. For question 1, we used a 7-point Likert scale adapted from the proposal of (Kolski, Loslever, & Sagar, 2012) in a case study about HCI specification (Figure 34). We defined each scale point²¹ for the context of this study (text presented in quotes in Figure 34) when analyzing the reports.

(-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".

Figure 34. Likert scale

For question 2, we looked for any kind of task specification (informal specification not using a method, or specification using a specific method such as those taught in class - CTT, HTA, MAD, and SADT & Petri Nets). In question 3, we classified the specification as presenting very generic modeling (named global modeling) or detailed modeling. For question 4, we checked if the task modeling considered all profiles (supervisor, rounds man, production engineer, maintenance technician, expert, overseer, and fire department employee) of the study project (see section 4.2.2). This question was completed by question 5 where we verify if each profile that was considered in the task model is also considered in the use case diagrams normally used for requirement specification.

To answer question 6, we collected the data from the responses to the Evaluation questionnaire (Annex C). This questionnaire is composed of several closed questions that we quantified the number of answers for each item being.

4.2.2. The HCI course in the Master Program

As previously presented, we considered the reports of students of a HCI course that is part of the Computer Science master at the University of Valenciennes. The subjects were students in the second year of the master. In the first iteration (Section 4.3), the students belong to five promotions: 2010/2011 (called here 2010) to 2014/2015 (called here 2014). For the second iteration (section 4.4) the students belong to two promotions: 2015/2016 (called here 2015) to 2016/2017 (called here 2016).

Table 27 shows the HCI course in the Master program to indicate when we performed the study. The number of the semester is considered from the student's admission to the university. That way a student in the seventh semester, for example, is doing the first year of the Master's degree as defined by the European System.

²¹ Our main objective with this scale was to evaluate the quality of the task models.

The study was conducted after 21 hours of classes (in the first year - 9 hours of the HCI course and 3 hours of supervised exercise, including task modeling using CTT, HTA, MAD, and SADT, and in the second year - 9 hours of the Adaptive Interactive Systems course). We can see in Table 27 that the students have a Software Engineering course (basic methods and models) in third year of Licence in engineering, and a UML (Unified Modeling Language) course in the first year of the Master's degree. The task modeling exercises and final project are the same in both iterations. The HCI course is composed of several lectures (divided into six main parts) and exercises classes covering different subjects, as presented in Figure 35.

Table 27. Information on Master's degree program

	CS (IF and FA) program	#Credits*
Third year of license in engineering	Software Engineering course (methods and models) – sixth semester	2
First year of Master	UML course – seventh semester	2
	HCI course including supervised exercises ** – eighth semester	2
Second year of Master	Adaptive Interactive System Design and Evaluation course including project – ninth semester	2

* European Credits Transfer Scale (ECTS) equivalent

** The exercises about task modeling in both programs are the same.

The lecture classes (in first year) are performed with goal of presenting the students with the particularities of interactive system development (specification, design and evaluation). Many illustrations are presented to explain the different concepts and techniques used in interactive system development. In addition, several references to web sites, books and guidelines are provided to support further investigation (resources are also available on an internal system). Concerning task modeling, the most important principles of CTT, MAD, HTA and DIANE are presented and specific references are provided. Moreover, the students resolve exercises in class using particularly CTT as an example.

<p>➤ Part 1. Definitions, basic concepts, examples</p> <p><i>HCI in general. Definition of interactive system with many examples from different domains. Explanation about what are task and activity. Examples of advanced interaction devices...</i></p> <p>➤ Part 2: Criteria for design and evaluation</p> <p><i>Heuristics of Shneiderman, Heuristics and maxims of Nielsen, ergonomic criteria (Scapin & Bastien, 1997) and (Vanderdonckt, 1995), International standards (ISO 13407, ISO 16982), definition of User eXperience...</i></p> <p>➤ Part 3. Analysis and system modeling (for normal and abnormal situations)</p> <p><i>Recalls concerning Cartesian approach (SADT), fluency graphs, object-oriented analysis (UML), and systemic approach (MERISE), Petri Nets, FMEA (Failure Mode Analysis, Effects and Criticality Analysis), fault trees...</i></p> <p>➤ Part 4: Software engineering (SE) and HCI: modeling elements</p> <p><i>SE enriched in terms of HCI (UCD), presentation of several task-modeling formalisms (CTT, MAD, HTA, and DIANE) (with exercises), Mockup and Prototyping are presented as essential in HCI, Interactive system architecture models, presentation modes. Practical exercises with CTT formalism.</i></p> <p>➤ Part 5: HCI evaluation: Representative methods</p> <p><i>Global typology, design tests, usability labs, Wizard of oz experiment, heuristic evaluation, evaluation grids, eye-tracking, cognitive walkthrough...</i></p> <p>➤ Part 6: adaptive UI</p> <p><i>Basic principles, examples with intelligent, plastic and personalized UI (for the CS program, this last part can be considered as an introduction of the course called "Adaptive Interactive System design and evaluation" (9 hours of lectures focused on different types of so-called adaptive or intelligent user interfaces – The HCI course was considered as a pre-requisite course for this one).</i></p>
--

Figure 35. Content of HCI course (adapted from (Gonçalves, Oliveira, & Kolski, 2017c))

The same experienced teacher (more than 20 years as full professor in the university) conducted the classes and the project during the seven years. As a learning method to support the project proposition the teacher used Project-Based Learning (PjBL) approach. PjBL has a constructivist orientation and focuses on student interest (Markham, 2012). The author (Markham, 2012) defines PjBL as “an extended learning process that uses inquiry and challenge to stimulate the growth and mastery of skills”. Warin et al. (2016) summarize some of the main features of PjBL as: (i) it is often multi-disciplinary; (ii) it takes weeks or months long; (iii) it includes the creation of product; (iv) it often involves real-world; and (v) fully authentic tasks and settings. Moreover, it usually takes the form of student projects that are realistic cases. In realistic and complex industrial cases, the students will implement what they have learned.

Using the PjBL approach, the teacher plays the role of a very skilled worker, since he knows the application domain in-depth: he was first a rounds man in the company; he was then promoted to be a supervisor, and finally promoted to become an overseer. He answers questions asked by each group (without information transmission between the groups for reasons of confidentiality). The students have no or little knowledge about the application domain.

Figure 36 illustrates the general steps followed in the course to produce the requirement specification. First of all, during three sessions lasting two hours each, in presence of the teacher, the students play the role of design teams, organized in groups of 3-4 members in a competitive context; each group represents a fictive company.

The groups are invited to specify the system for the control room and the HCI intended for the supervision of five inter-connected product mixing stations (presented in the previous section). The students compete to win since the students from the winning group obtain the best marks. They should produce a specification report (20 to 30 pages) about the design of an interactive system using a description of a real problem (see next section) where interactive actions are explicitly defined.

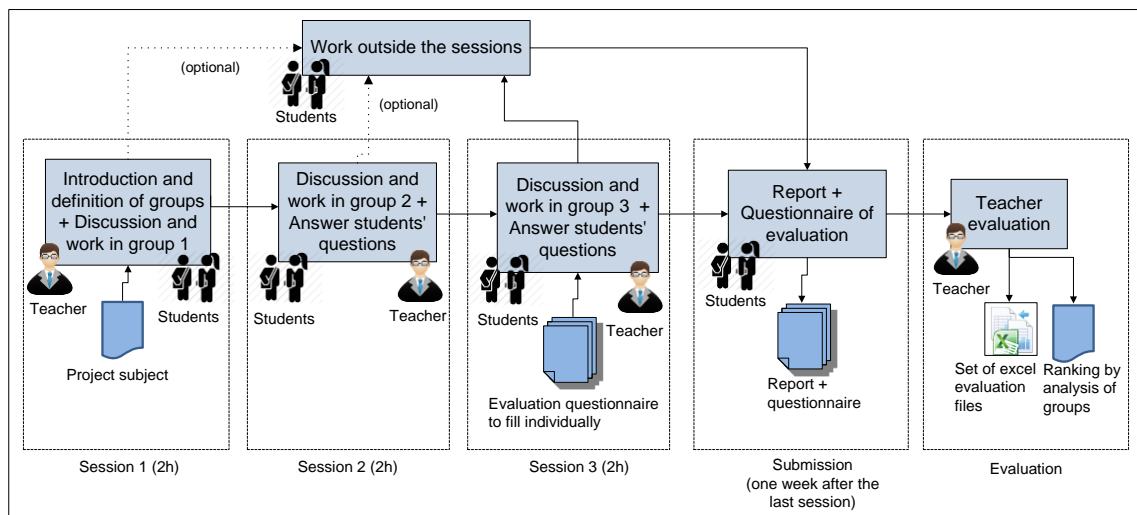


Figure 36. General steps in the development and evaluation of the requirement specification

After the first and second sessions the students can (optionally) work outside the sessions. However, after the third session the students must work outside the session to finalize the project. One week after the last session, the students must submit the report (a deadline must be respected, as for a real call for tenders), along with the answers to an evaluation questionnaire about the course (see Annex

C). Finally, in the last step the teacher evaluates the students' reports and communicates the final grade and a ranking of the groups.

4.2.3. The project of the study

The students receive a detailed description of a real problem involving an industrial process with five interconnected mixing stations (see Figure 37) that need a supervisory HCI. The real problem considers seven different profiles of potential user: supervisors, overseers, rounds men, production engineers, experts, maintenance technicians and fire department employees.

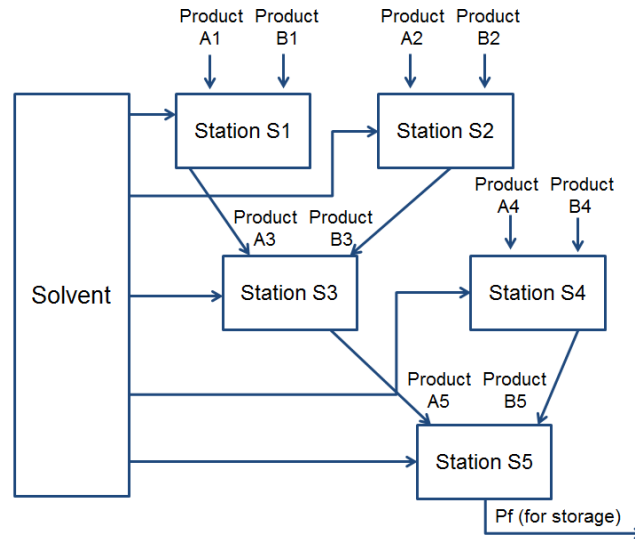


Figure 37. Five interconnected mixing stations (published in (Gonçalves et al., 2017c) and adapted from (Kolski, Sagar, & Loslever, 2004)

Orders cannot originate in the control room, only the supervisor is authorized to decide what actions can be executed on the process. Each action (e.g., start cycle, add solvent) is carried out by the rounds men following the instructions of the supervisors. A rounds man needs about 5 to 10 min to move from one station to another. The mixing process at each of the five stations (see Figure 37) called S1, S2, S3, S4 and S5, lasts 10, 10, 20, 20 and 50 min, respectively. The total process, starting with the basic products (A1, B1, A2, B2, A4 and B4), is composed of the mixtures from these five mixing stations and the result is a finished product (Pf). Several constraints to ensure the security and functionality of the system should be taken into account (e.g. run 24/24, need to exchange specific information while in dangerous situation, etc.). To summarize, different problems of safety, production, and quality have to be taken into account by the different types of user, which shows the different profiles of human-interaction with the system.

The system must be displayed on a single or several screens in a control room occupied by two human operators, called **supervisors**, working in rotating shifts (3 x 8h). The description of all profiles is presented in Table 28.

Table 28. Profiles and tasks

Profiles	Description
Supervisor	The supervisor remains in contact through support equipment and interactive devices with rounds man.
Rounds man	The rounds man is novice or more or less experienced equipped with mopeds.
Production Engineer	Production Engineer needs to perform statistical analyses before, during, or after certain productions, to make the best decisions possible. They are in charge of several processes (for them, this set of stations corresponds to one among other processes).
Maintenance Technician	Maintenance Technician is involved with curative or predictive maintenance objectives (including emergency repairs). Everyone must be informed of their presence in the field and operations concerned.
Expert	Several experts with over 25 years of experience in many types of processes are constantly reachable (in the country or abroad) to deal with possible questions day and night (if they are on-call) related to an ongoing problem, the progress of which they must be able to observe, irrespective of where they are. They are more or less expert in mechanics, electricity, chemistry, etc.
Overseer	The Overseer is responsible for the supervisors and rounds men and receives comprehensive instructions (objectives, changes ...) from production engineers. He/She needs to have an overall view of all activities and refers to his/her hierarchy.
Fire department employee	In the event of a problem, the fire department employee must be able to intervene effectively.

4.3. First iteration: Descriptive analysis

In this first iteration of the study (Gonçalves et al., 2017c), the students do not receive any recommendation to produce the requirement specification. That means they should provide their specification as they wish, following only a general recommendation of using the subjects presented in the lectures: the teacher explains that they are free to use what they learned in the current course or in other courses (particularly those presented in Table 27 - Software Engineering, UML and Human-Computer Interaction courses).

Before 2012, the Master's degree program in Computer Science (CS) was only composed of full-time students (CS-IF). From 2012, the program was divided into two groups: full-time (CS-IF) and block-release apprenticeship (CS-FA), where the students work part-time on the Master's degree and part-time in a company working in industrial projects.

We analyzed 43 reports: (i) 30 from CS-IF; (ii) and 13 from CS-FA. These reports were produced by 150 students: 106 students from CS-IF (2010-2014) and 44 students from CS-FA (2012-2014) for which we had all the answers of the feedback questionnaire. For the analysis we only considered complete projects for which we had the report and the answers for the students' evaluation questionnaire (Annex C).

In the next sections we will present the results and a descriptive analysis for each question.

4.3.1. Students' profile

In this section we present the results related to the students' profile information and then we present the result of each research question previously defined. For the analysis of the students' answer to the questionnaire, when the students left a blank answer we consider, for reasons of presentation, the

value “No Answer” as a scale point (this happens for 10 questionnaires – 9 with only one blank answer and 1 with more than one).

Concerning the gender of the respondents, we had 14 females and 136 males. Figure 38 presents the general results about the students’ evaluation.

We note that the majority of students consider themselves to be good workers (47% - work investment) and methodical (72% - working method). They prefer to work in pairs (51% - work preference) rather than individually (20% - work preference) and in a team (29% - work preference). Moreover, approximately 42% (freedom of action) prefer to be guided in part of the work (i.e., not all the time or have only general lines of the work), in the beginning (approximately 42% - freedom of action) than be guided from the start and throughout the work (15% - freedom of action).

To analyze the profile of the group of students that work on the same project, we calculated the mode for each item as presented in Table 29 for the 43 reports. In general, the working groups are good workers (35% - 15/43), methodical (76% - 33/43), that prefer to have the goal and the main lines of resolution, then let you do (39% - 17/43), but that would prefer to work in pairs (49% - 21/43).

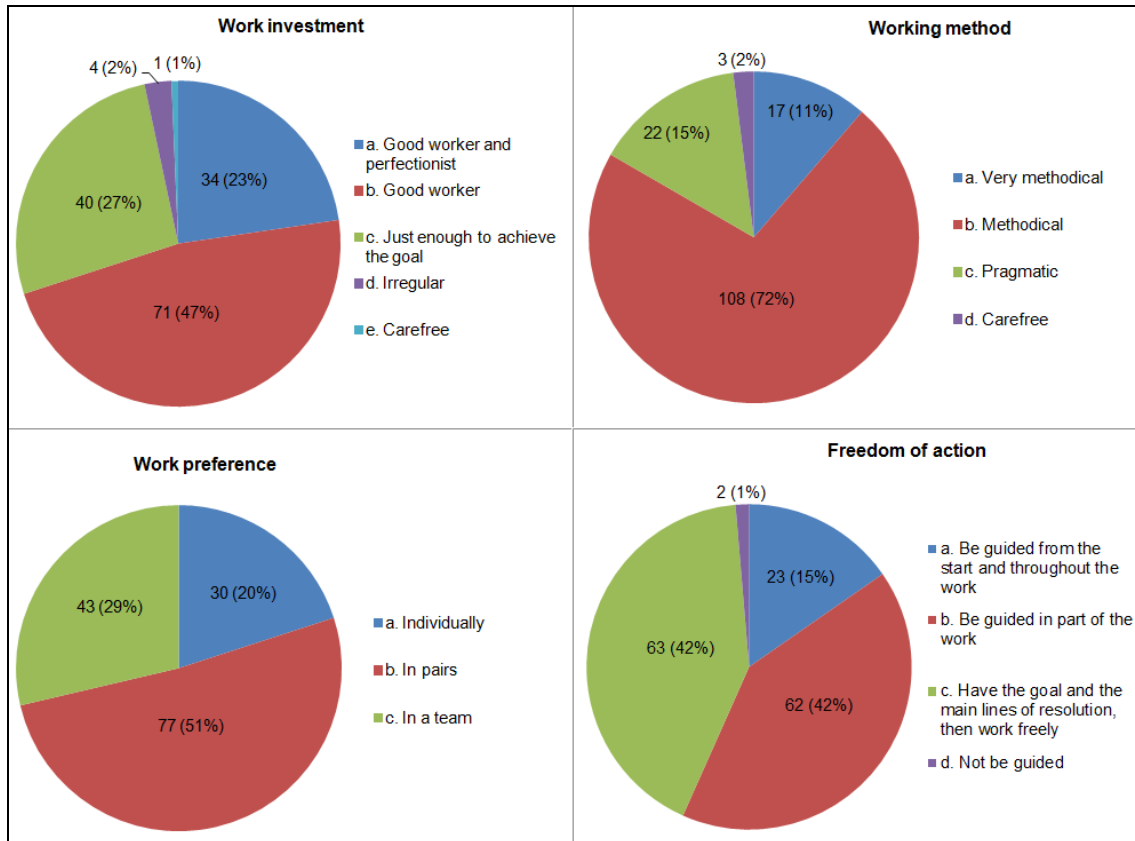


Figure 38. General students’ profile

Table 29. Mode of each item of the group profile

Options in the scale	Work investment	Working method	Work preference	Freedom of action
a	Good worker and perfectionist (5)	Very methodical (0)	Individually (5)	Be guided from the start and throughout the work (3)
b	<i>Good worker (15)</i>	<i>Methodical (33)</i>	<i>In pairs (21)</i>	Be guided in part of the work (14)
c	Just enough to achieve the goal (9)	Pragmatic (0)	In a team (8)	<i>Have the goal and the main lines of resolution, then let you do (17)</i>
d	Irregular (1)	Carefree (0)	-	Not be guided (0)
e	Carefree (0)	-	-	-
a & b	Good worker and perfectionist & Good worker (1)	Very methodical & Methodical (3)	Individually & In pairs (2)	Be guided from the start and throughout the work & Be guided in part of the work (0)
b & c	Good worker & Just enough to achieve the goal (5)	Methodical & Pragmatic (4)	In pairs & In a team (1)	Be guided in part of the work & Have the goal and the main lines of resolution, then let you do (4)
No mode	7	3	6	5
Total of reports	43	43	43	43

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

To answer this question, we considered any specification that describes the final user tasks as a result of task modeling.

Figure 39 presents the results of our findings for the 43 reports. We note that 53% (16/30) from CS-IF and 46% (6/13) from CS-FA **did not** present any result of task modeling.

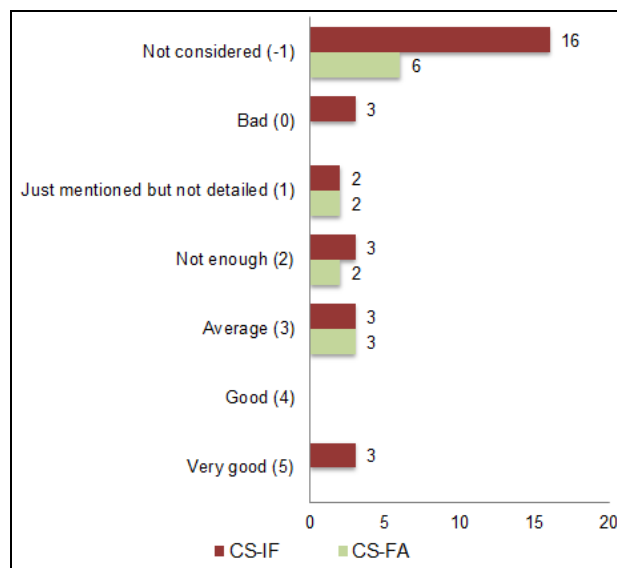


Figure 39. Results of Task modeling

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 modeling tasks of end users in the specification phase, which is the phase where the task modeling has more emphasis on the system development (Courage, Redish, & Wixon, 2009) and (Santoro, 2005).

We found that 49% of the reports (14/30 from CS-IF and 7/13 from CS-FA) presented task modeling and only a small part of the reports (9/43 – 21%) that considered task modeling was really relevant for this study. That means reports where task models were evaluated from 3 to 5 in the scale: 6 (approximately 20% of total) of 30 reports from CS-IF, and 3 (approximately 23% of total) of 13 reports from CS-FA. These results show that the task modeling activity was not performed well by students in the system specification.

4.3.3. Question 2: Which are the methods used for task modeling?

The results for this question, presented in Figure 40, show that for the 43 reports, only 21 reports (49%) presented some task modeling using different approaches: from informal specifications (not using a method - 9% (4/43)) to the use of different formalisms for task modeling (approximately 40% - 17/43). We expected to find the use of the taught modeling formalisms: CTT, HTA, MAD, and SADT and Petri Nets. However, after analyzing the first question we identified that the modeling formalisms used were not only those taught in class. For the informal specification, we found “simple list of tasks”.

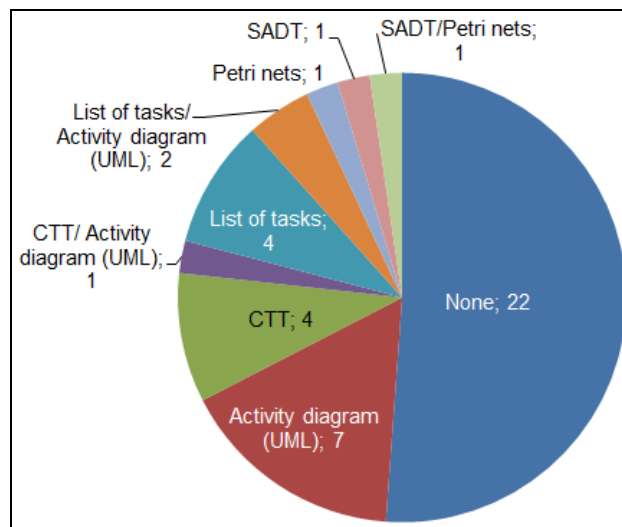


Figure 40. Overall result of task modeling methods

About the specifications that used the formalisms, we found: Activity diagram (from UML), CTT, SADT, and Petri Nets. Some reports present the use of two combined models, as follows: CTT/Activity diagram, list of tasks/activity diagram, and SADT/Petri Nets. We found that only 8/21 (4 reports with CTT, 1 report with CTT and activity diagram, 1 report with SADT, 1 report with Petri Nets and 1 report with SADT/Petri Nets) used the methods taught in class. That means that the formalisms taught in class were **not systematically** used by the students.

Looking in more detail, Figure 41 shows the methods used for CS-FA and CS-IF. We were surprised by the fact that there are students who do not use any formalism (22 out of 43 reports) for task

modeling. We expected that they would consider the importance of task modeling to specify interactive system and what was taught in class, because they are the future people that will work in the industry. This is especially the case for CS-FA students which already work part-time on industrial projects.

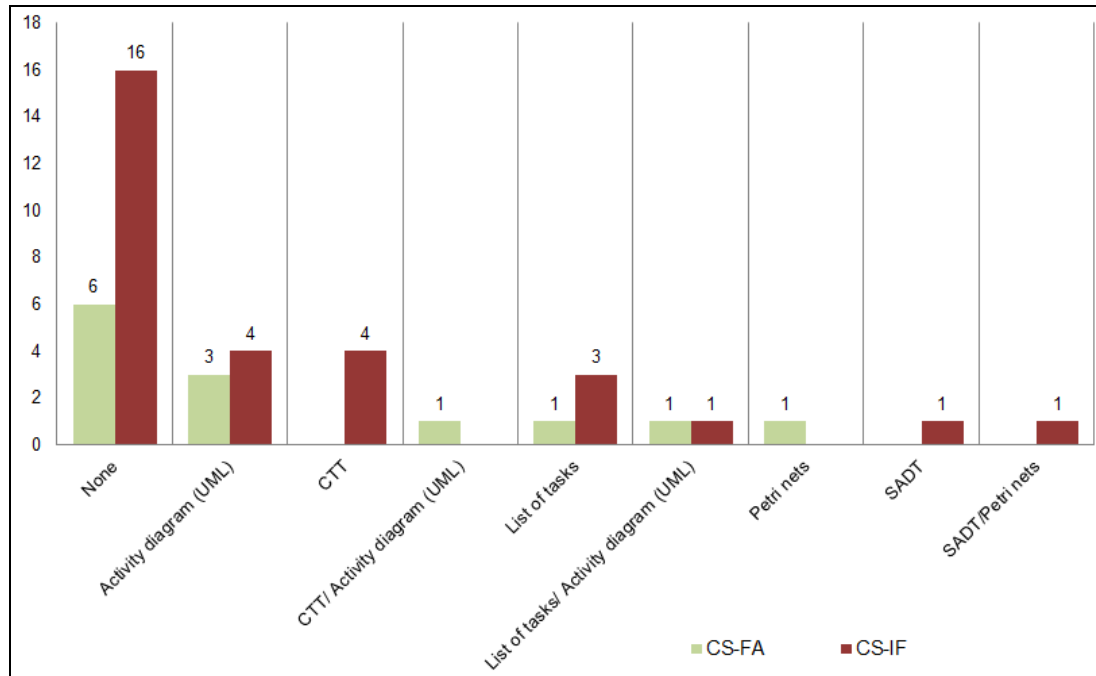


Figure 41. Detailed result of task modeling methods

4.3.4. Question 3: How detailed was the task modeling?

For this question we classified the result of task modeling as “global modeling” or “detailed modeling”. For instance, when using CTT we considered “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.

For 43 reports (30 of CS-IF and 13 of CS-FA), 21 presented task modeling and 14% of these (3/21) were detailed (all of them from CS-IF). These results found (see Figure 42) that even when the models are defined, they are not defined in detail, contrary to what we expected.

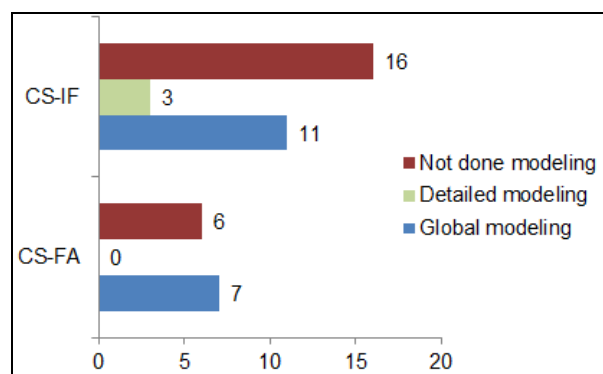


Figure 42. Task modeling details

We argue that task models must provide a level of detail that covers the different levels of user tasks that can contribute to the design of user interfaces (UI) which reflect the reality of the end users. These principles were explained in class²² and not followed by the students.

4.3.5. Question 4: Does the task modeling consider all profiles defined in the problem?

In this question, we found different results for the two groups. Figure 43 shows our findings for each user profile defined in the study. We noted that of 43 projects (13 from CS-FA and 30 from CI-IF) only 21 presented task models.

For the CS-FA program (7 reports out of 13) the profiles most described are: supervisor (6 times) and rounds man (4 times). For the CS-IF program (14 reports out of 30), the profiles are: supervisor (14 times) and rounds man (7 times). Nevertheless, we expected to find all profiles described in each one of the 43 reports, in different task models or in a generic one.

All user profiles were found in the reports by CS-IF, but only 1/14 reports that presented task modeling considered all user profiles in task modeling. On the other hand, for CS-FA, six user profiles were found in the reports, but no report (0/7) presents the seven user profiles in task modeling. These results show that the user profiles defined in the project were not identified in most task models. Moreover, usually every user profile should be represented in an associated task model (even if this model is included as part of a complete model for the whole system).

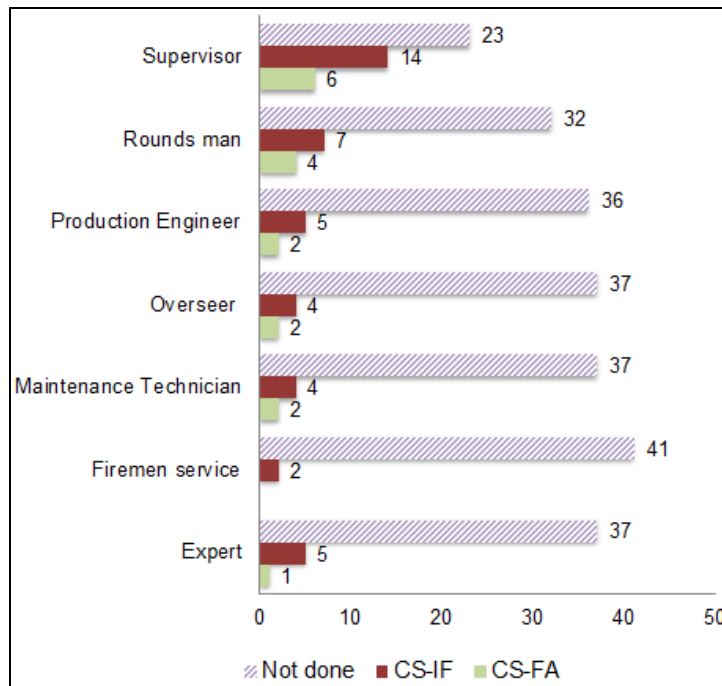


Figure 43. User profiles found in task models

²² Recall: During the previous year.

4.3.6. Question 5: Are the user profiles described in task modeling also described in use case diagrams?

To answer this question, we analyzed the use case diagrams presented in the requirement specifications. Table 30 summarizes the found results for the two groups. We note that: 100% (13/13) from CS-FA and 86% (26/30) from CS-FI presented use case diagrams. In contrast, few reports present all user profiles (3/13, i.e., 23% for CS-FA and 10/26, i.e., 38% for CS-FI). In general, 33% (13/39) of the reports that presented use case diagrams considered the seven user profiles.

Table 30. Results of Use Case Diagrams and profiles

	CS-FA	CS-FI	Total
Reports that presented Use Case Diagram	13/13 (100%)	26/30 (86%)	39/43 (90%)
Reports that presented all seven profiles	3/13 (23%)	10/26 (38%)	13/39 (33%)

Figure 44 presents the user profiles found for the two groups. Like in the previous question, we expected that all 43 specifications would present all user profiles defined in the use cases.

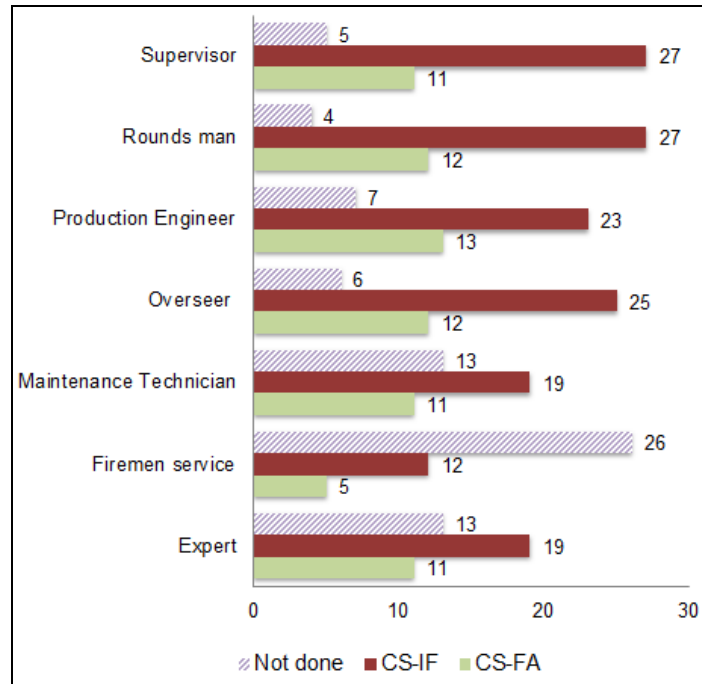


Figure 44. User profiles found in use case diagrams

In Figure 45 we present the general results for the user profiles found in use case diagrams and in task models. In both cases we expected to find 43 reports with use case diagrams and task models, but we identified only 39 reports that have use case diagrams and 21 reports that have task models. The user profiles represented the most in use case diagrams are: rounds men and supervisor. The supervisor was the user profile taken into account the most in task models.

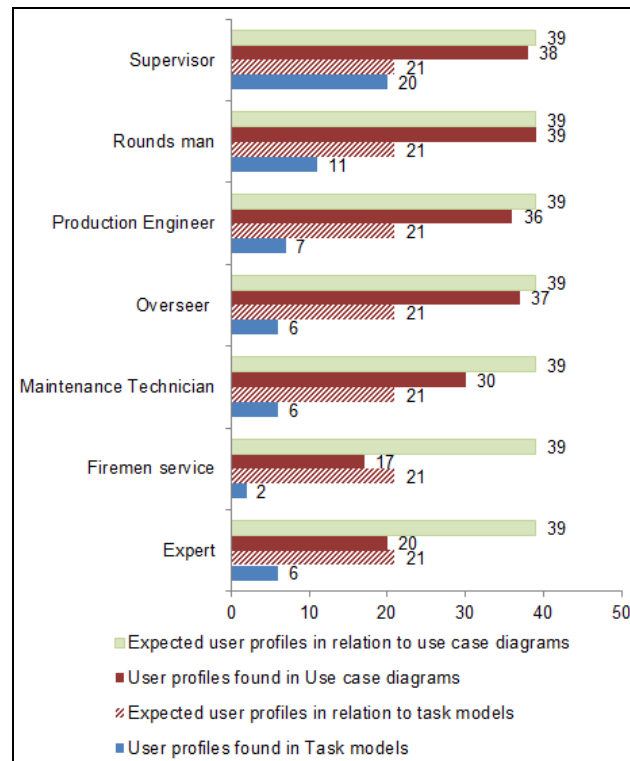


Figure 45. General results - user profiles found in Use Case diagrams and in Task models

4.3.7. Question 6: What are the students' feedback considering the project subject, evaluation and pedagogical issues?

For this question we analyzed three topics from the 150 responses to the individual evaluation questionnaire. About the mini-project (requirement specification - see Figure 46) most of the students (64% - initial interest) initially felt interested in the subject, considered that the subject was well detailed (54% - subject comprehension) for their comprehension and that it was "at the right level" (70% - difficulty of work) for their learning. However, they considered that the time spent in the supervised work classes was not relevant enough (44% - time for performance) or not at all sufficient (19% - time for performance) when compared with the time spent²³ to perform the whole project specification.

²³ The analysis of the reports shows that the students were motivated since they worked outside of the class to finally produce professional quality reports, sometimes comprising about thirty pages. They considered that the time in class was not enough and all compensated by a considerable work outside of the class.

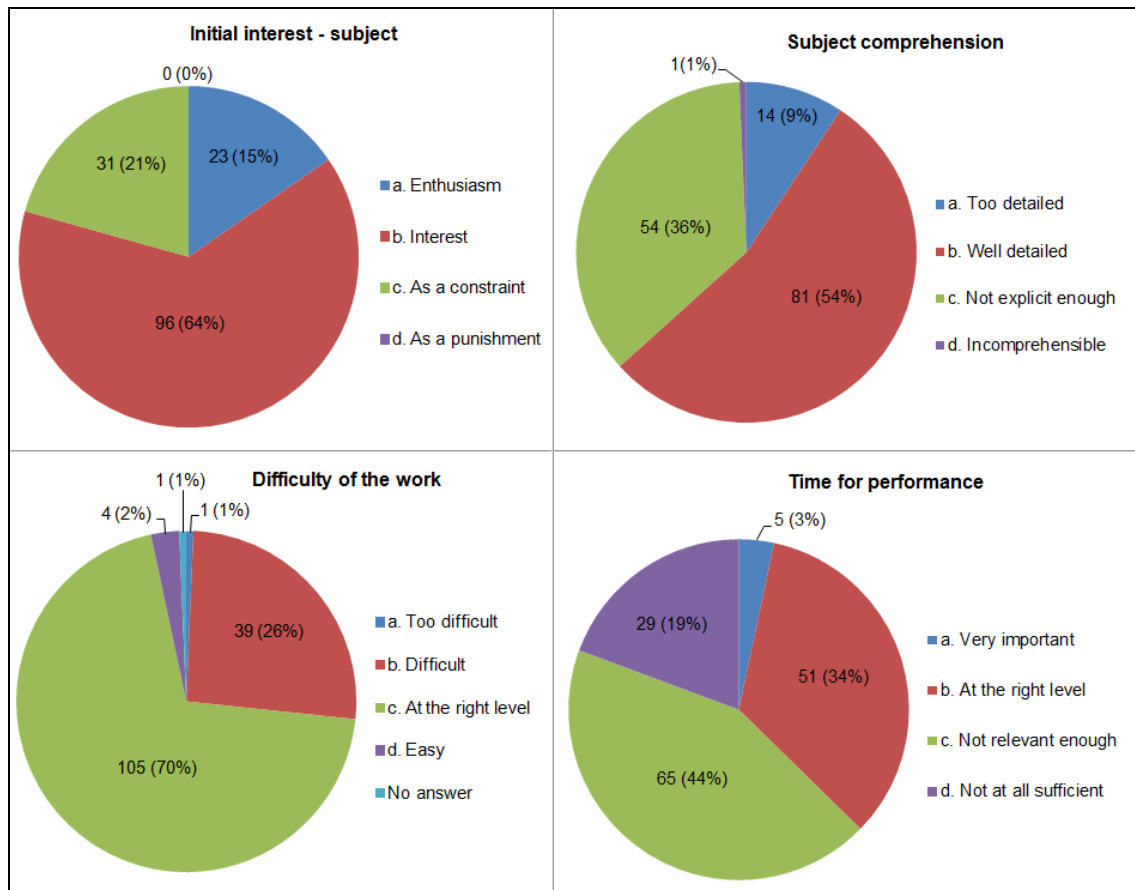


Figure 46. Students' opinion about the studied project

The pedagogy was evaluated considering twelve items as presented in Figure 47. We note that:

- more than half of the students (66% - 99, initial interest) felt interested in the use of a scenario/methodology even though it is obligatory;
- 69% (104, study of the scenario/method) declared they read the subject very carefully;
- 51% (77, understanding alone) of the students considered the scenario easy to understand by themselves, but 59% (89, understanding in group) consider that is easier to understand in a group;
- 56% (84, participation thanks to the scenario/method) of the students are almost certain that the scenario/method makes the supervised classes more motivating and encourages greater participation;
- the students also think that the use of scenario/method is relevant (63% - 95 students, utility of scenario/method), easy to apply (54% - 82 students, scenario/method understanding), that its application (78% - 117 students, quality of the report) has favored the quality of the report, and 56% (84 - scenario method/application) declared that they absolutely apply the scenario/method;
- 69% (103, knowledge provided by teachers) declare that the acquired knowledge (previously and in the teaching class) was at least largely sufficient to perform the project (52% largely and 17% absolutely) against 2% (3) that considered it not enough.

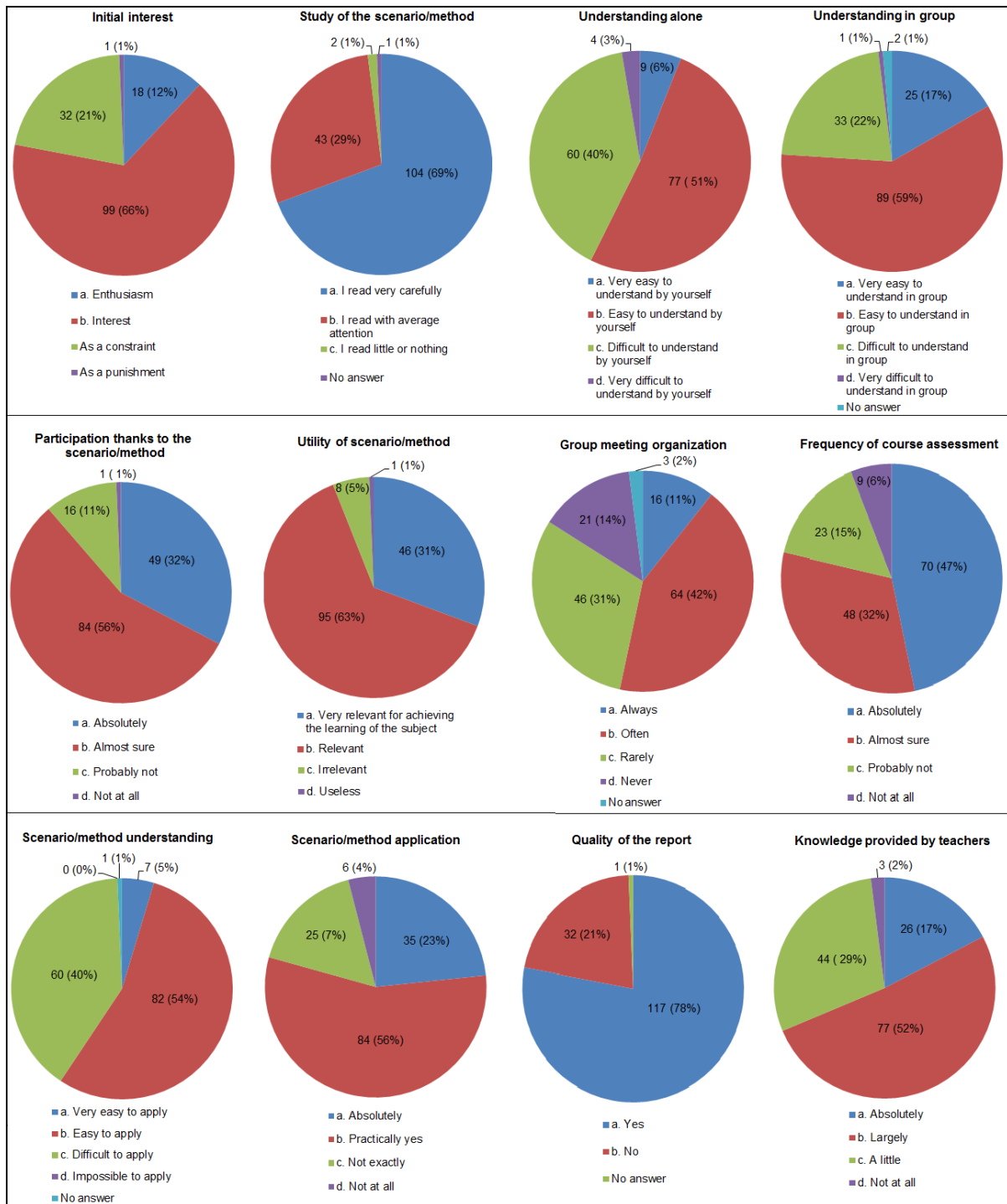


Figure 47. Students' opinion about the pedagogy

Finally about the evaluation (see Figure 48), 59% of the students considered the evaluation by the requirement specification “binding but supportable”, 82% considered that the evaluation system was “highly” or “absolutely pertinent” to promote learning, and the majority (74%) prefer being evaluated with the project instead of only an exam.

In summary, we can conclude with the evaluation of all the questions that the students were quite satisfied with the applied methodology and the assessment of their learning by using the project requirement specification.

Workload	Relevance	Preference of a single exam
a. Absolutely (18 - 21%)	a. Absolutely (48 - 32%)	a. Absolutely (8 - 5%)
b. Binding but supportable (88 - 59%)	b. Highly pertinent (76 - 50%)	b. Strongly (5 - 4%)
c. Binding but easy to integrate into your training workloads (26 - 17%)	c. Not very pertinent (22 - 15%)	c. A little (26 - 17%)
d. Not at all (17 - 11%)	d. Not at all (4 - 3%)	d. Not at all (111 - 74%)
No answer (1 - 1%)	-	-

Figure 48. Students' opinion about the evaluation

4.4. Second iteration: Descriptive analysis

In this second iteration (started in 2015), the students received a list of HCI approaches²⁴ - recall of methods and techniques (see Table 31) that could be used in the requirement specification. This list presents an intermediate result of the HCI approaches (categories) presented in Chapter 3 for requirements development process area since at that time (2015) the research presented in Chapter 3 was not still completed. However, it is important to highlight that the category being studied (Task Modeling category) is the same since the intermediary result. We remember that the students should provide their requirement specification as they wish, following or not this recall.

Table 31. Recall: Suggestions of Approaches to Designing Interactive Systems

#	HCI Approaches	Intention	Examples
1	Techniques to identify needs	Identify the tasks of the end user.	<ul style="list-style-type: none"> Elicitation techniques: <ul style="list-style-type: none"> ✓ Brainstorming ✓ Interviews ✓ Questionnaires ✓ Card Sorting ✓ Focus Groups ✓ Field Studies Techniques for Analysis and Documentation <ul style="list-style-type: none"> ✓ Persona ✓ Scenario (User stories) ✓ Storyboard
2	Methods of analysis and modeling of tasks	Identify stakeholder needs, expectations, constraints and interfaces.	<ul style="list-style-type: none"> CTT (Concur Task Tree) K-MAD (Kernel of Model for Activity Description) or MAD (Model for Activity Description) HTA (Hierarchical Task Analysis) SADT (Structured Analysis and Design Technique) or SADT coupled with Petri Nets GTA (Groupware Task Analysis)
3	Standards and Guidelines for the HCI design	Use standards and guidelines for HCI design and documentation.	<ul style="list-style-type: none"> Ergonomics criteria (Scapin and Bastien, 1993; Vanderdonckt, 1994) ISO/IEC 9126-1 (2001) ISO 9241-11 (1998) ISO/IEC 25000 (2014)
4	Prototype for HCI	Specify a prototype to design and validate the requirements of HCI.	<ul style="list-style-type: none"> Rapid Prototyping <ul style="list-style-type: none"> ✓ Offline techniques: Paper and pencil (paper sketches, storyboards), Mockups, Wizard of Oz, Video prototyping ✓ Online techniques using software tools: No interactive simulations, Interactive simulations

²⁴ The students received this document 30 minutes after receiving the subject of the project. In addition the teacher said that it was a complementary document that could be useful (without any imposition) to perform the requirements specification.

#	HCI Approaches	Intention	Examples
5	Operational Concepts and related Scenarios	Develop an operational concept for the use of the product, detailing the interaction of the product, the end user and the environment (in the form of scenarios).	<ul style="list-style-type: none"> • Characteristics <ul style="list-style-type: none"> ✓ Context awareness ✓ Adapting to context • Techniques <ul style="list-style-type: none"> ✓ Persona ✓ Scenarios ✓ Use cases (with scenarios) ✓ User Profile (detailed)
6	Techniques to validate requirements	Validate the requirements taking into account the needs and constraints, in terms of HCI.	<ul style="list-style-type: none"> • Proto Task (K-MAD) • Task Model Simulator (CTT) • Focus Group to validate requirements • Questionnaires • Verbalization (Thinking Aloud)

We analyzed 22 reports: 10 from CS-IF and 12 from CS-FA. These reports were produced by 72 students: 34 students from CS-IF (2015-2016) and 38 students from CS-FA (2015-2016) for which we had all the answers of the feedback questionnaire. For the analysis we only considered complete projects (the report and the answers for the students' evaluation questionnaires (Annex C).

In the next sections we will present the results found for the all questions of the study.

4.4.1. Students' profile

In this section we present the information about the students' profile. Then, in the next sections we present the results of each one of the research questions previously defined. For the analysis of the students' profile, we analyzed the data of the questionnaire and when the students left a blank answer we consider, for reasons of presentation, the value "No Answer" as a scale point (this happens for 6 questionnaires – 5 with only one blank answer and 1 with more than one).

Concerning the gender of the respondents, we had 7 females and 65 males. We present in Figure 49 the general results about the students' evaluation. The majority of students consider themselves to be methodical (76% - working method) and good workers (58% - work investment). They prefer to work in pairs (49% - work preference) rather than individually (8% - work preference). Moreover, the great majority (57% - freedom of action) prefer to be guided in part of the work, in the beginning (29% - freedom of action) than be guided from the start and throughout the work (10% - freedom of action).

We compute the mode for each item as presented in Table 32 for the 22 reports, with the objective to analyze the profile of the group of students that work on the same project. In general, the working groups are methodical (81% - 18/22), good workers (55% - 12/22), they prefer to be guided only in part of the work (59% - 13/22) and to work in pairs (63% - 14/22).

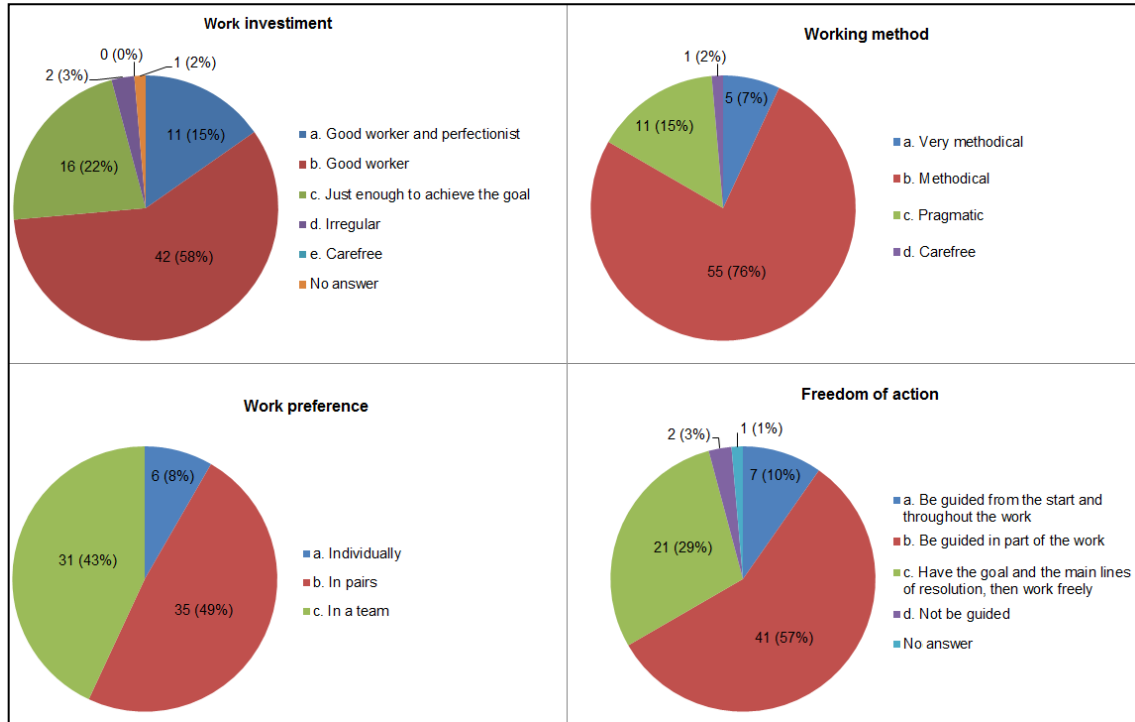


Figure 49. Students' profile

Table 32. Mode of the group profile

Options in the scale	Work investment	Working method	Work preference	Freedom of action
a	Good worker and perfectionist (2)	Very methodical (1)	Individually (1)	Be guided from the start and throughout the work (0)
b	Good worker (12)	Methodical (18)	In pairs (14)	Be guided in part of the work (13)
c	Just enough to achieve the goal (4)	Pragmatic (1)	In a team (6)	Have the goal and the main lines of resolution, then let you do (5)
d	Irregular (0)	Carefree (0)	-	Not be guided (0)
e	Carefree (0)	-	-	-
a & b	Good worker and perfectionist & Good worker (0)	Very methodical & Methodical (0)	Individually & In pairs (0)	Be guided from the start and throughout the work & Be guided in part of the work (0)
b & c	Good worker & Just enough to achieve the goal (1)	Methodical & Pragmatic (1)	In pairs & In a team (1)	Be guided in part of the work & Have the goal and the main lines of resolution, then let you do (1)
No mode	3	1	0	3
Total of reports	22	22	22	22

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

As in the first iteration, we considered to answer this question any specification that describes the final user tasks as a result of task modeling. We present in Figure 50 the results for the 22 reports. We note that only 20% (2/10) from CS-IF and 41% (5/12) from CS-FA **did not** present any result of task modeling. That means that 68% of the reports (8/10 from CS-IF and 7/12 from CS-FA) presented task modeling.

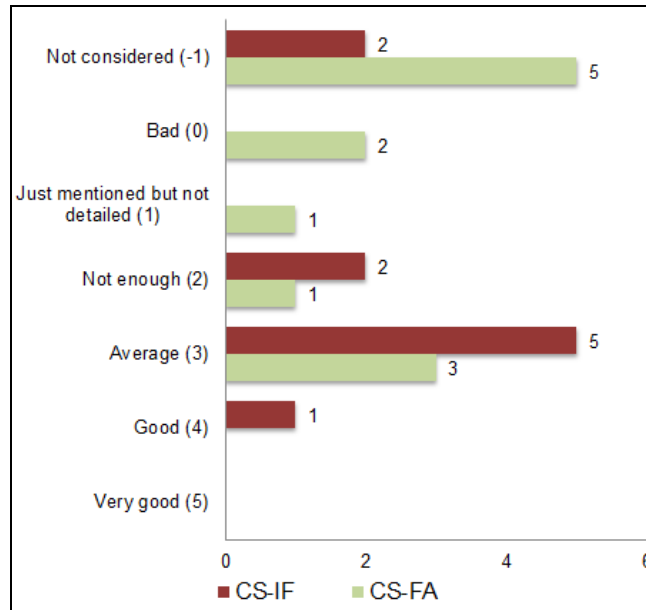


Figure 50. Task modeling

We found 9 reports (9/22 = 40%) that considered task modeling was really relevant for this study; meaning reports where task models were evaluated from 3 to 5 in the scale: 6 (60% of the total) of out 10 reports from CS-IF, and 3 (approximately 25% of total) of out 12 reports from CS-FA. In general, these results show that the task modeling activity was well performed by the students.

4.4.3. Question 2: Which are the methods used for task modeling?

The results for this question, presented in Figure 51, show that for the 22 reports, 15 (68%) present some task modeling using different formalisms for task modeling. Two reports present the use of two combined models, as follows: CTT/Activity diagram and CTT/SADT. We found that 13/22 (1 report with CTT/SADT, 1 report with CTT/Activity diagram, 10 reports with CTT, and 1 report with Petri Nets) used the methods taught in class. That means that the formalisms taught in class were **systematically** used by the students.

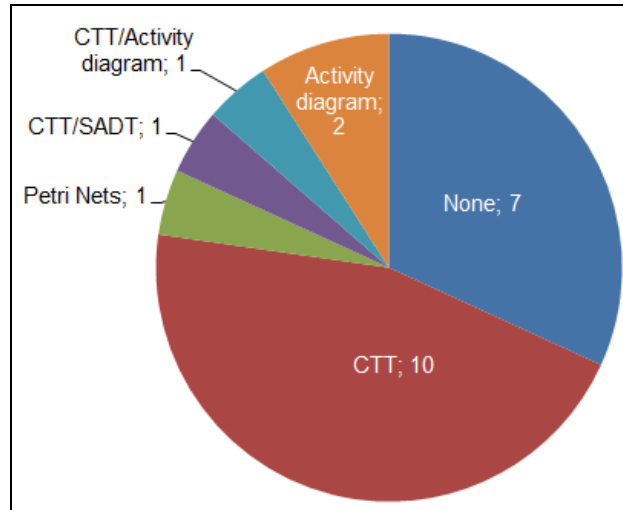


Figure 51. Task modeling methods

Looking in more detail, Figure 52 shows the methods used by CS-IF and CS-FA.

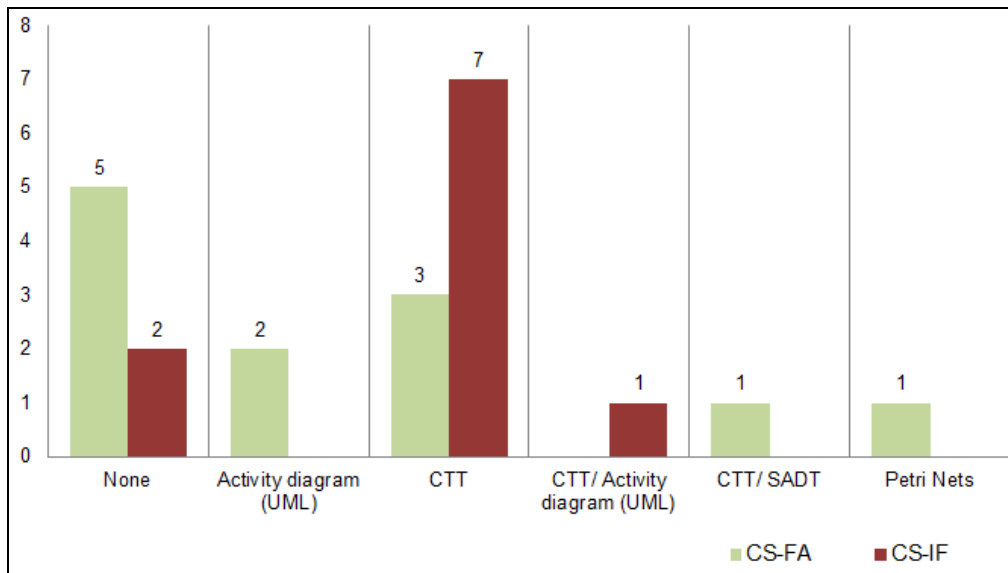


Figure 52. Details of the task modeling methods

4.4.4. Question 3: How detailed was the task modeling?

For this question we classified the result of task modeling as “global modeling” or “detailed modeling”. We have 15/22 reports that presented task modeling: 8 from CS-IF where 4 presented a “detailed task modeling”; and 7 from CS-FA where 2 presented a “detailed task modeling”. As conclusion, 40% (6/15) of the reports presented detailed task modeling. These results (see Figure 53) in general are good and for CS-FA, that presented few detailed models, we can explain that these models presented a good quality.

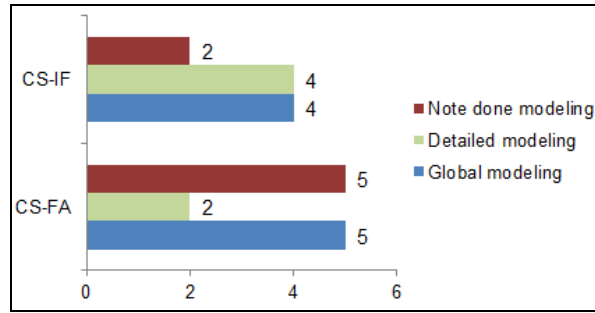


Figure 53. Details of the task modeling

4.4.5. Question 4: Does the task modeling consider all profiles defined in the problem?

We present in Figure 54 the result found for each user profile defined in the study. For the reports from CS-IF that presented task modeling (8 reports out of 10), only three profiles are considered: rounds man (6 times), supervisor (4 times) and technician (1 time). For CS-FA (5 reports out of 12) the profiles most described are: supervisor (2 times), rounds man (2 times), expert (2 times) and engineer (2 times). In addition, for 3 reports from CS-FA the models were considered global (it is not possible to identify the profiles). No report considered all the profiles defined in the study. The firemen profile was not considered by any report. The user profiles defined in the project were not identified in most task models.

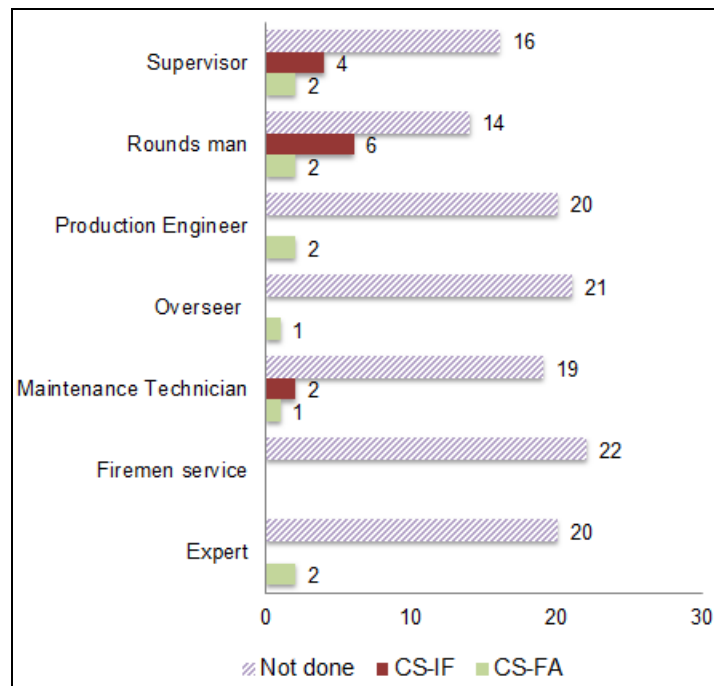


Figure 54. User profiles in task models

4.4.6. Question 5: Are the user profiles described in task modeling also described in use case diagrams?

To answer this question, we analyzed the use case diagrams in the requirement specifications. In Table 7 we summarize the found results for the two groups. For the CS-FA group only 41% (5/12) of the reports presented use case diagrams and 40% (2/5) of them presented all seven profiles. For CS-IF

group we found better results concerning the definition of use case diagrams (80% - 8/10), but only 25% (2/8) of them presented all user profiles. In general, only 31% (4/13) of the reports that presented use case diagrams considered the seven user profiles.

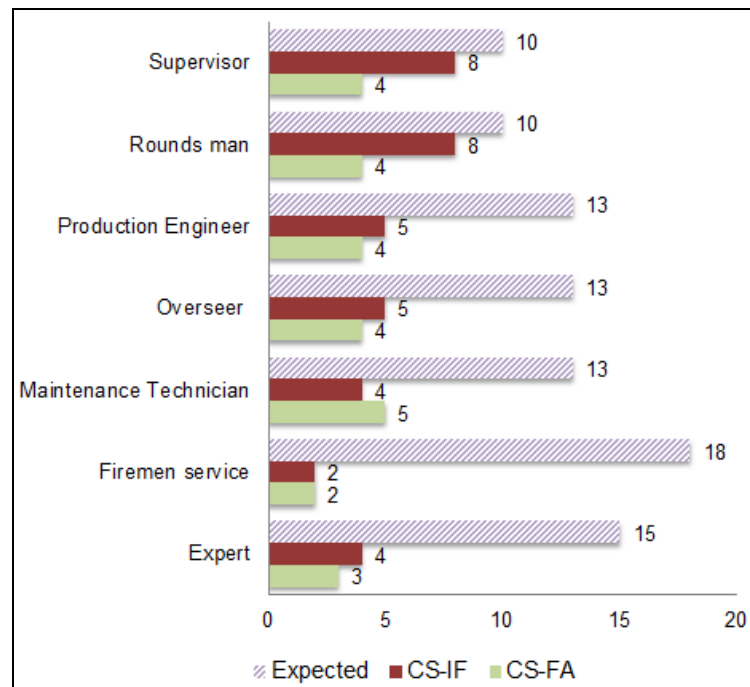


Figure 55. User profiles in use case diagrams

Figure 55 presents the user profiles found for the two groups. We expected that all 22 reports presented use case diagrams with all seven user profiles defined in the study.

Table 33. Use Case Diagrams and profiles

	CS-FA	CS-IF	Total
Reports that presented Use Case Diagram	5/12 (41%)	8/10 (80%)	13/22 (59%)
Reports that presented all seven profiles	2/5 (40%)	2/8 (25%)	4/13 (31%)

In Figure 56 we present the general results for the user profiles found in use case diagrams and in task models. In both cases we expected to find 22 reports with use case diagrams and task models, but we identified only 13 reports that have use case diagrams and 15 reports that have task models. The user profiles represented the most in use case diagrams are: supervisor, rounds man, production engineer, overseer and technician. The rounds man was the user profile taken into account the most in task models.

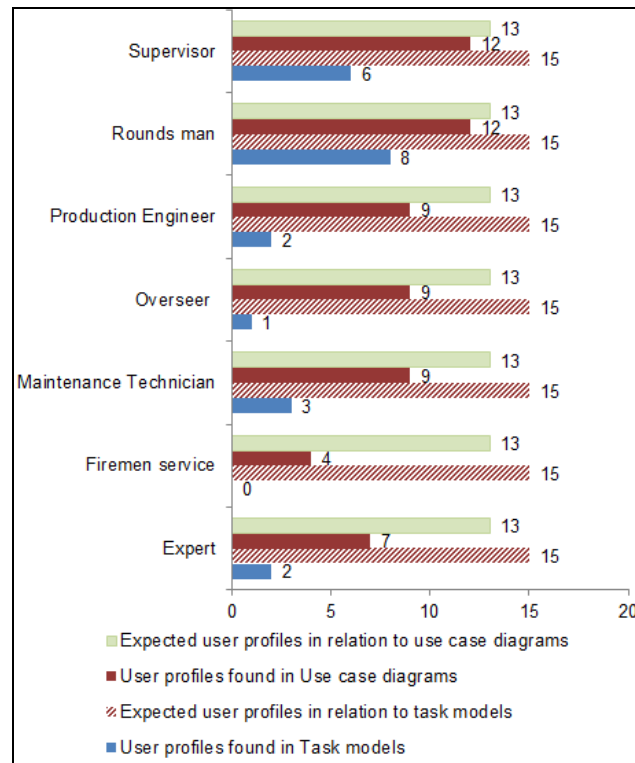


Figure 56. User profiles found in Use Case diagrams and in Task models

4.4.7. Question 6: What are the students' feedback considering the project subject, evaluation and pedagogical issues?

In this question we analyzed three topics from the 72 responses to the individual evaluation questionnaire. For the topic mini-project (requirement specification – see Figure 57), most of the students (78% - initial interest) initially felt interested in the subject and they considered that the subject was well detailed (64% - subject comprehension) for their comprehension. Regarding to the difficulty of the project most of the students said that it was “at the right level” (58% - difficulty of work) for their learning. However, some of them considered that the time spent in the supervised work classes was not relevant enough (46% - time for performance) or not at all sufficient (25% - time for performance).

The topic pedagogy was evaluated considering twelve items as presented in Figure 58. For these items we note that:

- more than half of the students (54 - 75%, initial interest) felt interested in the use of a scenario/methodology;
- 68% (49, study of the scenario/method) declared that they read the subject very carefully;
- 57% (41, participation thanks to the scenario/method) of the students are almost certain that the scenario/method makes the supervised classes more motivating and encourages greater participation;

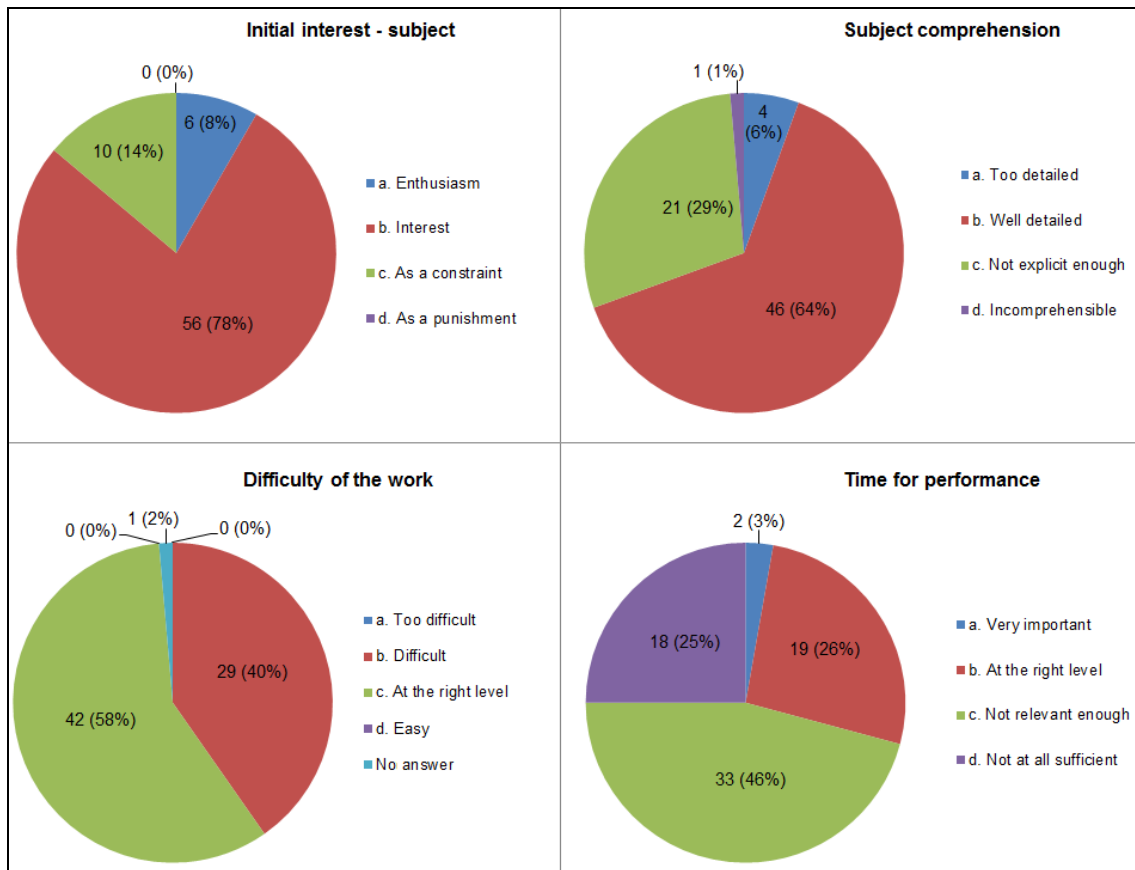


Figure 57. Opinion of the students about the studied project

- some students think that the use of scenario/method is relevant (76% - 55 students, utility of scenario/method), easy to apply (68% - 49 students, scenario/method understanding), that its application (83% - 60 students, quality of the report) has favored the quality of the report and 78% (56 - scenario method/application) declared that they absolutely apply the scenario/method;
- 77% (55 – 46 largely and 9 absolutely, knowledge provided by teachers) declare that the acquired knowledge (previously and in the teaching class) was at least largely sufficient to perform the project against 1% (1, knowledge provided by teachers) that considered it not enough.

Finally about the evaluation topic (see Figure 59), 55 (76%) students considered the evaluation by the requirement specification “binding but supportable”, 81% considered that the evaluation system was “highly” or “absolutely pertinent” to promote learning and the majority (72% - 52 students) prefer being evaluated with the project instead of only an exam.

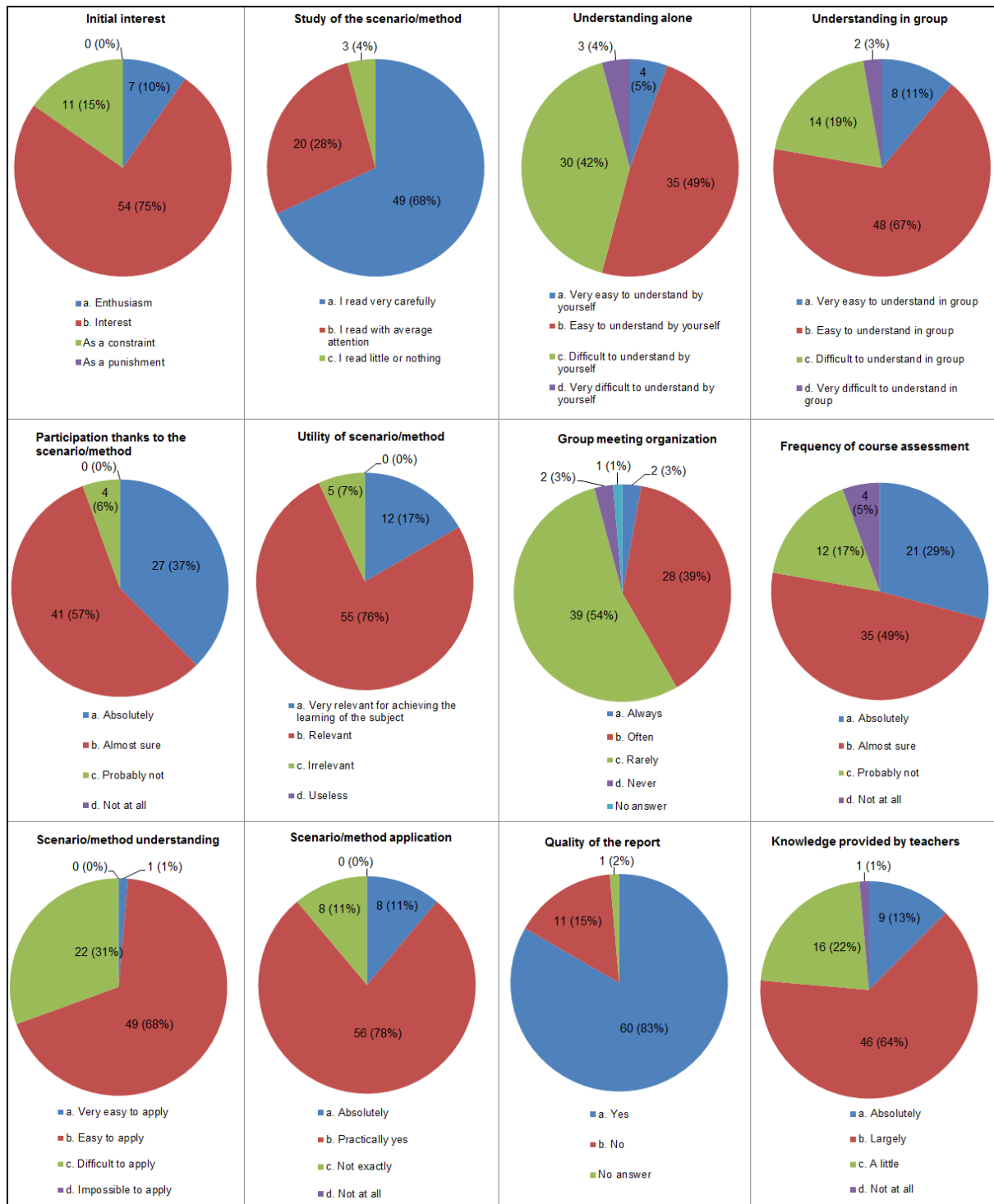


Figure 58. Opinion of the students about the pedagogy

Workload	Relevance	Preference of a single exam
a. Absolutely (8 - 11%)	a. Absolutely (14 - 19%)	a. Absolutely (5 - 7%)
b. Binding but supportable (55 - 76%)	b. Highly pertinent (45 - 62%)	b. Strongly (2 - 3%))
c. Binding but easy to integrate into your training workloads (7 - 10%)	c. Not very pertinent (9 - 13%)	c. A little (11 - 15%)
d. Not at all (0 - 3%)	d. Not at all (4 - 6%)	d. Not at all (52 - 72%)
-	-	No answer (2 - 3%)

Figure 59. Students' opinion about the evaluation

4.5. Discussion and comparison of the two iterations

In the first iteration of the study we involved 150 Master's degree students, and in the second 72. These students were placed in a situation close to industrial reality: they were asked to provide a specification document for a complex interactive system. In both cases, no specific modeling formalism was compulsory, but they could use any modeling method/approach they had learned in the recent years of university education (in HCI classes, software engineering modules, etc.). In the second iteration we gave a recall of HCI approaches²⁵ (see Table 31) that could be used in the requirement specification.

The students worked in groups and sixty-five (65) specification reports (43 in the first iteration and 22 in the second iteration) were analyzed regarding to task modeling. It was thus possible to analyze in depth how they exploited the task models that were taught and practiced with several exercises often commented on or mentioned in different courses. Their systematic exploitation was therefore expected in all the 65 requirement specifications.

We believed that students had understood the relevance or not of task modeling in the specification and whether its use is a constraint or not. The particularity of the first iteration was the complete absence of instructions concerning the task modeling approaches.

Another point to remember is that the HCI module (in computer science or any other domain) is only one module among others, whether students are interested in/passionate about it or not. They must be trained and every year throughout their curriculum they deal with many subjects: programming, database, software engineering, artificial intelligence, operational research, computer architecture, complexity, etc.

We recall that the objective of this study is to investigate whether suggesting HCI approaches (in this case, suggesting task modeling) can increase their use in practice. Thus, we analyzed the data from two iterations (without and with recall regarding to HCI approaches) looking for the task modeling presented in 65 requirement specification. We also compared the two iterations with the objective to show that if the students have a suggestion of HCI approaches they can produce more task modeling. Following we present a discussion about each question of the study and the comparison done to show the effective use of task modeling in requirement specification.

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

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% (21/43 - see Figure 60) of the reports performed task modeling. A small part (21%, i.e., 9 out of 43 - see Figure 60) of the reports presented a good result of task modeling. As said above, 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 (Courage et al., 2009) and (Hackos & Redish, 1998). For the second study we have better results.

²⁵ We remember that this document is a recall and informal, i.e., the teacher did not impose anything.

Regarding to the second iteration 68% (15/22) of the reports presented task modeling. In addition, from the 22 reports, 9 (40%) reports presented task modeling that we consider satisfactory for this study. We considered task models as satisfactory when they were evaluated from 3 to 5 in the defined 7-point Likert scale (see section 4.2.1) We can note an **improvement (from 49% to 68%)** regarding the percentage of reports (see Figure 60) 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 (see Figure 60) that presented satisfactory task modeling. We can say that the suggestions of HCI approaches improved the results.

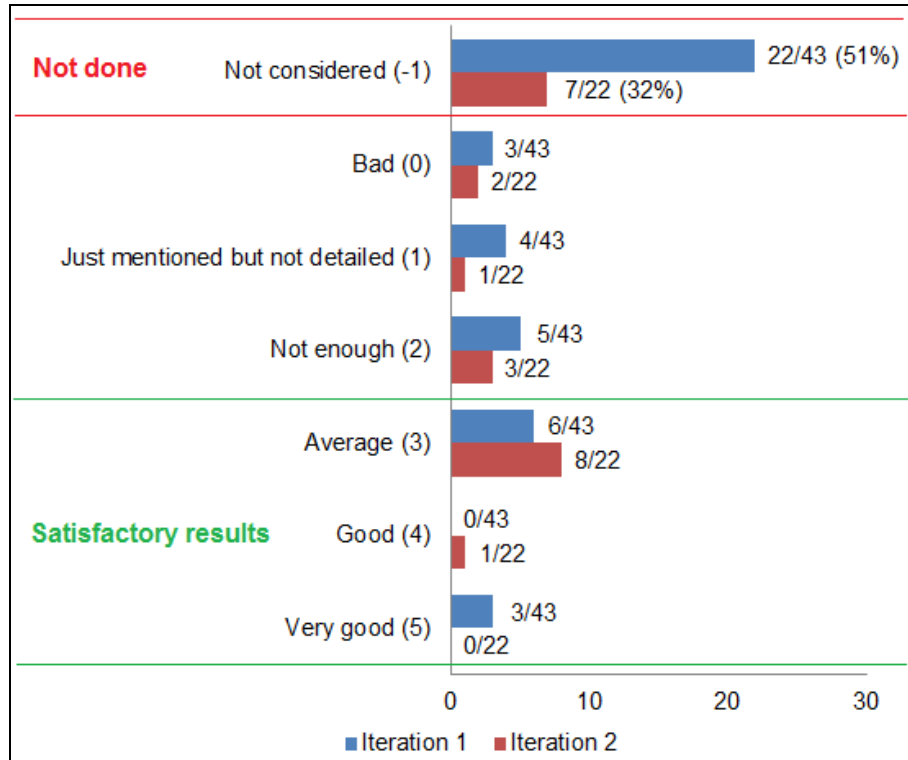


Figure 60. Comparison of the two iterations

Analyzing the answers of the students regarding the evaluation questionnaire (Annex C), we could suppose that maybe they do not have enough time and this is a variable that implies in the construction or not of task modeling. In question 6 (Mini project topic – time for performance, see Figure 46 for iteration 1), we found that 63% of the students think that the dedicated realization time compared to the work required was not sufficient or not relevant enough. We made the same analysis for the iteration 2 (see Figure 57, time for performance) and we found that 71% of the students think in the same way.

To confirm that time is an important variable, we calculated the **mode** for each group regarding to the time to perform the work. We present in Table 8 the result of the two iterations and we can note that in both cases the **mode** is “not relevant enough”. But the difference is that in the iteration 2 we had a higher percentage of reports that presented task modeling than in iteration 1. Thus, we can suppose that even though time has been an impediment to produce task modeling, the suggestion of HCI approaches (including task modeling) helped in the decision to construct this type of model.

We believe that the time must have been an important and decisive factor in choosing to perform task models or other models for the project (for instance, use cases). However, this does not explain why, when they have the choice, they prefer not to do task modeling.

Table 34. Mode of the groups – time for performance

Options in the scale	Iteration 1	Iteration 2
a. Very important	1	0
b. At the right level	11	6
c. Not relevant enough	14	8
d. Not at all sufficient	8	4
b & c. At the right level & Not relevant enough	3	1
c & d. Not relevant enough & Not at all sufficient	1	1
No mode	5	2
Total of reports	43	22

We also analyzed two other variables (see Figure 38 for iteration 1 and Figure 49 for iteration 2) that maybe have influenced in the construction of task modeling. When we analyze the **mode** of the groups regarding to the “work investment” (see Table 35) and “freedom of action” (see Table 36, we identified that the students of some groups declared be good workers and they prefer to be guided in part of the work. We can suppose that maybe these groups decided to work without being guided. For the iteration 2 we can say that in general the groups made a better job confirmed by the percentage (**68% against 49% for iteration 1**) of reports that present task modeling. Thus, once again we can suppose that the use of HCI approaches help to improve the production of task modeling.

Table 35. Mode of the groups – work investment

Options in the scale	Iteration 1	Iteration 2
a. Good worker and perfectionist	2	2
b. Good worker	12	12
c. Just enough to achieve the goal	4	4
d. Irregular	0	0
e. Carefree	0	0
a & b. Good worker and perfectionist & Good worker	0	0
b & c. Good worker & Just enough to achieve the goal	1	1
No mode	3	3
Total of reports	43	22

Table 36. Mode of the groups – freedom of action

Options in the scale	Iteration 1	Iteration 2
a. Be guided from the start and throughout the work	0	0
b. Be guided in part of the work	13	13
c. Have the goal and the main lines of resolution, then let you do	5	5
d. Not be guided	0	0
a & b. Be guided from the start and throughout the work & Be guided in part of the work	0	0
b & c. Be guided in part of the work & Have the goal and the main lines of resolution, then let you do	1	1
No mode	3	3
Total of reports	43	22

Finally, we analyzed another variable (difficulty of the work) that maybe have influenced in the construction of task modeling. Analyzing the **mode** of the groups (see Table 37) we identified that the majority of the groups declared that the level of difficulty of the work is right. For this, we expected to find task modeling in all reports. As previously explained, the best performance was in the second iteration where 68% of the reports produce task modeling. Thus, this improvement can to suggest that the use of HCI approaches helps to improve the production of task modeling.

Table 37. Mode of the groups – difficulty of the work

Options in the scale	Iteration 1	Iteration 2
a. Too difficult	0	0
b. Difficult	8	6
c. At the right level	27	14
d. Easy	0	0
b & c. Difficult & At the right level	5	2
c & d. At the right level & Easy	1	-
No mode	2	-
Total of reports	43	22

4.5.2. Question 2: Which are the methods used for task modeling?

For the second question we expected to find the use of the taught modeling methods: 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. 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) where 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) 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.

4.5.3. Question 3: How detailed was the task modeling?

In the first iteration we found that only 14% (3/21) of the reports presented “detailed task modeling”. We could suppose that maybe they do not retain and put in practice all the information given about the task modeling methods. However the majority of the students (69% - 103 students) answered that they had enough knowledge obtained previously or with the teacher in the course (see Figure 47, knowledge provided by teachers) to perform the required work.

For the second iteration, 40% (6/15) of reports presented “detailed task modeling”. Similarly to first iteration, 77% (55) of the students declared that the acquired knowledge (previously and in the teaching class) was at least largely sufficient to perform the project (see knowledge provided by

teachers in Figure 58). We can note an **improvement (from 14% to 40%)** regarding the percentage of reports that presented **detailed task modeling**.

4.5.4. Question 4: Does the task modeling consider all profiles defined in the problem?

Regarding the first iteration we have only 5% (1/21) of the reports that presented task modeling with the seven profiles defined in the problem description. In the second iteration, we did not find any report that presented task modeling with the all profiles. However, we found that the main profiles (rounds man, overseer and supervisor) were considered for several reports in both iterations (see Table 38). We believe that the students chose only the most demanded profiles in relation to the presented scenario. This is another point that can be studied in the future.

Table 38. Profiles found in Task models

Profiles in Task models	Iteration 1	Iteration 2
Expert	6/21	2/15
Firemen service	2/21	0/15
Maintenance Technician	6/21	3/15
Overseer	6/21	1/15
Production Engineer	7/21	2/15
Rounds man	11/21	8/15
Supervisor	20/21	6/15

4.5.5. Question 5: Are the user profiles described in task modeling also described in use case diagrams?

In the first iteration we found that 33% (13/39) of the use case diagrams presented all user profiles defined in the project. For the second iteration, we found that 31% (4/13) of the use case diagrams presented all user profiles. However, we found that the main profiles (rounds man, overseer and supervisor) were considered for several reports (see Table 39). We could suppose that maybe the students thought that not all profiles were important to the system specification.

Table 39. Profiles found in Use case diagrams

Profiles in Use case diagrams	Iteration 1	Iteration 2
Expert	30/39	7/13
Firemen service	17/39	4/13
Maintenance Technician	30/39	9/13
Overseer	37/39	9/13
Production Engineer	36/39	9/13
Rounds man	39/39	12/13
Supervisor	38/39	12/13

We can note that the results of the questions 4 and 5 were must better for the iteration 1. Moreover, the results were not as we expected for the two iterations. We believe that the students chose only the most demanded profiles in relation to the presented scenario. As said in §4.5.4, the user profile is another HCI approach that we plan to study in the future. Thus, we should revisit all reports to investigate the user profiles.

4.5.6. Conclusion of the iterations' comparison

In general, for the first iteration we considered that the specification reports were good, but 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 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 quality of the requirement specification. In Table 40 we observe that the worst grade, the best and the average for the requirement specification with task modeling was better than those that did not use it (see Table 40), for the first iteration. We can note that the average difference is small (5%) and the average value is low regarding to the minimum value (10) needed to validate the course. Moreover, one 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 40). Even if the best grade is smaller 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 40. Grade for projects with and without task modeling (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 recall with the HCI approaches was the variable that contributed for the improvement of the results. We plan to continue our long-term validation in academic environment with other HCI approaches (categories).

4.6. Threats of validity

To analyze the results, we considered the threats to validity proposed by Wohlin et al. (2012): construct validity, internal validity, conclusion validity, and external validity. We analyzed each one of them and the mitigation we have performed.

Threats to the **construct validity** illustrate the relation between theory and observation, and the questions of whether the treatment adequately reflects the cause; whether the result adequately reflects

the effects. In this study, the main aspect related to this treat is whether the final result (i.e., the requirement specification of the project) used as an evaluation for the course is adequate to evaluate the application of task modeling in requirement specifications. Since the project is used as an evaluation for the course, the students could feel anxious about being evaluated, apply what they know best, and not what they do not feel comfortable using. We considered the fact that the professor provides to the students a detailed description of a real problem, and the fact that the problem is discussed during three class sections mitigates this risk. Moreover, we decided to consider any representation as task modeling; from simple lists of tasks to adequate models using a notation presented in class. This representation would be analyzed in the same way considering the predefined scale (see section 4.2.1). The students are also assured that everything they did would be counted positively for them. In other words, the professor motivates the students to use whatever they want to specify the requirements of the proposed interactive system.

Threats to the **internal validity** draw from influences that can affect the independent variables with respect to causality without the researchers' knowledge. In this study the threat is associated to the students and their interest in learning and using task modeling. As previously presented, in all classes the teacher provides additional references for further study about all techniques, standards and theory presented in the class. However, we cannot guarantee that they will look at this material. To minimize the risk of them not doing so, the teacher defined a competitive environment where the best report will receive a bonus (the best mark). However, we cannot be sure they would do it. We therefore assumed 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 this study the application of task modeling. The results could be considered not reliable since we are using only a part (task modeling) of the final product (requirement specification) of the study. To allow us to draw the best conclusion and mitigate this risk, we consider a seven point Likert scale, where we can really identify to what extent the students perform task modeling. Another risk is the evaluation using this likert scale to be biased by the main evaluator (the author of this work). To minimize this risk we performed this analysis by pairs. Moreover, we used the answers of the evaluation questionnaire (Annex C) to get the opinion of the student about the project, the evaluation and the pedagogy used in the course. In this way, we can analyze the results comparing what is identified and the opinion of the students.

Finally, threats to the **external validity** are conditions that limit our ability to generalize the results of our experiment outside the scope of this study. This study was planned in the context of a Master's degree program (Computer Science) and, therefore, cannot be generalized. However, the study could be replicated in many other universities. Nevertheless, we argued that this study indicates that showing how to perform task modeling, that is indicating explicitly the approaches to do it, can imply in a large use of the HCI approach.

4.7. Synthesis and Conclusion

In this chapter we presented the results of a case study performed in academic environment to validate the effectiveness of use of one HCI category (Task modeling) proposed in this work. To that end we performed two iterations of the fourth phase (long-term validation in academic environment) of the research methodology defined for this thesis. We compared the results of the two iterations, one without the use of HCI approaches and one using the HCI approaches.

We analyzed 43 reports (from 2010 to 2014) for the first iteration, and 22 reports (from 2015 to 2016) for the second iteration. The first iteration was conducted with 150 students and the second iteration with 72 students from Computer Science Master's degree program at the University of Valenciennes, France. We analyzed all reports considering different aspects: the definition of task models using methods, the quality of these task models and the specification of profiles with the task models. Furthermore, 222 evaluation questionnaires (Annex C) were analyzed concerning the project subject, project evaluation, applied pedagogy and suggestions concerning the proposed pedagogy.

In general, the result of the first iteration (43 reports), presented in section 4.3, confirmed what we previously found with some other colleagues (Oliveira, Girard, Gonçalves, Lepreux, & Kolski, 2015): most students do not use task modeling naturally and spontaneously in software design; and it seems that they do not think or understand that task models add value to the requirement specification, even if their benefits are highlighted in class. The results showed that less than 70% of the project reports (49% for iteration 1 and 68% for iteration 2) presented results of task modeling. Several students complained about the large scope of theory in the course and probably they did not retain the knowledge acquired in previous courses. However, these possible justifications do not explain why, when the students have the choice of performing task modeling or not, they actually do not perform it. But the groups have generally felt very involved and have spent considerable time working outside of the sessions, in order to provide a professional quality report.

In the next chapter we will present the results of the fifth phase of this work.

***Chapter 5 – Survey about the perception of knowledge and use of HCI
approaches with SPCM models consultants***

5.1. Introduction

As presented in the General Introduction of this thesis, we believe that one way to introduce HCI issues in industry is to integrate HCI approaches to support the SPCM models that are already used in practice. With the definition of the set of HCI approaches that supports the SPCM models, we performed two empirical studies to investigate to what extent the SPCM models consultants know and use HCI approaches in relation to SE approaches. Our assumption is that the SPCM models consultants do not know and do not use HCI approaches as they know and use SE approaches. Confirming this assumption we can envision that with the definition of how to perform the practices of these models with HCI approaches, the users of SPCM may apply them in practice.

This chapter presents the study performed for the fifth phase (*Survey about the perception of knowledge and use of HCI approaches with SPCM models consultants*) of our research methodology (see item (v) in Figure 61) that was presented in General Introduction.

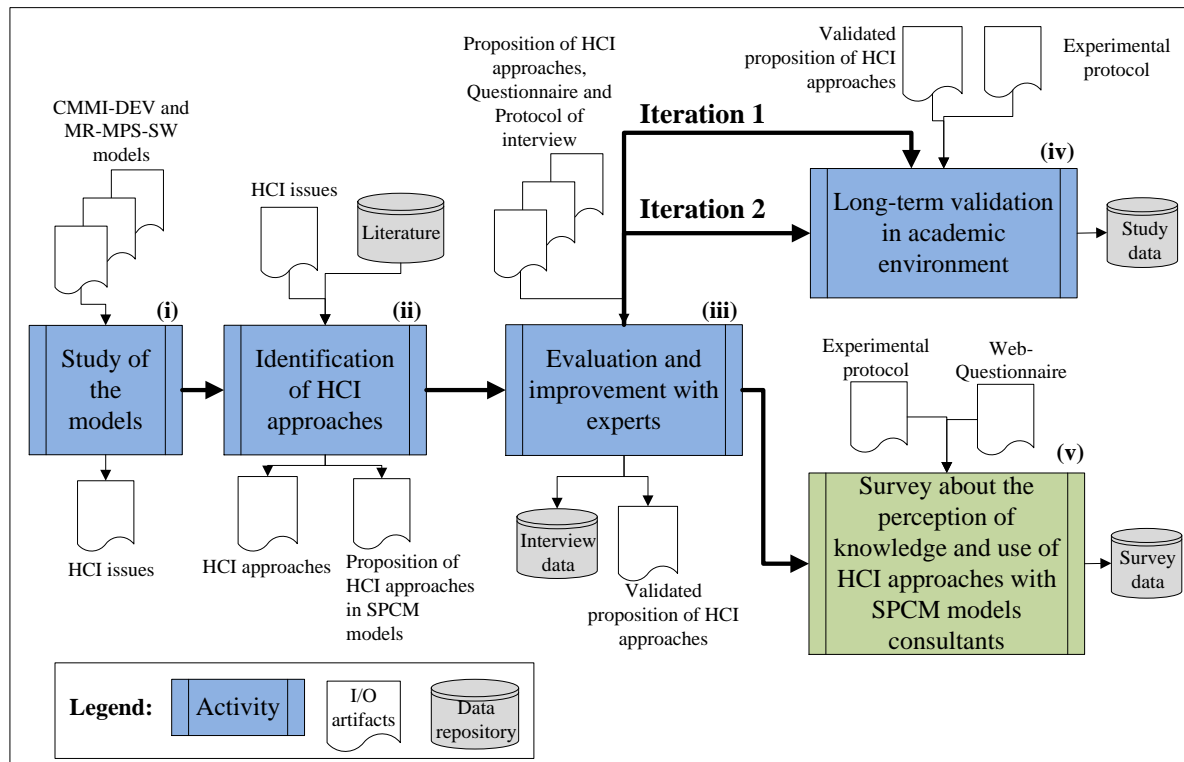


Figure 61. Research methodology

To perform the empirical studies we followed the procedures defined by Wholin et al. (2012) and Purchase (2012).

We start this chapter by presenting the context of this study by formalizing the objectives and hypothesis (section 5.2). Details of the subjects, instrument (web questionnaire) and results will be presented for each study separately. The first study developed in a Brazilian context is described in section 5.3. Preliminary results of the second study developed in an international context are presented in section 5.4. Section 5.5 discusses the threats of validity of these studies. We finish the chapter with a conclusion section (section 5.5).

5.2. Studies context

5.2.1. Objective

Taking profit of the large use of CMMI-DEV and MR-MPS-SW (particularly in Brazil) in industry, we decided to survey consultants of these models regarding the application of HCI in practice. The first study was performed in Brazil regarding the CMMI-DEV and MR-MPS-SW, and the second study was performed with the ten top countries that use CMMI-DEV.

The main objective of these studies is to investigate what is the perception of knowledge and use of the SPCM models consultants (in Brazil and other countries) about HCI approaches integrated to SPCM models practices. Moreover, we aim to compare the results from HCI approaches with Software Engineering (SE) approaches related to the same SPCM models practices. Our assumption is that the consultants do not know and do not use HCI approaches as they know and use SE approaches.

We formalized the goal of these studies according to Basili & Rombach (1988) as follows:

Analyze HCI and Software engineering (SE) approaches
for the purpose of characterization
with respect to the perception of knowledge and use of methods, techniques, standards and patterns
from the point of view of SPCM models consultants
in the context of SPCM models implementations.

In other words, the goal is to characterize the perception of knowledge and use of methods, techniques, standards and patterns of SE and HCI related to SPCM models implementations. We mean by SPCM implementations, the use of SPCM models in the development/maintenance of software projects in industry. The enterprises that decide to use SPCM models usually hire consulting to help them to introduce the practices and to train the staff. This is done because, in general, those enterprises are interested in being officially assessed by the institutes that manage the models (CMMI Institute²⁶ for CMMI-DEV and SOFTEX for MR-MPS-SW²⁷). The consultants are responsible to introduce the approaches to be used by the software developers of the enterprises in the development/maintenance of the software products. Therefore, we considered consultants as a good source to investigate what probably has being used in industry when applying SPCM models.

5.2.2. Hypothesis

The main assumption of our work is that HCI approaches are not sufficiently known and not sufficiently used in practice as SE approaches. To perform our studies we used the HCI approaches (identified and evaluated in the Chapter 3) to support the implementation of the practices of SPCM models. To make the evaluation of our assumption possible we should compare the perception of knowledge and use of these approaches with SE approaches. Therefore, we had to identify SE approaches that are usually applied with SPCM models.

²⁶ <http://partners.cmmiinstitute.com/find-partner-organization/>

²⁷ <http://www.softex.br/mpsbr/instituicoes-autorizadas/>

For the identification of the SE approaches, we analyzed the literature of SE (including CMMI-DEV and MR-MPS-SW), and for each HCI category we defined an equivalent category for SE considering software engineering point of view. Then, we looked for well-known approaches that could support each practice. This proposition was peer reviewed by five experts from software engineering. Our idea was to get more examples and confirm our propositions. We did not consider necessary a larger number of experts, since our goal was not to be exhaustive in terms of examples of the approaches but to quote some important ones. Moreover, the chosen approaches are classical ones from software engineering literature. Annex D shows the form used for the peer review. All the experts have a Ph.D. in software engineering and have experience in industrial projects. They suggest other examples of approaches for each category. Table 1 presents the fourteen HCI categories and the correspondent SE categories.

Table 41. HCI categories x SE categories

#	HCI categories	SE categories
1	Techniques to identify user needs	Techniques to identify needs
2	Techniques to identify user and organizational requirements	Techniques to identify requirements
3	Task Modeling	Software Modeling
4	Standards and Guidelines for HCI design	Standards and Guidelines for design
5	Prototype for HCI requirements	Prototype for requirements
6	Techniques to validate HCI requirements	Techniques to validate requirements
7	Architecture patterns for HCI	Architecture Patterns for SE
8	Design patterns for HCI	Design Patterns for SE
9	Techniques for interaction modeling	Interaction modeling for SE
10	Techniques for HCI documentation	Techniques for final documentation
11	Iterative and Evolutionary Prototypes (system versions)	Prototype (system versions)
12	Evaluation methods for HCI verification	Verification methods
13	Evaluation methods for HCI review	Review methods
14	Evaluation methods for HCI validation	Validation methods

Considering our assumption and the defined HCI and SE categories, we formalized two hypotheses to be investigated:

H1. SPCM models consultants do not know HCI approaches as they know Software engineering (SE) approaches when applying the same specific practice of CMMI-DEV engineering process area or its correspondent result of MR-MPS-SW.

H2. SPCM models consultants do not use HCI approaches as they use Software engineering (SE) approaches when applying the same specific practices of CMMI-DEV engineering process area or its correspondent result of MR-MPS-SW.

Hereafter when we mention “consultants know or do not know” means their *perception* of what they know or do not know.

For the first hypothesis (H1) we have:

- **Null hypothesis (H1₀):** SPCM models consultants know HCI approaches (KHCI) as they know software engineering approaches (KSE) when applying the same specific practice of CMMI-DEV engineering process area or its correspondent result of MR-MPS-SW.

H1₀: $KHCI_i - KSE_j = 0$; where i and j are each equivalent category of HCI and SE that supports each CMMI-DEV practice or result of MR-MPS-SW.

- **Alternative hypothesis (H1_A):** SPCM models consultants know HCI approaches less than SE approaches when applying the same specific practice of CMMI-DEV engineering process area or its correspondent result of MR-MPS-SW.

H1_A: $KHCI_i - KSE_j < 0$; where i and j are each equivalent category of HCI and SE that supports each CMMI-DEV practice or result of MR-MPS-SW.

Similarly, the second hypothesis (H2) related to the use of HCI approaches was formalized as follows:

- **Null hypothesis (H2₀):** SPCM models consultants use HCI approaches (UHCI) as use software engineering approaches (USE) when applying the same specific practice of CMMI-DEV engineering process area or its correspondent result of MR-MPS-SW.

H2₀: $UHCI_i - USE_j = 0$; where i and j are each equivalent category of HCI and SE that supports each CMMI-DEV practice or result of MR-MPS-SW.

- **Alternative hypothesis (H2_A):** SPCM models consultants use HCI approaches less than SE approaches when applying the same specific practice of CMMI-DEV engineering process area or its correspondent result of MR-MPS-SW.

H2_A: $UHCI_i - USE_j < 0$; where i and j are each equivalent category of HCI and SE that supports each CMMI-DEV practice or result of MR-MPS-SW.

5.3. Empirical study in the Brazilian context

In this section we will present the instrument (web questionnaire), the subjects, the execution and the results of an empirical study that was performed with Brazilian SPCM models consultants regarding the CMMI-DEV and MR-MPS-SW models.

5.3.1. Instrument: a web questionnaire

A web survey composed of two parts was developed. The first part was developed to collect the demographic data. The first four data fields (respondent identification, e-mail, formation degree, formation area) were designed to identify the respondent information. After that, the SPCM models consultants answered six following questions:

- Are you an official implementer of the MR-MPS-SW model?
- Are you affiliated to an Implementing Institution (II)²⁸?
- Did you take the official CMMI introduction course?
- How many years have you worked in Capability Maturity models implementations?
- What are the Capability Maturity model(s) and maturity level(s) that you have supported implementations? (a list of all levels was provided)
- Approximately, how many enterprises and projects did you support the implementation? (for the levels previously selected).

The second part is composed of questions about HCI and SE categories and their approaches. Figure 62 presents a screenshot of the main page (a) and part of the questions about HCI and SE categories and their approaches (b). The complete questionnaire is presented in Annex E.

For each category (HCI and SE category), the SPCM models consultants are invited to answer to what extent are their perception of knowledge and use about the categories when implementing the SPCM models practices. To answer this part of the questionnaire they use a Visual Analogue Scale – VAS (Wewers & Lowe, 1990). VAS is a continuous scale usually used in psychological studies. With this scale we can do all arithmetic calculus. It consists of a horizontal line with two anchor points. We used the classical anchor points, from (0) **None** to (10) **A lot**.

5.3.2. Subjects and planning

The subjects selected for this study are the Brazilian SPCM models consultants who work on the enterprises associated to CMMI institute and SOFTEX databases. Three enterprises are associated to CMMI institute and SOFTEX. Only SPCM models consultants that have implemented CMMI-DEV maturity level equal or greater than 3 and MR-MPS-SW maturity level equal or greater than D could participate in this study since our interest is the engineering process areas. The sampling of the population is selected for non-probability sampling technique, and it is a quota sampling where the subjects are selected from various elements of a population. Table 42 shows the information about the SPCM models partner enterprises.

²⁸ In Brazil the SOFTEX, who manages the MR-MPS-SW, has an official database of recognized institutions for implementing the models. All consultants affiliated to these institutions are trained in official courses and perform specific exam to get the grade of official “implementer” of the model.



[HOME](#)
[PART 1](#)
[PART 2](#)

Survey - Implementation of methods, techniques, standards and patterns of Human-Computer Interaction and Software Engineering

This survey aims to evaluate to what extent methods, techniques, standards and patterns of Software Engineering and Human-Computer Interaction are used by software developers that have implemented the maturity levels (A, B, C or D) of the **MR-MPS-SW** model (Reference Model MPS for Software) and/or the maturity levels (5, 4 or 3) of the **CMMI-DEV** model (Capability Maturity Model Integration for Development).

We would like to highlight that any publication generated from this survey will present only statistical results by summarizing the raw data and treating the answers anonymously. In other words, in any circumstances the answers provided in this survey will be published with personal information of the respondents or the institutions for which they work.

This research is part of a doctoral thesis which is being developed at University of Valenciennes and Hainaut-Cambresis and financed by the Brazilian government (Program Science without Borders/CAPEs).

The survey is divided into 2 parts (described below) and the estimated time to fill it is **40 minutes**.

- Part 1 - Characterization**
- Part 2 - Evaluation of the Implementation of methods, techniques, standards and patterns of Human-Computer Interaction and Software Engineering**


We really appreciate your help and time with this research.

Best regards,

Taísa Guidini Gonçalves
Kátia Marçal de Oliveira
Christophe Kolski

[NEXT PAGE](#)

(a)



[HOME](#)
[PART 1](#)
[PART 2](#)

Survey - Implementation of methods, techniques, standards and patterns of Human-Computer Interaction and Software Engineering

Part 2 - Evaluation of the Implementation of methods, techniques, standards and patterns of Human-Computer Interaction and Software Engineering

The item listed above present several methods, techniques, standards and patterns from Software Engineering (SE) and Human-Computer Interaction (HCI) that can support the implementation of the Processes of MR-MPS-SW or of the Process Areas of CMMI-DEV, according to the literature and experts.

Please, indicate your level of knowledge (*I know*) and level of use (*I used*) to each one of those methods, techniques, standards and patterns when in the implementation of Capability Maturity models in enterprises you worked.

#	Process (MR-MPS-SW)	Process Area (CMMI-DEV)	Potential methods, techniques, standards and patterns from Software Engineering (SE)	Potential methods, techniques, standards and patterns from Human-Computer Interaction (HCI)
1	Requirements Development <i>DRE1</i> As necessidades, expectativas e restrições do cliente, tanto do produto quanto de suas interfaces, são identificadas	Requirements Development <i>RD SP1.1</i> Elicit Needs	Techniques to identify needs Examples (see References), not limited to: • Brainstorming • Interviews • Questionnaires • Card Sorting • Focus Groups • Field Studies/Observation • Workshops • Protocol Analysis Answers I Know: None <input type="text"/> A lot I Used: None <input type="text"/> A lot	Techniques to identify user needs Examples (see References), not limited to: • Brainstorming • Interviews • Surveys/Questionnaires • Card Sorting • Focus Groups • Field Studies/Observation Answers I Know: None <input type="text"/> A lot I Used: None <input type="text"/> A lot

(b)

Figure 62. Web Survey

After building the questionnaire we executed a pilot of survey (pre-testing) with two (2) SPCM models consultants' to assess the instrument's survey. They are Brazilian SPCM models consultants with the same characteristics of the population selected for this study. The two consultants answered the web questionnaire and filled an evaluation form (Annex F) related to the instrument. The instrument was filled out by the consultants without the help of an instructor and the evaluation form was sent by e-mail.

The study was planned to be conducted off-line. The questionnaire was available in a web site in a way that the SPCM models consultants answered the questionnaire in his/her time and environment, not being monitored. The survey request was send by email. We contact each person of the partner enterprises to explain about the survey and ask if they have the profile to answer the questionnaire (have participated in consulting for implementation of CMMI-DEV maturity level equal or greater

than 3 and MR-MPS-SW maturity level equal or greater than D). We also confirmed with the coordinator of each partner enterprise which the members have the required profile.

Table 42. Partner Organizations

SPCM models partner enterprises	Total number of consultants	Study population	Models	CMMI Partner	SOFTEX Partner
A	17	8	CMMI-DEV and MR-MPS-SW	No	Yes
B	12	2	CMMI-DEV and MR-MPS-SW	No	Yes
C	18	4	CMMI-DEV and MR-MPS-SW	No	Yes
D	9	7	CMMI-DEV and MR-MPS-SW	Yes	Yes
E	11	3	CMMI-DEV and MR-MPS-SW	Yes	Yes
F	6	1	CMMI-DEV and MR-MPS-SW	No	Yes
G	14	6	CMMI-DEV and MR-MPS-SW	No	Yes
H	8	2	CMMI-DEV and MR-MPS-SW	Yes	Yes
I	1	1	CMMI-DEV	Yes	No
J	9	4	CMMI and MR-MPS-SW	No	Yes
K	9	2	MR-MPS-SW	No	Yes
Total	114	40			

5.3.3. Study execution and analysis of the results

The survey request was sent by email on 11st Nov 2016, with four reminders till 30th March 2017. We obtained 36 responses out of 40 (total of study population column in Table 42). Considering our population size (40) and sample size (36) with confidence level of 95%, as usual recommended, we have 5% of margin of error²⁹. The margin of error is a percentage that describes how closely the answer our sample gave is the “true value” in our population. In addition, we had a percentage of 90% as response rate, which was considered a reliable level.

5.3.3.1. Descriptive Data

We start our analysis with some descriptive data. Figure 63 shows the profile of the SPCM models consultants where 27 (75%) have worked as consultant in the enterprises; 5 have worked as consultant and they are employees in the enterprises; and 4 are employees of the enterprises.

²⁹ <https://www.surveymonkey.com/mp/margin-of-error-calculator/>

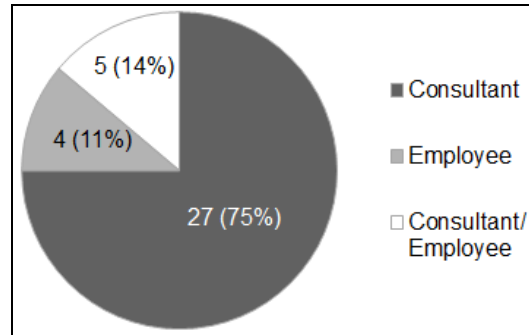


Figure 63. Consultant profiles

About their formation (Figure 64), most of them (50% - 18) have master degree (Master in Science - MSc), and approximately 14% (5/36) have a Master on Business Administration (MBA) formation.

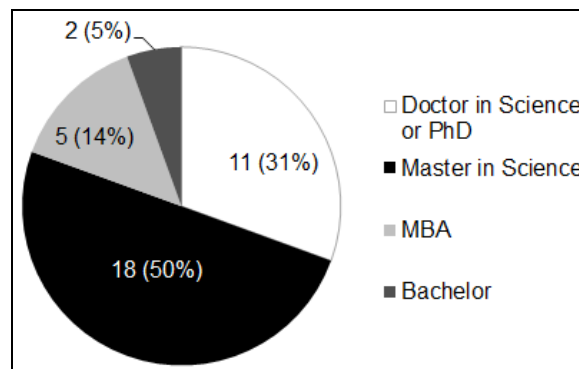


Figure 64. Formation degree

Figure 65 shows the different categories from formation area. We have 22 SPCM models consultants (61%) who's their domain is Software Engineering. Several of them (5) declared also other formation (e.g. civil engineering, electrical engineering, computer engineering, business administration, and mathematics) however they have worked as software engineers. Two of them correspond to MBA formation, 8 have a doctorate, and 12 correspond to Master in Science.

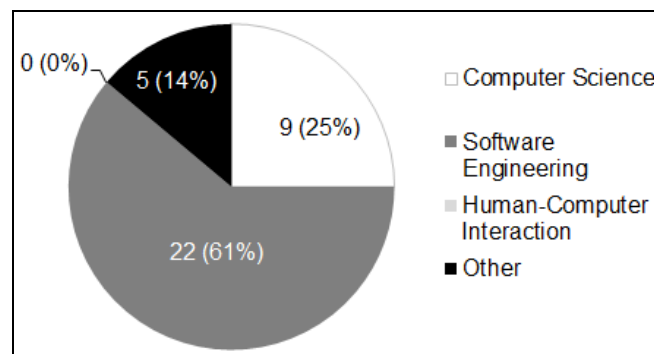


Figure 65. SPCM models consultants' formation area

The distribution regarding the work time (in years) with SPCM models implementation is presented in Figure 66. The work time with implementation models was between 5 and 25 years, and the mean was 12.85. The majority of SPCM models consultants were placed between 10 and 16 years of work time.

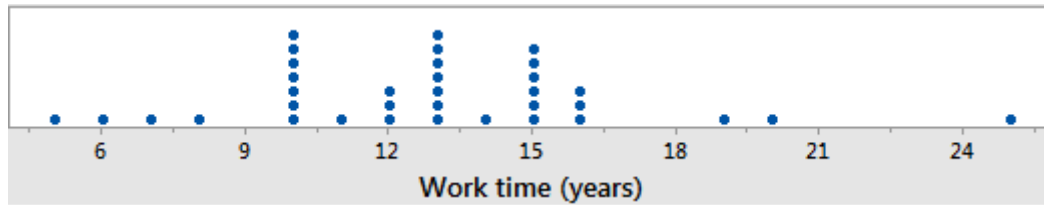


Figure 66. Distribution of work time

About the capability maturity models and levels (Table 43) supported in the implementations we can note that: (i) 9 of the SPCM models consultants have supported implementations in CMMI-DEV level 3 and MR-MPS-SW level C; (ii) 8 declared that they have supported implementations in CMMI-DEV level 3 and MR-MPS-SW levels C, D.

Table 43. SPCM Models and maturity levels

Models and levels	#
CMMI-DEV 5-4-3	1
CMMI-DEV 5-4-3 and MR-MPS-SW A-B-C-D	3
CMMI-DEV 5-4-3 and MR-MPS-SW B-C-D	1
CMMI-DEV 5-4-3 and MR-MPS-SW C-D	3
CMMI-DEV 5-3 and MR-MPS-SW A-C	2
CMMI-DEV 5-3 and MR-MPS-SW C-D	2
CMMI-DEV 5-3 and MR-MPS-SW C	1
CMMI-DEV 5 and MR-MPS-SW A	1
CMMI-DEV 3 and MR-MPS-SW C	9
CMMI-DEV 3 and MR-MPS-SW C-D	8
CMMI-DEV 3 and MR-MPS-SW D	2
MR-MPS-SW C	3

One consultant respondent said that to have implemented only CMMI-DEV levels 5, 4, 3, and three consultants said that to have implemented only MR-MPS-SW level C. In general, MR-MPS-SW level C and CMMI level 3 are the levels the most implemented in the organizations.

The SPCM model consultants declared the quantity of enterprises and projects in which they have performed CMMI-DEV and/or MR-MPS-SW implementations. The collected data correspond to the implementations performed in 671 enterprises regarding to 1405 projects.

5.3.3.2. Answering hypotheses

Before testing the hypothesis of the study (defined in section 5.2.2) we verified if there are significant differences among the **demographic data** (profile of SPCM models consultants, formation of SPCM models consultants, and formation area of SPCM models consultants) and the **variables of the study** (KHCI = Know of HCI, KSE = Know of SE, UHCI = Use of HCI, and USE = Use of SE). To that end we combined variance analysis test (ANOVA) with the F-test. The results (mean (μ), standard deviation (SD), F value (F), and p-value (p) of ANOVA with F-test were presented in Table 44. To execute the test we computed the mean from all questions/categories (1 to 14 – see Annex B) for each variable (KHCI, KSE, UHCI, and USE) and for each respondent (36 answers).

All statistical analysis presented in this thesis was supported by the use of Minitab³⁰ tool version 17, 2016/17.

Table 44. Mean of the demographic data per variable

Profile categories	N	KHCI			KSE			UHCI			USE		
		μ	SD	F 0,51 <i>p</i> 0,605	μ	SD	F 0,77 <i>p</i> 0,470	μ	SD	F 0,36 <i>p</i> 0,699	μ	SD	F 0,45 <i>p</i> 0,640
Consultant	27	4,64	2,69		7,27	2,01		3,28	2,04		5,88	1,87	
Consultant/ Employee	5	5,17	3,69		8,39	1,33		3,54	3,47		6,68	1,98	
Employee	4	3,29	3,33		7,70	1,32		2,31	2,78		6,42	2,02	
Formation categories	N	KHCI			KSE			UHCI			USE		
		μ	SD	F 0,27 <i>p</i> 0,849	μ	SD	F 1,42 <i>p</i> 0,255	μ	SD	F 0,41 <i>p</i> 0,749	μ	SD	F 1,05 <i>p</i> 0,382
Bachelor	2	3,47	0,90		4,86	0,70		2,68	1,38		3,90	0,47	
Doctor in Science	11	4,43	3,09		7,63	2,12		3,20	2,77		6,26	2,44	
Master in Science	18	4,50	3,19		7,64	1,81		2,98	2,31		6,03	1,60	
MBA	5	5,49	1,36		7,59	1,35		4,24	1,33		6,56	1,40	
Formation area categories	N	KHCI			KSE			UHCI			USE		
		μ	SD	F 1,15 <i>p</i> 0,328	μ	SD	F 1,92 <i>p</i> 0,162	μ	SD	F 0,41 <i>p</i> 0,669	μ	SD	F 0,33 <i>p</i> 0,723
CS	9	5,48	3,38		7,92	1,90		3,71	2,91		6,24	2,09	
Other	5	3,07	1,70		6,02	1,89		2,57	1,90		5,42	2,21	
SE	22	4,52	2,78		7,62	1,79		3,14	2,14		6,12	1,76	

N: the quantity of answers for each variable

In the combined analysis of the demographic data, using the variance analysis test (ANOVA) with the F-test, no significant differences were identified to any pair of the investigated variables, as follows:

- between **profile of SPCM models consultants** and **KHCI** ($F = 0.51$ with $p = 0.605$); **profile of SPCM models consultants** and **KSE** ($F = 0.77$ with $p = 0.470$);
- between **profile of SPCM models consultants** and **UHCI** ($F = 0.36$ with $p = 0.699$); **profile of SPCM models consultants** and **USE** ($F = 0.45$ with $p = 0.640$);
- between **formation** and **KHCI** ($F = 0.27$ with $p = 0.849$); **formation** and **KSE** ($F = 1.42$ with $p = 0.255$);
- between **formation** and **UHCI** ($F = 0.41$ with $p = 0.749$); **formation** and **USE** ($F = 1.05$ with $p = 0.382$);
- between **formation area** and **KHCI** ($F = 1.15$ with $p = 0.328$); **formation area** and **KSE** ($F = 1.92$ with $p = 0.162$);
- between **formation area** and **UHCI** ($F = 0.41$ with $p = 0.669$); **formation area** and **USE** ($F = 0.33$ with $p = 0.723$).

After that, to answer our hypotheses (H1 and H2) we execute some steps:

³⁰ <http://www.minitab.com/en-us/products/minitab/>

- (i) we calculate the mean (see Table 45) for each question/category (1 to 14 – see Annex B) of each variable (KHCI, KSE, UHCI, and USE) for the 36 answers collected;
- (ii) we execute two tests to verify if the data follow a normal distribution and if the data are homoscedastic (required to apply the paired t-test);
- (iii) we run the paired t-test considering the mean values (obtained in (i)) to test each hypothesis globally for all categories; and,
- (iv) we run the paired t-test considering the values for each category to test each hypothesis for the 36 answers of each category.

We execute the two tests (step (ii)) and we identify that the data (of each variable) follow a normal distribution and they also are homoscedastic. Thus, knowing that the data have a normal distribution and they are homoscedastic, we executed paired t-test (step (iii)) to analyze the mean difference between the paired observations (**Know of HCI** and **Know of SE**; **Use of HCI** and **Use of SE**) to test the two first hypotheses (H1 and H2). Table 46 presents the results of the test.

Table 45. Mean of the questions per variable

#	HCI categories	Mean KHCI	Mean UHCI	SE categories	Mean KSE	Mean USE
1	Techniques to identify user needs	7,3	6,2	Techniques to identify needs	8,9	8
2	Techniques to identify user and organizational requirements	6,7	5,2	Techniques to identify requirements	8,94	7,6
3	Task Modeling	3,6	1,9	Software Modeling	7,5	5,8
4	Standards and Guidelines for HCI design	4,2	2,32	Standards and Guidelines for design	6,1	4,3
5	Prototype for HCI requirements	5,7	3,9	Prototype for requirements	7,2	5,6
6	Techniques to validate HCI requirements	3,2	1,8	Techniques to validate requirements	7,3	5,5
7	Architecture patterns for HCI	3,8	2,7	Architecture Patterns for SE	7,6	6,3
8	Design patterns for HCI	2,7	1,6	Design Patterns for SE	5,5	3,8
9	Techniques for interaction modeling	1,4	0,8	Interaction modeling for SE	8	6,4
10	Techniques for HCI documentation	3,6	2,28	Techniques for final documentation	4,6	3
11	Iterative and Evolutionary Prototypes (system versions)	4,8	3,8	Prototype (system versions)	6,2	5,4
12	Evaluation methods for HCI verification	7,2	6,1	Verification methods	9,11	8,2
13	Evaluation methods for HCI review	4,1	2,2	Review methods	8,8	7,2
14	Evaluation methods for HCI validation	5,6	4	Validation methods	9,07	7,9
	Mean for all HCI categories	4,56	3,20	Mean for all SE categories	7,48	6,07

Table 46. Results of paired t-test

	N	Mean	Standard deviation	Standard error of the mean	T-value	p-value
KHCI	14	4,56	1,74	0,46	-6.91	< 0.0001
KSE	14	7,48	1,44	0,38		
<i>Mean difference of Know</i>	14	-2,92	1,58	0,42		
UHCI	14	3,20	1,70	0,45	-7.57	< 0.0001
USE	14	6,07	1,62	0,43		
<i>Mean difference of Use</i>	14	-2,87	1,41	0,37		

N: the quantity of answers for each variable

For this step we used the data from Table 45 (the 14 means of each variable), where in the paired t-test **KHCI** minus **KSE** and **UHCI** minus **USE** items were computed. As previously mentioned we considered $\alpha = 0.05$ which allows us to build a confidence interval of 0.95.

The results of the paired t-test (see Table 46) allow us to refute the null hypotheses (H_{10} : **KHCI** - **KSE** = 0; and H_{20} : **UHCI** - **USE** = 0) since $p < 0.0001$ which is less than $\alpha = 0.05$. We accept therefore, the alternative hypotheses H_{1A} : **KHCI** - **KSE** < 0; and H_{2A} : **UHCI** - **USE** < 0).

For the step (iv), we performed the same analysis for each item of the questionnaire (i.e., the categories). Table 47 shows the results of the paired t-test for the 14 items of each variable using the 36 responses. The results allow us to refute the null hypotheses (H_{10} and H_{20}) and to accept the alternative hypotheses (H_{1A} and H_{2A}) for all items except for the item 10 (Use); for which null hypothesis could not be rejected because the T-value (-1.14) > that the critical value (-1.697) and p (0.131) > α (0.05).

Table 47. Results of paired t-test for each item

Item	Know (KHCI – KSE) 36 responses			Item	Use (UHCI – USE) 36 responses		
	Mean	Standard deviation	T-value		Mean	Standard deviation	T-value
1	-1,556	3,341	-2,79***	1	-1,792	3,360	-3,20***
2	-2,239	3,033	-4,43***	2	-2,356	3,110	-4,54***
3	-3,867	2,751	-8,43***	3	-3,817	2,885	-7,94***
4	-1,919	2,573	-4,48***	4	-1,989	2,653	-4,50***
5	-1,447	3,095	-2,81***	5	-1,647	3,212	-3,08***
6	-4,103	3,356	-7,34***	6	-3,692	3,605	-6,14***
7	-3,711	3,100	-7,18***	7	-3,600	3,178	-6,80***
8	-2,839	3,066	-5,56***	8	-2,186	2,754	-4,76***
9	-6,600	3,126	-12,67***	9	-5,594	3,869	-8,68***
10	-1,022	3,414	-1,80**	10	-0,706	3,705	-1,14*
11	-1,461	2,774	-3,16***	11	-1,536	2,738	-3,37***
12	-1,917	3,768	-3,05***	12	-2,131	3,691	-3,46***
13	-4,633	3,639	-7,64***	13	-4,947	3,218	-9,22***
14	-3,500	4,048	-5,19***	14	-3,869	3,933	-5,90***

* $p = 0.131$; ** $p = 0.041$; *** $0.000 < p \leq 0.004$.

5.3.4. Discussion related to the literature

The most important finding of our study is the confirmation of the hypotheses H1 and H2. In this section we present relevant aspects of this finding.

❖ Perception of knowledge of HCI and SE approaches

The **hypothesis H1** was confirmed showing that the SPCM consultants do not know HCI approaches as know SE approaches when implementing the same specific practices of the SPCM models.

Analyzing Table 45, we note that the category **techniques to identify user needs** (category 1) presented the highest level of perception of knowledge for HCI (mean = 7.3). The mean of all the

categories for perception of knowledge of HCI was 4.56, which is a bit lower than the central point (5) of our scale. However, when we analyze the means of each category for **KHCI**, we note that only five categories have the mean greater than the central point (5) of our scale. The categories are presented in descending order in Table 48.

Table 48. HCI and SE rank for “knowledge”

Rank	Categories rank for HCI	Mean KHCI	Categories rank for SE	Mean KSE
1	Techniques to identify user needs	7,3	Verification methods	9,11
2	Evaluation methods for HCI verification	7,2	Validation methods	9,07
3	Techniques to identify user and organizational requirements	6,7	Techniques to identify requirements	8,94
4	Prototype for HCI requirements	5,7	Techniques to identify needs	8,9
5	Evaluation methods for HCI validation	5,6	Review methods	8,8

When we made the same analysis for each category of **KSE** we note that almost all categories (except the category 10 - techniques for final documentation) have the mean greater than the central point (5) of our scale. We show in Table 48 the categories that presented the five highest means. We note that:

- For the HCI categories (techniques to identify user needs, evaluation methods for HCI verification, and techniques to identify user and organizational requirements) the suggested methods were completely or partially the same to the ones proposed for its correspondent SE category (e.g., Brainstorming, Unit test, Scenario, respectively). We believe that this fact can explain the similarity between the rank of the categories **techniques to identify user and organizational requirements**, third position for HCI and **techniques to identify requirements**, third position for SE;
- for the SE category **validation methods** the suggested methods for SE imply the participation of the users or end users (such as acceptance test with users and tests of products (by end user/stakeholders));
- for the SE category **techniques to identify needs** the suggested methods were completely or partially the same for the HCI category **techniques to identify user needs**;
- the categories ranked (for both domains) in the fifth first positions are associated with the practices of the following process areas from the SPCM models: requirements development, verification and validation.

The categories **techniques for interaction modeling** (category 9 for HCI, mean = 1.4) and the category **techniques for final documentation** (category 10 for SE, mean = 4.6) are the categories with the lowest means regarding to the perception of knowledge.

We believe that the result found for the category **techniques for interaction modeling** is due to the fact that the HCI community is relatively young (Prates (2007); Souza, Baranauskas, Prates, & Pimenta (2008); Prates et al. (2013)) and the research for this type of technique is relatively new in

Brazil (Barbosa & Paula, 2003), and it is possible that the approaches are not yet being used in practice. A recently published paper presents initial plans to approximate industry and academy through the evangelization of HCI research and practice in Brazil, because HCI is not yet widely known and has not yet been formally adopted in the industry (Furtado et al., 2016). This approximation between industry and academy is also discussed by Scheiber et al. (2012) that argues that the universities were considered as important sources of knowledge regarding HCI by software producers.

❖ Use of HCI and SE approaches

The **hypothesis H2** was confirmed showing that the SPCM consultants do not use HCI approaches as use SE approaches when implementing the same specific practices of the SPCM models.

The category **techniques to identify user needs** presented the highest level of use of HCI (mean = 6.2). The global mean of all categories for use of HCI was 3.20 that is low than the central point (5) of our scale. When we analyzed the means of each category for **UHCI**, we noted that only three categories have the mean greater than the central point (5) of our scale. The categories are presented in descending order in Table 49:

Table 49. HCI and SE rank for “use”

Rank	Categories rank for HCI	Mean UHCI	Categories rank for SE	Mean USE
1	Techniques to identify user needs	6,2	Verification methods	8,2
2	Evaluation methods for HCI verification	6,1	Techniques to identify needs	8,0
3	Techniques to identify user and organizational requirements	5,2	Validation methods	7,9

When we made the same analysis for each category of **USE** we note that almost all categories (except for two of the SE categories: standards and guidelines for design, design patterns and techniques for final documentation) have the mean greater than the central point (5) of our scale. We show in Table 49 the categories that presented the three highest means.

The categories that presented the three first positions for **use** are also the categories that are placed in the three first positions for **perception of knowledge**, in both domains. As previously explained the suggested methods for these categories (HCI and SE) are completely or partially the same.

Similarly to perception of knowledge, the categories **techniques for interaction modeling** (category 9 for HCI, mean = 0.8) and the category **techniques for final documentation** (category 10 for SE, mean = 3.0) are the categories with the lowest means regarding to the **use**.

As previously presented in Chapter 2, Vukelja et al. (2007) asked the participants (134) about the use of software engineering methods in the software development. Regarding **modules and systems test**, they found that: (i) the modules are tested in 76.2% of cases; and (ii) the systems in 98.1% of cases. When we compare these results with our results, we found that the SE category **verification methods** was the category most used by the consultants (mean = 8.2). They also said that the documentation for the end user is written in 34.2% of cases in parallel with the development and at the end; unfortunately in 65.8% of cases only at the end. When we compare these results with ours, we found

that the SE category **techniques for final documentation** was the least used category by the consultants (mean = 3.0). Regarding usability tests, they found that only in 37.9% of the cases this type of test is conducted. When we compare these results with our, we found that the SE category **validation methods** was classified in third position (see Table 49) of our rank. **Acceptance test with users** approach is placed in this category.

Finally, we analyze the HCI approaches against the results presented in Chapter 2 (section 2.4.2, Table 15) to analyze the top 10 techniques/methods found in those works against our findings (see Table 50). To perform this analysis we considered the HCI category ranking (presented in Table 50) and we looked for the examples of approaches in that category that are the same than those presented in the studies (we indicate the HCI approach we considered in this analysis as example in the last column of Table 50).

Analyzing the rank of the HCI techniques/methods (Table 50) used in practice by UI/UX/UCD/usability practitioners, we found different results from some studies (UXPA Curitiba (2016); Ji & Yun (2006); Venturi et al. (2006)).

UXPA Curitiba (2016) and Venturi et al. (2006) shared three techniques that had the same classification (hi-fi prototypes, heuristic evaluation, and usability test – classified as second, fourth and fifth techniques in Table 50). In the other side, Ji & Yun (2006) and Venturi et al. (2006) did not share techniques with the same classification rank.

Similarly, we analyzed the rank of the HCI techniques/methods (Table 50) used in practice cited by three studies (Salgado et al. (2016); Hussein et al. (2012); Ji & Yun (2006)) and that were performed with software developers. In addition, we compared this analysis with our study.

Salgado et al. (2016) and us shared two techniques that had the same classification (see the techniques classified as first and tenth in Table 50 – these techniques correspond to requirements development and verification process areas, respectively). Hussein et al. (2012) and us shared two techniques (techniques placed in Table 50 as second and fourth position – these techniques correspond to verification and validation process areas, respectively). Finally, Ji & Yun (2006) and us shared two techniques with the same rank classification (see the techniques classified as third and sixth in Table 50 – the third was associated to requirements development process area, and the sixth to technical solution, product integration, verification and validation process areas).

The HCI categories (techniques/methods) for **use** classified by our study in the first five positions are:

- (i) *techniques to identify user needs;*
- (ii) *evaluation methods for HCI verification;*
- (iii) *techniques to identify user and organizational requirements;*
- (iv) *evaluation methods for HCI validation; and,*
- (v) *prototype for HCI requirements.*

The first, third and fifth categories were associated to requirements development process area; the second and fourth categories correspond to verification and validation process areas, respectively. These results were corroborated by some studies cited in this work that discussed the frequently use of HCI methods for validation (Ji & Yun (2006); Venturi et al. (2006); Hussein et al. (2012); Salgado et

al. (2016); UXPA Curitiba (2016)) and verification (Venturi et al. (2006); Hussein et al. (2012); UXPA Curitiba (2016)), and also the use of *techniques to identify user needs* in the initial phases of the software development (Ji & Yun (2006); Venturi et al. (2006); Hussein et al. (2012); Salgado et al. (2016); UXPA Curitiba (2016)).

Table 50. Rank of the HCI techniques/methods used in the practice

Rank	Salgado et al. (2016)	UXPA Curitiba (2016)	Hussein et al. (2012)	Ji and Yun (2006)		Venturi et al. (2006)	Gonçalves et al.
	<i>Subject: software developer</i>	<i>Subject: usability/UX professionals</i>	<i>Subject: software developer</i>	<i>Subject: development practitioners</i>	<i>Subject: UI/usability practitioners</i>	<i>Subject: UCD practitioners</i>	<i>Subject: SPCM models consultants (they have the profile of a software developer).</i>
1	User interviews	Lo-fi Prototypes	Task analysis	Task analysis	Task analysis	User interviews	Techniques to identify user needs - <i>e.g.</i> : Interviews
2	Lo-fi prototypes	Hi-fi prototypes	User Acceptance Test	Evaluate existing system	Evaluate existing system	Hi-fi prototyping	Evaluation methods for HCI verification - <i>e.g.</i> : acceptance test
3	Hi-fi prototypes	User interviews	User experience	User analysis/profiling	User analysis/profiling	Lo-fi prototyping	Techniques to identify user and organizational requirements - <i>e.g.</i> : user Profile (detailed)
4	Contextual analysis	Heuristic evaluation	Evaluate existing system	Surveys	Surveys	Expert or heuristic evaluation	Evaluation methods for HCI validation
5	Usability tests	Usability test	Surveys	Scenarios of use	Scenarios of use	Qualitative, quick and dirty usability test	Prototype for HCI requirements
6	Heuristic evaluation	Personas	Heuristics evaluation, usability expert evaluation	Screen mock-up test	Heuristics evaluation, usability expert evaluation	Observation of real usage	Iterative and Evolutionary Prototypes (system versions)
7	Personas	Survey	Scenarios of use	Navigation design	Navigation design	Scenarios	Architecture patterns for HCI
8	Survey	Contextual analysis	User analysis/profiling	Usability checklists	Usability checklists	Style guides	Standards and Guidelines for HCI design
9	Remote usability tests	UX Training	Lab usability testing	Participatory design	Focus group interview	Early human factors analysis	Techniques for HCI documentation
10	Guidelines/ Checklist review	Card sorting	Navigation design	Lab usability testing	Lab usability testing	Competitive analysis	Evaluation methods for HCI review

5.4. Empirical study in the international context

In this section we will present the instrument (web questionnaire), the subjects, the execution and the first results of a study (Gonçalves, Oliveira, & Kolski, 2017a) that has been performed with worldwide CMMI-DEV model consultants (consultants from ten different countries).

5.4.1. Instrument: a web questionnaire

We adapted the questionnaire (Annex E) developed for the first study described in section 5.3. The first part was developed to collect the demographic data. The first four data fields (respondent identification, e-mail, formation degree, formation area) were designed to identify the respondent. After that, the CMMI-DEV model consultants answered five questions as following described:

- Are you affiliated to a CMMI Partner Organization?
 - Which one? (optional)
 - What is your country of operation?
- Did you take the official CMMI introduction course?
- How many years have you worked in Capability Maturity models implementations?
- What is/are the maturity level(s) that you have supported in Capability Maturity model implementations?
- Approximately, how many enterprises and projects have you supported the implementation? (for the level(s) previously selected)

The second part was composed of questions about HCI and SE approaches related to each practice only of CMMI-DEV once we were in the international context.

Figure 67 presents a screenshot of the main page (a) and part of the questions about HCI and SE approaches (b).

5.4.2. Subjects and planning

The web questionnaire was applied with the CMMI-DEV model consultants of the partner enterprises which implementing the CMMI-DEV in the industry. The subjects are CMMI-DEV model consultants of the partner enterprises to CMMI Institute's database³¹. This database presents the data of 281 partner enterprises that implement and evaluate CMMI-DEV from different countries. To reduce the scope, we decided to select the ten top countries (CMMI Product Team, 2015) in terms of official CMMI appraisals in the last years.

Table 51 presents these countries that totalize 207 partner enterprises, our population for this study. Only CMMI-DEV model consultants that had implemented CMMI-DEV maturity level equal or greater than 3 could participate in this study since our interest is the engineering process areas. In addition, we selected only partner enterprises that indicate in the site of the CMMI institute English as a langue of communication since our questionnaire was in English.

³¹ <http://partners.cmmiinstitute.com/find-partner-organization/>



[HOME](#)
[PART 1](#)
[PART 2](#)

Survey - Implementation of methods, techniques, standards and patterns of Human-Computer Interaction and Software Engineering

This survey aims to evaluate to what extent methods, techniques, standards and patterns of Software Engineering and Human-Computer Interaction are used by software developers that have implemented the maturity levels (5, 4 or 3) of the **CMMI-DEV** model (Capability Maturity Model Integration for Development).

We would like to highlight that any publication generated from this survey will present only statistical results by summarizing the raw data and treating the answers anonymously. In other words, in any circumstances the answers provided in this survey will be published with personal information of the respondents or the institutions for which they work.

This research is part of a doctoral thesis which is being developed at University of Valenciennes and Hainaut-Cambresis and financed by the Brazilian government (Program Science without Borders/CAPES).

The survey is divided into 2 parts (described below) and the estimated time to fill it is **40 minutes**.

- **Part 1 - Characterization**
- **Part 2 - Evaluation of the Implementation of methods, techniques, standards and patterns of Human-Computer Interaction and Software Engineering**

We really appreciate your help and time with this research.

Best regards,

Taisa Guidini Gonçalves
Káthia Marçal de Oliveira
Christophe Kolski

[NEXT PAGE](#)

(a)



[HOME](#)
[PART 1](#)
[PART 2](#)

Survey - Implementation of methods, techniques, standards and patterns of Human-Computer Interaction and Software Engineering

Part 2 - Evaluation of the Implementation of methods, techniques, standards and patterns of Human-Computer Interaction and Software Engineering

The item listed above present several methods, techniques, standards and patterns from Software Engineering (SE) and Human-Computer Interaction (HCI) that can support the implementation of the Process Areas of CMMI-DEV, according to the literature and experts.

Please, indicate your level of knowledge (*I know*) and level of use (*I used*) to each one of those methods, techniques, standards and patterns when in the implementation of Capability Maturity models in enterprises you worked.

#	Process Area (CMMI-DEV)	Potential methods, techniques, standards and patterns from Software Engineering (SE)	Potential methods, techniques, standards and patterns from Human-Computer Interaction (HCI)
1	Requirements Development <i>RD SP1.1</i> Elicit Needs	<p>Techniques to identify needs</p> <p>Examples (see References), not limited to:</p> <ul style="list-style-type: none"> Brainstorming Interviews Questionnaires Card Sorting Focus Groups Field Studies/Observation Workshops Protocol Analysis <p>Answers</p> <p>I Know:</p> <p>None <input type="text"/> A lot</p> <p>I Used:</p> <p>None <input type="text"/> A lot</p>	<p>Techniques to identify user needs</p> <p>Examples (see References), not limited to:</p> <ul style="list-style-type: none"> Brainstorming Interviews Surveys/Questionnaires Card Sorting Focus Groups Field Studies/Observation <p>Answers</p> <p>I Know:</p> <p>None <input type="text"/> A lot</p> <p>I Used:</p> <p>None <input type="text"/> A lot</p>

(b)

Figure 67. Web questionnaire

Table 51. Partner enterprises to CMMI Institute

Country	Population (partner enterprises)
Brazil	4
China	18
France	6
India	23
Japan	5
Mexico	7
Republic of Korea	7
Spain	7
United Kingdom	10
United States	119
Total	207

The study was planned to be conducted off-line. The questionnaire was available in a web site in a way that the consultants answered the questionnaire in his/her time and environment.

5.4.3. Study execution and analysis of the results

The survey request was sent by email on 25th Nov 2016 with six reminders till 23th May 2017. Up to now we obtained 21 responses out of 207 (potential population). Considering the confidence level of 95%, as usually recommended, we have 20% of margin of error³² considering this sample size. As consequence, we cannot analyze statically the data and the hypothesis of the study. So, we performed only a descriptive analysis.

5.4.3.1. Descriptive data

About the profile of consultants, Figure 68 shows that 15 (71%) have worked as consultant in the enterprises; 4 are employees; and 2 have worked as consultant and they are employees.

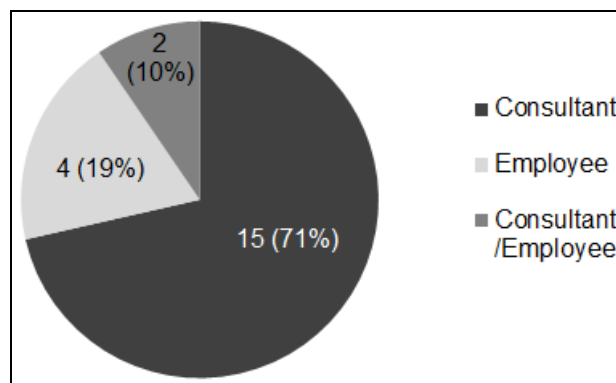


Figure 68. Profile of the consultants

For the formation of the consultants (see Figure 69), 33% (7/21) have Master of Business Administration (MBA) formation, 29% (6/21) have master degree, 24% (5/21) have bachelor and 14% (3/21) have doctorate.

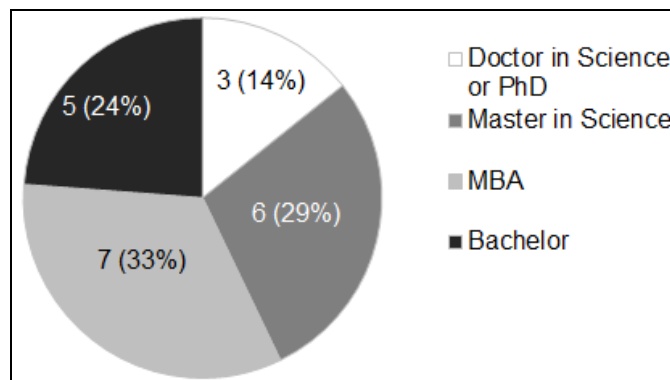


Figure 69. Formation of the consultants

Figure 70 shows the different categories about the formation area of the consultants. We have 9 consultants (43%) whose domain is Software Engineering. Two of them correspond to MBA formation, 8 have a doctorate, and 12 correspond to Master in Science.

³² <https://www.surveymonkey.com/mp/margin-of-error-calculator/>

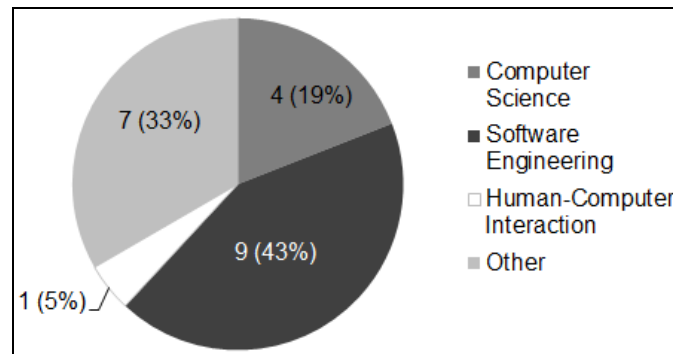


Figure 70. Formation area of the consultants

Regarding the consultants per country, the Figure 71 shows that: most of them (6/21) are of the United States; Four consultants are from Brazil; and 2 from France.

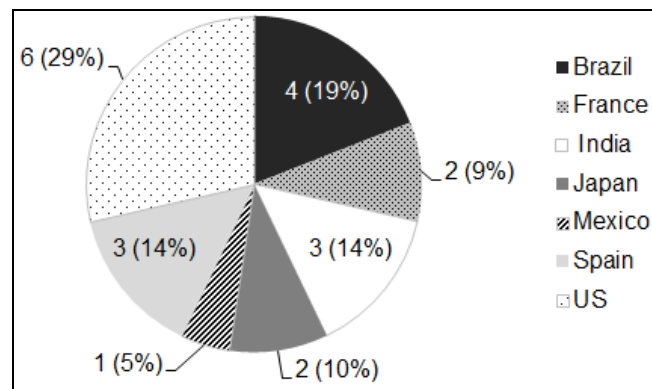


Figure 71. Consultants per country

The distribution about the work time (in years) with implementation models is presented in Figure 72. The work time with implementation models was between 9 and 28 years, and the mean was 18.04. The majority of consultants are placed between 15 and 35 years of work time.

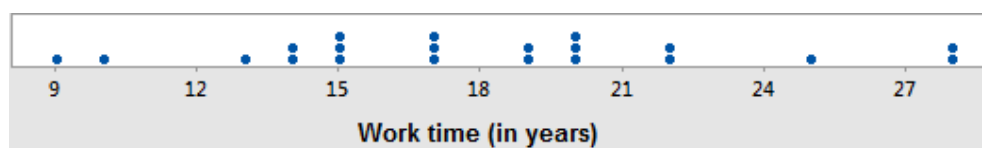


Figure 72. Distribution of work time with implementations

About the capability maturity level(s) supported in the implementations (see Figure 73), 57% of the consultants have supported implementations in CMMI-DEV level 5, 4 and 3, 14% have supported implementations in level 5 and 3, and 29% declared that they have supported only implementations in level 3.

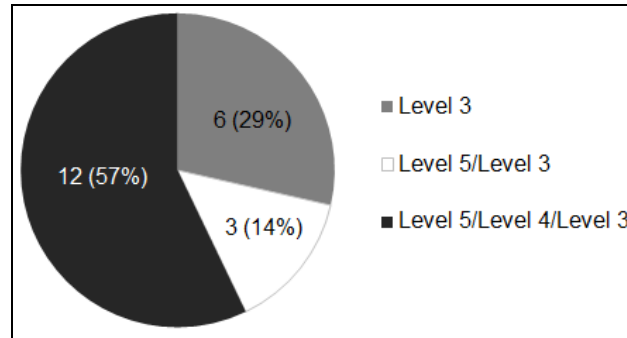


Figure 73. Maturity levels

5.4.3.2. Analysis of the results

As previously presented the consultants answered the questions about their **perception of knowledge** and **use** regarding to Human-Computer Interaction and Software Engineering categories using a Visual Analogue Scale (VAS) – “None” (0) to “A lot” (10). For this, they considered their experience in different enterprises implementing CMMI-DEV model in several projects. The SPCM model consultants declared the quantity of enterprises and projects in which they have performed the implementation of CMMI-DEV model. The collected data correspond to the implementations performed in 1228 enterprises regarding to 5025 projects.

For the collected data (for **perception of knowledge** and **use**), we calculated the mean of the 21 answers to each category and to each domain (HCI and SE) (see Table 52).

Table 52. Means for the data

#	Human-Computer Interaction			Software Engineering		
	Categories	Mean for Knowledge	Mean for Use	Categories	Mean for Knowledge	Mean for Use
1	Techniques to identify user needs	8,7	6,9	Techniques to identify needs	9	7,3
2	Techniques to identify user and organizational requirements	7,5	6,1	Techniques to identify requirements	8,5	7,3
3	Task Modeling	4,8	4	Software Modeling	6,7	5,5
4	Standards and Guidelines for HCI design	3,5	2,2	Standards and Guidelines for design	4,8	3,3
5	Prototype for HCI requirements	6,2	6,1	Prototype for requirements	6,4	5,1
6	Techniques to validate HCI requirements	5	5,6	Techniques to validate requirements	7,4	9,4
7	Architecture patterns for HCI	3,1	2,3	Architecture Patterns for SE	5,1	3,9
8	Design patterns for HCI	2,4	1,4	Design Patterns for SE	3,4	2,1
9	Techniques for interaction modeling	2,4	1,6	Interaction modeling for SE	6,5	4,9
10	Techniques for HCI documentation	4,8	4,1	Techniques for final documentation	4,6	3,6
11	Iterative and Evolutionary Prototypes (system versions)	4,5	3,6	Prototype (system versions)	5,3	4,3
12	Evaluation methods for HCI verification	8,6	7,9	Verification methods	8,8	8
13	Evaluation methods for HCI review	5,8	4	Review methods	9,2	8,2
14	Evaluation methods for HCI validation	6,7	5,5	Validation methods	8,8	8

Analyzing the **HCI categories** for the variable **perception of knowledge** we note that the categories 1 (techniques to identify user needs) and 12 (evaluation methods for HCI verification) – see the bars with green color in Figure 74 – presented the highest levels. We believe that this result is justified since the approaches exemplified in these categories are quite similar for SE and HCI.

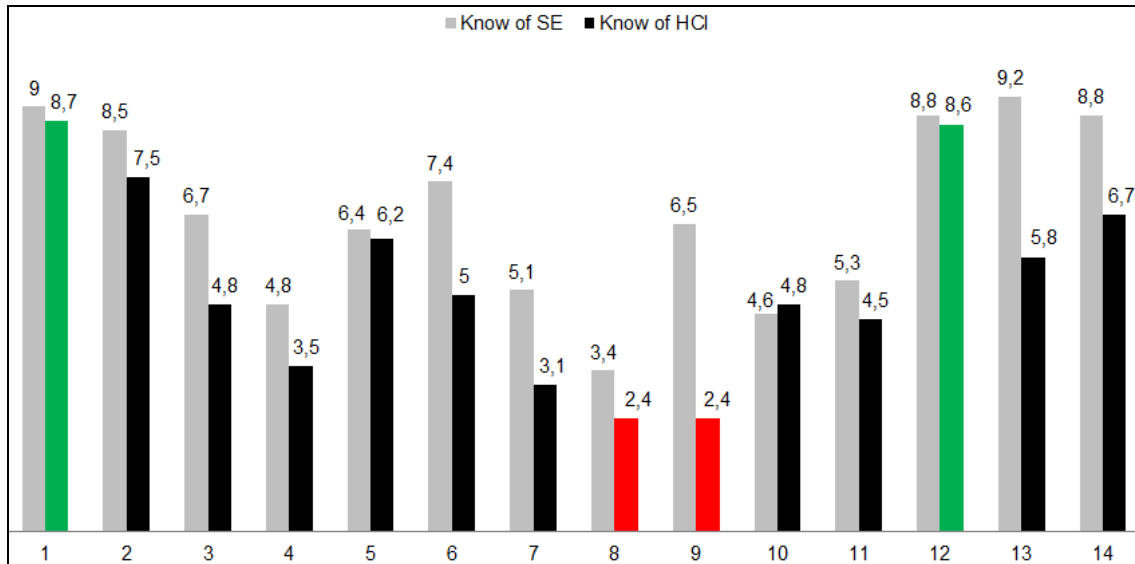


Figure 74. Results by category found for “perception of knowledge”

Regarding the **HCI categories** with the lowest level for **perception of knowledge**, we had the categories 8 (design patterns for HCI) and 9 (techniques for interaction modeling) – see the bars with red color in Figure 74. This result is, probably, due to the fact that the approaches related to **design patterns** and **interaction modeling for HCI** categories were recently proposed.

When we analyze the results for **SE categories**, we note that: (i) the **perception of knowledge** of the SPCM consultants is bigger in relation to the categories 1 (techniques to identify needs) and 13 (review methods); (ii) and their **perception of knowledge** is lower in relation to the categories 8 (design patterns for SE) and 10 (techniques for final documentation). As previously explained, the approaches exemplified in category 1 are quite similar for SE and HCI. The approaches from category 13 are related to review methods that are much known in SE domain.

Looking in more detail the **HCI** the categories with highest values, (i.e., categories 1 and 12 as indicated in Figure 74) by country (Figure 75), we note that **US** is the country with the highest level for “perception of knowledge” (category 1 - techniques to identify user needs) and the **Japan** for category 12 (evaluation methods for HCI verification). We note also that **France** has the second highest level for **perception of knowledge** (category 1), but it has also the lowest level for **perception of knowledge** considering the category 12.

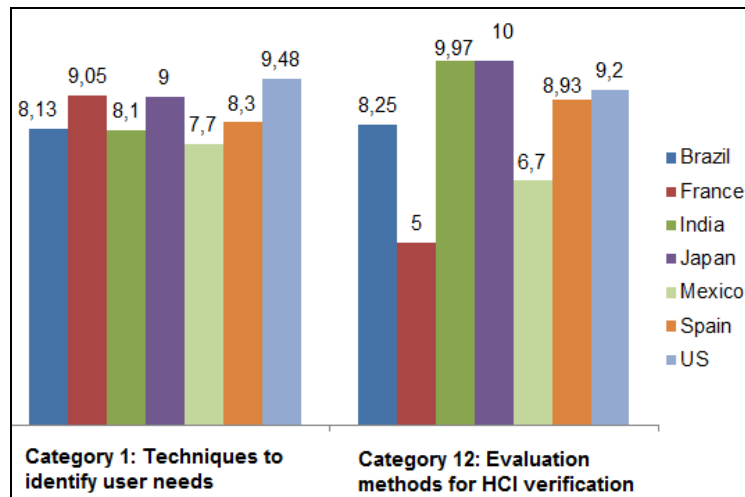


Figure 75. Results to highest levels for “perception of knowledge”

Similarly, looking in more detail the **HCI** categories with lowest values (categories 8 and 9, see Figure 74) by country (Figure 76), we note that **India** (category 8 - design patterns for HCI) and **Spain** (category 9 - techniques for interaction modeling) are the countries with the lowest level for “perception of knowledge”. We note also that **France** presented the second best place in relation to category 8, and the first best place is from **Japan**. The Japan also occupied the first best place for category 9 followed by **India**.

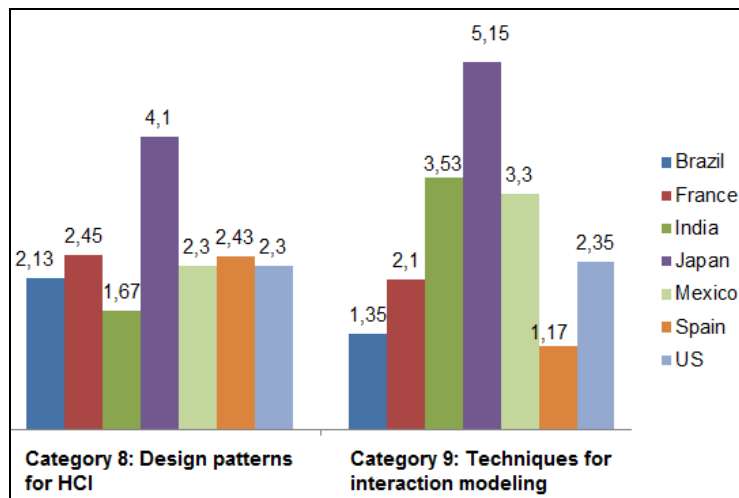


Figure 76. Results to lowest levels for “perception of knowledge”

Analyzing the **HCI categories** for the variable **use** we note that the categories 1 (techniques to identify user needs) and 12 (evaluation methods for HCI verification) – see the bars with green color in Figure 77 – presented the highest levels. The approaches exemplified in these categories are similar for HCI and SE.

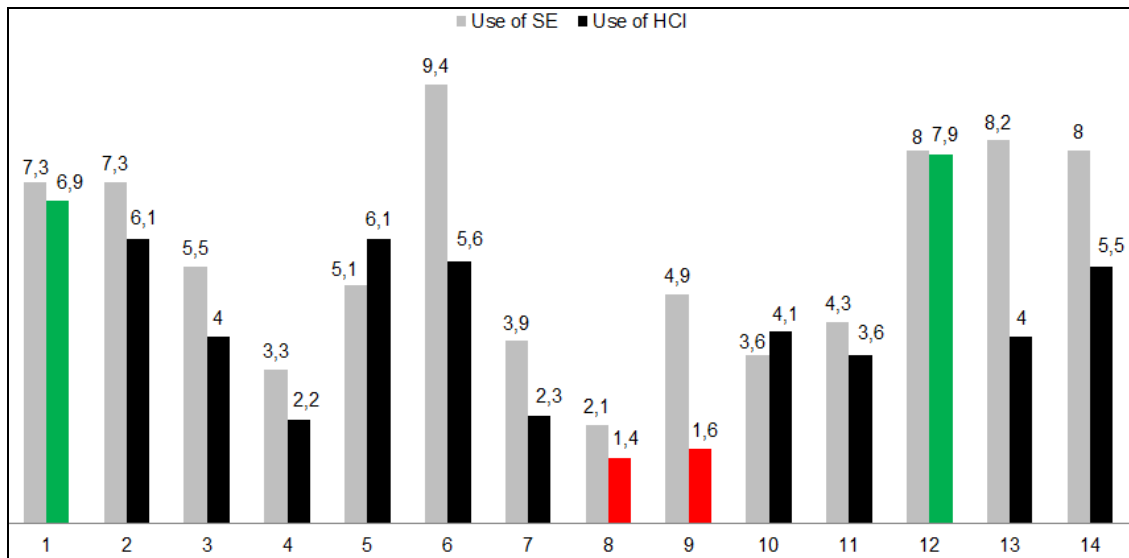


Figure 77. Results by category found for “Use”

Regarding the **HCI categories** with the lowest level for **use**, we had the categories 8 (design patterns for HCI) and 9 (techniques for interaction modeling) – see the bars with red color in Figure 77. As previously explained, the approaches regarding these categories were recently proposed.

When we analyze the results for **SE categories**, we note that: (i) the consultants **use** more the approaches from the categories 6 (techniques to validate requirements) and 13 (review methods); (ii) and they **use** less the approaches from the categories 4 (standards and guidelines for HCI design) and 8 (design patterns for SE). The approaches from category 13 are related to review methods that are much used in SE domain. Similarly, the category 6 presents approaches that are more used in SE domain to validate customer requirements than the approaches from category 14 (validation methods).

Looking in more detail the **HCI categories** 1 and 12 (the highest value) by country (Figure 78) we note that **Japan** is the country with the highest level for **use**. We note also that **France** has the second highest level for **use** (category 1- techniques to identify user needs), but it has also a lowest level for **use** considering the category 12 (evaluation methods for HCI verification).

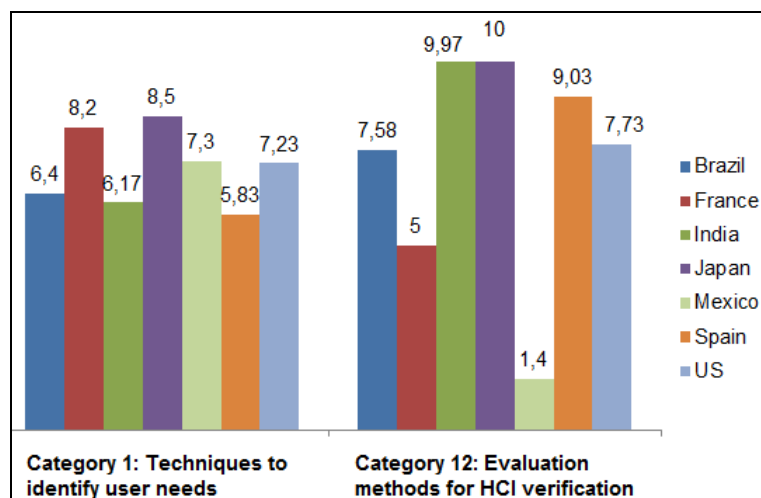


Figure 78. Results to highest levels for “Use”

Now, looking in more detail the **HCI** categories with lowest values (categories 8 and 9, see **Figure 77**) by country (Figure 79), we note that **Spain** is the country with the lowest level for **use**. We note also that **France** has the same lowest level for **use** considering the category 9 (techniques for interaction modeling). We note also that first best place for category 8 (design patterns for HCI) is from **Brazil**, and for category 9 is from **India**.

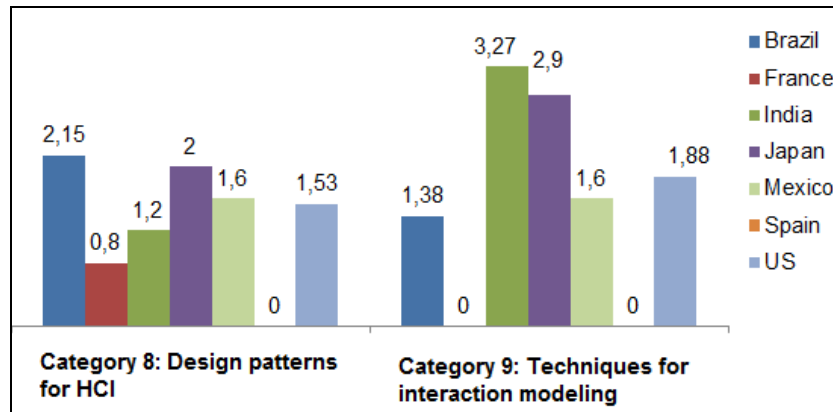


Figure 79. Results to lowest levels for “Use”

The results also show that the **perception of knowledge** and **use** for the category 10 (techniques for final documentation) is higher for HCI than for SE. This category is related to final documentation of the system, and we believe that this result is due to the importance done to final documentation for the end user (e.g., on-line helps, user manual, etc.). In addition, for the category 5 (prototype for HCI requirements) the **level of use** is higher for HCI when compared with SE. This was expected since this category is normally used in HCI domain to validate user requirements and user interfaces.

5.5. Threats of Validity Analysis

We considered the four threats of validity proposed by Wohlin et al. (2012) (construct, internal, external and conclusion validity) trying to define some mitigations as described below.

Threats to the **construct validity** illustrate from the relation between theory and observation and the questions of whether the treatment adequately reflects the cause; whether the result adequately reflects the effects. In our case, whether the items (HCI and SE approaches) to be evaluated adequately reflect the application for the practices of the SPCM models. To minimize this threat the web questionnaire was built using the original text and examples of the official documentation of CMMI-DEV and MR-MPS-SW models; the HCI approaches was evaluated with 20 domain experts after a carrying out ad hoc study of literature; the SE approaches was peer-reviewed by 5 domain experts after a carrying out ad hoc study of literature. Another threat to the construction validity concerns the interpretation of the practices of CMMI-DEV and MR-MPS-SW model for the customization of the instrument. This risk was accepted because we used the equivalence mapping (Softex, 2016b) of the models to build the instrument.

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, this threat is associated with the subjects involved in the study. The subjects were selected by quota sampling, SPCM models consultants that have implemented the maturity levels “A, B, C or D” of the MR-MPS-SW model and/or the maturity levels “5, 4 or 3” of the CMMI-DEV model. The subjects should characterize

their perception of knowledge and use related to the implementation of HCI and SE approaches in different projects. A potential risk is that a person who has not done implementations for the levels mentioned in the study answer the survey. To minimize this risk, we selected SPCM models consultants that are associated to partner enterprises of the CMMI institute and SOFTEX. In addition, we explicitly ask them if they have the required experience and we confirmed with the coordinator of each partner enterprises.

Threats to the **external validity** are conditions that limit our ability to generalize the results of our experiment outside the scope of our study. In the Brazilian context we answered our hypotheses. However, although Brazil is one of the top ten countries that apply CMMI-DEV, according to CMMI Institute, we cannot generalize the results of this study to any country. In the international context, we cannot generalize the results due since the sample did not allow us to validate statistically the hypotheses.

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. Analyzing our hypotheses, we identified the risk that a SPCM model consultant having answered that he/she does not know and/or does not use a category of HCI or SE approach because he/she does not recognize the approaches name. To minimize this threat, we included on the web questionnaire several examples and their bibliographic references. In this way, the SPCM models consultants could consult the list of bibliographic references in case of doubt. Another risk is that the consultants have answered each question regarding only the approaches examples rather than the category, which represent the kind of approaches to support each practice. To mitigate this risk we included the text “not limited to” before present the least of example. Moreover, to analyze this risk we asked the two consultants who participated in the pre-test what was the reasoning used to answer the questions. They answered that they considered the category in general although they have read the list of examples. Therefore, we considered that the risk was potentially weak. We considered that even that consultants could answer based intuitively in an average of their perception of knowledge and use of all examples in the list from a category they were evaluating both HCI and SE in the same way, therefore we could analyze the results of the evaluation one against the other. Therefore we decided to accept this risk. Finally, another threat of validity is about the formation area of the consultants in the conclusions. We are aware that formation (master or PhD) on Human-Computer Interaction is much younger than on software engineering therefore we have the risk that the answers were biased for the original formation of the SPCM models consultants that in majority declared have been formed in software engineering. Considering that usually HCI issues are usually integrated as courses in the masters and PhD on different formations (such as software engineering and computer systems in general), we accepted this risk.

5.6. Synthesis and Conclusion

In this chapter we presented the results of two empirical studies performed in industrial environment. The first one was performed in Brazil with SPCM models consultants (CMMI-DEV and MR-MPS-SW models). The second one was performed with CMMI-DEV model consultants from different countries.

The objective was to investigate the perception of knowledge and use of HCI and SE approaches by the SPCM models consultants in the context of SPCM models implementations. For this, we formalized hypotheses and surveyed SPCM models consultants of these models.

Analyzing the data (36 answers out of 40) for the first study, we concluded that the SPCM models consultants do not know and do not use HCI approaches as they know and use SE approaches when applying the same engineering specific practices of SPCM models.

For the second study (21 answers out of 207), we cannot be conclusive since the quantity of data is not sufficient to do statistics analysis. However, the initial results show that: (i) the categories **techniques to identify user needs** and **evaluation methods for HCI verification** presented the highest level of **perception of knowledge** and **use** of HCI; (ii) the results found in the first study regarding to the same categories found in the second study.

Finally, for now the studies show the need of dissemination of HCI approaches in industry. We believe that one way to do that is augmenting the number of hours of HCI classes in the computer science undergraduate courses.

In the next chapter we will present the contributions and the limitations of this thesis, and the future works.

General Conclusion

The work presented in this thesis contributes to the development of interactive systems, particularly those that are developed guided by software development process that follows software process capability (SPCM) models.

The development of usable and useful interactive systems in any field of application is related to the application of methods, techniques, standards, patterns and processes proposed to support it. In addition, the inclusion of the end-users throughout all the development process is essential to build quality systems. Human-Computer Interaction (HCI) and Software Engineering (SE) have presented in the literature some approaches and software development process that can contribute for the development of interactive systems. However, the application and combination of these approaches is a challenge for industry.

Nowadays, SPCM models are well used in the industry (for example, more than 10,000 official appraisals are reported by the CMMI models (CMMI Product Team, 2010)). Several studies have still done propositions for the high maturity levels (e.g., (Schots et al., 2015)). A collection of software engineering practices is proposed by these models with the objective to improve the software process of the companies. These models define what should be done but do not specify how it should be done. We argue that indicating specific approaches that support the development of interactive system considering HCI issues is a good way to specify “how” to perform what these models advocate. With this belief, this thesis investigated how to integrate HCI approaches to support the analysis, design, implementation, and evaluation of interactive systems when using SPCM models. To that end three specific research objectives were set:

- To identify HCI approaches to support interactive system development that follows CMM-DEV and/or MR-MPS-SW models;
- To evaluate and improve the identified HCI approaches with domain experts;
- To conduct empirical studies with the resulting proposal.

To address the first objective, the documentation of CMMI-DEV and MR-MPS-SW (international and Brazilian model respectively) were scanned and HCI issues were identified supporting the definition of a first proposition of HCI approaches integrated to those models (see chapter 3). This first proposition was evaluated and improved by experts from HCI domain (also presented in chapter 3), and empirical studies in academic environment (see chapter 4) and industrial context (see chapter 5) were conducted.

Contributions of this thesis

This thesis allowed us to make the following contributions:

- (i) The definition of fourteen categories of HCI approaches with a set of examples of specific methods, techniques, standards and patterns to support interactive system development that follows CMM-DEV and/or MR-MPS-SW models.

The HCI approaches were identified from an in-depth analysis of the CMMI-DEV and MR-MPS-SW documentation (engineering process areas) and the HCI literature. This proposition was reviewed and improved with domain experts and can be used by the organizations to support the different phases of development of interactive systems using SPCM models ((Gonçalves et al., 2017b), (Gonçalves et al., 2016a), (Gonçalves et al., 2016b) and (Gonçalves et al., 2015)).

- (ii) A set of suggestions about how to use the HCI approaches in engineering practices of SPCM models for the development of interactive systems.

The thesis makes evident which HCI approaches support the implementation of engineering practices (e.g., 26 practices for the CMMI-DEV). Based on the reviews with the expert and in literature we described some general suggestions explaining how the HCI approaches integrated in engineering practices can be used to support the development of interactive systems following SPCM models. The suggestions covered only the engineering practices integrated in our approach. They can be used by the organizations that develop interactive systems using SPCM models, especially the organizations that wish to reach level 3 (CMMI-DEV) or level D/C (MR-MPS-SW).

- (iii) The definition and implementation of an instrument (questionnaire) to perform empirical studies about the perception of knowledge and use of the HCI approaches in industrial context.

We developed a web questionnaire (Annex E) focusing in all engineering practices considered in our approach and in the HCI categories proposed for these practices. These studies, performed with SPCM models consultants, allow us to collect evidence regarding to the perception of knowledge and use of the HCI categories in industry. This questionnaire was implemented for collecting results using a web site.

- (iv) An observational study in academic environment to validate our proposition.

The observational study ((Gonçalves et al., 2017c) and (Oliveira et al., 2015)) allows us to collect evidence regarding to the use of one category (Task modeling) of our proposition. Two iterations were performed. The first one (iteration 1 – without our list of HCI approaches) shows that most students (more than 50%) did not use task modeling naturally and spontaneously in software design. In contrast, the second one (iteration 2 – with our list of HCI approaches) shows that 68% of the students' reports presented task modeling.

- (v) The evidence that users of SPCM models do not know and use HCI approaches, as they know and use software engineering approaches.

The questionnaires previously defined were answered by SPCM models consultants in an international and national (Brazilian) context. The results of the study performed in Brazil, confirm our main supposition that the users, in particular consultants, of SPCM models do not know and use HCI approaches, as they know and use software engineering approaches. The survey conducted in the international context considered 10 different countries but was not conclusive. Nevertheless, it suggests that the perception of knowledge and use of HCI approaches is not yet enough to reach the practice in industry ((Gonçalves et al., 2017a)).

- (vi) A state-of-the-art about theory and practice of HCI issues.

This state of art reviews the Usability Capability/Maturity (UCM) models proposed in literature; it shows that they are not applied in practice. It also presents works that have proposed the integration of HCI issues with Software Engineering, and works that have investigated the practice of HCI in the industry.

Limitations

As any research study, this thesis has several limitations. The first research limitation is related to the analysis performed regarding to the SPCM models. Our research is limited for engineering process

areas from both chosen models (CMMI-DEV and MR-MPS-SW). We did not consider process management, project management and support process areas. We are aware that these categories presented a strong relationship with HCI issues in the organization that can impact the software development activities.

The second limitation of this thesis is related to the HCI approaches (explicit examples) that are proposed. It was not our intention to be exhaustive since new approaches continue to be proposed in literature (for instance, new approaches have been proposed for intelligent or adaptive user interfaces). Our intention was to identify the large categories of methods, techniques, standards and patterns based on the HCI issues found in the analysis of the SPCM models. These categories can be continually enriched with other HCI approaches.

The third research limitation is related to the validation of the HCI approaches categories in practice. We start a long-term validation in academy performing two observational studies in an academic environment to validate one of the HCI categories (Task modeling) proposed in this work. However, these studies were not performed in a SPCM implementation context. We considered that this study is a start point of a long-term validation project not only in academy but, if possible, in industry.

The fourth research limitation is related to the empirical studies performed to validate our research hypothesis. For the first study, the results confirm our main supposition that the users of SPCM models do not know and use HCI approaches, as they know and use software engineering approaches. However, we cannot generalize the results for all users of SPCM models because the context of this study is Brazil. For the second study (with 10 different countries) the results were not conclusive since we did not get the necessary amount of data to do a statistical analysis.

Future works

This thesis opens several research perspectives of which we highlight the main ones as follows.

A first perspective is to continue our long-term validation. For example, we can continue the analysis of the requirement specification reports regarding the user profiles of the system, or the prototypes performed. The idea is to perform several observational studies in academic environment about all proposed HCI categories.

A second perspective is to define a software process for requirement specification of interactive systems using our HCI approaches. Then, we plan to apply this software process in the same academic environment context of the study presented in this thesis to allow the validation of our proposition for the complete process for requirements specification.

A third perspective is to perform observational studies in industrial environment and in the context of SPCM implementations. The objective is to validate the effective use of our HCI categories in this context. As previously explained we defined fourteen HCI categories that are composed of several HCI approaches.

A fourth perspective is to define a software process for the development for specific interactive systems. We recall that SPCM models are usually used to establish defined software processes in organizations. Based on that, we have worked on the definition of a software development process for a specific kind of interactive system (application on an interactive tabletop with tangible objects). Defining specific software processes introducing the approaches to be used to support the activities can help in the validation of the proposition in practice.

A fifth perspective is to develop a capability maturity model for user-centered design. The idea is to combine one or more of the usability capability/maturity (UCM) models presented in the state-of-the-art and a SPCM model (for instance CMMI-DEV). The proposition is to develop a capability maturity model for user-centered design that presents the same maturity levels, the same structure of CMMI-DEV including our HCI approaches and the practices defined in UCM models. In this way the enterprises that develop interactive systems could be evaluated regarding to not only software engineering issues but also HCI engineering issues.

A sixth perspective is to investigate the needs to adapt the official evaluation model (SCAMPI - Standard CMMI Appraisal Method for Process Improvement) of the CMMI according to the new model for user-centered design, and to validate the model with domain experts.

Finally, we affirm that all these perspectives open up potential future research to lead to a better interactive systems development in industry.

References

- Ab Latif, R., Mohamed, R., Dahlan, A., & Mat Nor, M. (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.
- Abed, M., Bernard, J. M., & Angué, J. C. (1991). Task analysis and modelization by using SADT and Petri Networks. In *Proceedings of Tenth European Annual Conference on Human Decision Making and Manual Control*. Liege, Belgium.
- Acuña, S. T., Castro, J. W., & Juristo, N. (2012). A HCI technique for improving requirements elicitation. *Information and Software Technology*, 54(12), 1357–1375. <https://doi.org/http://dx.doi.org/10.1016/j.infsof.2012.07.011>
- Alpert, S. R. (2003). Getting Organized: Some outstanding questions and issues regarding interaction design patterns. In *Workshop of Perspectives on HCI Patterns: Concepts and Tools in CHI'03*.
- Alyahyan, L., Alnafjan, K., & Aldabbas, H. (2016). A framework for integrating usability practices into small-sized software development organizations. *International Journal of Software Engineering & Applications*, 7(5), 13–24. <https://doi.org/10.5121/ijsea.2016.7502>
- Anderson, J., Fleek, F., Garrity, K., & Drake, F. (2001). Integrating Usability Techniques into Software Development. *IEEE Software*, 18(1), 46–53. <https://doi.org/10.1109/52.903166>
- Anzalone, D., Manca, M., Paternò, F., & Santoro, C. (2015). Responsive Task Modelling. In *Proceedings of the 7th ACM SIGCHI Symposium on Engineering Interactive Computing Systems* (pp. 126–131). New York, NY, USA: ACM. <https://doi.org/10.1145/2774225.2775079>
- Ardito, C., Buono, P., Caivano, D., Costabile, M. F., & Lanzilotti, R. (2014). Investigating and promoting UX practice in industry: An experimental study. *International Journal of Human-Computer Studies*, 72(6), 542–551. <https://doi.org/http://doi.org/10.1016/j.ijhcs.2013.10.004>
- Assila, A., Oliveira, K. M., & Ezzedine, H. (2016). Standardized Usability Questionnaires: Features and Quality Focus. *Electronic Journal of Computer Science and Information Technology*, 6(1), 15–31.
- 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, B. Eggen, E. Aarts, & J. L. Crowley (Eds.), *Ambient Intelligence: Second European Symposium, EUSAI 2004* (pp. 291–302). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-30473-9_28
- Barbosa, S. D. J., & Paula, M. G. (2003). Designing and Evaluating Interaction as Conversation: A Modeling Language Based on Semiotic Engineering. In J. A. Jorge, N. J. Nunes, & J. F. e Cunha (Eds.), *Interactive Systems. Design, Specification, and Verification: 10th International Workshop, DSV-IS 2003, Funchal, Madeira Island, Portugal, June 11-13, 2003*. (pp. 16–33). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-39929-2_2
- Basili, V. R., & Rombach, H. D. (1988). The TAME Project: Towards Improvement-oriented Software Environments. *IEEE Trans. Softw. Eng.*, 14(6), 758–773. <https://doi.org/10.1109/32.6156>
- Bass, L., Clements, P., & Kazman, R. (2012). *Software Architecture in Practice* (3rd ed.). Addison-Wesley Professional.
- 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.
- Beaudouin-Lafon, M., & Mackay, W. E. (2009). Prototyping tools and techniques. In A. Sears & J. A. Jacko (Eds.), *Human-Computer Interaction: Development Process* (pp. 121–143). NW: CRC Press.
- 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. <https://doi.org/http://dx.doi.org/10.1016/j.jss.2004.06.004>
- Bell Canada. (1994). *Trillium Model for Telecom Product Development & Support Process Capability* (No. Release 3.0).
- Bellini, E., Nesi, P., Pantaleo, G., & Venturi, A. (2016). Functional resonance analysis method based-decision support tool for urban transport system resilience management. In *IEEE International Smart Cities Conference (ISC2)*. Trento, Italy: IEEE. <https://doi.org/10.1109/ISC2.2016.7580833>

- Bevan, N. (2003a). New ISO Standards for Usability, Usability Reports and Usability Measures. In *Proceedings of the International Conference on Human-Computer Interaction. Human-Computer Interaction: Theory and Practice (Part 1)* (Vol. 1, pp. 434–438). Lawrence Erlbaum.
- Bevan, N. (2003b). UsabilityNet methods for user centred design. In *Proceedings of HCI International - Human-Computer Interaction: theory and Practice* (Vol. 1, pp. 434–438). Lawrence Erlbaum.
- Bevan, N. (2005). Cost-Benefit Framework and Case Studies. In R. G. Bias & D. J. Mayhew (Eds.), *Cost-Justifying Usability: An Update for an Internet Age* (Second edition, pp. 575–600). San Francisco: Morgan Kaufmann. <https://doi.org/10.1016/B978-012095811-5/50020-1>
- Bevan, N. (2009). International Standards for Usability Should Be More Widely Used. *Journal Usability Studies*, 4(3), 106–113.
- Bevan, N., Carter, J., Earthy, J., Geis, T., & Harker, S. (2016). New ISO Standards for Usability, Usability Reports and Usability Measures. In *Proceedings of the 18th International Conference on Human-Computer Interaction. Theory, Design, Development and Practice - Part I* (Vol. 9731, pp. 268–278). New York, NY, USA: Springer-Verlag New York, Inc. https://doi.org/10.1007/978-3-319-39510-4_25
- Boehm, B. W. (1988). A Spiral Model of Software Development and Enhancement. *IEEE Computer*, 21(5), 61–72. <https://doi.org/10.1109/2.59>
- Borchers, J. (2001). *A Pattern Approach to Interaction Design*. New York, NY, USA: John Wiley & Sons, Inc.
- Brown, J. (1996). Evaluation of the Task-Action Grammar Method for Assessing Learnability in User Interface Software. In *Proceedings of the 6th Australian Conference on Computer-Human Interaction (OZCHI '96)* (p. 308–). Washington, DC, USA: IEEE Computer Society.
- Caffiau, S., Girard, P., Scapin, D. L., Guittet, L., & Sanou, L. (2008). Assessment of Object Use for Task Modeling. In P. Forbrig & F. Paternò (Eds.), *Engineering Interactive Systems: Second Conference on Human-Centered Software Engineering, HCSE 2008, and 7th International Workshop on Task Models and Diagrams, TAMODIA 2008*. (pp. 14–28). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-85992-5_2
- Camara, F., & Calvary, G. (2015). Worth-Centered Design in Practice: Lessons from Experience and Research Agenda. In J. Abascal, S. Barbosa, M. Fetter, T. Gross, P. Palanque, & M. Winckler (Eds.), *Human-Computer Interaction – INTERACT 2015: 15th IFIP TC 13 International Conference* (Vol. Part IV, pp. 123–139). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-22723-8_10
- Carvalho, P. V. R. de. (2011). The use of Functional Resonance Analysis Method (FRAM) in a mid-air collision to understand some characteristics of the air traffic management system resilience. *Reliability Engineering & System Safety*, 96(11), 1482–1498. <https://doi.org/https://doi.org/10.1016/j.res.2011.05.009>
- Céret, E., Dupuy-Chessa, S., Calvary, G., Front, A., & Rieu, D. (2013). A taxonomy of design methods process models. *Information and Software Technology*, 55(5), 795–821. <https://doi.org/https://doi.org/10.1016/j.infsof.2012.11.002>
- Chagas, L. F., de 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 A. Mitasiunas, T. Rout, R. V. O'Connor, & A. Dorling (Eds.), *Software Process Improvement and Capability Determination: Proceedings of 14th International Conference, SPICE 2014* (pp. 177–189). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-13036-1_16
- Chapman, L., & Plewes, S. (2014). A UX Maturity Model: Effective Introduction of UX into Organizations. In A. Marcus (Ed.), *Proceedings of Third International Conference of Design, User Experience, and Usability* (pp. 12–22). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-07638-6_2
- Chung, E. S., Hong, J. I., Lin, J., Prabaker, M. K., Landay, J. A., & Liu, A. L. (2004). Development and Evaluation of Emerging Design Patterns for Ubiquitous Computing. In *Proceedings of the 5th Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques* (pp. 233–242). New York, NY, USA: ACM. <https://doi.org/10.1145/1013115.1013148>
- CMMI Product Team. (2010). *CMMI® for Development* (Version 1.3 No. CMU/SEI-2010-TR-033). Pittsburgh, PA, USA: Software Engineering Institute, Carnegie Mellon University. Retrieved from <http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=9661>
- CMMI Product Team. (2015). *Maturity Profile Report*. Retrieved from <http://cmmiinstitute.com/resources/>
- Cockton, G, Woolrych, A., & Lavery, D. (2009). Inspection-based evaluations. In A. Sears & J. A. Jacko (Eds.), *Human-Computer Interaction: Development Process* (pp. 273–292). NW: CRC Press.

- Cockton, Gilbert. (2006). Designing Worth is Worth Designing. In *Proceedings of the 4th Nordic Conference on Human-computer Interaction: Changing Roles* (pp. 165–174). New York, NY, USA: ACM. <https://doi.org/10.1145/1182475.1182493>
- Courage, C., & Baxter, K. (2005). *Understanding your users: a practical guide to user requirements, methods, tools, and techniques* (1st Edition). San Francisco, CA: Morgan Kaufmann Publishers.
- Courage, C., Redish, J. G., & Wixon, D. (2009). Task Analysis. In A. Sears & J. A. Jacko (Eds.), *Human-Computer Interaction: Development Process* (pp. 33–53). NW: CRC Press.
- 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.
- da Silva, A. C., Silva, J. C. A., Penteado, R. A. D., & da Silva, S. R. P. (2004). Aplicabilidade de Padrões de Engenharia de Software e de IHC no Desenvolvimento de Sistemas Interativos. In *Anais do IV Congresso Brasileiro de Computação* (pp. 118–123).
- da Silva, A. C., Silva, J. C. A., Penteado, R. A. D., & da Silva, S. R. P. (2005). Integrando visões de IHC e de ES por Padrões no desenvolvimento por prototipação descartável. In *Proceedings of the Latin American Conference on Human-computer Interaction* (pp. 223–234). New York, NY, USA: ACM. <https://doi.org/10.1145/1111360.1111383>
- DATECH. (2002). *DATEch-Prufbaustein, Usability-Engineering-Prozess* (No. Version 1.2 (in German)). Frankfurt/Main.
- Diaper, D., & Stanton, N. A. (2004). *The Handbook of Task Analysis for Human-Computer Interaction*. Mahwah, NJ: Lawrence Erlbaum Associates.
- Dumas, J. S., & Fox, J. E. (2009). Usability testing: Current practice and future directions. In A. Sears & J. A. Jacko (Eds.), *Human-Computer Interaction: Development Process* (pp. 231–250). NW: CRC Press.
- Dyba, T. (2000). An Instrument for Measuring the Key Factors of Success in Software Process Improvement. *Empirical Software Engineering*, 5(4), 357–390. <https://doi.org/10.1023/A:1009800404137>
- Earthy, J. (1998). *Usability Maturity Model: Human Centredness Scale* (INUSE Project deliverable D5.1.4(s) No. Version 1.1). London: Lloyd's Register.
- Earthy, J. (1999). *Usability Maturity Model: Processes* (INUSE Project deliverable D5.1.4 No. Version 2.2). London: Lloyd's Register.
- Earthy, J. (2000a). *Quality In Use processes and their integration - Part 2 Assessment Model*. London: Lloyd's Register.
- Earthy, J. (2000b). *Quality In Use processes and their integration Part 1 - Reference Model*. London: Lloyd's Register.
- Earthy, J. (2001). *HFI CMM Study - Process risk assessors version of HSL model* (No. Work package 3). London: Lloyd's Register. Retrieved from <http://www.processforusability.co.uk/HFIPRA/>
- Earthy, J., Bowler, Y., Forster, M., & Taylor, R. (1999). A Human Factors Integration Capability Maturity Model. In *Proceedings of the International Conference on People in Control* (pp. 320–326). IET. <https://doi.org/10.1049/cp:19990208>
- 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. <https://doi.org/10.1007/BF00127446>
- Estorilio, C., Vaz, G. R. M., Lisboa, F. C. de, & Bessa, L. de O. F. (2015). The relationship between industrial process maturity and quality certification. *Computer Standards & Interfaces*, 39, 22–33. <https://doi.org/http://dx.doi.org/10.1016/j.csi.2014.11.002>
- Fathy, S., & Kornysheva, E. (2016). Approches de Design Rationale: Cadre de Référence. In *Actes du XXXIVème Congrès INFORSID* (pp. 83–98). Grenoble, France.
- Fenton, N. E., & Pfleeger, S. L. (1998). *Software Metrics: A Rigorous and Practical Approach* (2nd ed.). Boston, MA, USA: PWS Publishing Co.
- Ferre, X., Juristo, N., & Moreno, A. M. (2005a). Framework for Integrating Usability Practices into the Software Process. In F. Bomarius & S. Komi-Sirviö (Eds.), *Product Focused Software Process Improvement: 6th International Conference, PROFES 2005* (pp. 202–215). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/11497455_17
- Ferre, X., Juristo, N., & Moreno, A. M. (2005b). Which, When and How Usability Techniques and Activities Should Be Integrated. In A. Seffah, J. Gulliksen, & M. C. Desmarais (Eds.), *Human-Centered Software*

- Engineering — Integrating Usability in the Software Development Lifecycle* (pp. 173–200). Springer Netherlands. https://doi.org/10.1007/1-4020-4113-6_10
- Fischer, H. (2012). Integrating Usability Engineering in the Software Development Lifecycle Based on International Standards. In *Proceedings of the 4th ACM SIGCHI Symposium on Engineering Interactive Computing Systems* (pp. 321–324). New York, NY, USA: ACM. <https://doi.org/10.1145/2305484.2305541>
- Fischer, H., Streng, B., & Nebe, K. (2013). Towards a Holistic Tool for the Selection and Validation of Usability Method Sets Supporting Human-Centered Design. In A. Marcus (Ed.), *Design, User Experience, and Usability. Design Philosophy, Methods, and Tools: Second International Conference, DUXU 2013, Held as Part of HCI International 2013* (Vol. Part I, pp. 252–261). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-39229-0_28
- Flanagan, G. A. (1996). Usability Management Maturity, Part 1: Self-Assessment - How Do You Stack Up? *SIGCHI Bulletin*.
- Flanagan, G. A., & Rauch, T. L. (1995). Usability Management Maturity, Part 1: Self-Assessment - How Do You Stack Up? In *Conference Companion on Human Factors in Computing Systems* (p. 336). New York, NY, USA: ACM. <https://doi.org/10.1145/223355.223705>
- Foley, J. D., & Van Dam, A. (1982). *Fundamentals of Interactive Computer Graphics*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.
- Folmer, E., Welie, M. van, & Bosch, J. (2006). Bridging patterns: An approach to bridge gaps between SE and HCI. *Information and Software Technology*, 48(2), 69–89. <https://doi.org/https://doi.org/10.1016/j.infsof.2005.02.005>
- Fundação Nacional da Qualidade. (2011). *Cr terios de Excel ncia: Cr terios de Excel ncia 2011 - Avalia  o e diagn stico da gest o organizacional* (19th ed.). S o Paulo.
- Furtado, E. S., Conte, T., Barbosa, S. D. J., Melo, P., Maciel, C., & Souza, C. R. B. (2016). Evangelizing HCI Research and Practice in Brazil. *Interactions*, 23(6), 78–78. <https://doi.org/10.1145/3004226>
- Garz s, J., Pino, F. J., Piattini, M., & Fern ndez, C. M. (2013). A maturity model for the Spanish software industry based on ISO standards. *Computer Standards & Interfaces*, 35(6), 616–628. <https://doi.org/http://doi.org/10.1016/j.csi.2013.04.002>
- Giese, M., Mistrzyk, T., Pfau, A., Szwillus, G., & von Detten, M. (2008). AMBOSS: A Task Modeling Approach for Safety-Critical Systems. In P. Forbrig & F. Patern  (Eds.), *Engineering Interactive Systems: Second Conference on Human-Centered Software Engineerin* (pp. 98–109). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-85992-5_8
- Goldberg, A. (1984). *SMALLTALK-80: The Interactive Programming Environment*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.
- Gon alves, T. G., Oliveira, K. M., & Kolski, C. (2015). HCI Engineering Integrated with Capability and Maturity Models. In *Proceedings of the 27th Conference on L'Interaction Homme-Machine* (p. 36 (1-6)). New York, NY, USA: ACM. <https://doi.org/10.1145/2820619.2825017>
- Gon alves, T. G., Oliveira, K. M., & Kolski, C. (2016a). HCI engineering integrated with capability maturity models: A study focused on requirements development. In *Tenth IEEE International Conference on Research Challenges in Information Science, RCIS 2016, Grenoble, France, June 1-3, 2016* (pp. 633–644). <https://doi.org/10.1109/RCIS.2016.7549319>
- Gon alves, T. G., Oliveira, K. M., & Kolski, C. (2016b). Verification and Validation of Interactive Systems in the CMMI-DEV Capability Maturity Model. In *Proceedings of the 28th Conference on Interaction Homme-Machine* (pp. 48–58). New York, NY, USA: ACM. <https://doi.org/10.1145/3004107.3004124>
- Gon alves, T. G., Oliveira, K. M., & Kolski, C. (2017a). A study about HCI in practice of interactive system development using CMMI-DEV. In *Proceedings of the 29th Conference on Interaction Homme-Machine* (pp. 169–177). New York, NY, USA: ACM.
- Gon alves, T. G., Oliveira, K. M., & Kolski, C. (2017b). Identifying HCI Approaches to support CMMI-DEV for Interactive System Development. *Computer Standards & Interfaces*. <https://doi.org/10.1016/j.csi.2017.12.003>
- Gon alves, T. G., Oliveira, K. M., & Kolski, C. (2017c). The use of task modeling in interactive system specification. *Cognition, Technology & Work*, 19(Issue 2–3), 493–515. <https://doi.org/10.1007/s10111-017-0427-1>
- 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 International Conference on Human-Centered Software Engineering, HCSE 2016, and 8th International Conference on Human Error, Safety, and System Development, HESSD 2016* (pp. 301–313). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-44902-9_19
- Gupta, A. (1997). *The Humanware Process Improvement Framework: Interfacing User Centred Design and the Product Creation Process at Philips*. Philips.
- Hackos, J. T., & Redish, J. C. (1998). *User and Task Analysis for Interface Design*. John Wiley & Sons, Inc.
- Hao, H. M., & Jaafar, A. (2011). Usability in Practice: Perception and Practicality of Management and Practitioners. In *Proceedings of International Conference on Pattern Analysis and Intelligence Robotics* (Vol. 2, pp. 211–216). <https://doi.org/10.1109/ICPAIR.2011.5976928>
- Hartson, H. R., Siochi, A. C., & Hix, D. (1990). The UAN: A User-oriented Representation for Direct Manipulation Interface Designs. *ACM Transactions on Information System*, 8(3), 181–203. <https://doi.org/10.1145/98188.98191>
- 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. <https://doi.org/10.1002/smr.560>
- 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(6), 841–858. <https://doi.org/http://dx.doi.org/10.1016/j.jss.2005.08.023>
- Hix, D., & Hartson, H. R. (1993). *Developing User Interfaces: Ensuring Usability Through Product and Process*. New York, NY, USA: John Wiley & Sons, Inc.
- 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. <https://doi.org/https://doi.org/10.1016/j.infsof.2017.01.007>
- Hussein, I., Mahmud, M., & Md. Tap, A. O. (2012). User Experience Design (UXD): a survey of user interface development practices in Malaysia. In *Proceedings of SEANES 2012*. Langkawi, Malaysia.
- Hussein, I., Mahmud, M., & Yeo, A. W. (2010). HCI practices in Malaysia: A reflection of ICT professionals' perspective. In *Proceedings of International Symposium on Information Technology* (pp. 1549–1554). IEEE. <https://doi.org/10.1109/ITSIM.2010.5561470>
- Hussein, I., Seman, E. A. A., & Mahmud, M. (2009). Perceptions on Interaction Design in Malaysia. In N. Aykin (Ed.), *Internationalization, Design and Global Development: Third International Conference, IDGD 2009, Held as Part of HCI International 2009* (pp. 356–365). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-02767-3_40
- International Organization for Standardization. (1998). *Ergonomic requirements for office work with visual display terminals (VDTs) – Part 11: Guidance on usability (ISO 9241-11:1998)*. Geneva, Switzerland.
- International Organization for Standardization. (1999). *Human-Centred Design Process for Interactive System (ISO 13407:1999)*. Geneva, Switzerland.
- International Organization for Standardization. (2000). *Ergonomics – Ergonomics of human-system interaction – Human-centred lifecycle process descriptions (ISO/TR 18529:2000)*. Geneva, Switzerland.
- International Organization for Standardization. (2002). *Ergonomics of human-system interaction – Usability methods supporting human centred design (ISO/TR 16982:2002)*. Geneva, Switzerland.
- International Organization for Standardization. (2003). *Ergonomics of human-system interaction – Specification for the process assessment of human-system issues (ISO/PAS 18152:2003)*. Geneva, Switzerland.
- International Organization for Standardization. (2010a). *Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems (ISO 9241-210:2010)*. Geneva, Switzerland.
- International Organization for Standardization. (2010b). *Ergonomics of human-system interaction – Specification for the process assessment of human-system issues (ISO/TS 18152:2010)*. Geneva, Switzerland.
- International Organization for Standardization/International Electrotechnical Commission. (1995). *Information technology – Software life cycle processes (ISO/IEC 12207:1995)*. Geneva, Switzerland.
- International Organization for Standardization/International Electrotechnical Commission. (2001). *Software engineering – Product quality – Part 1: Quality model (ISO/IEC 9126-1:2001)*. Geneva, Switzerland.

- International Organization for Standardization/International Electrotechnical Commission. (2003). *Information Technology – Process Assessment – Part 2: Performing an assessment (ISO/IEC 15504-2: 2003)*. Geneva, Switzerland.
- International Organization for Standardization/International Electrotechnical Commission. (2004a). *Information Technology – Process Assessment – Part 1: Concepts and vocabulary (ISO/IEC 15504-1)*. Geneva, Switzerland.
- International Organization for Standardization/International Electrotechnical Commission. (2004b). *Information Technology – Process Assessment – Part 4: Guidance on use for Process Improvement and Process Capability Determination (ISO/IEC 15504-4)*. Geneva, Switzerland.
- International Organization for Standardization/International Electrotechnical Commission. (2008a). *Information technology – Process assessment – Part 7: Assessment of organizational maturity (ISO/IEC TR 15504-7:2008)*. Geneva, Switzerland.
- International Organization for Standardization/International Electrotechnical Commission. (2008b). *Systems and software engineering – Software life cycle processes (ISO/IEC 12207:2008)*. Geneva, Switzerland.
- International Organization for Standardization/International Electrotechnical Commission. (2008c). *Systems and software engineering – System life cycle processes (ISO/IEC 15288:2008)*. Geneva, Switzerland.
- International Organization for Standardization/International Electrotechnical Commission. (2014). *Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Guide to SQuaRE (ISO/IEC 25000:2014)*. Geneva, Switzerland.
- International Organization for Standardization/International Electrotechnical Commission. (2015). *Information technology – Process assessment – Process measurement framework for assessment of process capability (ISO/IEC 33020)*. Geneva, Switzerland.
- Ivory, M. Y., & Hearst, M. A. (2001). The State of the Art in Automating Usability Evaluation of User Interfaces. *ACM Computing Surveys*, 33(4), 470–516. <https://doi.org/10.1145/503112.503114>
- Ji, Y. G., & Yun, M. H. (2006). Enhancing the Minority Discipline in the IT Industry: A Survey of Usability and User-Centered Design Practice. *International Journal of Human–Computer Interaction*, 20(2), 117–134. https://doi.org/10.1207/s15327590ijhc2002_3
- Jokela, T. (2001). Review of Usability Capability Assessment Approaches. In *Proceedings of the 24th Information Systems Research Seminar*. Scandinavia, Norway.
- Jokela, T. (2004). *The KESSU Usability Design Process Model* (No. Version 2.1). University of Oulu.
- Jokela, T. (2008). Characterizations, Requirements, and Activities of User-Centered Design—the KESSU 2.2 Model. In E. L.-C. Law, E. T. Hvannberg, & G. Cockton (Eds.), *Maturing Usability: Quality in Software, Interaction and Value* (pp. 168–196). London: Springer London. https://doi.org/10.1007/978-1-84628-941-5_8
- Jokela, T. (2010). Usability Maturity Models: Making your Company User-Centered. *User Experience Magazine*, 9(1). Retrieved from http://uxpamagazine.org/usability_maturity_models/
- Jokela, T., & Lalli, T. (2003). Usability and CMMI: Does a Higher Maturity Level in Product Development Mean Better Usability? In *CHI '03 Extended Abstracts on Human Factors in Computing Systems* (pp. 1010–1011). New York, NY, USA: ACM. <https://doi.org/10.1145/765891.766121>
- 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. <https://doi.org/10.1080/01449290500168079>
- Juristo, N., López, M., Moreno, A. M., & Sánchez, M. I. (2003). Improving software usability through architectural patterns. In *Proceedings of ICSE 2003 Workshop on Bridging the Gaps Between Software Engineering and Human-Computer Interaction* (pp. 12–19). Portland, USA.
- 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 *Proceedings of 9th International Conference on the Quality of Information and Communications Technology* (pp. 28–37).
- Kalinowski, M., Weber, K., Santos, G., Franco, N., Duarte, V., & Travassos, G. (2015). Software Process Improvement Results in Brazil Based on the MPS-SW Model. *Software Quality Professional*, 17(4), 15–28.
- Kieffer, S., & Vanderdonckt, J. (2016). STRATUS: A Questionnaire for Strategic Usability Assessment. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing* (pp. 205–212). New York, NY, USA: ACM. <https://doi.org/10.1145/2851613.2851912>

- Kitchenham, B., & Charters, S. (2007). *Guidelines for performing systematic literature reviews in software engineering* (Technical Report EBSE 2007-001) (p. 57p.). Keele University and Durham University.
- Kolski, C., Ezzedine, H., & Abed, M. (2001). Développement du logiciel : des cycles classiques aux cycles enrichis sous l'angle des IHM. In C. Kolski (Ed.), *Analyse et Conception de l'IHM. Interaction Homme-machine pour les SI* (Vol. 1, pp. 23–49). Paris: Hermès.
- Kolski, C., & Loslever, P. (1998). An HCI-Enriched Model for Supporting Human-Machine Systems Design and Evaluation. *IFAC Proceedings Volumes*, 31(26), 419–424. [https://doi.org/http://dx.doi.org/10.1016/S1474-6670\(17\)40129-7](https://doi.org/http://dx.doi.org/10.1016/S1474-6670(17)40129-7)
- Kolski, C., Loslever, P., & Sagar, M. (2012). The performance of future designers on the specification of supervisory HCI: case study of a simulated work situation. *Cognition, Technology & Work*, 14(2), 107–128. <https://doi.org/10.1007/s10111-010-0169-9>
- Kolski, C., Sagar, M., & Loslever, P. (2004). Experiment based on participative ergonomics and performed as part of a supervision engineers' training course. In E. F. Fallon & W. Karwowski (Eds.), *Proceedings of HAAMAHA'2004 - Human and Organisational Issues in the Digital Enterprise* (pp. 336–346). Galway: National University of Ireland.
- Lacerda, T. C., & Wangenheim, C. G. von. (2018). Systematic literature review of usability capability/maturity models. *Computer Standards & Interfaces*, 55, 95–105. <https://doi.org/https://doi.org/10.1016/j.csi.2017.06.001>
- Lazar, J., Feng, J. H., & Hochheiser, H. (2010). *Research Methods in Human-Computer Interaction*. Wiley Publishing.
- Lepreux, S., Abed, M., & Kolski, C. (2003). A Human-centred Methodology Applied to Decision Support System Design and Evaluation in a Railway Network Context. *Cogn. Technol. Work*, 5(4), 248–271. <https://doi.org/10.1007/s10111-003-0128-9>
- Limbourg, Q., & Vanderdonckt, J. (2004). Comparing Task Model for User Interface Design. In D. Diaper & N. A. Stanton (Eds.), *The Handbook of Task Analysis for Human-Computer Interaction* (pp. 135–154). Mahwah, NJ: Lawrence Erlbaum Associates.
- Linstone, H. A., & Turoff, M. (2002). *The Delphi Method: techniques and applications*. New Jersey. Retrieved from <https://web.njit.edu/~turoff/pubs/delphibook/delphibook.pdf>
- Llerena, L., Rodríguez, N., Castro, J. W., & Acuña, S. T. (2017a). Adoption of the Focus Groups Technique in the Open Source Software Development Process. In M. Kurosu (Ed.), *Proceedings of International Conference on Human-Computer Interaction - User Interface Design, Development and Multimodality - Part I* (pp. 325–340). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-58071-5_25
- Llerena, L., Rodríguez, N., Castro, J. W., & Acuña, S. T. (2017b). Adoption of the User Profiles Technique in the Open Source Software Development Process. In J. Mejia, M. Muñoz, Á. Rocha, T. San Feliu, & A. Peña (Eds.), *Trends and Applications in Software Engineering: Proceedings of CIMPS 2016* (pp. 201–210). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-48523-2_19
- Maguire, M. (2001). Methods to support human-centred design. *International Journal of Human-Computer Studies*, 55(4), 587–634. <https://doi.org/http://dx.doi.org/10.1006/ijhc.2001.0503>
- 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. <https://doi.org/10.1080/10447311003781409>
- Marcus, A., Gunther, R., & Sieffert, R. (2009). Validating a Standardized Usability/User-Experience Maturity Model: A Progress Report. In M. Kurosu (Ed.), *Proceedings of the International Conference on Human Centered Design* (pp. 104–109). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-02806-9_13
- Markham, T. (2012). *Project Based Learning Design and Coaching Guide*. HeartIQ Press.
- Martinie, C., & Palanque, P. (2015). Designing and Assessing Interactive Systems Using Task Models. In I. T. Monteiro & M. S. Silveira (Eds.), *Tutorials of the 14th Brazilian Symposium on Human Factors in Computing Systems*. (pp. 29–58). SBC.
- Martinie, C., Palanque, P., Navarre, D., Winckler, M., & Poupart, E. (2011). Model-based Training: An Approach Supporting Operability of Critical Interactive Systems. In *Proceedings of the 3rd ACM SIGCHI Symposium on Engineering Interactive Computing Systems* (pp. 53–62). New York, NY, USA: ACM. <https://doi.org/10.1145/1996461.1996495>

- Mayhew, D. J. (1999). *The Usability Engineering Lifecycle: a practitioner's handbook for user interface design*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Molina, A. I., Redondo, M. A., & Ortega, M. (2007). Applying Task Modeling and Pattern-based techniques in Reengineering Processes for Mobile Learning User Interfaces: A case study. *Journal of Computers*, 2(4), 23–30.
- Mostafa, D. (2013). *Maturity Models in the Context of Integrating Agile Development Processes and User Centred Design* (PhD thesis). University of York, York, United Kingdom.
- 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 International 2007* (Vol. Part I, pp. 194–203). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-73105-4_22
- 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). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-92698-6_8
- Nebe, K., Zimmermann, D., & Paelke, V. (2008). Integrating Software Engineering and Usability Engineering. In S. Harning (Ed.), *Advances in Human Computer Interaction* (pp. 331–350). InTech. https://doi.org/10.1007/978-3-540-92698-6_8
- Nielsen, J. (1993). *Usability Engineering*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Nielsen, J. (1994). Usability Inspection Methods. In J. Nielsen & R. L. Mack (Eds.) (pp. 25–62). New York, NY, USA: John Wiley & Sons, Inc. Retrieved from <http://dl.acm.org/citation.cfm?id=189200.189209>
- Nielsen, J. (1997). The Use and Misuse of Focus Groups. *IEEE Software*, 14(1), 94–95. <https://doi.org/10.1109/52.566434>
- Nielsen, J. (2006a). Corporate UX Maturity: Stages 1-4. Retrieved September 1, 2017, from <https://www.nngroup.com/articles/ux-maturity-stages-1-4/>
- Nielsen, J. (2006b). Corporate UX Maturity: Stages 5-8. Retrieved September 1, 2017, from <https://www.nngroup.com/articles/ux-maturity-stages-5-8/>
- Nigay, L., & Coutaz, J. (1995). A Generic Platform for Addressing the Multimodal Challenge. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '95* (pp. 98–105). New York, NY, USA: ACM Press/Addison-Wesley Publishing Co. <https://doi.org/10.1145/223904.223917>
- 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). <https://doi.org/10.1109/CLEI.2013.6670641>
- Ogunyemi, A. A., Lamas, D., Adagunodo, E. R., Loizides, F., & Rosa, I. B. D. (2016). Theory, Practice and Policy: An Inquiry into the Uptake of HCI Practices in the Software Industry of a Developing Country. *International Journal of Human-Computer Interaction*, 32(9), 665–681. <https://doi.org/10.1080/10447318.2016.1186306>
- Ogunyemi, A. A., Lamas, D., Stage, J., & Lárusdóttir, M. (2017). Assessment Model for HCI Practice Maturity in Small and Medium Sized Software Development Companies. In A. Mas, A. Mesquida, R. V. O'Connor, T. Rout, & A. Dorling (Eds.), *Software Process Improvement and Capability Determination. Communications in Computer and Information Science* (Vol. 770, pp. 55–69). Cham: Springer.
- Oktaba, H., Alquicira, C., Angélica, E., Ramos, S., Martínez, A., & Quintanilla, G. (2005). *Modelo de Procesos para la Industria de Software: MoProSoft* (No. Versión 1.3). UNAM- Secretaría de Economía.
- Oktaba, H., & Vázquez, A. (2008). MoProSoft®: A Software Process Model for Small Enterprises. In H. Oktaba & M. Piattini (Eds.), *Software Process Improvement for Small and Medium Enterprises: Techniques and Case Studies* (pp. 170–176). Hershey: New York: Information Science Reference, IGI Global. <https://doi.org/10.4018/978-1-59904-906-9>
- Oliveira, K. M., Girard, P., Gonçalves, T. G., Lepreux, S., & Kolski, C. (2015). Teaching Task Analysis for User Interface Design: Lessons Learned from Three Pilot Studies. In *Proceedings of the 27th Conference on L'Interaction Homme-Machine* (p. 31(1-6)). New York, NY, USA: ACM. <https://doi.org/10.1145/2820619.2825011>
- Paternò, F. (2001). Towards a UML for Interactive Systems. In *Proceedings of the 8th IFIP International Conference on Engineering for Human-Computer Interaction* (pp. 7–18). London, UK, UK: Springer-Verlag.

- Paternò, F. (2004). ConcurTaskTrees: An Engineered Notation for Task Models. In D. Diaper & N. A. Stanton (Eds.), *The Handbook of Task Analysis for Human-Computer Interaction* (pp. 483–501). Mahwah, NJ: Lawrence Erlbaum Associates.
- Paternò, F., Mancini, C., & Meniconi, S. (1997). ConcurTaskTrees: A Diagrammatic Notation for Specifying Task Models. In S. Howard, J. Hammond, & G. Lindgaard (Eds.), *Human-Computer Interaction INTERACT '97: IFIP TC13 International Conference on Human-Computer Interaction*. (pp. 362–369). Boston, MA: Springer US. https://doi.org/10.1007/978-0-387-35175-9_58
- Paulk, M. C., Curtis, B., Chrissis, M. B., & Weber, C. V. (1993). Capability maturity model, version 1.1. *IEEE Software*, 10(4), 18–27. <https://doi.org/10.1109/52.219617>
- Paulk, M. C., Weber, C. V., Curtis, B., & Chrissis, M. B. (1995). *The Capability Maturity Model: Guidelines for Improving the Software Process*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.
- Peres, A. L., Silva, T. S. D., Silva, F. S., Soares, F. F., Carvalho, C. R. M. D., & Meira, S. R. D. L. (2014). AGILEUX Model: Towards a Reference Model on Integrating UX in Developing Software Using Agile Methodologies. In *Proceedings of the 2014 Agile Conference* (pp. 61–63). Washington, DC, USA: IEEE Computer Society. <https://doi.org/10.1109/AGILE.2014.15>
- Pfaff, G. E. (1985). *User interface management system*. Springer-Verlag.
- Prates, R. O. (2007). HCI Brazilian Community - After 10 Years. In *Proceedings of the 11th IFIP TC 13 International Conference on Human-computer Interaction - Volume Part II* (pp. 631–632). Berlin, Heidelberg: Springer-Verlag.
- Prates, R. O., Barbosa, S., Silveira, M. S., Souza, C. S., Baranauskas, C., Maciel, C., ... Kujala, T. (2013). HCI Community in Brazil—sweet 16! *Interactions*, 20(6), 80–81. <https://doi.org/10.1145/2530983>
- Prates, R. O., Souza, C. S., & Barbosa, S. (2000). A method for Evaluating the Communicability of User Interfaces. *Interactions*, 7(1), 31–38.
- Preece, J., Sharp, H., & Rogers, Y. (2015). *Interaction Design: Beyond Human-Computer Interaction* (4th Edition). John Wiley & Sons.
- Purchase, H. C. (Ed.). (2012). Six principles for conducting experiments. In *Experimental Human-Computer Interaction: A Practical Guide with Visual Examples* (pp. 199–202). Cambridge: Cambridge University Press. <https://doi.org/10.1017/CBO9780511844522.009>
- Ragosta, M., Martinie, C., Palanque, P., Navarre, D., & Sujan, M. A. (2015). Concept Maps for Integrating Modeling Techniques for the Analysis and Re-Design of Partly-Autonomous Interactive Systems. In *Proceedings of the 5th International Conference on Application and Theory of Automation in Command and Control Systems* (pp. 41–52). New York, NY, USA: ACM. <https://doi.org/10.1145/2899361.2899366>
- Raza, A., Capretz, L. F., & Ahmed, F. (2012). An open source usability maturity model (OS-UMM). *Computers in Human Behavior*, 28(4), 1109–1121. <https://doi.org/http://dx.doi.org/10.1016/j.chb.2012.01.018>
- Rodríguez, F. D., Acuña, S. T., & Juristo, N. (2015). Design and programming patterns for implementing usability functionalities in web applications. *Journal of Systems and Software*, 105, 107–124. <https://doi.org/http://dx.doi.org/10.1016/j.jss.2015.04.023>
- Ross, D. T. (1977). Structured Analysis (SA): A Language for Communicating Ideas. *IEEE Trans. Softw. Eng.*, 3(1), 16–34. <https://doi.org/10.1109/TSE.1977.229900>
- Royce, W. W. (1970). Managing the Development of Large Software Systems: Concepts and Techniques. In *Proceedings of WESCON* (Vol. 26, pp. 1–9). IEEE Computer Society Press.
- Salah, D., Paige, R., & Cairns, P. (2014a). An Evaluation Template for Expert Review of Maturity Models. In A. Jedlitschka, P. Kuvaja, M. Kuhrmann, T. Männistö, J. Münch, & M. Raatikainen (Eds.), *Product-Focused Software Process Improvement: 15th International Conference, PROFES 2014, Helsinki, Finland, December 10-12, 2014. Proceedings* (pp. 318–321). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-13835-0_31
- Salah, D., Paige, R., & Cairns, P. (2014b). Integrating Agile Development Processes and User Centred Design—A Place for Usability Maturity Models? In S. Sauer, C. Bogdan, P. Forbrig, R. Bernhaupt, & M. Winckler (Eds.), *Proceedings of International Conference on Human-Centered Software Engineering: 5th IFIP WG 13.2* (pp. 108–125). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-44811-3_7
- Salah, D., Paige, R., & Cairns, P. (2016). A Maturity Model for Integrating Agile Processes and User Centred Design. In P. M. Clarke, R. V. O'Connor, T. Rout, & A. Dorling (Eds.), *Software Process Improvement and*

- Capability Determination: Proceedings of 16th International Conference, SPICE 2016* (pp. 109–122). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-38980-6_9
- Salgado, A., Amaral, L. A., Freire, A. P., & Fortes, R. P. M. (2016). Usability and UX Practices in Small Enterprises: Lessons from a Survey of the Brazilian Context. In *Proceedings of the 34th ACM International Conference on the Design of Communication* (p. 18:1–18:9). New York, NY, USA: ACM. <https://doi.org/10.1145/2987592.2987616>
- Santoro, C. (2005). *A Task Model-Based Approach for the Design and Evaluation of Innovative User Interfaces*. Louvain: Presses Universitaires de Louvain, SIMILAR.
- Sauro, J., & Lewis, J. R. (2012). *Quantifying the User Experience - Practical Statistics for User Research*. Morgan Kaufmann, MA.
- 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. <https://doi.org/10.1080/014492997119806>
- Scheiber, F., Wruk, D., Oberg, A., Britsch, J., Woywode, M., Maedche, A., ... Plach, M. (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). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-31371-4_3
- Schots, N. C. L., Magalhães, R. F., Gonçalves, T. G., Busquet, R. H., Rocha, A. R. C., Santos, G., & Oliveira, K. M. (2015). A Knowledge-based Environment for Software Process Performance Analysis. *CLEI Electronic Journal*, 18(2), 1–26.
- Sears, A., & Jacko, J. A. (2009a). *Human-Computer Interaction: Design issues, solutions, and applications*. NW: CRC Press.
- Sears, A., & Jacko, J. A. (2009b). *Human-Computer Interaction: Development Process*. NW: CRC Press.
- Seffah, A. (2015). *Patterns of HCI Design and HCI Design of Patterns* (1st ed.). Springer International Publishing. Retrieved from 10.1007/978-3-319-15687-3
- Seffah, A., Desmarais, M. C., & Metzker, E. (2005). HCI, Usability and Software Engineering Integration: Present and Future. In A. Seffah, J. Gulliksen, & M. C. Desmarais (Eds.), *Human-Centered Software Engineering — Integrating Usability in the Software Development Lifecycle* (pp. 37–57). Springer Netherlands. https://doi.org/10.1007/1-4020-4113-6_3
- Seffah, A., & Gaffar, A. (2007). Model-based user interface engineering with design patterns. *Journal of Systems and Software*, 80(8), 1408–1422. <https://doi.org/http://dx.doi.org/10.1016/j.jss.2006.10.037>
- Shneiderman, B., Plaisant, C., Cohen, M., & Jacobs, S. (2009). *Designing the User Interface: Strategies for Effective Human-Computer Interaction* (5th Edition). Pearson.
- Silva, F. S., Soares, F. S. F., Peres, A. L., Azevedo, I. M. de, Vasconcelos, A. P. L. F., Kamei, F. K., & Meira, S. R. de L. (2015). Using CMMI together with agile software development: A systematic review. *Information and Software Technology*, 58, 20–43. <https://doi.org/https://doi.org/10.1016/j.infsof.2014.09.012>
- Silva, W., Valentim, N. M. C., & Conte, T. (2015). Integrating the Usability into the Software Development Process - A Systematic Mapping study. In *Proceedings of the 17th International Conference on Enterprise Information Systems - Volume 3* (pp. 105–113). Portugal: SCITEPRESS - Science and Technology Publications, Lda. <https://doi.org/10.5220/0005377701050113>
- Silveira, M. S., Barbosa, S. D. J., & Souza, C. S. de. (2004). Designing online help systems for reflective users. *Journal of the Brazilian Computer Society*, 9, 25–38.
- Softex. (2016a). *MPS.BR - Brazilian Software Process Improvement, Implementation Guide - Part 4: Rationale for the Implementation of Level D of the MR-MPS-SW:2016*. Retrieved from <http://www.softex.br/mpsbr>
- Softex. (2016b). *MPS.BR - Brazilian Software Process Improvement, Implementation Guide - Part 11: Implementation and Evaluation of MR-MPS-SW: 2016 in conjunction with CMMI-DEV v1.3*. Retrieved from <http://www.softex.br/mpsbr>
- Softex. (2016c). *MPS.BR - Brazilian Software Process Improvement, MR-MPS-SW - General Guide*. Retrieved from <http://www.softex.br/mpsbr>
- Souza, C. S., Baranauskas, M. C. C., Prates, R. O., & Pimenta, M. S. (2008). HCI in Brazil: Lessons Learned and New Perspectives. In *Proceedings of the VIII Brazilian Symposium on Human Factors in Computing Systems* (pp. 358–359). Porto Alegre, Brazil, Brazil: Sociedade Brasileira de Computação.

- Souza, C. S., Leitão, C. F., Prates, R. O., & Silva, E. J. (2006). The Semiotic Inspection Method. In *Proceedings of VII Brazilian Symposium on Human Factors in Computing Systems* (pp. 148–157). New York, NY, USA: ACM. <https://doi.org/10.1145/1298023.1298044>
- Staggers, N., & Rodney, M. (2012). Promoting Usability in Organizations with a New Health Usability Model: Implications for Nursing Informatics. In *NI 2012: Proceedings of the 11th International Congress on Nursing Informatics*. Retrieved from <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3799150/>
- Sward, D., & Macarthur, G. (2007). Making user experience a business strategy. In *Proceedings of the Workshop on Towards a UX Manifesto* (pp. 35–40). Lancaster, UK.
- Taylor, B., Gupta, A., McClelland, I., van Gelderen, T., & Hefley, B. (1998). HumanWare process improvement – institutionalising the principles of user-centred design. In *Tutorial in XIII Human-Computer Interaction Conference on People and Computers*.
- Thanh-Diane, N., Vanderdonckt, J., & Seffah, A. (2016). UIPLML: Pattern-based engineering of user interfaces of multi-platform systems. In *IEEE Tenth International Conference on Research Challenges in Information Science* (Vol. 1, pp. 273–284). Grenoble, France: IEEE. <https://doi.org/10.1109/RCIS.2016.7549348>
- Thayer, R. H., & McGettrick, A. D. (1993). *Software Engineering: A European Perspective*. Los Alamitos, CA, USA: IEEE Computer Society Press.
- Tidwell, J. (1999). *A Pattern Language for Human-Computer Interface Design*. Retrieved from http://ewriting.narod.ru/SamplesAndWP/Pattern_Language_for_HCI.pdf
- Tidwell, J. (2010). *Designing Interfaces - Patterns for Effective Interaction Design* (2nd Edition). O'Reilly Media.
- Trujillo, M. M., Oktaba, H., Ventura, T., & Torres, R. (2013). From MoProSoft Level 2 to ISO/IEC 29110 Basic Profile: Bridging the Gap. *CLEI Electronic Journal*, 16(1), Paper 2.
- UXPA Curitiba. (2016). *Profile of UX professionals in Brazil 2015*. UXPA - User Experience Professionals Association Curitiba. Retrieved from <http://uxpacuritiba.blogspot.fr/>
- Van Tyne, S. (2009). Corporate User-Experience Maturity Model. In M. Kurosu (Ed.), *Proceedings of the first International Conference on Human Centered Design* (pp. 635–639). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-02806-9_74
- 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.
- Vanderdonckt, J. (1995). *Guide ergonomique des interfaces homme-machine*. Namur, Belgium: Presses Universitaires de Namur.
- Vasmatazidis, I., Ramakrishnan, A., & Hanson, C. (2001). Introducing Usability Engineering into the Cmm Model: An Empirical Approach. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 45(24), 1748–1752. <https://doi.org/10.1177/154193120104502410>
- Veer, G. C. van der, Lenting, B. F., & Bergevoet, B. A. J. (1996). GTA: Groupware task analysis — Modeling complexity. *Acta Psychologica*, 91(3), 297–322. [https://doi.org/10.1016/0001-6918\(95\)00065-8](https://doi.org/10.1016/0001-6918(95)00065-8)
- Venturi, G., Troost, J., & Jokela, T. (2006). People, Organizations, and Processes: An Inquiry into the Adoption of User-Centered Design in Industry. *International Journal of Human-Computer Interaction*, 21(2), 219–238. https://doi.org/10.1207/s15327590ijhc2102_6
- Vermeeren, A. P. O. S., Law, E. L.-C., Roto, V., Obrist, M., Hoonhout, J., & Väänänen-Vainio-Mattila, K. (2010). User Experience Evaluation Methods: Current State and Development Needs. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries* (pp. 521–530). New York, NY, USA: ACM. <https://doi.org/10.1145/1868914.1868973>
- Vukelja, L., Müller, L., & Opwis, K. (2007). Are Engineers Condemned to Design? A Survey on Software Engineering and UI Design in Switzerland. In C. Baranauskas, P. Palanque, J. Abascal, & S. D. J. Barbosa (Eds.), *Human-Computer Interaction – INTERACT 2007: 11th IFIP TC 13 International Conference* (Vol. Part II, pp. 555–568). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-74800-7_50
- Wangenheim, C. G., Hauck, J. C. R., Salviano, C. F., & von Wangenheim, A. (2010). Systematic Literature Review of Software Process Capability/Maturity Models. In *Proceedings of International SPICE Conference on Process Improvement and Capability dEtermination*.
- Warin, B., Talbi, O., Kolski, C., & Hoogstoël, F. (2016). Multi-Role Project (MRP): a new Project-based Learning Method for STEM. *IEEE Transactions on Education*, 59(2), 137–146.

- Wewers, M. E., & Lowe, N. K. (1990). A critical review of visual analogue scales in the measurement of clinical phenomena. *Research in Nursing & Health*, 13(4), 227–236. <https://doi.org/10.1002/nur.4770130405>
- Winckler, M. A., & Pimenta, M. S. (2004). Análise e Modelagem de Tarefas. In *Tutorial in Congresso Brasileiro de Fatores Humanos em Sistemas Computacionais*. Curitiba, Brazil.
- Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2012). *Experimentation in Software Engineering*. Springer-Berlin Heidelberg.
- Woltjer, R., & Hollnagel, E. (2008). Modeling and evaluation of air traffic management automation using the functional resonance accident model (FRAM). In *Proceedings of 8th International Symposium of the Australian Aviation Psychology Association*. Sydney, Australia.
- World Wide Web Consortium. (2014). MBUI - Task Models. Retrieved from <https://www.w3.org/TR/task-models/>.
- World Wide Web Consortium. (2015). WAI Guidelines and Techniques. Retrieved from <https://www.w3.org/WAI/guid-tech.html>.
- Young, W., & Leveson, N. G. (2014). An Integrated Approach to Safety and Security Based on Systems Theory. *Communications of the ACM*, 57(2), 31–35. <https://doi.org/10.1145/2556938>
- Yousuf, M. I. (2007). Using Experts' Opinions Through Delphi Technique. *Practical Assessment, Research & Evaluation*, 12(4), 1–7.

Annexes

Annex A. Analysis of CMMI-DEV

A.1. Specific Practices of Requirements Development process area

Requirements Development (RD)		Type of citation
Specific Practice	Information of HCI	
SP1.1: Elicit stakeholder needs, expectations, constraints, and interfaces for all phases of the product lifecycle.	<p><i>Subpractice 1:</i> “Engage relevant stakeholders using <u>methods for eliciting needs, expectations, constraints, and external interfaces.</u>”</p> <p><i>Examples of techniques:</i> “Questionnaires, interviews, and <u>scenarios obtained from end users</u>”, “<u>end-user task analysis</u>” and “<u>prototypes and models.</u>”</p>	Explicit
SP1.2: Transform stakeholder needs, expectations, constraints, and Interfaces into prioritized customer requirements.	<p><i>Subpractice 1:</i> “Translate stakeholder needs, expectations, constraints, and interfaces into documented <u>customer requirements.</u>”</p> <p><i>Subpractice 2:</i> “Establish and maintain a prioritization of customer functional and <u>quality attribute requirements.</u>”</p> <p><i>Example Work Products:</i> “Prioritized <u>customer requirements</u>”</p>	Implicit
SP2.1: Establish and maintain product and product component requirements, which are based on the customer requirements.	<i>Subpractice 3:</i> “Develop <u>architectural requirements</u> capturing <u>critical quality attributes</u> and quality attribute measures necessary for establishing the product architecture and design.”	Implicit
SP2.2: Allocate the requirements for each product component.	-	-
SP2.3: Identify interface requirements.	-	-
SP3.1: Establish and maintain operational concepts and associated scenarios.	<i>Subpractice 4:</i> “Develop a <u>detailed operational concept</u> , as products and product components are selected, that defines the interaction of the product, the end user, and the environment, and that satisfies the operational, maintenance, support, and disposal needs.”	Explicit
SP3.2: Establish and maintain a definition of required functionality and quality attributes.	<i>Subpractice 2:</i> “Identify desirable functionality and <u>quality attributes.</u> ”	Implicit
SP3.3: Analyze requirements to ensure that they are necessary and sufficient.	<i>Subpractice 1:</i> “ <u>Analyze stakeholder needs, expectations, constraints, and external interfaces</u> to organize them into related subjects and remove conflicts.”	Implicit
SP3.4: Analyze requirements to balance stakeholder needs and constraints.	<i>Subpractice 1:</i> “ <u>Use proven models, simulations,</u> and prototyping to analyze the balance of stakeholder needs and constraints.”	Explicit

Requirements Development (RD)		Type of citation
Specific Practice	Information of HCI	
SP3.5: Validate requirements to ensure the resulting product will perform as intended in the end user's environment.	<p><i>Subpractice 3:</i> “Assess the design as it matures in the context of the requirements validation environment to identify validation issues and expose unstated needs and customer requirements.”</p> <p><i>Example of technique:</i> “<u>Prototyping</u>”.</p>	Explicit

A.2. Specific Practices of Technical Solution process area

Technical Solution (TS)		Type of citation
Specific Practice	Information of HCI	
SP1.1: Develop alternative solutions and selection criteria.	<i>Subpractice 4:</i> “Identify re-usable solution components or applicable <u>architecture patterns</u> .”	Implicit
SP1.2: Select the product component solutions based on selection criteria.	<i>Subpractice 1:</i> “Evaluate each alternative solution/set of solutions against the selection criteria established in the context of the <u>operational concepts and scenarios</u> .” <i>Additional information:</i> “Develop timeline scenarios for product operation and user interaction for each alternative solution.”	Implicit
SP2.1: Develop a design for the product or product component.	<i>Subpractice 1:</i> “Establish and maintain criteria against which the design can be evaluated.” Example of quality attribute: “ <u>Usable</u> ”.	Explicit
	<i>Subpractice 2:</i> “Identify, develop, or acquire the <u>design methods</u> appropriate for the product.” Examples of techniques and methods: “ <u>Prototypes</u> ”.	Implicit
	<i>Subpractice 3:</i> “Ensure that the design adheres to applicable <u>design standards and criteria</u> .” <i>Example of design standard:</i> “Operator interface standards.”	Implicit
	<i>Additional information:</i> Examples of architecture definition tasks include: “Selecting <u>architectural patterns</u> that support the functional and quality attribute requirements, and instantiating or composing those patterns to create the product architecture”.	
SP2.2: Establish and maintain a technical data package.	-	-
SP2.3: Design product component interfaces using established criteria.	-	-
SP2.4: Evaluate whether the product components should be developed, purchased, or reused based on established criteria.	-	-
SP3.1: Implement the designs of the product components.	<i>Subpractice 1:</i> “Use effective methods to implement the product components.” Example of method: “Use of applicable <u>design patterns</u> ”	Implicit
SP3.2: Develop and maintain the end-use documentation.	<i>Subpractice 3:</i> “Adhere to the applicable <u>documentation standards</u> .” <i>Example of documentation standards:</i> “Consistency with a designated style manual”. <i>Example Work Products:</i> “End-user training materials”, “User's manual”.	Implicit

A.3. Specific Practices of Product Integration process area

Product Integration (PI)		<i>Type of citation</i>
<i>Specific Practice</i>	<i>Information of HCI</i>	
SP1.1: Establish and maintain a product integration strategy.	<i>Additional information:</i> “A product integration strategy addresses items such as: using models, <u>prototypes</u> , and simulations to assist in evaluating an assembly, including its interfaces.”	Explicit
SP1.2: Establish and maintain the environment needed to support the integration of the product components.	-	-
SP1.3: Establish and maintain procedures and criteria for integration of the product components.	-	-
SP2.1: Review interface descriptions for coverage and completeness.	-	-
SP2.2: Manage internal and external interface definitions, designs, and changes for products and product components.	-	-
SP3.1: Confirm, prior to assembly, that each product component required to assemble the product has been properly identified, behaves according to its description, and that the product component interfaces comply with the interface descriptions.	-	-
SP3.2: Assemble product components according to the product integration strategy and procedures.	-	-
SP3.3: Evaluate assembled product components for interface compatibility.	-	-
SP3.4: Package the assembled product or product component and deliver it to the customer.	-	-

A.4. Specific Practices of Verification process area

Verification (VER)		Type of citation
Specific Practice	Information of HCI	
SP1.1: Select work products to be verified and verification methods to be used.	<i>Subpractice 4:</i> “Define <u>verification methods</u> to be used for each selected work product.” <i>Additional information:</i> “Verification for systems engineering typically includes <u>prototyping</u> , modeling, and simulation to verify adequacy of system design (and allocation).”	Implicit / Explicit
SP1.2: Establish and maintain the environment needed to support verification.	<i>Subpractice 3:</i> Identify <u>verification equipment and tools</u> .	Implicit
SP1.3: Establish and maintain verification procedures and criteria for the selected work products.	<i>Subpractice 2:</i> “Develop and refine <u>verification criteria</u> as necessary.” <i>Example of source for verification criteria:</i> “ <u>Standards</u> .”	Implicit
SP2.1: Prepare for peer reviews of selected work products.	<i>Subpractice 1:</i> “Determine the type of peer review to be conducted.” <i>Examples of types of peer reviews:</i> “ <u>Inspections, Structured walkthroughs</u> .”	Implicit
SP2.2: Conduct peer reviews of selected work products and identify issues resulting from these reviews.	<i>Additional information:</i> “Peer reviews should address the following guidelines: there should be sufficient preparation, the conduct should be managed and controlled, consistent and sufficient data should be recorded (an example is conducting a <u>formal inspection</u>), and action items should be recorded.”	Implicit
SP2.3: Analyze data about the preparation, conduct, and results of the peer reviews.	<i>Subpractice 1:</i> “Record data related to the preparation, conduct, and results of the peer reviews.” <i>Additional information:</i> “Typical data are product name, product size, composition of the peer review team, <u>type of peer review</u> , preparation time per reviewer, length of the review meeting, number of defects found, type and origin of defect, and so on.”	Implicit
SP3.1: Perform verification on selected work products.	<i>Subpractice 4:</i> “Document the “as-run” <u>verification method</u> and deviations from available methods and procedures discovered during its performance.”	Implicit
SP3.2: Analyze results of all verification activities.	<i>Subpractice 2:</i> “Based on the established verification criteria, identify products that do not meet their requirements or identify problems with <u>methods, procedures, criteria, and the verification environment</u> .”	Implicit

A.5. Specific Practices of Validation process area

Validation (VAL)		Type of citation
Specific Practice	Information of HCI	
SP1.1: Select work products to be verified and verification methods to be used.	<i>Subpractice 3:</i> “Select the product and product components to be validated.” <i>Examples of products and product components that can be validated:</i> “ <u>User interfaces</u> , User manuals.”	Explicit
	<i>Subpractice 4:</i> “Select the <u>evaluation methods</u> for product or product component validation.” <i>Examples of validation methods:</i> “Discussions with end users perhaps in the context of a formal review, <u>Prototype demonstrations</u> .”	Explicit
SP1.2: Establish and maintain the environment needed to support validation.	<i>Subpractice 3:</i> “Identify <u>test equipment and tools</u> .”	Implicit
SP1.3: Establish and maintain procedures and criteria for validation.	<i>Subpractice 2:</i> “Document the environment, operational scenario, procedures, inputs, outputs, and <u>criteria for the validation</u> of the selected product or product component.” Example of source for validation criteria: “ <u>Standards</u> .”	Implicit
SP2.1: Perform validation on selected products and product components.	<i>Additional information:</i> “Validation activities are performed and the resulting data are collected according to <u>established methods, procedures, and criteria</u> .”	Implicit
SP2.2: Analyze results of validation activities.	<i>Subpractice 2:</i> “Based on the established validation criteria, identify products and product components that do not perform suitably in their intended operating environments, or identify problems with <u>methods, criteria, or the environment</u> .”	Implicit

Annex B. 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 was carried out an analysis of the Software Process Capability and Maturity Model (Capability Maturity Model Integration (CMMI-DEV) from the point 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 our answer, please.***

Respondent information

Name:

Date:

Formation and Profession:

The working period in the HCI area:

CMMI Model and Engineering Process Areas

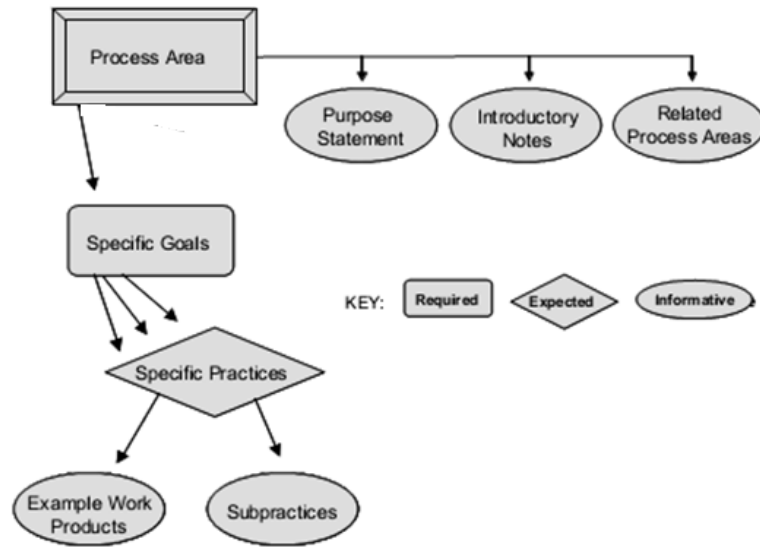


Figure 1. CMMI Model Components (CMMI Product Team, 2010)

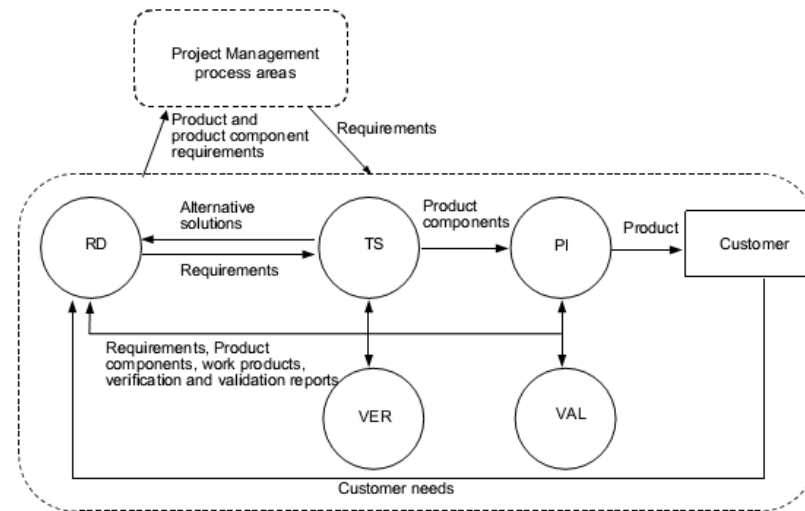


Figure 2. Engineering Process Areas (CMMI Product Team, 2010)

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 <i>SG 1 Develop Customer Requirements</i> Stakeholder needs, expectations, constraints, and interfaces are collected and translated into customer requirements.	SP 1.1 Elicit Needs Elicit stakeholder needs, expectations, constraints, and interfaces for all phases of the product lifecycle.	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 <i>SG 1 Develop Customer Requirements</i> Stakeholder needs, expectations, constraints, and interfaces are collected and translated into customer requirements.	SP 1.2 Transform Stakeholder Needs into Customer Requirements Transform stakeholder needs, expectations, constraints, and interfaces into prioritized 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 <i>SG 2 Develop Product Requirements</i> 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 <i>SG 3 Analyze and Validate Requirements</i> 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	Standards and Guidelines for design and documentation of				

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	
<i>SG 3 Analyze and Validate Requirements</i> The requirements are analyzed and validated.	Definition of Required Functionality and Quality Attributes Establish and maintain a definition of required functionality and quality attributes.	HCI Examples: • Ergonomic Criterion (Scapin and Bastien, 1993) • ISO/IEC 9126-1 (2001) • ISO 9241-11 (1998) • ISO/IEC 25000 (2014)				
	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 <ul style="list-style-type: none"> <input type="checkbox"/> Offline techniques: Paper and pencil (paper sketches, storyboards), Mockups, Wizard of Oz, Video prototyping <input type="checkbox"/> Online techniques using software tools: No interactive simulations, Interactive simulations, Scripting languages 				
Technical Solution <i>SG 1 Select Product</i>	SP 1.1 Develop Alternative Solutions and	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 <i>SG 2 Develop the Design</i> 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)				
	SP 3.1 Implement the Design	Design patterns for HCI				
Technical Solution <i>SG 3 Implement the Product</i>						

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 <i>SG 1 Prepare for Product Integration</i> 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 <input type="checkbox"/> Offline techniques: Paper and pencil (paper sketches, storyboards), Mockups, Wizard of Oz, Video prototyping <input type="checkbox"/> 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 <input type="checkbox"/> User interface toolkits <input type="checkbox"/> User interface builders <input type="checkbox"/> 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 <input type="checkbox"/> Exploratory tests <input type="checkbox"/> Assessment tests <input type="checkbox"/> Validation or verification tests <input type="checkbox"/> 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 <input type="checkbox"/> User interface toolkits <input type="checkbox"/> User interface builders <input type="checkbox"/> 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 <input type="checkbox"/> Exploratory tests <input type="checkbox"/> Assessment tests <input type="checkbox"/> Validation or verification tests <input type="checkbox"/> 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)				
	SP 2.1 Perform Validation Perform validation on	Evaluation methods for HCI verification tests				
Validation <i>SG 2 Validate Product or</i>						

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 <input type="checkbox"/> Exploratory tests <input type="checkbox"/> Assessment tests <input type="checkbox"/> Validation or verification tests <input type="checkbox"/> 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 <input type="checkbox"/> Exploratory tests <input type="checkbox"/> Assessment tests <input type="checkbox"/> Validation or verification tests <input type="checkbox"/> Comparison tests • Validation by HCI expert(s)				
Verification	SP 1.1 Select Work	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 <input type="checkbox"/> Exploratory tests <input type="checkbox"/> Assessment tests <input type="checkbox"/> Validation or verification tests <input type="checkbox"/> 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 <input type="checkbox"/> User interface toolkits <input type="checkbox"/> User interface builders <input type="checkbox"/> 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 <input type="checkbox"/> Exploratory tests <input type="checkbox"/> Assessment tests <input type="checkbox"/> Validation or verification tests <input type="checkbox"/> 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)				
	SP 2.1 Prepare for Peer Reviews Prepare for peer reviews of	Evaluation methods for HCI review				
Verification <i>SG 2 Perform Peer Reviews</i> Peer reviews are performed on						

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				
	SP 3.1 Perform Verification Perform verification on selected work products.	Evaluation methods for HCI verification tests Examples: • Usability tests <input type="checkbox"/> Exploratory tests <input type="checkbox"/> Assessment tests <input type="checkbox"/> Validation or verification tests <input type="checkbox"/> Comparison tests • Validation by HCI expert(s)				
Verification <i>SG 3 Verify Selected Work Products</i> Selected work products are verified against their specified requirements.	SP 3.2 Analyze Verification Results Analyze results of all verification activities.	Evaluation methods for HCI verification tests Examples: • Usability tests <input type="checkbox"/> Exploratory tests <input type="checkbox"/> Assessment tests <input type="checkbox"/> Validation or verification tests <input type="checkbox"/> Comparison tests • Validation by HCI expert(s)				

Other suggestions:

Annex C. Evaluation questionnaire³³

Preamble: We started with a course that could be described as classic, associated with different supports. Then, in supervised work classes, I proposed an active pedagogy, supported by the performance of a collective mini-project. To improve this pedagogy, I would like to know how you feel about it. With this in mind, I would like you to fill out the following questionnaire. The questionnaire responses will be used only for research purposes and anonymously. Thank you in advance for your help.

General profile

1. Gender:

2. Work investment - About your work investment in the master, you consider yourself as:

- a. Good worker and perfectionist
- b. Good worker
- c. Just enough to achieve the goal (The average in an exam for example)
- d. Irregular
- e. Carefree

3. Working method - You evaluate yourself as:

- a. Very methodical
- b. Methodical
- c. Pragmatic
- d. Carefree

4. Work preference - When you have the choice, you prefer to work:

- a. Individually
- b. In pairs
- c. In a team

5. Freedom of action - When doing the work, you prefer to:

- a. Be guided from the start and throughout the work
- b. Be guided in part of the work
- c. Have the goal and the main lines of resolution, then work freely
- d. Not be guided

The proposed mini-project

1. Initial interest - You can say that the theme of the project initially aroused:

- a. Enthusiasm
- b. Interest
- c. As a constraint
- d. As a punishment

2. Subject comprehension - About your comprehension, you think the subject was:

- a. Too detailed
- b. Well detailed
- c. Not explicit enough
- d. Incomprehensible

3. Difficulty of the work - You consider the work to be done:

- a. Too difficult
- b. Difficult
- c. At the right level
- d. Easy

4. Time for performance - Compared to the work required to complete the mini-project, you consider the time for performance spent in the supervised work classes was:

- a. Very important
- b. At the right level
- c. Not relevant enough
- d. Not at all sufficient

³³ This evaluation questionnaire was proposed by Bruno Warin (University of Littoral Côte d'Opale, Calais, France).

The pedagogy

- 1. Initial interest** - You can say that the obligation to respect a scenario/methodology initially aroused:
 - a. Enthusiasm
 - b. Interest
 - c. As a constraint
 - d. As a punishment
- 2. Study of the scenario/method** – Did you read the scenario/method (in relation to project subject)?
 - a. I read very carefully
 - b. I read with average attention
 - c. I read little or nothing
- 3. Understanding alone** – You think the scenario/method (project subject) is:
 - a. Very easy to understand by yourself
 - b. Easy to understand by yourself
 - c. Difficult to understand by yourself
 - d. Very difficult to understand by yourself
- 4. Understanding in group** - You think the scenario/method (project subject) is:
 - a. Very easy to understand in a group
 - b. Easy to understand in a group
 - c. Difficult to understand in a group
 - d. Very difficult to understand in a group
- 5. Participation thanks to the scenario/method** - Compared to sessions where the teacher presents the knowledge to learn on the “blackboard” (video presentation), do you think the scenario/method makes the supervised classes more motivating and encourages greater participation?
 - a. Absolutely
 - b. Almost sure
 - c. Probably not
 - d. Not at all
- 6. Utility of scenario/method** - You think the scenario/method is:
 - a. Very relevant for achieving the learning of the subject / subjects studied in class
 - b. Relevant
 - c. Irrelevant
 - d. Useless
- 7. Group meeting organization** - Were the group meetings organized (designation of a facilitator, a rapporteur, agenda, duration, time of individualized speech, etc.):
 - a. Always
 - b. Often
 - c. Rarely
 - d. Never
- 8. Frequency of course assessment** - Do you think that regular assessments encourage better learning than an overall assessment at the end of the course?
 - a. Absolutely
 - b. Almost sure
 - c. Probably not
 - d. Not at all
- 9. Scenario/method understanding** - You think the scenario/method is:
 - a. Very easy to apply
 - b. Easy to apply
 - c. Difficult to apply
 - d. Impossible to apply
- 10. Scenario/method application** - Did you apply the scenario/method?
 - a. Absolutely
 - b. Practically yes
 - c. Not exactly
 - d. Not at all
- 11. Quality of the report** – Has the application of scenario/method favored the quality of the final product (the report)?
 - a. Yes
 - b. No

12. Knowledge provided by teachers - Is the knowledge acquired by your group or the course given by the teacher before the project sufficient to do the required work?

- a. Absolutely
- b. Largely
- c. A little
- d. Not at all

The evaluation

1. Workload - Does the system of evaluation by report seem cumbersome?

- a. Absolutely
- b. Binding but supportable
- c. Binding but easy to integrate into your training workload
- d. Not at all

2. Relevance – Does the evaluation system seems relevant to promote learning?

- a. Absolutely
- b. Highly pertinent
- c. Not very pertinent
- d. Not at all

3. Preference of a single exam - Would you have preferred a global exam instead of an exam and the project report?

- a. Absolutely
- b. Strongly
- c. A little
- d. Not at all

Annex D. Questionnaire for Peer review

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 Peer review

Domain: Methods, techniques, standards and patterns of Software Engineering

Taísa Guidini Gonçalves
Kathia Oliveira
Christophe Kolski

September 2016

Questionnaire for Peer review - Methods, techniques, standards and patterns of Software Engineering

This peer review aims to improve the set of methods, techniques, standards and patterns of Software Engineering suitable to support the practices defined in five process areas from CMMI-DEV (**Requirements Development (RD)**, **Technical Solution (TS)**, **Product Integration (PI)**, **Validation (VAL)** and **Verification (VER)**). Based on [1, 7, 8] we defined fourteen groups of approaches with several examples of methods, techniques, standards and patterns.

Do you suggest any other example of approach?

Respondent information

Name:

Date:

Formation and Profession:

Working period in the SE domain:

Process Area and Specific Practice (CMMI-DEV)	Potential methods, techniques, standards and patterns from Software Engineering (SE)	What else?
Requirements Development <i>SP 1.1 Elicit Needs</i>	Techniques to identify needs Examples: ([1] pp. 329 – [8] pp. 11, 12) • Brainstorming • Interviews • Field Studies/Observation • Questionnaires	
Requirements Development <i>SP 1.1 Elicit Needs</i> <i>SP 1.2 Transform Stakeholder Needs into Customer Requirements</i> <i>SP 3.1 Establish Operational Concepts and Scenarios</i>	Techniques to identify requirements Examples: ([1] pp. 329, 336 – [8] pp. 11, 12) • Scenario • Use cases • User stories • Storyboards • Task Analysis • Quality Function Deployment • FAST (Facilitated Application Specification Techniques) technique: JAD, The Method	
Requirements Development <i>SP 1.1 Elicit Needs</i> <i>SP 1.2 Transform Stakeholder Needs into Customer Requirements</i> <i>SP 2.1 Establish Product and Product Component Requirement</i> <i>SP 3.3 Analyze Requirements</i>	Software Modeling Examples: ([1] pp. 329, 338 – [8] pp. 12) • Business case analysis • Suitable UML diagrams (see Table 53) • HTA (Hierarchical Task Analysis) • SADT (Structured Analysis and Design Technique)	
Requirements Development <i>SP 1.2 Transform Stakeholder Needs into Customer Requirements</i> <i>SP 2.1 Establish Product and Product</i> <i>SP 3.2 Establish a Definition of Required Functionality and Quality Attributes</i>	Standards and Guidelines for design	

Process Area and Specific Practice (CMMI-DEV)	Potential methods, techniques, standards and patterns from Software Engineering (SE)	What else?
Technical Solution <i>SP 2.1</i> Design the Product or Product Component Verification <i>SP 1.3</i> Establish Verification Procedures and Criteria Validation <i>SP 1.3</i> Establish Validation	Examples: ([1] pp. 331, 332, 337, 381, 382, 398, 405 – [8] pp. 12, 48, 58) • ISO/IEC 9126-1 (2001) • ISO/IEC 25000 (2014) • Accessibility standards and guidelines (WAI-W3C)	
Requirements Development <i>SP 1.2</i> Transform Stakeholder Needs into Customer Requirements <i>SP 3.5</i> Validate Requirements	Prototype for requirements Examples: ([1] pp. 340 – [8] pp. 18) • Paper Prototyping/Sketches • Storyboards • Wireframes • Mockups • Wizard of Oz • Video prototyping	
Requirements Development <i>SP 3.4</i> Analyze Requirements to Achieve Balance <i>SP 3.5</i> Validate Requirements	Techniques to validate requirements Examples: ([1] pp. 339, 340 – [8] pp. 18) • Analysis • Simulations • Demonstrations • Thinking Aloud	
Technical Solution <i>SP 1.1</i> Develop Alternative Solutions and Selection Criteria <i>SP 2.1</i> Design the Product or Product Component <i>SP 3.1</i> Implement the Design	Architecture Patterns for SE Examples: ([1] pp. 378, 381, 388 – [8] pp. 36, 38, 40) • MVC (Model-View-Controller) Model • 3-Tier Model • Pipes and Filters • Suitable UML diagrams (see Table 53)	
Technical Solution	Design Patterns for SE	

Process Area and Specific Practice (CMMI-DEV)	Potential methods, techniques, standards and patterns from Software Engineering (SE)	What else?
SP 3.1 Implement the Design	<p>Examples: ([1] pp. 388 – [8] pp. 40)</p> <ul style="list-style-type: none"> • Design Patterns: Elements of Reusable Object-Oriented Software (Gamma <i>et al.</i>, 1994) • GRASP - General Responsibility Assignment Software Patterns (Larman, 2004) • Head First Design Patterns (Freeman <i>et al.</i>, 2004) • Patterns of Enterprise Application Architecture (Fowler, 2002) 	
Technical Solution SP 1.1 Develop Alternative Solutions and Selection Criteria	Interaction modeling for SE <p>Examples: ([1] pp. 329 – [8] pp. 11, 12)</p> <ul style="list-style-type: none"> • Suitable UML diagrams (see Table 53) 	
Technical Solution SP 3.2 Develop Product Support Documentation	Techniques for final documentation <p>Examples: ([1] pp. 390 – [8] pp. 41)</p> <ul style="list-style-type: none"> • Style manual • ISO/IEC 26514 (2008) 	
Technical Solution SP 2.1 Design the Product or Product Component Product Integration SP 1.1 Establish an Integration Strategy Verification SP 1.1 Select Work Products for Verification Validation SP 1.1 Select Products for Validation	Prototype (system versions) <p>Examples: ([1] pp. 382, 395, 396, 404)</p> <ul style="list-style-type: none"> • User interface toolkits • User interface builders • User interface development environments 	
Verification SP 1.1 Select Work Products for Verification SP 1.2 Establish the Verification Environment SP 3.1 Perform Verification SP 3.2 Analyze Verification Results	Verification methods <p>Examples: ([1] pp. 404, 405, 409, 410 – [8] pp. 56, 59, 60)</p> <ul style="list-style-type: none"> • Unit test • Integration test • System test • Acceptance test • Installation test 	

Process Area and Specific Practice (CMMI-DEV)	Potential methods, techniques, standards and patterns from Software Engineering (SE)	What else?
Verification <i>SP 2.1</i> Prepare for Peer Reviews <i>SP 2.2</i> Conduct Peer Reviews <i>SP 2.3</i> Analyze Peer Review Data	Review methods	
	Examples: ([1] pp. 406, 407, 408, 409 – [8] pp. 56, 59, 60) • Inspections • Structured walkthroughs • Pair programming • Guidelines review • Audits	
Validation <i>SP 1.1</i> Select Products for Validation <i>SP 1.2</i> Establish the Validation Environment <i>SP 2.1</i> Perform Validation <i>SP 2.2</i> Analyze Validation Results	Validation methods	
	Examples: ([1] pp. 396, 397, 399 – [8] pp. 47, 48, 50) • Formal review • Tests of products (by end user/ stakeholders) • Analyses of product • Functional demonstrations	

References

- 1 CMMI Product Team. 2010. CMMI® for Development (CMMI-DEV), V1.3, (CMU/SEI-2010th-TR-033 ed.). Pittsburgh, USA: Carnegie Mellon University.
- 2 Fowler, M. 2002. Patterns of Enterprise Application Architecture. Addison-Wesley.
- 3 Freeman, E., Freeman, E., Bates, B., Sierra, K. 2004. Head First Design Patterns. O'Reilly.
- 4 Gamma, E., Helm, R., Johnson, R., Vlissides, J. 1994. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley.
- 5 ISO/IEC. 2008. ISO/IEC 26514:2008 – Systems and software engineering – Requirements for designers and developers of user documentation.
- 6 Larman, C. 2004. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. New Jersey: Prentice Hall.
- 7 Softex. 2016. MPS.BR – Melhoria de Processo do Software Brasileiro – Guia Geral MPS de Software. Retrieved April, 2016 from <http://www.softex.br>.
- 8 Softex. 2016. MPS.BR – Melhoria de Processo do Software Brasileiro – Guia de Implementação - Parte 4: Fundamentação para Implementação do Nível D do MR-MPS-SW:2016. Retrieved April, 2016 from <http://www.softex.br>.

Table 53. Suitable UML diagrams

Process area	UML Diagrams
RD - SP 1.1	Use case, Activity diagram
RD - SP 1.2	Use case, Activity diagram
RD - SP 2.1	Use case, Activity diagram, Class diagram, Sequence diagram, State machine diagram, Communication diagram
RD - SP 3.3	Use case, Activity diagram, Class diagram, Sequence diagram, State machine diagram, Timing diagram, Communication diagram
TS - SP 1.1	Component diagram, Interaction overview diagram
TS - SP 2.1	Class diagram, Component diagram, Deployment diagram
TS - SP 3.1	Use case, Timing diagram

Annex E. Web Questionnaire

Survey - Implementation of methods, techniques, standards and patterns of Human-Computer Interaction and Software Engineering

This survey aims to evaluate to what extent methods, techniques, standards and patterns of Software Engineering and Human-Computer Interaction are used by software developers that have implemented the maturity levels (A, B, C or D) of the MR-MPS-SW model (Reference Model MPS for Software) and/or the maturity levels (5, 4 or 3) of the CMMI-DEV model (Capability Maturity Model Integration for Development).

We would like to highlight that any publication generated from this survey will present only statistical results by summarizing the raw data and treating the answers anonymously. In other words, in any circumstances the answers provided in this survey will be published with personal information of the respondents or the institutions for which they work.

This research is part of a doctoral thesis which is being developed at University of Valenciennes and Hainaut-Cambresis and financed by the Brazilian government (Program Science without Borders/CAPES).

The survey is divided into 2 parts (described below) and the estimated time to fill it is 40 minutes.

- Part 1 - Characterization
- Part 2 - Evaluation of the Implementation of methods, techniques, standards and patterns of Human-Computer Interaction and Software Engineering

We really appreciate your help and time with this research.

Best regards,

Taísa Guidini Gonçalves
Káthia Marçal de Oliveira
Christophe Kolski

Part 1 - Characterization

For the characterization, please indicate the items listed below:

Respondent identification:	Enterprise employee						Consultant of software process capability maturity models						
E-mail													
Formation degree:	Doctor in Science (D.Sc.) or PhD		Master in Science (M.Sc.)		Specialist or MBA degree				Bachelor degree				
Formation area:	Software Engineering		Computer Science		Human-Computer Interaction				Other				
Are you an official implementer of the MR-MPS-SW model?	Yes						No						
Are you affiliated to an Implementing Institution (II)?	Yes						No						
	What?												
Did you take the official CMMI introduction course?	Yes						No						
How many years have you worked in Capability Maturity models implementations?													
Capability Maturity model(s) and maturity level(s) that you have supported implementations:	CMMI-DEV						MR-MPS-SW						
	5		4		3		A		B		C		D
Approximately, how many enterprises and projects you supported the implementation? (for the levels previously selected)	Enterprises						Projects						

Part 2 - Evaluation of the Implementation of methods, techniques, standards and patterns of Human-Computer Interaction and Software Engineering

The item listed above present several methods, techniques, standards and patterns from Software Engineering (SE) and Human-Computer Interaction (HCI) that can support the implementation of the Processes of MR-MPS-SW or of the Process Areas of CMMI-DEV, according to the literature and experts.

Please, indicate your level of knowledge (*I know*) and level of use (*I used*) to each one of those methods, techniques, standards and patterns when in the implementation of Capability Maturity models in enterprises you worked.

Example of scale of answer to each question

Answers
I Know:
None ----- A lot
I Used:
None ----- A lot

Question	Process (MR-MPS-SW)	Process Area (CMMI-DEV)	Potential methods, techniques, standards and patterns from Software Engineering (SE)	Potential methods, techniques, standards and patterns from Human Computer-Interaction (HCI)
1	Requirements Development DRE1 As necessidades, expectativas e restrições do cliente, tanto do produto quanto de suas interfaces, são identificadas	Requirements Development RD SP1.1 Elicit Needs	Techniques to identify needs	Techniques to identify user needs
			Examples (see References), not limited to: <ul style="list-style-type: none"> • Brainstorming • Interviews • Questionnaires • Card Sorting • Focus Groups • Field Studies/Observation • Workshops • Protocol Analysis 	Examples (see References), not limited to: <ul style="list-style-type: none"> • Brainstorming • Interviews • Surveys/Questionnaires • Card Sorting • Focus Groups • Field Studies/Observation
2	Requirements Development DRE1 As necessidades, expectativas e restrições do cliente, tanto do produto quanto de suas interfaces, são identificadas DRE2 Um conjunto definido de requisitos do cliente é especificado e priorizado a partir das necessidades, expectativas e restrições identificadas DRE6 Conceitos operacionais e cenários são desenvolvidos	Requirements Development RD SP1.1 Elicit Needs RD SP1.2 Transform Stakeholder Needs into Customer Requirements RD SP3.1 Establish Operational Concepts and Scenarios	Techniques to identify requirements	Techniques to identify user and organizational requirements
			Examples (see References), not limited to: <ul style="list-style-type: none"> • Scenario • User stories • Storyboards • Task Analysis • Use cases • Quality Function Deployment • FAST (Facilitated Application 	Examples (see References), not limited to: <ul style="list-style-type: none"> • Scenario • User stories • Storyboards • Task Analysis • Persona • Context-of-use analysis • User Profile (Detailed)

Question	Process (MR-MPS-SW)	Process Area (CMMI-DEV)	Potential methods, techniques, standards and patterns from Software Engineering (SE)	Potential methods, techniques, standards and patterns from Human Computer-Interaction (HCI)
			Specification Techniques): JAD, The Method	• Requirements specification templates (e.g. VOLERE, IEEE, RESCUE)
3	Requirements Development DRE1 As necessidades, expectativas e restrições do cliente, tanto do produto quanto de suas interfaces, são identificadas DRE2 Um conjunto definido de requisitos do cliente é especificado e priorizado a partir das necessidades, expectativas e restrições identificadas DRE3 Um conjunto de requisitos funcionais e não-funcionais, do produto e dos componentes do produto que descrevem a solução do problema a ser resolvido, é definido e mantido a partir dos requisitos do cliente DRE7 Os requisitos são analisados, usando critérios definidos, para balancear as necessidades dos interessados com as restrições existentes	Requirements Development RD SP1.1 Elicit Needs RD SP1.2 Transform Stakeholder Needs into Customer Requirements RD SP2.1 Establish Product and Product Component Requirements RD SP3.3 Analyze Requirements	Software Modeling	Task Modeling
			Examples (see References), not limited to: • HTA (Hierarchical Task Analysis) • SADT (Structured Analysis and Design Technique) • Business case analysis • Suitable UML diagrams (see UML diagrams)	Examples (see References), not limited to: • HTA (Hierarchical Task Analysis) • SADT (Structured Analysis and Design Technique) • CTT (Concur Task Tree) • K-MAD (Kernel of Model for Activity Description) • GTA (Groupware Task Analysis) • HAMSTERS notation • Task Model Standard (W3C)
4	Requirements Development DRE2 Um conjunto definido de requisitos do cliente é especificado e priorizado a partir das necessidades, expectativas e restrições identificadas DRE3 Um conjunto de requisitos funcionais e não-funcionais, do produto e dos componentes do produto que descrevem a solução do problema a ser resolvido, é definido e mantido a partir dos requisitos do cliente DRE4 Os requisitos funcionais e não-funcionais de cada componente do produto são refinados, elaborados e alocados Design and Construction of the Product PCP3 O produto e/ou componente do produto é projetado e documentado Validation	Requirements Development RD SP1.2 Transform Stakeholder Needs into Customer Requirements RD SP2.1 Establish Product and Product Component Requirements RD SP3.2 Establish a Definition of Required Functionality and Quality Attributes Technical Solution TS SP2.1 Design the Product or Product Component Validation VAL SP1.3 Establish Validation Procedures and Criteria Verification	Standards and Guidelines for design	Standards and Guidelines for HCI design
			Examples (see References), not limited to: • ISO/IEC 25000 (2014) • ISO/IEC 9126-1 (2001) • Accessibility standards and guidelines (WAI-W3C) • Domain-Specific Standards (Eg. security, critical systems, ...)	Examples (see References), not limited to: • ISO/IEC 25000 (2014) • ISO/IEC 9126-1 (2001) • ISO 9241-11 (1998) • Ergonomic Criterion (Scapin and Bastien, 1993; Vanderdonckt, 1994) • Accessibility standards and guidelines (WAI-W3C) • Nielsen's Heuristics • Golden Rules of Interface Design

Question	Process (MR-MPS-SW)	Process Area (CMMI-DEV)	Potential methods, techniques, standards and patterns from Software Engineering (SE)	Potential methods, techniques, standards and patterns from Human Computer-Interaction (HCI)
	VAL3 Critérios e procedimentos para validação dos produtos de trabalho a serem validados são identificados e um ambiente para validação é estabelecido Verification VER3 Critérios e procedimentos para verificação dos produtos de trabalho a serem verificados são identificados e um ambiente para verificação é estabelecido	VER SP1.3 Establish Verification Procedures and Criteria		
5	Requirements Development DRE2 Um conjunto definido de requisitos do cliente é especificado e priorizado a partir das necessidades, expectativas e restrições identificadas DRE8 Os requisitos são validados	Requirements Development RD SP1.2 Transform Stakeholder Needs into Customer Requirements RD SP3.5 Validate Requirements	Prototype for requirements Examples (see References), not limited to: • Paper Prototyping/Sketches • Storyboards • Wireframes • Mockups • Wizard of Oz • Video prototyping	Prototype for HCI requirements Examples (see References), not limited to: • Paper Prototyping/Sketches • Storyboards • Wireframes • Mockups • Wizard of Oz • Video prototyping
6	Requirements Development DRE7 Os requisitos são analisados, usando critérios definidos, para balancear as necessidades dos interessados com as restrições existentes DRE8 Os requisitos são validados	Requirements Development RD SP3.4 Analyze Requirements to Achieve Balance RD SP3.5 Validate Requirements	Techniques to validate requirements Examples (see References), not limited to: • Thinking Aloud • Analysis • Simulations • Demonstrations • User Testing (using Prototypes) • Perspective base-reading	Techniques to validate HCI requirements Examples (see References), not limited to: • Thinking Aloud • Proto Task (K-MAD) • Task Model Simulator (CTT) • Focus Group for evaluate requirements
7	Design and Construction of the Product PCP1 Alternativas de solução e critérios de seleção são desenvolvidos para atender aos requisitos definidos de produto e componentes de produto PCP3 O produto e/ou componente do produto é projetado e documentado PCP6 Os componentes do produto são implementados e verificados de acordo com o que foi projetado	Technical Solution TS SP1.1 Develop Alternative Solutions and Selection Criteria TS SP2.1 Design the Product or Product Component TS SP3.1 Implement the Design	Architecture Patterns for SE Examples (see References), not limited to: • MVC (Model-View-Controller) Model • Service-Oriented Architecture (SOA) • 3-Tier Model • Pipes and Filters • Suitable UML diagrams (see UML diagrams)	Architecture patterns for HCI Examples (see References), not limited to: • MVC (Model-View-Controller) Model • Arch Model (Bass et al., 1991) • Language Model • SEEHEIM Model (Pfaff, 1985) • PAC (Presentation-Abstraction-Control) Model • PAC-AMODEUS Model • CAMELEON-RT • Frameworks
8	Design and Construction of the Product	Technical Solution	Design Patterns for SE	Design patterns for HCI

Question	Process (MR-MPS-SW)	Process Area (CMMI-DEV)	Potential methods, techniques, standards and patterns from Software Engineering (SE)	Potential methods, techniques, standards and patterns from Human Computer-Interaction (HCI)
	PCP6 Os componentes do produto são implementados e verificados de acordo com o que foi projetado	TS SP3.1 Implement the Design	Examples (see References), not limited to: • Design Patterns: Elements of Reusable Object-Oriented Software • GRASP - General Responsibility Assignment Software Patterns • Head First Design Patterns • Patterns of Enterprise Application Architecture	Examples (see References), not limited to: • A Pattern Language for Human-Computer Interface Design • A Pattern Approach to Interaction Design • Pattern Languages in Interaction Design: Structure and Organization • Designing interfaces
9	Design and Construction of the Product PCP1 Alternativas de solução e critérios de seleção são desenvolvidos para atender aos requisitos definidos de produto e componentes de produto	Technical Solution TS SP1.1 Develop Alternative Solutions and Selection Criteria	Interaction modeling for SE	Techniques for interaction modeling
			Examples (see References), not limited to: • Suitable UML diagrams (see UML diagrams)	Examples (see References), not limited to: • MoLIC (Modeling Language for Interaction as Conversation) • UAN (User Action Notation) • TAG (Task-Action Grammar)
10	Design and Construction of the Product PCP7 A documentação é identificada, desenvolvida e disponibilizada de acordo com os padrões estabelecidos	Technical Solution TS SP3.2 Develop Product Support Documentation	Techniques for final documentation	Techniques for HCI documentation
			Examples (see References), not limited to: • Style manual • ISO/IEC 26514 (2008)	Examples (see References), not limited to: • Style guide • Architecture for help • Training Program
11	Design and Construction of the Product PCP3 O produto e/ou componente do produto é projetado e documentado Product Integration ITP1 Uma estratégia de integração, consistente com o projeto (design) e com os requisitos do produto, é desenvolvida e mantida para os componentes do produto Validation VAL1 Produtos de trabalho a serem validados são identificados VAL2 Uma estratégia de validação é desenvolvida e implementada, estabelecendo cronograma, participantes envolvidos, métodos para validação e qualquer material a ser utilizado na validação Verification	Technical Solution TS SP2.1 Design the Product or Product Component Product Integration PI SP1.1 Establish an Integration Strategy Validation VAL SP1.1 Select Products for Validation Verification VER SP1.1 Select Work Products for Verification	Prototype (system versions)	Iterative and Evolutionary Prototypes (system versions)
			Examples (see References), not limited to: • User interface toolkits • User interface builders • User interface development environments	Examples (see References), not limited to: • User interface toolkits • User interface builders • User interface development environments

Question	Process (MR-MPS-SW)	Process Area (CMMI-DEV)	Potential methods, techniques, standards and patterns from Software Engineering (SE)	Potential methods, techniques, standards and patterns from Human Computer-Interaction (HCI)
	VER1 Produtos de trabalho a serem verificados são identificados VER2 Uma estratégia de verificação é desenvolvida e implementada, estabelecendo cronograma, revisores envolvidos, métodos para verificação e qualquer material a ser utilizado na verificação			
12	Verification VER1 Produtos de trabalho a serem verificados são identificados VER2 Uma estratégia de verificação é desenvolvida e implementada, estabelecendo cronograma, revisores envolvidos, métodos para verificação e qualquer material a ser utilizado na verificação VER3 Critérios e procedimentos para verificação dos produtos de trabalho a serem verificados são identificados e um ambiente para verificação é estabelecido VER4 Atividades de verificação, incluindo testes e revisões por pares, são executadas VER6 Resultados de atividades de verificação são analisados e disponibilizados para as partes interessadas	Verification VER SP1.1 Select Work Products for Verification VER SP1.2 Establish the Verification Environment VER SP3.1 Perform Verification VER SP3.2 Analyze Verification Results	Verification methods Examples (see References), not limited to: <ul style="list-style-type: none"> • Unit test • Integration test • System test • Acceptance test • Installation test 	Evaluation methods for HCI verification Examples (see References), not limited to: <ul style="list-style-type: none"> • Unit test • Integration test • System test • Acceptance test • Installation test
13	Verification VER2 Uma estratégia de verificação é desenvolvida e implementada, estabelecendo cronograma, revisores envolvidos, métodos para verificação e qualquer material a ser utilizado na verificação VER4 Atividades de verificação, incluindo testes e revisões por pares, são executadas VER6 Resultados de atividades de verificação são analisados e disponibilizados para as partes interessadas	Verification VER SP2.1 Prepare for Peer Reviews VER SP2.2 Conduct Peer Reviews VER SP2.3 Analyze Peer Review Data	Review methods Examples (see References), not limited to: <ul style="list-style-type: none"> • Inspections • Structured walkthroughs • Guidelines review • Pair programming • Audits 	Evaluation methods for HCI review Examples (see References), not limited to: <ul style="list-style-type: none"> • Semiotic inspection • Formal usability inspection • Consistency inspection • Cognitive walkthrough • Groupware walkthrough • Guidelines review • Metaphors of human thinking (MOT) • Heuristic evaluation

Question	Process (MR-MPS-SW)	Process Area (CMMI-DEV)	Potential methods, techniques, standards and patterns from Software Engineering (SE)	Potential methods, techniques, standards and patterns from Human Computer-Interaction (HCI)
14	Validation VAL1 Produtos de trabalho a serem validados são identificados VAL2 Uma estratégia de validação é desenvolvida e implementada, estabelecendo cronograma, participantes envolvidos, métodos para validação e qualquer material a ser utilizado na validação VAL3 Critérios e procedimentos para validação dos produtos de trabalho a serem validados são identificados e um ambiente para validação é estabelecido VAL4 Atividades de validação são executadas para garantir que o produto esteja pronto para uso no ambiente operacional pretendido VAL6 Resultados de atividades de validação são analisados e disponibilizados para as partes interessadas	Validation VAL SP1.1 Select Products for Validation VAL SP1.2 Establish the Validation Environment VAL SP2.1 Perform Validation VAL SP2.2 Analyze Validation Results	Validation methods Examples (see References), not limited to: • Acceptance test with users • Formal review • Tests of products (by end user/stakeholders) • Analyses of product • Functional demonstrations	Evaluation methods for HCI validation Examples (see References), not limited to: • Usability testing • Communicability test • Standardized usability questionnaires • Post-experience interviews • User experience evaluation

Annex F. Form of evaluation

Form of evaluation of the pilot testing of the instrument

Survey - Implementation of methods, techniques, standards and patterns of Human-Computer Interaction (HCI) and Software Engineering (SE)

The aim of this pilot testing is to verify the facility to answer the questions about the application of HCI and SE methods, techniques, standards and patterns in the industry. To that end, please verify if the questions are clear, with no ambiguity. Moreover, if the layout used for the questionnaire is easy to understand.

In case of negative answer (**No**), please precisely justify in a way that we could correct it.

Date:
xx/xx/2016

Name:
xxxxxxxxxx

#	Questions	Answer	Justification
1	How long did you take to answer the questionnaire (in minutes)?		
2	Are the questions clear and easy to understand?		
3	Is the layout easy to understand?		
4	Are the instructions of the survey appropriate and consistent?		
5	Do you have any suggestion/criticism related to the survey?		

We appreciate your cooperation with this research.

Taísa Guidini Gonçalves
Káthia Marçal de Oliveira
Christophe Kolski