

Tailoring ISO/IEC 12207 for Usability Engineering*

CRISTINA MANRESA-YEE** and ESPERANÇA AMENGUAL**

Department of Mathematics and Computer Science, University of the Balearic Islands

Ctra. de Valldemossa, Km. 7.5, 07122—Palma de Mallorca, Spain. E-mail: {cristina.manresa, eamengual}@uib.es

Usability is a key aspect in diverse courses in Computer Engineering Education. Although usability has been analysed and discussed in both Software Engineering and Human-Computer Interaction disciplines, there is still a need to provide a global view of usability between these two perspectives that have evolved separately. Aiming at emphasizing on usability engineering, we have compiled the usability practices and techniques reported in the literature and present a mapping between these practices and the software life cycle processes defined in ISO/IEC 12207:2008 *usability process view*. The reported results can help lecturers to plan and coordinate their usability-related courses and give the students a complete, consistent and broad picture about the integration of usability practices and techniques into software development processes.

Keywords: usability; usability engineering; software engineering; human computer interaction; ISO/IEC 12207; engineering education

1. Introduction

Both Software Engineering (SE) and Human-Computer Interaction (HCI) disciplines are concerned with the concept of usability, but they have evolved separately regarding to its definition and how to ensure it in software systems [1–3].

At a research and industry level, endeavors to integrate usability in software development have been reported [4–7]. Mining the literature, we find works that integrate Usability Engineering (UE) techniques in the software development processes [4–6] or combine the quality benefits of UE with the systematic processes of the old version of ISO 12207 [7]. Further, from the International Organization for Standardization, the ISO/IEC TR 25060:2010 *Common Industry Format (CIF) for Usability: General Framework for Usability-related Information* [8] is being developed to partially bridge the HCI and SE gap regarding usability concerns.

This lack of coordination between disciplines is also present in the education domain. Pyla et al. [10] already argued there is a need of education and training regarding the importance of communication, collaboration and coordination between the SE and UE processes. Undergraduate students in Computer Science engineering should have a broad and connected view on this issue. The curricula for this particular degree always include a SE subject in the initial years. Later, they may include a compulsory or optional course related to the design, development and evaluation of usable user interfaces such as Usability engineering, User-Centred Design (UCD) or User interfaces design.

Based on this situation students may encounter two different challenges. First, if students do not take one of the later subjects, they will lack of methodologies, activities and techniques to engineer usability into software development projects. Second, even if students have enrolled both subjects, they may not link their previous knowledge on software processes with the new concepts to ensure usability. Further, in this latter case, lecturers should be coordinated to present a global view between courses.

In the Degree in Computer Engineering at the University of the Balearic Islands (Spain), we face the second scenario. There is a 6-ECTS ‘Software Engineering’ course compulsory for second-year students taught during a semester, four hours per week, where software development processes and methodologies are explained, specifically ISO/IEC 12207:2008 [9]. Then, in the fourth and last year, there is a compulsory 6-ECTS course, ‘Distributed Web Applications and User Interfaces’ taught during a semester, four hours per week, where UCD and usability techniques are introduced. Authors of this work are the lecturers of these courses.

The general aim of this paper is to conduct a systematic literature to map UCD practices and techniques that are actually being used with the software life cycle processes detailed in ISO/IEC 12207:2008 *usability process view*. This mapping will help fulfilling these specific aims:

- Provide a planning instrument for SE lecturers, whose study curricula do not have a compulsory course on interfaces design. In this case, while explaining the processes, they could go introducing usability techniques for students to have additional practical tools to apply.

** Both authors contributed equally to this work.

- Provide a coordination instrument for SE and HCI lecturers to plan the courses taking into account the previous knowledge of the students, coordinate both courses, divide content and give a general framework beyond the individual course.

It is not an objective of this work to describe the mentioned usability techniques, but to map them into the software processes.

The remainder of the paper is organized as follows. In the next section, the process view for usability provided by ISO/IEC 12207 *International Standard for the software life cycle processes* is introduced. Section 3 describes the method used to perform the review. Then, in section 4 the results are presented and discussed. Finally, the paper concludes highlighting the implications of this work in the context of computer engineering education.

2. The ISO/IEC 12207 usability process view

ISO/IEC 12207 is the current International Standard for Software life cycle processes. The purpose of the standard is to provide a defined set of processes to facilitate communication among acquirers, suppliers and other stakeholders in the life cycle of a software product. It establishes a common framework for software life cycle processes, with a well-defined terminology, that can be referenced by the software industry. It contains processes, activities, and tasks that are to be applied

during the acquisition of a software product or service and during the supply, development, operation, maintenance and disposal of software products. Figure 1 shows the ISO/IEC 12207 Process Reference Model (PRM).

This PRM classifies the software life cycle processes into two major categories: System Context Processes and Software Specific Processes. The first category contains processes for dealing with a standalone software product or a software system. The second contains the software specific processes to be used in the implementation of a software product that is an element of a larger system. Within these two categories, processes are classified into seven process groups:

- Agreement processes define the activities necessary to establish an agreement between two organizations.
- Organization Project-Enabling processes support projects and ensure the satisfaction of organizational objectives and establishment of agreements.
- Project processes describe processes concerned with planning, assessment and control.
- Technical processes are used to define the requirements for a system, to transform them into an effective product, to permit consistent reproduction of the product where necessary, to use the product, to provide the required services, to sustain the provision of those services and to

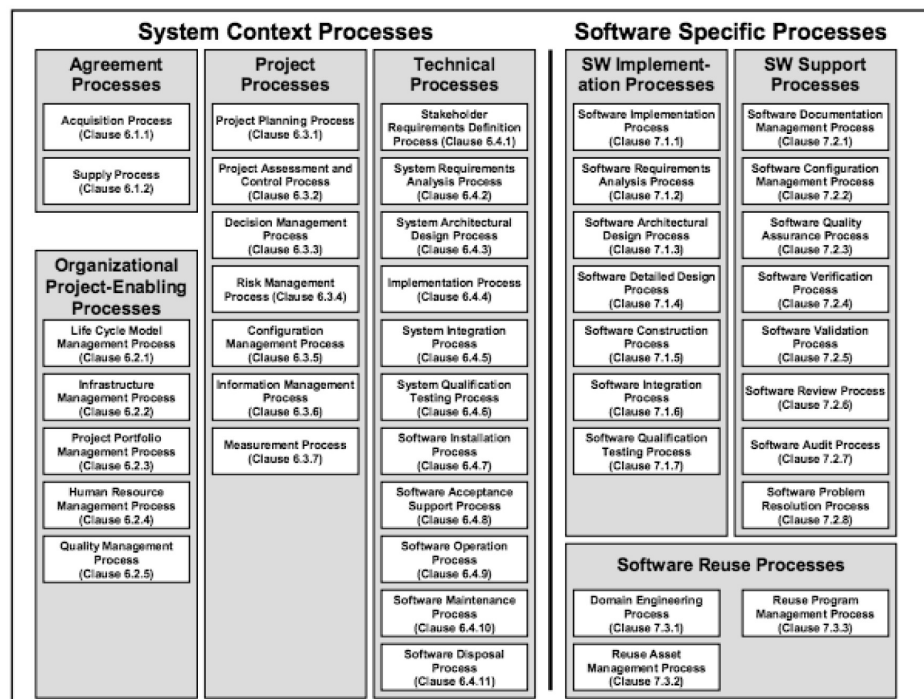


Fig. 1. ISO/IEC 12207 Life Cycle Process groups (extracted from ISO/IEC 12207:2008).

dispose of the product when it is retired from service.

- Software Implementation processes are used to produce a specified system element (software item) implemented in software.
- Software Support processes provides a specific focused set of activities for performing a specialized software process.
- Software Reuse processes supports an organization's ability to reuse software items across project boundaries.

At this point it is important to highlight that the PRM does not detail the life cycle processes in terms of methods or procedures required to meet the requirements and outcomes of a process. It either prescribes a specific system or software life cycle model, development methodology, method, model or technique. As highlighted in the standard, the parties are responsible for selecting and applying the software development methods and for performing

the activities and tasks suitable for the software project.

Annex A of the standard provides the requirements for tailoring the International Standard to adapt the processes to satisfy particular circumstances or factors such as usability which can be considered a particular engineering interest. For this purpose, another annex, Annex E, provides a process view for usability, intended to illustrate how a project might assemble processes, activities and tasks of ISO/IEC 12207 to provide focused attention to the achievement of a usable product.

The purpose of the Usability Process View is to ensure the consideration of the interests and needs of the stakeholders in order to enable optimizing and training, increased productivity and quality of work, improved human working conditions and reducing the chance of user rejection of the system. This process view can be implemented using the processes listed in Table 1.

Table 1. ISO/IEC 12207 Usability Process View

Process category	Process title	Usability purpose
Organizational Project-Enabling Processes	Portfolio Management	Establishment and maintenance of a focus on user issues, championing of a human-centered approach
	Infrastructure Management	Specification of how human-centered design activities fit into the whole systems lifecycle process and the organization
Project Processes	Project planning	Selection of human centered methods and techniques planning the involvement of users and other stakeholders, planning of human-centered design activities
	Project Assessment and Control	Monitoring the extent of achievement of the requirements and communicating the results to stakeholders and managers, ensuring a human-centered approach in the design team
	Information Management	Specification, development and maintenance of artifacts for documenting and communicating the extent of achievement. For usability this is detailed by ISO/IEC 25062 Common Industry Format for usability test reports
	Measurement	Defining an approach that relates measures to desired characteristics. For software these are detailed in ISO/IEC 25020: Software engineering—Software product Requirements and Evaluation (SQuaRE)—Measurement reference model and guide
Technical Processes	Stakeholders Requirements Definition	Identification and documentation of the context of use and the interaction between the users and the systems
	Systems Requirements Analysis	Specification an evaluation of the context of use and the usability and human centered design requirements
	System Architectural Design	Incorporation of design criteria to address the targets for usability and the ergonomic requirements
	System Integration	Planning the integration, including the considerations for user training and the assurance that the achievement of targets for usability and accordance with ergonomic requirements are verified and recorded
	Software Operation	Assuring that the usability requirements are appropriately achieved involves monitoring the system
	Software Maintenance	Sustains the capabilities of the system, including its usability properties
SW Implementation Processes	Software Requirements Analysis	Specification of the usability and software ergonomic requirements

3. Literature review

A large number of publications address fundamental issues about usability in software development. In order to compile the usability practices or activities reported in the literature we performed an extensive search for research papers dealing with this topic. This was done by following the guidelines reported in [11] for systematic literature reviews.

The search was conducted taking the following keywords as a basis: “usability, practice, software, development, activity, process”. These keywords were considered to obtain search strings that were adapted to each search engine provided by the databases listed in Table 2.

As a result of the search, an initial set of 824 papers were obtained for further evaluation. Each identified study was evaluated to decide whether or not it should be considered relevant. The selection of studies was based on the inclusion criteria (IC) and exclusion criteria (EC) outlined in Table 3.

After applying the inclusion criteria (IC1, IC2 and IC3) defined in Table 3, only 61 of the 824 discovered articles were considered as relevant articles. Applying the criterion EC1 for the exclusion of duplicated articles, we obtained 50 articles. Applying the criterion EC2 for the exclusion of articles published before 2000, we had 40 papers. After excluding theoretical papers (EC3), finally 25 have been considered primary studies.

An exhaustive analysis of these primary papers yielded to identify 91 usability practices together with a total of 76 associated techniques. In order to organize them and eliminate duplications we performed a second review resulting in 18 different practices and 33 techniques which are classified in

the following section. These practices and techniques have been linked to each corresponding ISO/IEC 12207 usability process.

4. Results and discussion

In order to support usability education in SE and provide a basis for the integration of usability within the software development processes we have classified usability processes, practices and techniques into two categories. On the one hand we provide a list of usability management practices that could be performed at two different levels: organizational and project. On the second hand we identify technical practices for usability.

4.1 Usability management practices

At *organizational level*, the usability process view contains two processes: *Portfolio Management* and *Infrastructure Management*. The main goal of the first process is to establish and maintain a focus on user issues, championing of a human-centered approach. The second process is aimed at specifying how human centered design activities fit into the whole systems lifecycle processes and the organization. Table 4 summarizes usability practices reported in the reviewed papers that could be directly linked to these two organizational processes.

Focusing on the *project level*, 64% of the reviewed papers report some usability management practices that should be done in each particular project. Results also reveal that measurement is considered a key aspect in the usability literature where both practices and techniques are reported in 50% of the primary papers. Table 5 shows the usability practices and techniques that support project level processes.

Table 2. Selected databases

Name	Web Site
IEEE Xplore Digital Library	http://ieeexplore.ieee.org/Xplore/home.jsp
ACM Digital Library	http://portal.acm.org/
Google Scholar	http://scholar.google.com
Springer Link	http://www.springerlink.com
ScienceDirect	http://www.sciencedirect.com
Wiley Online Library	http://onlinelibrary.wiley.com/
CiteSeerX	http://citeseerx.ist.psu.edu

Table 3. Definition of inclusion and exclusion criteria

Criterion	Description
IC1	Include papers whose title is related to usability in software development
IC2	Include papers that contain the keywords
IC3	Include papers whose abstract is related to usability in software development
EC1	Exclude duplicated papers
EC2	Exclude papers published before 2000
EC3	Exclude theoretical papers

Table 4. Usability management practices at organizational leve

Software process	Usability practices	Primary papers
Portfolio Management	Promote a user-centred development process	1, 10, 15, 16, 19
Infrastructure Management	Create a development process with a focus on usability	1, 4, 6, 7, 10, 12, 19, 23
	Support and educate teams for usability	1, 4, 6, 7, 10

Table 5. Usability management processes, practices and techniques at project level

Software process	Usability practices	Techniques	Primary papers
Project Planning	Plan for user participation Plan usability testing		6, 7, 8
Project Assessment and Control	Register usability issues Prioritize usability issues Provide support for usability matters Measure and control the progress of usability activities		10, 12, 16, 21
Information Management	Perform usability studies at user sites Evaluate usability	Prototyping Use Case Evaluation (UCE) Workshops Walkthroughs Heuristics Satisfaction questionnaires Exploratory inspections	1, 6, 7, 8, 9, 15, 18, 19, 21, 23, 24, 25
Measurement			

4.2 Technical practices for usability

According to ISO 12207, Technical Processes define the activities that enable organizational and project functions to optimize the benefits and reduce the risks that arise from technical decisions and actions. These activities enable products and services to achieve their functional and non-functional requirements such as usability. As shown in Table 1, the Usability Process View suggests the implementation of usability practices in 6 technical processes covering the whole life cycle of a project.

The purpose of the requirements definition and analysis processes is to define the requirements for a system that can provide the services needed by users and other stakeholders in a defined environment and transform them into a set of desired system technical requirements that will guide the design of the system. Reported practices are aimed at identifying and characterizing users' profiles, understanding and specifying the context of use, specifying usability requirements, and modelling systems tasks.

The implementation of these practices can be supported by different techniques which have been demonstrated to be successful in reported case studies and practical experiences. Table 6 shows the list of practices and techniques which have been described for dealing with usability at the requirements stage.

The System Architectural Design process main goal is to identify which system requirements should be allocated to which elements of the system.

During this process human-centered, design activities should be identified and performed and human factors and ergonomic knowledge and techniques should be incorporated in system design. Although this process is widely considered in a relevant number of papers, only two practices have been reported for this process in particular with a big amount of available techniques as shown in Table 7.

There are an important number of techniques which support the detailed design (see Table 7). These have been mapped to the *Software Detailed Design* process defined in ISO 12207 although this process is not considered in the usability process view. In our opinion, this process, whose purpose is to provide a design for the software that can be verified against the requirements and the software architecture, and which includes the definition of the external interfaces of each software unit, should be referenced in the usability process view.

To conclude with the technical processes it is necessary to mention that we have not found any papers that relate usability with *System Integration*, *Software Operation* and *Software Maintenance* processes. However, although there is no explicit connection between usability and these processes in the literature, from a practical point of view the usability practices and techniques reported for other processes are also applicable to system integration, software operation and software maintenance:

- Usability issues during system integration should be oriented to planning the user training and

Table 6. Usability requirements processes, practices and techniques

Software process	Usability practices	Techniques	Primary papers
Stakeholders Requirements Definition	Identify and characterize users' profiles Define usability objectives	Field studies Interviews	1, 6, 7, 8, 18, 19, 21, 23
System Requirements Analysis	Understand and specify the context of use	Meetings Usability Context Analysis (UCA) Contextual inquiry	19, 20, 23, 24
System/Software Requirements Analysis	Specify usability requirements	Use cases Scenarios Personas Elicitation patterns Quality Attribute Workshop (QAW) Systematic creativity	1, 2, 5, 6, 9, 13, 14, 17, 18, 19, 22, 23
	Model the system tasks	Ethnography Task analysis Hierarchical Task Analysis (HTA) Functionality matrix	17, 18, 19

Table 7. Usability design processes, practices and techniques

Software process	Usability practices	Techniques	Primary papers
System Architectural Design	Define the architecture	Usability patterns	2, 5, 22
Software Detailed Design	Produce design solutions iteratively	Prototyping Sketching Mock-ups Brainstorming Parallel design Storyboards Affinity diagramming Card Sorting Wizard-of-Oz	1, 2, 6, 7, 8, 17, 18, 19

assurance that the achievement of targets for usability are verified and recorded. Therefore this process has a strong connection with usability measurement that we have discussed in the previous section.

- The same argument is applicable to software operation that assures that the usability requirements are appropriately achieved by monitoring the system.
- The software maintenance process is aimed at sustaining the capabilities of the system, including its usability properties. Consequently, usability practices and techniques that could be applicable to this process are in definitive the same practices performed for other technical, managerial and measurement processes also considered in the previous section.

5. Conclusions

Usability is an important issue addressed by differ-

ent disciplines. Different endeavors, at research, industry and education level, have been carried out to integrate the SE and HCI approaches since the need for usability has grown in software development.

As lecturers of courses related with usability, we think that is essential to outline the current context of usability integration in the software life cycle processes. Particularly, in our educational context there is lack of coordination between our courses, due to the need of a consensus addressing usability issues. Therefore, we have done an effort to classify the existent usability practices and techniques and map them with the software development processes reported in the current version of ISO/IEC 12207 [9].

This work, besides the information provided by the classification and mapping, contributes as an instrument for computer engineering courses planning and coordination, specifically for those courses where usability is an important topic. Moreover, the

results of this work can increase students' awareness of usability, give them a general view of the usefulness of usability activities in software development and provide them with a usability engineering focus among courses.

The knowledge gained with this work has improved our usability expertise and we expect to put the lessons learnt into practice the next course. We hope our experience will also be useful to other SE and HCI educators.

Acknowledgements—This work is supported by the project TIN12-35427 granted by the Spanish MINECO, FEDER funding and PID 151614 granted by the ICE (UIB).

References

1. ISO/IEC 25000:2014, Systems and software engineering—Systems and software Quality Requirements and Evaluation (SQuaRE)—Guide to SQuaRE, *International Organization for Standardization*, 2014.
2. ISO 9241-210:2010, Ergonomics of human-system interaction—Part 210: Human-centred design for interactive systems, *International Organization for Standardization*, 2010.
3. ISO 9241-11:1998, Ergonomic requirements for office work with visual display terminals (VDTs)—Part 11: Guidance on usability, *International Organization for Standardization*, 1998.
4. X. Ferre, Integration of Usability Techniques into the Software Development Process. *Proc. ICSE 200*, Portland, Oregon, 2003, pp. 28–35.
5. K. Nebe and V. Paelke, Key Requirements for Integrating Usability Engineering and Software Engineering. *Proc. HCI 2011*. LNCS 6776. Springer, Heidelberg, 2011, pp. 114–120.
6. H. Fischer, K. Nebe and F. Klompmaier, A Holistic Model for Integrating Usability Engineering and Software Engineering Enriched with Marketing Activities, *Proc. HCI 2011*. LNCS 6776. Springer, Heidelberg, 2011, pp. 28–37.
7. K. Nebe, D. Zimmermann and V. Paelke, Integrating Software Engineering and Usability Engineering, *Advances in Human-Computer Interaction*, I-Tech Education and Publishing KG, Vienna, 2008, pp. 331–350.
8. ISO/IEC TR 25060:2010. Systems and software engineering—Systems and software product Quality Requirements and Evaluation (SQuaRE)—Common Industry—Format (CIF) for usability: General framework for usability-related information, *International Organization for Standardization*, 2010.
9. ISO/IEC 12207:2008. Systems and software engineering—Software life cycle processes, *International Organization for Standardization*, 2008.
10. P. S. Pyla, M. A. Perez-Quinones, J. D. Arthur and H. R. Hartson, What we should teach, but don't: Proposal for cross pollinated HCI-SE Curriculum, *Proc. of ASEE/IEEE Frontiers in Education Conference*, 2004, pp. S1H/17S1H/22.
11. B. Kitchenham, Guidelines for Performing Systematic Literature Reviews in Software Engineering Version 2.3, *Technical Report EBSE-2007-01*, Software Engineering Group, School of Computer Science and Mathematics, Keele University, and Department of Computer Science, University of Durham, July 2007.
12. J. Gulliksen, I. Boivie and B. Göransson, Usability professionals—current practices and future development, *Interacting with Computers*, **18**(4), 2006, pp. 568–600.
13. N. Juristo, A. Moreno, M. I. Sanchez-Segura and M. C. Calani, A Glass Box Design: Making the Impact of Usability on Software Development Visible, *Proc. INTERACT 2007*, LNCS 4663, 2007, pp. 541–554.
14. N. Juristo, A. M. Moreno and M. I. Sanchez-Segura, Guidelines for Eliciting Usability Functionalities, *IEEE Transactions on Software Engineering*, **33**(11), 2007, pp. 744–758.
15. N. Karster and P. Volker, Usability-Engineering-Requirements as a Basis for the Integration with Software Engineering, *New Trends on Human-Computer Interaction*, LNCS 5610, 2009, pp. 652–659.
16. N. Karster and P. Volker, Key Requirements for Integrating Usability Engineering and Software Engineering, *Human-Computer Interaction. Design and Development Approaches*, LNCS 6761, 2011, pp. 114–120.
17. S. Kenia, E. Furtado and H. Mendonça, UPi: a software development process aiming at usability, productivity and integration, *Proc. CLIHC 2005*, 2005, pp. 76–87.
18. J. C. Lee and D. S. McCrickard, Towards Extreme(ly) Usable Software: Exploring Tensions Between Usability and Agile Software Development, *Proc. AGILE 2007*, 2007, pp. 59–71.
19. M. Maguire, Methods to support human-centred design, *International Journal of Human-Computer Studies*, **55**, 2001, pp. 587–634.
20. S. Mills, Contextualising design: Aspects of using usability context analysis and hierarchical task analysis for software design, *Behaviour and Information Technology*, **26**(6), 2007, pp. 499–506.
21. C. S. A. Peixoto and A. E. A. da Silva, A Conceptual Knowledge Base Representation for Agile Design of Human-Computer Interface, *Proc. IITA 2009*, 1, 2009, pp. 156–160.
22. T. Rafla, P. N. Robillard and M. Desmarais, A method to
1. J. Anderson, F. Fleak, K. Garrity and F. Drake, Integrating usability techniques into software development, *IEEE Software*, **18**(1), 2001, pp. 46–53.
2. L. Bass and E. J. Bonnie, Linking usability to software architecture patterns through general scenarios, *Journal of Systems and Software*, **66**(3), 2003, pp. 187–197.
3. R. Bellamy, B. John and S. Kogan, Deploying CogTool: integrating quantitative usability assessment into real-world software development, *Proc. ICSE 2011*, 2011, pp. 691–700.
4. Boivie, C. Åborg, J. Persson and M. Löfberg, Why usability gets lost or usability in in-house software development, *Interacting with Computers*, **15**(4), 2003, pp. 623–639.
5. E. Folmer, J. van Gurp and J. Bosch, A framework for capturing the relationship between usability and software architecture, *Software Process: Improvement and Practice*, **8**(2), 2003, pp. 67–87.
6. B. Göransson, J. Gulliksen and I. Boivie, The Usability Design Process—Integrating User-centered Systems Design in the Software Development Process, *Software Process: Improvement and Practice*, **8**(2), 2003, pp. 111–131.
7. J. Heiskari, M. Kauppinen, M. Runonen and T. Mannisto, Bridging the Gap Between Usability and Requirements Engineering, *Proc. IEEE RE'09*, 2009, pp. 303–308.
8. J. W. Helms, J. D. Arthur, D. Hix and H. R. Hartson, A field study of the Wheel—a usability engineering process model, *Journal of Systems and Software*, **79**(6), 2006, pp. 841–858.
9. K. Hornbæk, R. T. Høegh, M. B. Pedersen and J. Stage, Use Case Evaluation (UCE): A Method for Early Usability Evaluation in Software Development, *Proc. INTERACT 2007*, LNCS 4662, 2007, pp. 578–591.
10. R. Th. Høegh, Case study: integrating usability activities in a software development process, *Behaviour & Information Technology*, **27**(4), 2008, pp. 301–306.
11. R. Th. Høegh and J. J. Jensen, A case study of three software projects: can software developers anticipate the usability problems in their software?, *Behaviour & Information Technology*, **27**(4), 2008, pp. 307–312.

- elicit architecturally sensitive usability requirements: its integration into a software development process, *Software Quality Journal*, **15**, 2007, pp. 117–133.
23. J. Scholtz and E. Morse, Using Consumer Demands to Bridge the Gap between Software Engineering and Usability Engineering, *Software Process: Improvement and Practice*, **8**(2), 2003.
 24. J. Winter and K. Rönkkö, SPI success factors within product usability evaluation, *Journal of Systems and Software*, **83**(11), 2010, pp. 2059–2072.
 25. L. Zhao, F. P. Deek and A. James, Exploratory inspection—a user-based learning method for improving open source software usability, *Journal of Software Maintenance and Evolution: Research and Practice*, **22**(8), 2010, pp. 653–675.

Cristina Manresa-Yee received her degree in Computer Science and her Ph. D. in Computer Science from the University of Balearic Islands. She is currently an Associate Professor at the University of the Balearic Islands. Her research interests include human-computer interaction and usability engineering. She is a member of the ACM SIGCHI.

Esperança Amengual is currently an Assistant Professor in software engineering and web engineering at the University of the Balearic Islands. She has participated in the QuaSAR project, a software process improvement programme in software small companies at the Balearic Islands. Her research interests include software process improvement, teamwork and usability engineering. She is member of the Software Quality Group of ATI Spain and a certified ISO/IEC 15504 assessor.