

The Influence of Process Quality on Product Usability: A Systematic Review

Gabriel Alberto García-Mireles

Universidad de Sonora, Departamento de Matemáticas,
Hermosillo, Sonora, México, 83000
mireles@gauss.mat.uson.mx

M^a Ángeles Moraga, Félix García and Mario Piattini

Universidad de Castilla-La Mancha, Instituto de Tecnologías y Sistemas de Información,
Grupo Alarcos,
Ciudad Real, España, 13071
{MariaAngeles.Moraga, Felix.Garcia, Mario.Piattini}@uclm.es

Abstract

When organizations implement software process improvement programs they are seeking to increase productivity, reduce costs and enhance the quality of the resulting software. In fact, the quality of software products depends, to a great extent, on the processes used for their development and/or maintenance. However, despite this belief concerning the influence of the process on product quality, the specific process-product relationship has been poorly addressed in literature. This paper therefore analyzes the influence that software processes can have on software product usability, which is one of the most important quality characteristics since it influences how the user perceives the product. To this end, we present a systematic literature review examining the relationship between usability and software processes, which has allowed us to identify the processes that relevant studies consider to be important in enhancing product usability.

Keywords: software process, usability, systematic literature review.

1 Introduction

In order to survive, software development companies change their processes to produce software at a lower cost, with higher quality, and a reduced cycle time. In fact, one common argument for investing in software process improvement (SPI) initiatives is to increase the quality of software [1].

In existing SPI literature, the most frequently used perspective is the manufacturing-based view of quality [2]. The manufacturing-based quality approach pursues decreasing post-release defects in software, as reported by Agrawal and Chari [3] who reviewed various empirical studies that showed how successful SPI programs reduced these defects. Another research work reported that SPI initiatives implemented in companies were able to reduce the defect density by between 10 and 94%, and also that there was a correlation between the levels of product quality and the level of process maturity (up to level three) [4]. Moreover, the general perception among those software developers who have participated in SPI programs in emerging countries is positive as regards the improvement in productivity and product quality, where quality is measured as a count defects function [5].

Although the level of software process maturity seems to influence the quality of the software product, few papers address the specific impact of their relationship. In this respect, in [6] a positive relationship between certification processes models, such as ISO 9000 and CMM, and quality software characteristics such as reliability, testability, usability, efficiency and integrity is reported. Surveys presented in [7, 8], in which managers and software developers participated, establish correlations between an SPI initiative and the usability and maintainability of software. However, none of the existing studies addresses how to implement an SPI effort focused on the product quality characteristics that they are intended to improve. Further research is required to establish, on the one hand, the relationships between process quality and product software quality, and on the other, the relationships between software quality and project constraints and business goals [9].

In accordance with the issues identified above, this paper analyzes the influence that software processes could have on the quality of the resulting software products. This analysis focuses on the usability of products, which is one of the most relevant quality characteristics of a product from a user perspective. Usability is defined as the "degree to which a product or system can be used by specified users to achieve specific goals with effectiveness, efficiency and satisfaction in a specified context of use" [10]. Experts agree that processes, methods, and tools are required to improve product software usability [11].

The analysis presented herein is based on the results of a systematic literature review (SLR) carried out to identify the processes that are important to promote better product usability. The remainder of this paper is structured as follows: Section 2 describes the methodology used to perform the SLR, along with some relevant statistics concerning the studies selected. Section 3 then goes on to summarize the major contributions of the selected primary studies. Section 4 describes how to integrate usability practices into processes according to the results obtained from the SLR. Section 5 describes the limitations of this SLR. Finally, our conclusions and future work are presented in Section 6.

2 Systematic Literature Review of the Influence of Software Processes on Product Usability

The SLR conducted followed the methodology proposed by Kitchenham and Charters [12]. The following research questions were formulated to satisfy the objectives of this study:

- RQ1: What is the state of research as regards the relationship between software processes and usability?
- RQ2: Which processes and activities should be considered in order to improve the usability of the product developed?

In accordance with the SLR methodology followed, the inclusion criteria considered for the selection of primary studies were: the study had been published as a journal article or in conference proceedings up to and including 2010, it was written in English, and its contents described processes and/or activities related to product usability. The exclusion criteria were applied to those papers that did not explicitly address the relationship between process and product usability, such as papers which analyzed the degree of integration of usability practices in industry.

Table 1: SLR search string

Terms	Strings
Usability	usability
Software	software OR product
Software process	"software process" OR "development process" OR "development method"

Table 1 shows the proposed strings, which were joined with the logical AND operator to produce a single search string. The searches were carried out in the following databases: Scopus, IEEE Computer Society, ACM Computer Library, Science@Direct and Springer. We also ensured that these databases included the proceedings of the EuroSPI, ICSP and PROFES conferences, since they are relevant in the area of software processes.

The articles retrieved from the search engines are shown in Table 2. 504 articles were retrieved (including duplicates). The first study selection stage consisted of applying the inclusion criteria to each paper by considering the title, keywords and abstract. As a result, 168 papers were selected as candidates. Finally, in a second stage, 18 primary studies were selected as a result of a thorough reading of the candidates to verify that they effectively met both specified criteria. Table 3 lists the papers selected as primary studies.

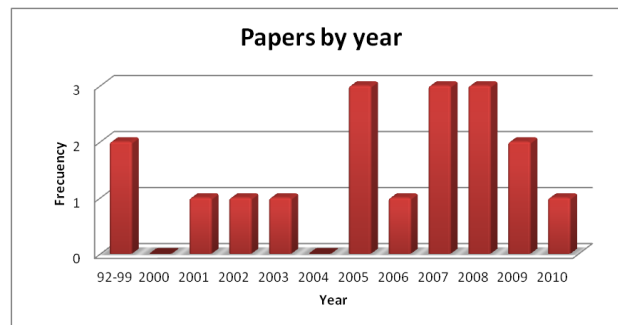
Table 2: Summary of studies from this SLR

Database	Retrieved	Candidates	Primaries
Scopus	238	20	1
Science @Direct	40	30	3
IEEE	133	48	6
ACM	44	25	1
Springer	49	45	7
Totals	504	168	18

Table 3: List of primary studies

Id.	Year	Type	Database
anderson2001 [13]	2001	Journal	IEEE
biel2010 [14]	2010	Journal	Sciadirect
bonacin2009 [15]	2009	Proceedings	Springer
castro2008 [16]	2008	Proceedings	IEEE
constantine2002 [17]	2002	Journal	IEEE
ferre2005 [18]	2005	Proceedings	Springer
glissmann2005 [19]	2005	Proceedings	IEEE
goransson2003 [20]	2003	Journal	Scopus
helms2006 [21]	2006	Journal	Sciadirect
hornbaek2007 [22]	2007	Proceedings	Springer
hussain2008 [23]	2008	Proceedings	Springer
lif1999 [24]	1999	Journal	Sciadirect
memmel2007 [25]	2007	Proceedings	Springer
moreno2009 [26]	2009	Proceedings	Springer
nielsen1992 [27]	1992	Journal	IEEE
rafla2007 [28]	2007	Journal	Springer
sousa2005 [29]	2005	Proceedings	ACM
wang2008 [30]	2008	Proceedings	IEEE

Fig. 1 shows the trends of publication of the primary studies selected. As can be observed, there is an ongoing interest in this topic, particularly in the last five years, but it is not uniformly distributed throughout the years. The maximum number of papers per year denotes that research into the topic is relatively scarce. With regard to the type of publication, more than half of the studies (10 out of 18, 56%) were published in conferences (Table 3).

**Figure 1:** Articles found by year

3 Analysis of SLR results: Usability Focused Software Processes

In this section, the analysis of the primary studies extracted is used to describe the practices that should be considered when usability is the product quality factor sought. The results in this section are classified by considering the underlying process reference model of each proposal. The process models used in the proposals are as follows:

- Rational Unified Process (RUP) [13, 20, 29],
- General Process reference models [18, 21, 27].
- Domain-oriented processes, such as Web applications or mobile devices [17, 19, 26]
- Agile methods [15, 23, 25, 30].
- The last category consists of proposals that have an impact on the usability requirements specification [14, 16, 22, 24, 28].

The following sections summarize the most relevant aspects of each proposal according to the aim of this SLR and the aforementioned classification.

3.1 RUP-Based Proposals

These proposals focus their efforts on the conception and elaboration phases of the RUP life cycle. They take advantage of its iterative and incremental nature to evaluate prototypes that are developed with different levels of detail. Anderson et al. [13] propose a RUP adaptation with which to integrate the following usability activities: creation of concepts, requirements elicitation, requirements analysis and design of the product vision. Three new roles are also identified, which were called usability engineer, user interface designer, and usability tester. The authors additionally suggest that management support is required to identify and to support software development tasks which improve product usability.

Goransson et al. [20] point out that it is important to involve the user, understand the context of system use, apply an iterative development approach, and continually evaluate the new prototypes. The activities included in RUP are: creating a usability design plan, carrying out a competitive analysis, conducting user studies, developing conceptual design, developing interaction design, developing detailed design, developing user assistance, evaluating usability, monitoring usability activities, and refining the usability design plan.

Lastly, Sousa et al. [29] present UPi (Unified Process for Interactive Systems Development) as a unified process for developing interactive systems. Their proposal is based on use cases, task models, usability requirements and usability design patterns. In the design phase, a usability plan is elaborated which defines values and usability patterns for each interface element. Prototypes are then built to facilitate the users' assessment of the patterns. The authors highlight the adaptation of the elaboration phase to the usability focus.

3.2 General Reference Models Proposals

This category of proposals is based upon the general software development life cycle phases in order to show those activities and practices that contribute towards improving the level of usability. In this respect, Ferré et al. [18] developed a framework to provide software developers with a selection of those HCI (Human Computer Interaction) techniques that are appropriate for organizations that aim to improve of the usability of their products. The practices are organized by type of activity and the timing of the iterations of the development cycle. These authors found a great similarity between the HCI activities and those related to software requirements, since in both fields it is very important to: know the context of use, identify the users and their tasks, specify usability requirements, develop the product concept, elaborate low cost prototypes, and then validate them with user representatives. They point out that interaction design can be carried out in parallel with and independently of software design. They also note that usability evaluation requires a different type of software test which moves away from typical software engineering, such as expert evaluation, usability testing, and follow-up studies in installed systems.

Helms et al. [21] propose an iterative process model called Wheel whose aim is to achieve flexibility in terms of iterative progression in the usability engineering life cycle, and the ability to customize instances of the process model to accommodate changes in budget, schedule or resources. The use of Wheel in a project signifies that developers must select product forms to support the software product development. The product forms are: product design, user model/tasks, usage scenarios, screen design, low-fidelity prototype, hi-fidelity prototype, integrated software, installed product, operational system, and close-down product. Developers must then select appropriate techniques or methods to be applied in each activity: analysis, design, implementation, and evaluation. The model also includes an assessment activity that controls the iterations. The proposal was applied in a company in which the selected project had required the design of two user interfaces for communication with Internet connected devices.

The "Usability Engineering" proposal [27] promotes a practical process that can easily be incorporated into the product development process using established methods to achieve usable interfaces. The essential elements are the empirical tests involving user, prototyping and iterative design. It includes three phases: pre-design, design and post design. In the pre-design phase, the goal is to understand users and their tasks. This can be achieved by doing market research, task analysis, and by building prototypes. In the design phase, an iterative approach is implemented to refine the user interface with the results of empirical assessment. Finally, the post-design stage aims to obtain data from the next version of the product and recommend follow-up studies.

3.3 Proposals Based on Processes for a Particular Domain

This category includes studies whose context is software development by considering the usability of the product, but from the particular characteristics of the technology used. Glissman et al. [19] present a process for developing mobile device applications considering the principles of HCI. The model has five stages: identify needs, establish requirements, conceptual design, physical design and implementation. The authors argue that conceptual design is a key aspect, since both the capabilities of the device and potential application solutions have to be analyzed and

checked with regard to the usability goals defined. The screens' appearance and the navigation structure are considered in the physical design.

Constantine and Lockwood [17], meanwhile, describe a flexible, model-driven engineering approach for Web applications called usage-centered design. The process differentiates human and non-humans actors. The preliminary steps identify the essential purpose, and exploratory modeling is then carried out. Role modeling, task modeling, tasks clustering, interface preliminary design and interaction modeling, abstract prototyping, detailed design, and construction are carried out in the first iteration. The subsequent iterations refine previous work products. The authors suggest that special training and close coordination between UI designers and programmers are required.

Finally, Moreno et al. [26] integrate accessibility issues in Web engineering processes in order to provide a methodological framework named AWA (Accessibility for Web Applications). Their proposal is supported by an accessibility mechanism identified in the Web Content Accessibility Guidelines (WCAG). They also suggest the use of usability techniques in the early stages of development, specifically when gathering and analyzing requirements to generate conceptual, navigation, and presentation models.

3.4 Proposals Based on Agile Methods

Some proposals are based on agile methods, particularly Extreme Programming (XP). The general approach consists of introducing some new practices to achieve higher levels of software usability, but by maintaining agility. Wang and Shi [30] thus propose the software development managed by the user model (UMDD) approach, which is user-centered and considers four phases: user modeling, designer modeling, implementation modeling, and usability evaluation. The objective of user modeling is to build the user interface according to the users' prior experience, their knowledge, and their preconceptions of the tasks. The activities performed in this proposal are requirements definition, user interface modeling, user interface prototyping, and early usability assessment. The requirements can be described as role model, task model and domain model. The other relevant phase, usability evaluation, sets goals, methods and evaluation criteria.

Bonacin et al. [15] present an agile method-based process that encourages user participation through participatory design and organizational semiotics methods. The lifecycle of the proposal imitates the idea of XP. This includes activities such as expert analysis, low cost prototyping, user acceptance tests, and user workshops. The authors highlight some lessons learned about the benefits of face to face meetings to evaluate the interface designs. They additionally emphasize the balance between the quality standards required for usability and accessibility and, on the other hand, the speed of the development process.

Hussain et al. [23] describe a development process in which XP is integrated with user-centered design. They note that similarities between XP and UCD (User Centered Design) are targeting the end user, continuous testing and iterative development. The process begins with the creation of stories and the subsequent construction of a paper prototype, both of which the client then evaluates. These results are used to create and implement a final prototype according to the customer's wishes. Usability testing is performed by usability experts and end users. Meanwhile, Memmel et al. [25] present a proposal which integrates low-cost and high cost prototyping in Agile XP. They note that the former offers immediate feedback whereas the latter supports problem identification in the screen content.

3.5 Usability Requirements Focused Proposals

This set of proposals intends to obtain a more complete and accurate usability requirements specification. To achieve this goal, the authors highlight the technical changes that affect the processes in the system/software analysis stage. Hornbaek et al. [22] describe a method for use case assessment called Use Case Evaluation (UCE). This consists of three activities: inspection of use cases, use case assessment and assessment documentation. They use a heuristic evaluation based on usability inspection guidelines. They suggest that this method may be complementary to the evaluation method called thinking aloud.

Biel et al. [14] have designed a method that analyzes the architecture of mobile applications using the software architecture analysis of usability requirements realization (SATURN) method. The evaluator must follow these steps: describe the context, determine scenario analysis, evaluate scenarios, interpret results, and review methods and tools. In order to compare the findings, the researchers then carry out a usability evaluation on a working prototype. The results of the case study show that 74% of usability problems could be related to problems in architectural design, 30% of problems were identified in the architecture analysis, and the rest were found during the usability evaluation.

Lif [24] presents "User interface Modeling" (UIM), which is a method for gathering interface requirements. This method involves elaborating actors, goals, and work models by using an iterative top-down approach. The actor model describes each category of users while the goal model lists their high level goals. The work model is a

specification of work situations (whatever the user needs to achieve one goal without sequential restrictions). The models are produced on paper and distributed to all session participants in order to evaluate them. The method was validated through action research in a Swedish company. The drawbacks reported were the method's dependence on a leader, and the difficulties in identifying the magnitude of all the additional tasks associated with a particular work situation.

Castro et al. [16] adapted the "Persona" technique, which is used to describe user profiles, and integrated it into the analysis phase of software process development. The eleven activities identified in "Persona" were mapped onto requirements engineering activities: gathering, analysis, specification and validation.

Finally, Rafla et al. [28] propose the Usability managed Quality Attributes Workshop with the aim of discovering and documenting usability requirements using the usability properties of Folmer [31] and the scenarios of Bass [32] as guidelines. Users are expected to identify those requirements that are relevant to the software. They then need to write a specific associated example. The intention is to identify high-level tasks that users expect from the system. The most representative scenarios are then selected, and detailed use cases are elaborated. The validation technique was conducted with undergraduate students, and the results show that the guidelines are useful in the articulation of usability requirements.

4 Mapping Usability Practices onto ISO 12207 Processes

In order to provide a homogeneous process reference model with which to promote product usability, the relevant practices identified in each primary study and presented in the previous section were integrated into the ISO/IEC 12207:2007 [33] process reference model.

Table 4: Mapping ISO 12207 processes with usability practices from primary studies

Article ↓	Process →												
	6.4.1 Stakeholder req.	6.4.2 Syst. Req. Analysis	6.4.3.Syst. Architectural	6.4.4.Implementation	7.1.1 Implementation	7.1.2 Sw. Req. Analysis	7.1.3 Sw. Architectural	7.1.4 Sw. Detailed D.	7.1.5 Sw. Construction	7.1.6 Sw. Integration	7.1.7 Sw. Qualification	6.4.8 Sw. Acceptance S.	6.4.9 Sw. Operation
anderson2001	x	x		x		x	x	x	x			x	x
biel2010				x			x				x		
bonacin2009	x	x		x	x				x				
castro2008	x	x		x		x							
constantine2002	x	x		x		x	x	x	x		x		
ferre2005	x	x	x	x		x	x					x	x
glissmann2005	x	x	x	x				x					
goransson2003	x	x		x	x		x	x				x	x
helms2006	x	x	x	x		x	x	x	x	x	x	x	x
hornbaek2007				x		x							
hussain2008		x		x		x			x	x	x		
lif1999	x	x		x		x	x						
memmel2007	x	x											
moreno2009	x	x		x		x	x		x		x		
nielsen1992	x	x	x	x	x	x	x	x	x		x	x	x
rafla2007	x	x		x		x							
sousa2005	x	x		x		x	x	x	x		x	x	
wang2008	x	x	x	x		x	x		x		x	x	
Totals	15	16	5	17	3	13	11	7	9	2	8	7	5

As one of the fundamental premises of ISO/IEC 12207:2007 states, software always exists in a system context; we therefore consider the system context technical processes and the software implementation processes with which to integrate the recommended practices found in literature. The technical processes are used to define the system requirements, product deployment, operation, maintenance, and retirement. This kind of processes has the goal, among others, of optimizing the system usability. The software implementation processes are meanwhile used to produce a piece of software that meets the requirements derived from system requirements.

Table 4 presents the processes identified in the ISO/IEC 12207 standard which may be candidates for the inclusion of usability practices. As can be observed in Table 4, there is a higher incidence of practices in the processes of system requirements analysis, requirements identification of participants, requirements analysis and architectural design software. This supports the findings of Ferré et al. [18], who found that user centered design and software engineering practices are focused on the same tasks in the early stages of development, namely: specify the context of use, analyzing users and their tasks, specify usability requirements, develop a product concept, producing low-fidelity prototypes, and validate them with representative users. However, there are few proposals which can be mapped onto design, construction, integration, acceptance testing, and operation processes.

The relevant practices associated with the system technical processes are presented in Fig. 2. These practices emphasize knowing the users, including their goals and needs in order to seek and set out the most efficient way to adapt the technology to their work environment, and providing users with more effective and efficient software interaction. The acceptance support and operation processes additionally focus on training and usability testing with users, in order to understand the level at which the system meets the requirements established in system analysis.

Technical Processes (from ISO 12207)		
Process	Description	Suggested usability oriented activities
6.4.1. Stakeholder requirements definition	Identify stakeholders, groups, user context, and usability requirements	Identify stakeholders groups, describe user profiles, define usability requirements, specify software/hardware platform
6.4.2. System requirements analysis	Describe functions and systems capabilities. Verify requirements	Analyze usability requirements
6.4.3. System architectural design	Human centered design activities are identified and performed. Architecture evaluation	User task redesign
6.4.4. Implementation	Develop system item. Replaced by 7.1.1 Software implementation	See software implementation
6.4.7. Software installation	Product is installed in target environment	Log usability issues
6.4.8 Software acceptance support	Customer is supported to achieve the trust that product meets system requirements. Training is provided	Log usability issues
6.4.9. Software operation	Software product is operated in target environment and support is provided to customers	Gather user feedback

Figure 2: Activities and techniques included in primary studies, and their mapping onto the system technical processes of ISO 12207

The techniques and/or practices associated with software implementation processes are shown in Fig. 3. Firstly, planning activities and the selection of appropriate methods with which to support usability by considering an iterative approach are recommended. Then, the use cases and alternative scenarios are suggested for the documentation of user tasks and the response to error conditions. In order to address interface design tasks, interaction and navigation models are highlighted, followed by the construction of prototypes, at different levels of

fidelity, in order to organize interaction objects and information, and to refine the expected behavior. In the design and construction processes, design patterns, standards, and automatic code generation are recommended. Finally, during the testing processes, various usability evaluation techniques could be used, such as heuristic evaluation and empirical tests of usability.

Software implementation processes (from ISO 12207)		
Process	Description	Suggested usability oriented activities
7.1.1. Software implementation	Software development planning	UCD task planning
7.1.2 Software requirements analysis	Establish quality characteristics: interaction, fault sensitive areas requiring training. Verify requirements	Design conceptual model
7.1.3 Software architectural design	Build high leve software structure and verify it	Standarize user interface design
7.1.4 Software detailed design	Specify component features. Verify software detailed design	User interface detailed design
7.1.5. Software construction	Develop and document each software item. Test each component	Build user interfaces
7.1.6. Software integration	Develop integration plan and test requirements. Perform plan and test	Integrate user interfaces
7.1.7. Software qualification testing	Perform qualification testing based on specified requirements	Verify software usability

Figure 3: Activities and techniques included in primary studies, and their mapping onto software implementation processes of ISO 12207

5 Limitations

The systematic review presented in this paper has some limitations related to publication bias and the quality of the primary studies. In the first case, although we cannot access all published articles, we used five well known databases to access journal articles or conference papers, and the more relevant conference proceedings on software process were also included.

Another issue related to this SLR is the quality of the primary articles. In order to analyze the rigor and potential industrial relevance of the papers we used the model proposed by Ivarsson and Gorscheck [34]. Rigor is related to the research method used, and how the research is reported and potential industrial relevance is related to the context in which the technology was assessed. Rigor is considered to be the extent to which the following aspects are reported: description of the context (description of the organization), study design (research planning), and validity (construct, internal and external validity threats). All of the above are scored according to three levels: weak (0), medium (0.5), and strong (1). A strong scoring is assigned when the descriptions are sufficiently detailed to understand how an evaluation could be replicated. A medium score is assigned when there is a brief description, while a weak score is obtained when no description is provided.

Table 5 presents the results concerning rigor. The best score obtained was 2 (three papers), followed by 1.5 (two papers), while a score of 1 was obtained by three papers and another of 0.5 by one paper. The others obtained a score of 0. These results were owing to the fact that: few papers reported contextual aspects, such as an organization's characteristics, a subject profile or a classification of the software product domain; eleven of them did not report the design study; and very few of them described the procedures which were applied. Finally, few studies discussed the construct, internal or external validity of research.

Table 5: Rigor scoring

Paper	Context	Study Design	Validity	Rigor
anderson2001	1	0	0,5	1,50
biel2010	0,5	1	0,5	2,00
bonacin2009	0	0	0	0,00
castro2008	0	0	0	0,00
constantine2002	0	0	0	0,00
ferre2005	0	0	0	0,00
glissmann2005	0	0	0	0,00
goransson2003	0	0	0	0,00
helms2006	0,5	0,5	1	2,00
hornbaek2007	1	0,5	0,5	2,00
hussain2008	0,5	0,5	0,5	1,50
lif1999	0	0,5	0,5	1,00
memmel2007	0	0	0	0,00
moreno2009	0	0	0	0,00
nielsen1992	0	1	0	1,00
rafla2007	0,5	0,5	0	1,00
sousa2005	0	0	0	0,00
wang2008	0,5	0	0	0,50

On the other hand, the potential industrial relevance is described by the realism of the environment in which data were gathered and by the research method used [34]. The realism of the environment is formed of subjects (practitioners, students, or researches), scale (toy example for industrial scale applications), and context (academia or industry). In relation to the research method, the model considers the methods which facilitate research in real environments, such as action research or case studies, as being the most suitable means to contribute to relevance. All the items in this category are scored as contributes (1) or does not contribute (0).

Table 6 presents the results related to potential industrial relevance. Practitioners only took part in the evaluation in three papers. With regard to the context, eight studies reported that they took place in industry, while six reported that they were conducted in projects similar to industrial ones. The most widely used research methods (Fig. 4) were the application of technology (38%) and case studies (33%). The remaining papers reported their research methods as being action research (6%), experiment (6%), and survey (6%).

According to the model, case studies, field studies, surveys, and action research are scored with the value 1. The papers with a relevance that is greater than two are anderson2001, glissmann2005, and lif1999.

Table 6: Potential industrial relevance scoring

Paper	Subjects	Context	Scale	Research method	Relevance
anderson2001	1	1	1	Application	3,00
biel2010	0	0	0	Case study	1,00
bonacin2009	0	0	0	Application	0,00
castro2008	0	0	0	Case study	1,00
constantine2002	0	0	0	Application	0,00
ferre2005	0	1	0	Application	1,00
glissmann2005	0	1	1	Case study	3,00
goransson2003	0	0	0	None	0,00
helms2006	0	1	0	Field study	2,00
hornbaek2007	0	0	0	No described	0,00
hussain2008	0	0	0	Application	0,00
lif1999	1	1	1	Action Research	4,00
memmel2007	0	1	1	Application	2,00
moreno2009	0	0	0	Case study	1,00
nielsen1992	1	1	1	Survey	4,00
rafla2007	0	0	0	Experiment	0,00
sousa2005	0	1	1	Case study	3,00
wang2008	0	0	0	Application	0,00

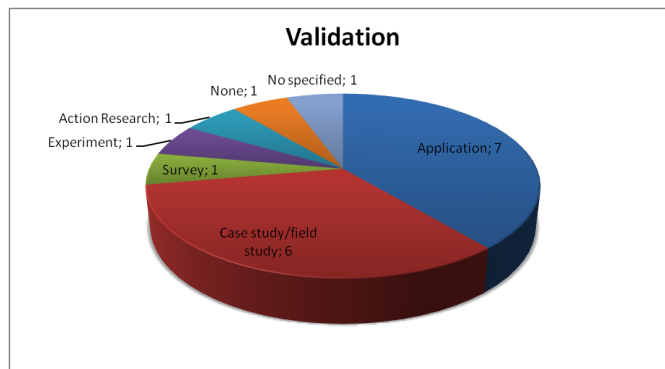


Figure 4: Empirical validation types reported in the primary studies

Finally, the distribution of the papers with regard to rigor and relevance is presented in the bubble chart (Fig. 5). The biggest bubbles are at the bottom, showing a rigor score of zero. Only five articles have a relevance that is greater than two, but they have little rigor.

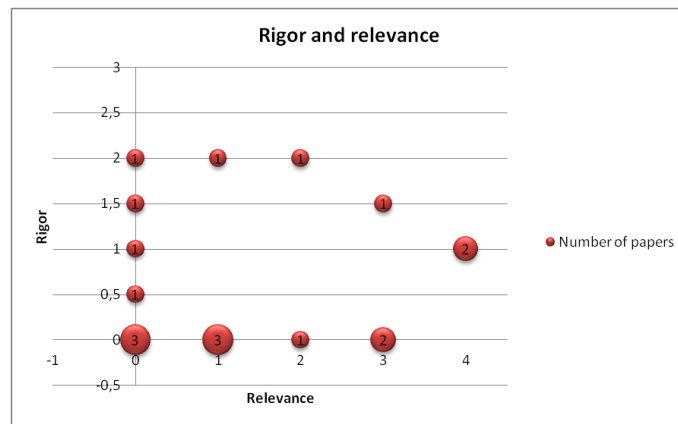


Figure 5: Rigor and relevance of primary papers

The model shows that the rigor and relevance of primary papers are distributed in a wide range, while the majority of them are below the midpoint of both scales. This result was expected given the low number of papers which propose a process (or method) by which to improve software product usability. In addition, some researchers have reported that papers in Software Engineering emphasize conceptual analysis or propose new technology, but there are few empirical studies [35, 36]. In fact, some research papers demonstrate that a technology works or report lessons learned without considering study validity [36].

6 Conclusions

In this article we have presented the results of a systematic review to discover what software development processes should advocate in order to improve the usability of the final products. The proposed classification of primary articles helps to understand the areas which have been researched with regard to usability and software processes. The areas that have in which most interest has been shown are based on RUP adaptations, general software engineering framework mappings, agile method adaptation and the introduction of usability-oriented techniques in the analysis phase.

The proposals which introduce user centered design practices agree on the need to gather usability requirements from the user, understand the context of use, and develop interfaces and interaction mechanisms through prototyping and user evaluation. Moreover, the set of primary articles oriented towards usability requirements specification have researched the adaptation of usability techniques to support requirements elicitation and to improve usability evaluation of software architecture.

The activities and practices in the primary studies were mapped onto the ISO/IEC 12207 system technical and software implementation process in order to identify the processes that an organization must change if it intends to improve the usability of its software product. We found that suggested practices have an impact on stakeholder requirements gathering, system requirements analysis, software requirements analysis, and high-level software design process. We also found, to a lesser extent, that some of the practices suggested affect the testing qualification, acceptance testing, and operation processes.

The analysis of the primary studies in this SLR has contributed towards the identification of tasks that could be integrated into the software development process. However, there is still little empirical evidence of their level of effectiveness when applied to real projects. Furthermore, while some proposals considered ISO standards with regard to integrating support tasks for usability, we found no references to software product quality standards such as ISO 9126. From an engineering perspective, it is important to measure quality attributes, but very few proposals include product or process measurement. The design of processes that integrate usability into full software development, as is suggested in [11], therefore remains valid.

As a result, it can be concluded that the number of relevant papers is low. Indeed, the reporting of rigor and relevance could be improved. This therefore motivates further studies through which to discover the relationships between process capability and product usability.

As a future work, we intend to broaden this analysis in order to determine the impact of usability practices on other categories of the processes in ISO/IEC 12207, such as those related to quality assurance. Moreover, this work is focused on the activities in the process, but is necessary to consider other elements such as the associated artifacts and roles. What is more, empirical studies are needed to assess the impact of practices on software usability, and indicators with which to measure the degree of usability achieved should be developed. Finally, the analysis of the relationship between the development process and software quality characteristics can be extended to other factors, such as those described in ISO / IEC 25010.

Acknowledgments. This work has been funded by the GEODAS-BC project (Ministerio de Economía y Competitividad, and Fondo Europeo de Desarrollo Regional FEDER, TIN2012-37493-C03-01) and PEGASO/MAGO (Ministerio de Ciencia e Innovación MICINN and FEDER, TIN2009-13718-C02-01).

References

- [1] M. Staples and M. Niazi, "Systematic review of organizational motivations for adopting CMM-based SPI," *Information and Software Technology*, vol. 50, pp. 605-620, 2008.
- [2] B. Kitchenham and S. Pfleeger, "Software quality: the elusive target " *IEEE Software*, vol. 13, pp. 112-21, 1996.
- [3] M. Agrawal and K. Chari, "Software effort, quality, and cycle time: a study of CMM level 5 projects," *IEEE Transactions on Software Engineering*, vol. 33, pp. 145-156, 2007.
- [4] J. Herbsleb, D. Zubrow, D. Goldenson, W. Hayes, and M. Paulk, " Software quality and the Capability Maturity Model," *Communications of the ACM* vol. 40, pp. 31-40, 1997.
- [5] S. De Oliveira, R. Valle, and C. Mahler, "A comparative analysis of CMMI software project management by Brazilian, Indian, and Chinese companies," *Software Quality Journal*, vol. 18, pp. 177-194, 2010.
- [6] N. Ashrafi, "The impact of software process improvement on quality: in theory and practice," *Information and Management*, vol. 40, pp. 677-690, 2003.
- [7] J. Zhou and T. Stalhane, "Web-based system development: status in the Norwegian IT organizations," vol. Bomarius e Iida (Eds.): PROFES 2004, LNCS 3009, pp. 363-377, 2004.
- [8] J. Y. C. Liu, V. J. Chen, C.-L. Chan, and T. Lie, "The impact of software process standardization on software flexibility and project management performance: Control theory perspective," *Information and Software Technology*, vol. 50, pp. 889-896, 2008.
- [9] J. Allen, B. Kitchenham, and M. Konrad, "Theme Q. The relationships between processes and product qualities," in *A Process Research Framework*, E. Forrester, Ed., ed Carnegie Mellon: Software Engineering Institute, 2006, pp. 19-28.
- [10] ISO, "ISO/IEC FCD 25010: Systems and software engineering - system and software product quality requirements and evaluation (SQuaRE) - System and software quality models," ed, 2010.
- [11] A. Seffah and E. Metzker, "The obstacles and myths of usability and software engineering," *Comm. Of the ACM*, vol. 47, pp. 71-76, 2004.
- [12] Kitchenham and S. Charters, "Guidelines for performing Systematic Literature Reviews in Software Engineering " vol. Technical Report EBSE/EPIC-2007-01., 2007.
- [13] J. Anderson, F. Fleak, K. Garrity, and F. Drake, "Integrating usability techniques into software development," *Software, IEEE*, vol. 18, pp. 46-53, 2001.
- [14] B. Biel, T. Grill, and V. Gruhn, "Exploring the benefits of the combination of a software architecture analysis and an usability evaluation of a mobile application," *Journal of Systems and Software*, vol. 83, pp. 2031-2044, 2010.
- [15] R. Bonacin, B. Calani, and M. Rodriguez, "An Agile Process Model for Inclusive Software Development," vol. 24 LNBIP, pp. 807 - 818, 2009.
- [16] J. W. Castro, S. T. Acuña, and N. Juristo, "Integrating the Personas Technique into the Requirements Analysis Activity," in *Computer Science, 2008. ENC '08. Mexican International Conference on*, 2008, pp. 104-112.
- [17] L. L. Constantine and L. A. D. Lockwood, "Usage-centered engineering for web applications," *IEEE Software*, vol. 19, pp. 42-50, 2002.
- [18] X. Ferre, N. Juristo, and A. M. Moreno, "Framework for integrating usability practices into the software process," in *Lecture Notes in Computer Science 3547*, 2005, pp. 202-215.
- [19] S. Glissmann, S. Smolnik, R. Schierholz, L. Kolbe, and W. Brenner, "Proposition of an m-business procedure model for the development of mobile user interfaces," in *Mobile Business, 2005. ICMB 2005. International Conference on*, 2005, pp. 308-314.

- [20] 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*, vol. 8, pp. 111-131, 2003.
- [21] 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*, vol. 79, pp. 841-858, 2006.
- [22] K. Hornbæk, R. T. Høegh, M. B. Pedersen, and J. Stage, "Use case evaluation (UCE): A method for early usability evaluation in software development," vol. 4662 LNCS, ed, 2007, pp. 578-591.
- [23] Z. Hussain, M. Lechner, H. Milchrahm, S. Shahzad, W. Slany, M. Umgeher, and P. Wolkerstorfer, "Agile user-centered design applied to a mobile multimedia streaming application," vol. 5298 LNCS, ed, 2008, pp. 313-330.
- [24] M. Lif, "User-interface modelling--adding usability to use cases," *International Journal of Human-Computer Studies*, vol. 50, pp. 243-262, 1999.
- [25] T. Memmel, H. Reiterer, and A. Holzinger, "Agile methods and visual specification in software development: A chance to ensure universal access," vol. 4554 LNCS, ed, 2007, pp. 453-462.
- [26] L. Moreno, P. Martínez, and B. Ruiz-Mezcua, "Integrating HCI in a web accessibility engineering approach," vol. 5616 LNCS, ed, 2009, pp. 745-754.
- [27] J. Nielsen, "The usability engineering life cycle," *Computer*, vol. 25, pp. 12-22, 1992.
- [28] T. Rafla, P. N. Robillard, and M. Desmarais, "A method to elicit architecturally sensitive usability requirements: Its integration into a software development process," *Software Quality Journal*, vol. 15, pp. 117-133, 2007.
- [29] K. Sousa, E. Furtado, and H. Mendonça, "UPi - A software development process aiming at usability, productivity and integration," in *ACM International Conference Proceeding Series 124*, 2005, pp. 76-87.
- [30] X. Wang and Y. Shi, "UMDD: User model driven software development," in *5th International Conference on Embedded and Ubiquitous Computing*, 2008, pp. 477-483.
- [31] E. Folmer, J. van Gurp, and J. Bosch, "A Framework for Capturing the Relationship between Usability and Software Architecture," *Software Process Improvement and Practice*, vol. 8, pp. 67-87, 2003.
- [32] L. Bass, B. John, and J. Kates, "Achieving Usability through Software Architecture," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA2001.
- [33] ISO, "ISO/IEC FDIS 12207 Systems and software engineering - software life cycle processes," ed, 2007.
- [34] M. Ivarsson and T. Gorschek, "A method for evaluating rigor and industrial relevance of technology evaluations," *Empirical Software Engineering*, vol. 16, pp. 365-395, 2011.
- [35] R. Glass, K. Vessey, and V. Ramesh, "Research in software engineering: an analysis of the literature," *Information and Software Technology*, vol. 44, pp. 491-506, 2002.
- [36] D. I. K. Sjøberg, T. Dybå, and M. Jørgensen, "The future of empirical methods in software engineering research," in *FoSE 2007: Future of Software Engineering 2007*, pp. 358-378.