Towards an approach matching CMD and DSR to improve the Academia-Industry software development partnership

A case of agile and UX integration

Joelma Choma
UFSCar – Campus Sorocaba
Rdv. João L Santos, Km 110
Sorocaba-SP, Brazil
jchoma@ufscar.br

Luciana A.M. Zaina
UFSCar – Campus Sorocaba
Rdv. João L Santos, Km 110
Sorocaba-SP, Brazil
lzaina@ufscar.br

Tiago Silva da Silva
ICT – UNIFESP
Universidade Federal de São Paulo
São José dos Campos-SP, Brazil
silvadasilva@unifesp.br

***Abstract*—The interest on promoting the Academia-Industry partnership in the software development field has been increasingly encouraged by means of research approaches that support the cooperation between researchers and practitioners. The main focus is driven by the collaborative work where the scientific research work meets the real needs of the Industry. Aiming to contribute to this effort we present an approach called SoftCoDeR (Software Cooperative Design Research) that combines CMD (Cooperative Method Development), a method of Action Research, to concepts of DSR (Design Science Research). Our proposal supports the conduction of projects integrating the viewpoints of Industry and Academia, aiming to add User eXperience (UX) methods into agile practices. We carried out two studies applying the SoftCoDeR approach in a software Industry, work in close cooperation with UX and SCRUM teams for building and evaluating artifacts based on UX methods to support practical activities of both teams.**

Keywords—Action Research; Design Science; Software Engineering; SCRUM; HCI; Interaction Design; User Experience.

I. Introduction

The need for sharing experiences and knowledge between Academia and software Industry has been becoming increasingly evident. However, the Academia-Industry interaction has demonstrated difficulties on establishing an effective relationship, because they usually have different viewpoints.

Mechanisms have been proposed to balancing the needs of research and industry throughout the development software process [1], and also to promote a strategic alignment between Academia and Industry, from which research should be driven by the needs and interests of the software Industry [2].

From this perspective, researchers need to develop skills to better understand practical problems observed in real scenarios from the experience of teams within the software development environment. Furthermore, researchers engaged in projects with Industry are encouraged to perform an active role during the investigation in order to maximize the use of the research results in the practice.

The active cooperation among researchers and practitioners is considered an important factor for the successful introduction of new technology in the organization [3]. In this context, technology refers to any artifacts produced by activities within software development, including concepts, tools, techniques, methods and methodologies. Artifacts are essential to the software process because they can report all information created over a period of development activities.

This paper presents an approach called SoftCoDeR (Software Cooperative Design Research) to encourage the partnership between Industry and Academia, bringing closer researchers to practitioners of software engineering. In our first experiments, the approach proved to be feasible in a research project which aimed to integrating User eXperience (UX) methods into agile practices. One of the research project goals was the development of artifacts to integrate the viewpoint of UX into SCRUM agile methodology. The SoftCoDeR approach allowed the researchers to work closely in cooperation with practitioners of the software development area, in order to link scientific knowledge (Academia) to the real needs of the Industry.

Usually the teams (UX and SCRUM) make use of different types of artifacts to support their activities and to communicate information produced throughout the development process. Artifacts defined in SCRUM are intended to maximize the transparency of information and support decisions during a software development project [4]. While the UX artifacts are to provide information regarding interaction and user requirements, and to offer design solutions [5]. Our experience was based on the building of artifacts which brought value for both UX and SCRUM teams during the software development process.

The remainder of this paper is organized as follows: Section II introduces the basis of our proposal - an overview of two research methods and related works; Section III outlines the proposal approach; Section IV presents the approach applied in practice; Section V discusses the outcomes of the prac-

978-1-4673-9272-3/15 $31.00 © 2015 IEEE
DOI 10.1109/SBES.2015.18

tices; and finally Section VI presents the conclusions and points out directions for further works.

II. Background

In this section, we initially provide an overview of two methods that based the proposal approach. Therefore we present related works.

A. Action Research and Cooperative Method Development

Action Research (AR) is an iterative process involving researchers and practitioners acting together on a particular cycle of activities, including problem diagnosis, action intervention, and reflective learning, as defined by Avison et al. [6].

AR has historical roots in the sociotechnical movement, whose origin is associated with Kurt Lewin's application of experimental logic and social psychological theory for practical social problems in the 1940s [7]. Since then, AR has evolved and has been applied in several scientific areas (*e.g.* nursing, education, management, information systems). Santos and Travassos [8] state that there is an increasing tendency on using AR in SE to address different research topics. According to them, AR has been seen as an alternative methodology that stimulates in-depth investigation of the practices of SE in an industrial context, intensifying the execution of relevant studies and adding value to the results. However, they have identified that the rigour and control of AR studies should be improved in SE studies.

In our proposal, we have used the concepts of the approach proposed by Dittrich [9] called Cooperative Method Development (CMD), which is an adaptation of AR that focuses on software development practices. CMD combines qualitative social science fieldwork, with problem-oriented methods, technique and process improvement. The CMD research process is modelled as evolutionary cycles consisting of three phases: (1) *Understanding Practice*, (2) *Deliberate Improvements*, and (3) *Implement and Observe Improvements*. The research begins with a phase of *Understanding Practice* with qualitative empirical investigations aiming at understanding and explaining practices from a practitioner's viewpoint, identifying aspects that are problematic for those involved in project. The second phase of *Deliberate Improvements* results on measures, that can be expected to improve the situation identified in the first phase, addressed by the cooperation between researchers and the practitioners. In the third phase *Implement and Observe Improvements*, firstly the improvements are implemented, and the researchers follow the improvements as participant observers. Afterwards, the results are evaluated together with the practitioners involved. The three phases can be repeatedly applied in the same context.

B. Design Science Research

Design Science (DS) originally arose from Herbert Simon's work entitled "The Sciences of the Artificial" published in 1969, whose concept is anchored to many professional disciplines like architecture, economics and engineering [10]. DS addresses the research by building and evaluating artifacts, which are designed to meet business needs identified in a particular environment [11]. The artifacts typically produced by DS Research (DSR) are related to concepts, models, methods, and instantiations [12]. DSR is an approach commonly adopted in the area of Information Systems (IS), but it has been implicitly practiced in SE for decades [13].

Hevner et al. [14] proposed a conceptual framework combining DS and Behavioral Science to drive research in the field of IS. In another paper, Hevner [15] defined three cycles to enhance the understanding of DSR. These research cycles within DSR framework proposed by the author are shown in Figure 1. According to him, the *Relevance Cycle* connects the contextual *Environment* of the research project *(a)* to the *DSR* activities *(b)*. The main focus of the *Relevance Cycle* is (i) to capture problems to be addressed or requirements for the research and (ii) to provide design solution artifacts for the *environment* of study, and evaluation in the application domain. The *Rigor Cycle* connects the *DSR* activities *(b)* with the *Knowledge Base (c)* that supplies the research project, ensuring innovation by providing existing knowledge for the research. The *Knowledge Base (c)* consists of foundations, existing experiences and expertise, and existing artifacts and processes. The main focus of *Rigor Cycle* is to provide applicable knowledge for *DSR* activities *(b)* and to feedback the updated knowledge to enrich the *Knowledge Base*. The *Design Cycle* (located in the center of Figure 1) operates between the activities of *building and evaluating* to design artifacts and processes from the research. These three cycles must be clearly identifiable in a research project, since design activities must be convincingly based on relevance and rigor.

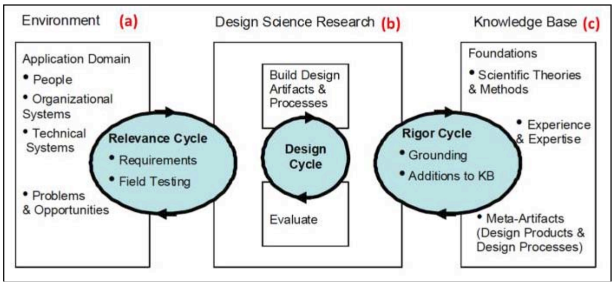


Fig. 1. Design Science Research Cycles. Source [15]

C. Related Work

According to Dittrich et al. [16], CMD has been applied in different research projects outlining different aspects within software engineering (*e.g.* design of flexible and adaptable software; agile development for e-government applications; and the integration of interaction design and software development). Recently, Al-Baik and Miller [17] have implemented CMD in an investigate study in order to understand the potential reasons behind the relatively low success rate of lean initiatives in IT organizations. As a result, they proposed a new model to classify wastes in IT organizations. Ardito et al. [18] applied CMD - in companies from southern Italy - aiming to inquire the impact of UX methods into the software development practices, involving practitioners, and looking the aspects from inside the companies. The authours confirmed that many companies either neglect usability and UX, or can not properly consider them.

Regarding the DSR, Adikari et al. [19] adopted the Hevner et al.'s framework [14] to develop an approach called Little Design Up Front (LDUF) aiming to integrate the User-

Centered Design perspective into Agile Requirements Engineering. The LDUF approach was evaluated in two agile projects; and in the evaluation of the approach, the qualities of easier to learn, and easier to use were pointed out by the users who used the products developed from LDUF. Recently, Rodríguez et al. [20] have also applied Hevner et al.'s framework in their projects to explore the main elements that characterize the combination of Agile and Lean methods in software development. They illustrate how DSR can be applied in empirical software engineering, reporting in lessons learned that such research methods have a good potential for supporting collaboration between Industry and Academia.

Some approaches combining AR and DSR have been proposed to drive research in the field of IS [21] [22] [23]. Baskerville et al. [21] proposed Soft DSR approach that incorporates aspects of Soft Systems Methodology (that is a form of AR) into a DSR process. Sein et al. [22] presented a method called Action Design Research (ADR) that defines four stages for performing a research: (1) Problem Formulation; (2) Building, Intervention, and Evaluation; (3) Reflection and Learning; and (4) Formalization of Learning. Both approaches - Soft DSR and ADR - are classified as "design-oriented action research". In a different way, Wieringa and Morali [23] proposed a method called Technical Action Research (TAR) that follows an artifact-driven approach, in which the AR is used only in the evaluation of the artifacts' viability in a given context.

III. The SoftCoDeR Approach

From the natural compatibility between AR and DSR we can observe that both are (1) interventionist methodologies (intervenes rather than study a phenomenon after the fact); and (2) they are closely related to problem-solving, and involving the evaluation of the solutions proposed [24]. Moreover, both methods are based on the argument that the research should ever contribute for both to Academia as well as to Industry. According to Baskerville [10], although 'doing' design science may look a lot like 'doing' action research, there are fundamental differences between them such as: (1) AR is focused on problem solving through social and organizational change, while DSR is focused on problem solving by creating and positioning an artifact in a natural setting; and (2) AR is clearly centered on discovery-through-action, while DSR is clearly centered on discovery-through-design.

Iivari and Venable [13] identified the following levels of overlapping for activities performed in the processes of AR and DSR: (1) non-overlapping; (2) somewhat overlapping; and (3) significantly overlapping. Non-overlapping occurs when the DSR is adopted for solving a purely technical problem, or even to solve a socio-technical problem in which there is no interest in discovery-through-action. The second case - somewhat overlapping - concerns to the use of AR to evaluate DSR, as *e.g.* when an action researcher is interested on evaluating an existing technological solution without engaging himself on the construction of artifacts. The significant overlap is when the researcher adopted AR aiming to discovery something through action, and additionally s/he is working on the construction of innovative artifacts addressed for client's problem solving. In this case, the research interest includes both the building and evaluation of artifacts, while working in close collaboration with the practitioners of the company.

Considering the aforementioned perspectives of combining AR and DSR, we have chosen to follow a similar approach to the proposals of Baskerville et al. [21] and Sein et al. [22], with regards to the research activities that are significantly overlapping. However, we chose the CMD method - based on AR that was proposed by Dittrich [9] - because it is a more adherent method to the SE practices, focusing on shop floor software development practices. In addition, the CMD approach provides a structured research process, which clearly defines three phases of research for understanding the practice, and for the deliberation, implementation, and evaluation of improvements. The CMD approach provides further guidance such as the use of ethnographic studies for understanding a practice from a practitioners' viewpoint. Interviews, document analysis, and workshops are some of the data gathering techniques commonly used as qualitative methods, for the empirical fieldwork. The CMD approach does not exclude quantitative methods, whether they appear to be useful to complement the qualitative data (*e.g.* descriptive statistics). Besides, the CMD encourages short iterative cycles to put the improvements in practice. This is totally adherent to the agile principals.

The DSR concepts were added to the proposal approach in order to provide guidelines to produce artifacts of value, an extremely relevant issue to the agile teams that will be used to support the activities of SE professionals. The DSR draws the attention of researchers both for aspects regarding problem relevance, as well as those that concern the research's rigor. The objective of DSR is to develop artifacts to solve problems that are relevant within a specific context, based on the application of rigorous methods both on building, as well as in the evaluation of artifacts.

Figure 2 shows our approach, which we have called SoftCoDeR - Software Cooperative Design Research.

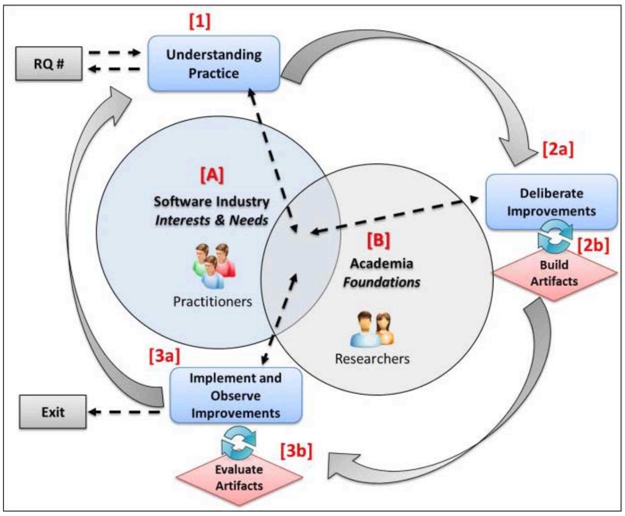


Fig. 2. SoftCoDeR - Software Cooperative Design Research

In the SoftCoDeR approach we have considered the three steps of CMD in terms of an iterative cycle: Understanding Practice (1), Deliberate Improvements (2a), and Implement and Observe Improvements (3a). From the best practices of producing valuable artifacts (DSR), we added an activity to Build

Artifacts (2b), and to Evaluate Artifact (3b). The *intersection* (A∩B) of information between *Industry (interests and needs)* and *Academia (foundations)* will provide guidelines through all the stages of the research cycle.

The first step of the SoftCoDeR consists of *Understanding Practices* (1) from the practitioners' viewpoint, aiming to identify the *needs* for new solutions or improvements that should be aligned to the *interests of Industry* (A). The *foundations* from the *Academia* (B) contribute so that the *researcher* identifies concepts that are used (or not) in practice in contrast to other theories and technologies that have already been employed in other companies. Ethnographic studies and interviews are the qualitative techniques more used to gather data on practices of software development. A *research question* (RQ) can be anticipated by researchers early in this phase, or can be formulated during the first stage. The RQ should emerge based on issues observed about the practice, in order to drive the next steps (2 and 3).

The trigger for the next step is the RQ formulated in the previous phase, which should guide the action to *Deliberate Improvements* (2a). During this second stage, Academia and Industry members meet to analyze the results gathered by the researcher regarding the Understanding Practice (1). Together (Academia and Industry members) they build a scenario which reproduces the problem-situation that they target to solve based on the RQ previously defined. Before proposing any solution, practitioners and researcher should thoughtfully examine the scenario; and then the answers come up from the intersection of information and knowledge of Industry and Academia experiences. In this step the *artifacts* of value, required to support the practitioners' work, are identified according to two viewpoints: technical function (the academic view), and of pragmatic function (the Industry view). Subsequently, these *artifacts* should be elaborated and *Built* (2b) by researchers and practitioners. This step is cyclical to allow the discussion to improve the artifacts.

In the third step, Industry members define and choose a practical project through which it will be possible to *Implement and Observe Improvements* (3a), and *Evaluate Artifacts* (3b). The artifacts can be evaluated in terms of practicality, efficiency or effectiveness, for example. Many evaluation methods can be adopted individually or in combination such as: laboratory experiments, field experiments, simulations, case studies, surveys, and ethnographic studies [11]. In some cases, an experimental validation is strongly recommended before to put the improvements in practice, especially to minimize the impact of introducing the new technology (artifact) in practice. The evaluation phase should provide feedback regarding the built artifacts, and any other improvements that were implemented. New interactions may be required to refine the artifacts, or even to suggest new improvements. The SoftCoDeR approach has an iterative and cyclical nature. New research question can be generated in each cycle based on the lessons learned. Thus, the cycle can be incremental, adding information for each new cycle or spiral, since the information can arise during the evaluation phase.

IV. The SoftCoDeR in Practice

This section presents the general settings of the environment in which the SoftCoDeR has been applied. We also describe the two cycles applied for answering the specific research questions involving the building and evaluating of three artifacts that integrate UX methods into the environment of agile development software.

A. Project Context and Research Theme

The issues addressed in this article are partial results of a research project that has been developed in partnership with a company that develops ERP (Enterprise Resource Planning) systems. Currently, the company has 700 employees, approximately 2,000 customers and more than 50,000 users. The company has been active in several market segments such as Manufacturing, Logistics and Agribusiness. In recent years, the executives of the company have concentrated its efforts to evolve their software development processes, incorporating best practices for the development of user interfaces (UI) by user-centered design. They believe that integrating UX practices into the development of ERP systems is a key aspect to fill an important gap of the development process, and consequently a way to stand out in their market segment.

The research theme is Interaction Design addressed to ERP Systems. The objective of the research is to propose methods and artifacts suitable to the ERP systems domain integrating UX concepts within the SCRUM. We (researchers) have worked in collaboration with the company's practitioners in many *actions* aiming to propose for instance (i) mechanisms and techniques for identifying the needs of users; (ii) standards and guidelines for the development of user interfaces; (iii) effective usability evaluation methods for ERP systems; and (iv) how to improve the communication of usability issues between SCRUM and UX teams.

In recent years, UX has been spread widely and well received by Academia and Industry. This increasing interest in UX can be explained by the fact that it is a broader concept than the concept of usability [25]. In addition, the insertion of UX methods into the software development process has been seen as an important competitive advantage for companies. UX is a concept that encompasses all aspects of user interaction; focusing to solve problems in an interdisciplinary, holistic and targeted way for achieving a deep understanding of behavior, cognition, abilities, desires and human context [26]. User eXperience Design (UXD) aims to provide a pleasant experience for users by means of simplicity, elegance, efficiency, effectiveness and emotional satisfaction of the people. The Human–Computer Interaction (HCI) field, in turn, provides techniques and methodologies that allow us to study and develop the user experience from the perspective of the Interaction Design.

The integration of UX methods into software development processes based on agile methodologies is not an easy task. Indeed, agile software development practices have features that foster the integration, *e.g.*, iterative processes, focus on the quality, and work in collaboration with different stakeholders [27]. However, there are many issues in which it can complicate such integration [28], *e.g.*, promoting synergy and establishing a good communication between the agile and the UX teams [5]. In particular, this issue involves dealing with the difference of mindset between agile developers and UX designers, who use different practices and express in their particular way from their knowledge areas.

Next we describe two cycles performed from the SoftCoDeR approach that have brought results and contributions to the research on interaction design addressed to ERP Systems.

B. First Cycle

Our first experience with the SoftCoDeR approach arose from an issue that the industry partner reported to us: the difficulty of communication between UX and SCRUM teams due to differences in vocabulary. Taking this issue into account, we consider the use of concepts that could approximate the different viewpoints. Our first insight was the use of two concepts – personas [29] and Nielsen's heuristics [30] - as a possible solution. Considering the problem-situation, and the use of the concepts mentioned, we have defined two research questions that guided the first cycle of research:

- RQ1: Can the concepts of personas and Nielsen's heuristics level out the awareness and concerns on usability aspects of UX designers and developers (programmers and testers)?
- RQ2: How could personas and Nielsen's heuristics concepts be used as a common vocabulary for the communication between the UX and SCRUM teams?

1) Understanding Practice

We started based on Personas and Nielsen's heuristics concepts, which are recognized by the UX experts as good practices to keep the software development focus on the needs of the end user, and on the software usability [31] [32]. Personas are used in the software development industry to describe a group of real users and their features through hypothetical archetypes. The use of personas can guide the development of interaction scenarios and/or be used to describe the typical tasks in usability testing [29]. The application of the personas technique has proved that designing for a small set of personas can meet a significant number of users due to their similar goals and features. The construction of personas should be grounded in the research on target users, which sometimes can require a long time. However, personas have been adapted to be light and lean, aiming to help agile developers and other stakeholders to elicit the user requirements [32].

Nielsen's heuristics [30] are usability guidelines commonly used to support the critical analysis in usability inspections (heuristic evaluation), but also can be used to drive the design of interactive interfaces, and also to confirm problems identified in usability testing [31]. The ten Nielsen's heuristics are: Visibility of system status (H1); Match between system and the real world (H2); Control and freedom for user (H3); Consistency and standards (H4); Error prevention (H5); Recognition rather than recall (H6); Flexibility and efficiency of use (H7); Aesthetic and minimalist design (H8); Help users recognize, diagnose, and recover from errors (H9); Help and documentation (H10).

Considering our main goal in the research project of Interaction Design to ERP Systems, we have been worked the mindset of the practitioners about the importance of the UX in the software development process through workshops evolving the UX team and SCRUM team. In this cycle, and to help us on investigation of our first RQ (RQ1), we carried out a workshop entitled "Usability Heuristics for ERP system" where we introduced the Nielsen's heuristics and personas concepts. After that, we interviewed the practitioners who had attended in the workshop.

The workshop's goal was spreading the concepts and practices of the application of personas and Nielsen's heuristics in order to identify the value that this concept may have to the participants. The 59 practitioners who participated in the workshop were from different areas: developers (36), testers (12), analysts (5), technical leaders (4) and software architects (2). Most participants had less than five years of experience in software development, and 56% had less than five years of experience in developing ERP systems. Only 11 participants (18.64%) had more than ten years of experience in software development and 9 participants (15.25%) had experience in the development of ERP systems. During the workshop the participants could work with the concepts in practical examples in ERP system scenarios.

Aiming to investigate the impact of the workshop in the daily practice of the practitioners, we carried out, a few weeks after the workshop, an interview to observe: (i) the expertise and skills obtained from the workshop; and (ii) the contribution of the workshop in changing the participants' viewpoint on the usability aspects in the software development, especially in ERP systems. Table 1, we present an overview of the interviewees' profile.

TABLE I. OVERVIEW OF INTERVIEWS

Interviewee	*Role*	*Software development experience*	*ERP Systems development experience*	*Interview length*
P1	Tester	4 years	2 years	17min39s
P2	Developer	8 years	5 years	16min53s
P3	Tester	13 years	9 years	22min45s
P4	Developer	11 years	4 years	15min23s
P5	Tester	3 years	3 years	17min02s
P6	Developer	6 years	3 years	10min08s
P7	Architect	9 years	9 years	20min42s
P8	Tester	4 years	2 years	16min46s
P9	Architect	27 years	27 years	19min06s
P10	Developer	4 years	2 years	15min12s

The interviews were audio recorded with the prior permission of the interviewees, and later transcribed to a report. During 1 day, we performed a semi-structured interview, including questions that summarize three aspects: (i) their viewpoint of usability aspects, (ii) their perception for practical application of the Nielsen's heuristics and personas in ERP systems, and (iii) their evaluation regarding the improvements on ERP systems brought by the application of usability issues. We analyzed qualitatively the participants' answers, categorizing them in the three points listed above.

Regarding the first observed aspect about *(i) viewpoint of usability aspects*, the majority of the respondents say that they are more careful and pay attention in the simple details that can improve the usability of the system. They reported to be more concerning on the needs and perceptions of the users (the personas). Some participant's comments were:

"(...) I pay more attention and recognize that simple points are important improvements which we should do (...) the workshop opens my mind of user interaction." [P4]

"(...), now when we look at a new software interface we try to imitate the persona's interaction, we are able to identify usability problems." [P8]

"(...) we are working on changes of error messages of a project because we noted the messages had no significance to the user (...)" [P2]

With regards on the second aspect, *(ii) their perception for practical application of the Nielsen's heuristics and personas in ERP system*; all respondents agree with the usefulness of the heuristics and personas in the practice of ERP development. They believe that heuristics can guide the usability verification and aid the developer in finding alternatives to solve usability issues. The personas have a psychological influence, because they have a concrete person that represents the users. Some participant's comments were:

"(...) the heuristics and personas made my work easier (...) we developed a new user interface applying the concepts of colors in a caption. We had not decided to use them randomly; we had looked for color patterns." [P8]

"(...) In my opinion, the heuristics showed to me that my concerns should not be only about the functional aspects, I also have to consider the screen, the content arrangement, pattern, etc. (...) After seeing the personas, I become more critical about the user interaction issues." [P3]

"(...) the heuristics could be used as a checklist during the software tests." [P7]

Finally, the third aspect *(iii) their evaluation regarding the improvements on ERP systems brought by the application of usability issues*; although the ERP system was complex, the respondents believe that it is possible to make the user experience more enjoyable. Some examples of comments were:

"Some people and some professionals believe that in big and complex systems, like ERP systems, the usability is not important. They think the user is familiar with the idea of a difficult system. On the other side, I do not agree. Even though ERP systems are complex and require complicated processes, I think our job, as developers, is to facilitate the user interaction, through the usability heuristics application, for example." [P1]

"The ERP system is really complex, I have no doubt about it, but it is always possible to improve the user interaction." [P9]

Overall, we noted that the subjects and activities of the workshops have been enough to positively influence the participants' mindset about the importance of usability in ERP systems. We also observed that Nielsen's heuristics can set important topics on usability in people's mind. Based on the interviews outcomes, we found evidences to answer our first question (RQ1). The participants considered that personas and Nielsen's heuristics can guide product development and can also be used as a common vocabulary among the developers, testers and UX designers.

2) Deliberate Improvements and Build Artifact

Considering the outcomes of the workshop and the interviews, researchers met with project leaders and the UX designer to evaluate impact of the use of artifacts applying these two concepts - personas and Nielsen's heuristics - in the company projects. The project leaders pointed out that the hard efforts on building of personas - requiring a prior knowledge of the potential users - could make the personas adoption infeasible. However, we found that the company had a knowledge of the users' profile from its after-sales department. Then the personas could be designed from this previous knowledge. And many of the personas could be reused in different projects. Then the leaders agreed with us. Another important practical point we discovered was that the UX teams usually carried out usability tests with proposed interfaces, collected the problems and reported the proposal for the SCRUM team from a low-fidelity prototype stating the details orally or by annotations in natural language. According to UX and SCRUM teams, this mechanism of communication has been caused misunderstanding between the teams.

From the discussions between Academia and Industry, and also to answer the second research question (RQ2), we have proposed building two artifacts (2b):

a) Lean Personas: represents a set of end-users, described briefly, from photo, name, age, expertise area, knowledge (skills and abilities), goals (motives and concerns), and usage patterns (context and usage habits), based on simplified structure proposal for personas specification presented by Miller and Williams [33]. We called them Hypotheses of Personas (HP).

b) Protocol for communication of design solutions and usability recommendations: the new artifact is generated from the analysis of the results of the usability testing session. Firstly, the issues identified during the usability testing session are listed, removing the duplicate issues (the same issues pointed out by more than one persona), and false positives (issues that were not considered real). Secondly, each issue is analyzed, and solutions and/or recommendations are proposed for them. Thirdly, the personas affected by each issue should be pinpointed. Lastly, each issue is mapped to Nielsen's heuristics. Formalizing, the protocol items can be represented in Equation (1).

$$\textbf{Protocol} = \{\textbf{item: item} \equiv \textbf{description, NHsubset, HPsubset}\} \tag{1}$$

where:

$$\text{description} \in \{\text{recommendation, solution}\} \tag{1a}$$

$$\text{NHsubset} = \{nh : nh \in NH\} \tag{1b}$$

$$NH = \{H1, H2, \ldots, H10\} \tag{1c}$$

$$\text{HPsubset} = \{hp : hp \in HP\} \tag{1d}$$

$$HP = \{hypPersona1, \ldots, hypPersona\ j\} \tag{1e}$$

NH = Nielsen's Heuristics HP = Hypotheses of Personas

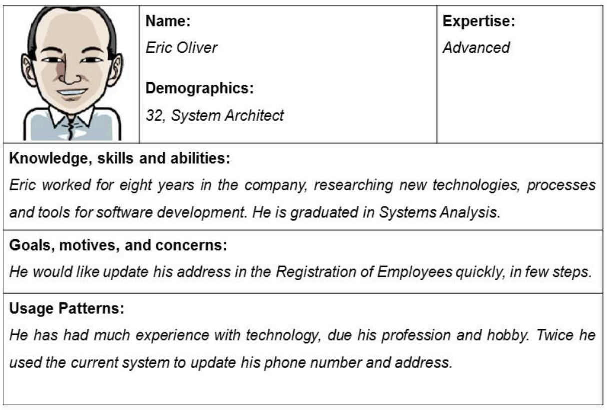
3) Implementing and Observing Improvements and Evaluating Artifacts

The new artifacts were presented to project leaders, UX designer, and SCRUM team in a previously scheduled meeting. Aiming to validate the two built artifacts - Lean Personas and Protocol - we decided along with practitioners evaluate them (3b) through two case studies, in which usability testing was performed on actual redesign project. The project goal was to develop design for a high fidelity prototype to the submodule of Human Resources (Web-based) used for the registration of employees. Noteworthy, the UX designer and SCRUM team that attended the validation, also had knowledge of the Nielsen's heuristics and personas, since they had attended in the workshops.

The use of the new artifacts (personas and protocol) was implemented only in the second case study, in order to compare their use advantages. Thus, in the first case study, we did not interfere in the teams' work; acting in observer's role (for 5 days).The UX designer recruited five users of different profiles, and performed the usability testing playing the role of test moderator. After the test, she analyzed ten issues identified during this testing, and then she stated the issues on a report written in natural language, and delivered it to the SCRUM team during an informal meeting. The SCRUM team has fixed six of the ten reported issues. They explained that the other four were recurrent issues. The UX designer did not agree with them.

In the second case study, we participated along with the UX and SCRUM teams in the usability test, using personas and the protocol to report the usability issues. For this intervention, we used the new version of the functional prototype fixed by the SCRUM team. Before the recruitment of the users, the UX designer built the three lean personas - using the template and checking some details with the after-sales department. The personas were created to represent three different levels of expertise regarding the use of technology: basic, intermediate and advanced. The personas were used to recruit six end-users, all participating for the first time. Figure 3 shows an example of lean persona described by UX designer.

The usability tests were performed and the outcomes were analyzed in a single day. Following steps to review usability issues and generate the protocol based on the Equation 1, the UX designer described the solutions and recommendation according to the template showed in Figure 4; and the report was delivered to the SCRUM team without any explanation. In this usability test were revealed three new issues, and the four issues, which had not been fixed, were confirmed once again. Differently from the first case study, the SCRUM team - who had no doubts regarding the results and recommendations - fixed all the items reported through the protocol.



Name:
Eric Oliver

Demographics:
32, System Architect

Expertise:
Advanced

Knowledge, skills and abilities:
Eric worked for eight years in the company, researching new technologies, processes and tools for software development. He is graduated in Systems Analysis.

Goals, motives, and concerns:
He would like update his address in the Registration of Employees quickly, in few steps.

Usage Patterns:
He has had much experience with technology, due his profession and hobby. Twice he used the current system to update his phone number and address.

Fig. 3. Lean persona described by UX designer

Description	Nielsen's Heuristics	Hypotheses of Personas		
		A	B	C
All users needed help to relate the acronyms of their identity card to the interface labels. We do not recommend using the acronyms adopted by identity card, adding to them a hint with brief non-technical information.	[H6]			X
LEGEND: | A = Basic | B = Intermediate | C = Advanced |				

Fig. 4. Template used to describe solutions and recommendation to usability issues

After the second case study, we met the SCRUM team to collect their opinion about the new artifacts. The team commented that the protocol for reporting the items through heuristics allowed them to perceive the real impact of each identified issue, so they do not had doubts on the adjustments they should fix. With regards to the personas, they agree that they could see the application focusing on the target audience.

C. Second Cycle

When the protocol (artifact) was put into practice in a project with the participation of Product Owners (PO), we noted that they had difficulties in understanding the protocol. Observing the practice, the researchers found that for the PO, the use of User Story (US) was more natural. Considering this issue, the researchers have suggested an adaptation of US incorporating the vision of the user experience. The second research cycle was started in order to answer the following question:

- RQ3: How could personas and Nielsen's heuristics concepts be incorporated into the US?

1) Understanding Practices

In the first phase, the Academia members have carried out a technical literature survey, aiming to investigate the use of US in agile practices. US is one the most popular artifacts created for conveying agile requirements. US is used to capture customer requirements, emphasizing user goals, briefly explaining the user perception and focusing on what is needed to be done

[34]. They are often written on index cards or sticky notes, and typically follow a simple template, as showed in Figure 5.

As a **<type of user>**, I want **<some goal>** so that **<some reason>**

Fig. 5. User Story template

Acceptance criteria are commonly added to US addressing to guide acceptance testing. Acceptance testing is the process of verifying whether stories were developed such that each works exactly the way the customer expected it to work. It is a mechanism used to guarantees the functionality of the software requirements [35].

As a result of literature review, some recommendations were found regarding dealing with usability, UX and US together. Moreno and Yagüe [36] have identified three ways to incorporate functional usability requirements with US: (1) adding new stories to represent the requirements that are directly derived from usability (called usability stories); (2) adding or modifying tasks in US (detailing as needed); and (3) adding or modifying acceptance criteria. According these authors, there are usability recommendations that: have positive impact on the final quality of use of software systems; can be considered as functional usability requirements that complement the traditional requirements; and can be documented as US - the usability stories - because both are similar. Barksdale et al. [37] have used conceptual maps to design a complete picture to the user, detailing the connections between US and scenarios. Scenarios is a design artifact used to describe how a particular user uses the system for a specific task, considering the context and environment where it will be held [38]. The conceptual mapping was initially proposed to mitigate agile and UX team conflict to communicate usability and interaction requirements on an agile project. Sohaib and Khan [39] also recommended the use of scenarios along with US in exploration phase, and heuristic evaluation during acceptance testing.

In addition to the technical literature review, the researchers did an ethnographic study in order to understand how US were being developed by POs. It was found that these were developed by traditional methods as in Figure 5.

2) Deliberate Improvements and Build Artifact

From the outcomes of the literature review and the ethnographic study, we proposed to the teams a grammar - incorporating personas and Nielsen's heuristics - that should be adopted on writing the US. The grammar was driven to answer the third research question (RQ3), and we called the new story template of UserX Story. Figure 6 shows the initial proposal to the UserX Story (*Build Artifact – 1st version*). In first version of artifact, the traditional grammar was modified, by replacing (1) <type of user> by <personas>; and (2) <some reason> by <Nielsen's heuristic(s)> to highlight from usability point of view the positive impact of user interaction (users' goals).

UserX Story (template)
As a **<Persona>** I want / need **<goal>,** so that **<Nielsen's heuristic>** will be met

Fig. 6. Initial proposal to UserX Story template

The initial proposal was discussed between the researchers, POs, Scrum Master and UX designer. From the discussions between Academia and Industry, the artifact had been improved (*Build Artifact – 2nd version*). The second version of UserX Story expands the US from the outcomes of the data collected in the user research phase. User research focuses on understanding user behaviors, needs, and motivations through observation techniques, task analysis, and other feedback methodologies, *e.g.* the usability testing performed on the first cycle. The UserX Stories are told from the perspective of the *persona*, who needs a particular condition for interaction. Such a condition can meet multiple personas. The stories describe an interactive process wherein the persona has a *goal* to achieve, for this s/he *acts* on the interface (*interaction*), to perform *tasks* (steps / features to effect the action) in a particular *context* (usage pattern). The persona will assess whether the objective was achieved interpreting system *feedback*. Aiming to verify whether stories were developed such that it exactly met the user interaction needs, the acceptance criteria should describe the *action*, the *set of conditions*, and the Nielsen's *heuristics* (action/feedback) that will be satisfied once the goal is successfully achieved. Figure 7 shows second version to template of UserX Story:

UserX Story (template)
As a **<persona>** I want / need **<goal>,** for this **<interaction>,** through/ when **[<task> / <context>].** I evaluate that my goal was achieved when **<feedback>.**
Acceptance criteria: Checks **<action>** through **<set of conditions>**to satisfy **<Nielsen's heuristic(s)>** of action, and **< Nielsen's heuristic(s)>** of feedback.

Fig. 7. UserX Story template (refined version)

The workshop entitled "Interaction Stories" was organized in order to make a warm up for the writing of stories adding the vision of UX through the use of personas and Nielsen's heuristics. Six POs attended the workshop. All the participants had more than ten years of experience in the IT field. However, five of the six participants had little experience with agile methodology (less than a year).

The workshop was divided into (i) explanation of the concept of personas; heuristic inspection using Nielsen's heuristics; and US; (ii) presentation of the artifacts Lean Personas and UserX Story; and (iii) an exercise of writing stories from the proposed template, including the acceptance criteria. Some data collected with end-users during a workshop about "User Research" techniques previously performed with another group of participants, were provided as supplementary material to support the proposed exercise. Some personas were also included in this material. At the end of the workshop activities we discussed with the POs the next step for implementing the stories in real projects.

3) Implementing and Observing Improvements and Evaluating Artifacts

In this step, the POs had a period of one month to implement the UserX Story in to one of their projects (3a). After this period, researchers carried out individuals' interviews with the POs to collect their experiences with the implementation of UserX Stories (*Evaluate Artifact*). Figure 8 shows an implemented UserX Story for the redesign of part related for the issuance of reports for a Tax Bookkeeping sub-module.

UserX Story - Tax Bookkeeping sub-module
As a <*Leo Walker*> **I need to** <*issue financial reporting and balance sheets, filtering by agents*>, **for this** < *the system allow me to choose the agent that I want to filter* >, **through/ when** [<*for issuing the report*> / < *regardless of the organization to which I am placed in the system, it being subsidiary or consolidator*>]. **I evaluate that my goal was achieved when** <*the report only list the launches carried for the selected agent* >
Acceptance criteria: **Checks** < *the system will validate if that agent code can be used for the selected organization* > **through** < *filtering by agent code* **> to satisfy** <*H5*> **of action, and** < *H9*> **of feedback**. **Checks** < *the system should display agent name next to the chosen code* > **through** < *choosing an agent, either by agent code or searching* > **to satisfy** <*H1*> **of action, and** <*H6*> **of feedback**.

Fig. 8. UserX Story and acceptance criteria implemented by PO

Most of the POs approved the use of the interaction stories as in the proposed template. They have shared the UserX Story with the SCRUM teams, who reacted positively. However, two POs had not implemented the UserX Stories in their projects, since they were working on small changes that were related exclusively to business rules (legal requirements), and such changes would have had no impact on user interaction. Further tests will be needed to evaluate the cycle in which UserX Stories are written from items reported in the protocol for the communication of usability recommendations. Firstly, the items reported in the protocol should be discussed between UX designer and SCRUM team. Then, whether the SCRUM team agrees with the recommended item, the item will be written in UserX Story form with the participation of the UX designer. Otherwise, the recommendation is discarded, for example by technical limitations preventing it to be implemented. Currently, we are investigating the level of detail with the acceptance criteria; and also we have been working in the drafting of guidelines to better target the use of the interactive stories.

V. Discussion

The CMD applied by Al-Baik and Miller [17] somehow encapsulates (in the second and third step of the method) the process of design and development of an artifact (model) to identify and eliminate waste based upon the time and resources available to lean initiatives in IT contexts. In the SoftCoDeR approach, however, the design process of artifacts is highlighted, showing that the activities to build (2b) and evaluate (3b) artifacts are connected to the "Relevance Cycle" to meet the interests and needs of Industry (A), as well as being connected to the "Rigor Cycle" to be grounded on academic foundations (B), following from the guidelines of the DSR.

Rodríguez et al. [20] presented their research through the DSR, but the reported activities could configure perfectly action research cycles. The artifacts were built in close collaboration with practitioners, and the research crossed the phases to diagnose the problem, propose solutions, and to implement and assess improvements. In this case, researchers have chosen to complement the research with focus group sessions rather than adopt action research as a more comprehensive methodology. The SoftCoDeR approach seems to be feasible for research work with such characteristics, and especially when it occurs in environments that adopt agile methods. Furthermore, the iterative nature of the proposal approach is compatible with agile environments because it intends to deliver valuable artifacts, adapting quickly to changes, evolving from new research questions. Differently to the cases discussed above, we can state that the works presented by Adikari et al. [19], and Ardito et al. [18] do not fit with an approach combining the two research methods.

Regarding the approaches that combine AR and DSR [21] [22] [23], we point out that all these initiatives were originated in the field of IS, which focus is to seek solutions to business needs from a systemic thinking. The research in SE, in turn, is closely related to software development activities, which can be better understood from the practitioners' viewpoints. Therefore, the choice of the CMD method contributes to an approach that is intended to guide research within the software development environment (shop floor), and heavily influenced by cooperation between researchers and practitioners.

Nevertheless, some limitations of our approach is related to application context. Besides this, the dependence of the company's schedule brings difficulty to keep the pace of iterations. First of all, researchers need the company's prior consent to interact with the practitioners during a sufficient period of iterations to understand the existing practice, discuss improvements, and evaluate what has been implemented. However, a greater number of iterations, sometimes it can be difficult to keep a schedule without unexpected events involving the practitioners. Sometimes the practitioners need to solve urgent matter, breaking the iterations evolution.

VI. Conclusions and Futher Work

By means of SoftCoDeR, an approach matching CMD and DSR methods, we have been able to achieve our main research goals. The main contribution of this approach is to encourage the Academia-Industry partnership, which the core proposal has been to guide the academic knowledge to meet the real needs of the Software Industry. The authors believe that the approach allows the deeply observation of problems that still have not been thoroughly investigated academically. In this work, for instance, we detected an issue in the communication carried out between UX and agile teams, who have different perspectives on usability and the user research.

As a future work, we are planning to apply this approach to new research projects in order to verify whether this proposal can be extended to other contexts, including other market segments; or in a different development process than agile methodology.

ACKNOWLEDGMENT

We would like to thank the company and the practitioners who were partners of this research project.

REFERENCES

[1] F. Shull, J. Carver and G. H. Travassos, "An empirical methodology for introducing software processes," *ACM SIGSOFT Software Engineering Notes,* vol. 26, n. 5, pp. 288-296, 2001.

[2] G. Santos, A. Rocha, T. Conte, M. Barcellos and R. Prikladnicki, "Alignment between Academy and Industry: A Virtuous Cycle to Promote Innovation," *In 26th Brazilian Symposium on Software Engineering (SBES) (in Portuguese)*, Natal, Brazil, 2012.

[3] D. Rombach and R. Achatz, "Research Collaborations between Academia and Industry," *In Proceeding of FOSE'07 - Future of Software Engineering*, Minneapolis, Minnesota, 2007.

[4] K. Schwaber and J. Sutherland, The scrum guide, Scrum Alliance, 2011.

[5] J. M. Brown, G. Lindgaard and R. Biddle, "Collaborative Events and Shared Artefacts: Agile Interaction Designers and Developers Working Toward Common Aims," *In Agile Conference AGILE 2011*, Salt Lake City, Utah, 2011.

[6] D. Avison, F. Lau, M. Myers and P. Nielsen, "Action Research," *Communications of the ACM,* vol. 42, n. 1, pp. 94-97, 1999.

[7] W. Carr, "Philosophy, methodology and action research," vol. 40, n. 4, pp. 421-435, 2006.

[8] P. S. M. Santos and G. H. Travassos, "Action Research Can Swing the Balance in Experimental Software Engineering," *Advances in Computers,* vol. 83, pp. 205-276, 2011.

[9] Y. Dittrich, "Doing 'Empirical Research in Software Engineering – finding a path between understanding, intervention and method development'," *In Social thinking – Software practice*, R. K. Floyd, Ed., MIT Press, 2002.

[10] R. Baskerville, "What design science is not," *European Journal of Information Systems,* vol. 17 , n. 5, pp. 441-443, 2008.

[11] J. Venable, "A Framework for Design Science Research Activities," *In Proceedings of the Information Resource Management Association Conference*, Washington, DC, USA, 2006.

[12] S. T. March and G. F. Smith, "Design and natural science research on information technology," *Decision support systems,* vol. 15, n. 4, pp. 251-266, 1995.

[13] J. Iivari and J. Venable, "Action research and design science research– seemingly similar but decisively dissimilar," *In ECIS 2009 - 17th European Conference on Information Systems*, Verona, Italy, 2009.

[14] A. R. Hevner, S. T. March, J. Park and S. Ram, "Design Science in Information Systems Research," *MIS Quarterly,* vol. 28, n. 1, pp. 75-105, 2004.

[15] A. Hevner, "A Three Cycle View of Design Science Research," *Scandinavian Journal of Information Systems,* vol. 19, n. 2, pp. 97-92, 2007.

[16] Y. Dittrich, K. Rönkkö, J. Eriksson, C. Hansson and O. Lindeberg, "Cooperative method development," *Empirical Software Engineering,* vol. 13, n. 3, pp. 231-260, 2008.

[17] O. Al-Baik and J. Miller, "Waste identification and elimination in information technology organizations," *Empirical Software Engineering,* vol. 19, n. 6, pp. 2019-2061, 2014.

[18] C. Ardito, P. Buono, D. Caivano, M. F. Costabile and R. Lanzilotti, "Investigating and promoting UX practice in industry: An experimental study," *International Journal of Human-Computer Studies,* vol. 72, n. 6, pp. 542-551, 2014.

[19] S. Adikari, C. McDonald and J. Campbell, "Little design up-front: A design science approach to integrating usability into agile requirements engineering," *In Human-computer interaction. New trends*, 2009.

[20] P. Rodríguez, P. Kuvaja and M. Oivo, "Lessons learned on applying design science for bridging the collaboration gap between industry and academia in empirical software engineering," *In Proceedings of the 2nd International Workshop on Conducting Empirical Studies in Industry*, Hyderabad, India, 2014.

[21] R. Baskerville, J. Pries-Heje and J. Venable, "Soft design science methodology," *In DESRIST'09*, Malvern, Pa, USA, 2009.

[22] M. Sein, O. Henfridsson, S. Purao, M. Rossi and R. Lindgren, "Action design research," *MIS Quarterly,* vol. 35, n. 1, pp. 35-56, 2011.

[23] R. Wieringa and A. Morali, "Technical action research as a validation method in information systems design science," *In Design Science Research in Information Systems. Advances in Theory and Practice*, Las Vegas, Nevada, USA, 2012.

[24] R. Cole, S. Purao, M. Rossi and M. Sein, "Being proactive: where action research meets design research," *In ICIS 2005 Proceedings*, Las Vegas, NV, USA, 2005.

[25] E. Law, V. Roto, A. P. Vermeeren, J. Kort and M. Hassenzahl, "Towards a shared definition of user experience," *In CHI'08*, 2008.

[26] J. Nielsen and D. Norman, "The Definition of User Experience," 2013. [Online]. Available: http://www.nngroup.com/articles/definition-user-experience/.

[27] M. Isomursu, A. Sirotkin, P. Voltti and M. Halonen, "User Experience Design Goes Agile in Lean Transformation - A Case Study," *In Agile Conference (AGILE)*, Dallas, Texas, 2012.

[28] D. Salah, R. F. Paige and P. Cairns, "A systematic literature review for agile development processes and user centred design integration," *In Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering (EASE '14)*, New York, 2014.

[29] A. Cooper, R. Reimann and D. Cronin, About Face 3: The Essentials of Interaction Design, Wiley Publishing, 2007.

[30] J. Nielsen, "How to Conduct a Heuristic Evaluation," 1995b. [Online]. Available: http://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation.

[31] A. Følstad, E. L. C. Law and K. Hornbæk, "Outliers in usability testing: How to treat usability problems found for only one test participant?," *In Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through*, 2012.

[32] L. Caballero, A. M. Moreno and A. Seffah, "Persona as a Tool to Involving Human in Agile Methods: Contributions from HCI and Marketing," *Human-Centered Software Engineering,* pp. 283-290, 2014.

[33] G. Miller and L. Williams, " Personas: Moving Beyond Role-Based Requirements Engineering," In Microsoft and North Carolina State University, 2006.

[34] I. Inayat, S. S. Salim, S. Marczak, M. Daneva and S. Shamshirband, "A systematic literature review on agile requirements engineering practices and challenges," *Computers in Human Behavior,* 2014.

[35] M. Cohn, User Stories Applied for Agile Software Development, 13 ed., Boston, MA: Addison-Wesley, 2009.

[36] A. M. Moreno and A. Yagüe, "Agile User Stories Enriched with Usability," *In Agile Processes in Software Engineering and Extreme Programming*, Malmö, Sweden, 2012.

[37] J. Barksdale, E. Ragan and D. McCrickard, "Easing TeamUsability: A Concept Mapping Approach," *In Agile Conference*, Chicago, IL, 2009.

[38] Y. Jia, M. K. Larusdottir and Å. Cajander, "The Usage of Usability Techniques in Scrum Projects," *In International Conference on Human-Centered Software Engineering*, Toulouse, France, 2012.

[39] O. Sohaib and K. Khan, "Incorporating Discount Usability in Extreme Programming," *International Journal of Software Engineering and Its Applications,* vol. 5, pp. 51 - 62, 2011.