Choma *et al. Journal of Software Engineering Research and Development* (2016) 4:8
DOI 10.1186/s40411-016-0034-5

Journal of Software Engineering Research and Development

RESEARCH Open Access

SoftCoDeR approach: promoting Software Engineering Academia-Industry partnership using CMD, DSR and ESE

Joelma Choma[1*], Luciana A. M. Zaina[1] and Tiago S. da Silva[2]

*Correspondence:
jh.choma@hotmail.com
[1]Department of Computer Science, Universidade Federal de Sorocaba (UFSCar) – Campus Sorocaba, Rdv. João L Santos, Km 110, Sorocaba, SP, Brazil
Full list of author information is available at the end of the article

Abstract

Background: The Academia-Industry partnership has been increasingly encouraged in the software development field. The main focus of the initiatives is driven by the collaborative work where the scientific research work meets the real needs of the Industry.

Methods: Aiming to contribute to this effort we have proposed 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).

Results: We have applied the SoftCoDeR approach in a software development company to support a research and development project aiming to integrate User Experience practices into the agile software development process.

Conclusions: In this paper, we present new findings about this approach that has been extended with the use of Experimental Software Engineering (ESE) practices to conduct experimental validations of artifacts before putting them in industrial practice.

Keywords: Action research, CMD, DSR, Software engineering, Experimental software engineering, Scrum, HCI, User experience, Interaction design

Introduction

Although the need for sharing experiences and knowledge between Academia and software Industry has become increasingly evident, both have demonstrated difficulties on establishing an effective relationship, mainly because they usually have different viewpoints. We use the term Academia to refer to the environment or community concerned with the scientific research (Pearsall and Hanks 1998). Mechanisms have been proposed to balancing the academic and industrial needs, and also to promote a strategic alignment between the interests of Academia and software Industry (Santos et al. 2012). From this perspective, researchers need to develop skills - with active role – to understand and deal with practical problems observed from the experience of teams of software developers 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 into an organization (Rombach

and Achatz 2007). 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.

Recently, we have employed a research approach called SoftCoDeR (Software Cooperative Design Research) in research projects related to improvement of software development processes involving construction and validation of support artifacts (Choma et al. 2015a). Initially, this approach was applied in a research and development project in a software company, which was intended to incorporate User eXperience practices (UX) into its development process. In order to link scientific knowledge to the real needs of the Industry, the SoftCoDeR approach allowed researchers to work closely in cooperation with practitioners of the software development area.

In Choma et al. (2015a), we reported two SoftCoDeR cycles that were applied to build and validate some artifacts aiming to improve the work between UX and Scrum teams, improving the communication between them. These first two cycles of the SoftCoDeR approach were performed within the industrial environment with the intense participation of both teams. Due to a new need that arose in the research project partner industry, a third cycle was applied. However, from the third cycle our research approach has been extended with the use of Experimental Software Engineering (ESE) practices to conduct experimental validations of artifacts before putting them in industrial practice. In this paper, we present details about this research cycle; also, we discuss about lessons learned, implications and limitations of the approach extended with ESE.

The remainder of this paper is organized as follows: "Background" section introduces the basis of our proposal – an overview of two research methods and related work; "The SoftCoDeR approach" section outlines the proposal approach; "Results" section presents the approach applied in practice; "Discussion" section discusses the outcomes of the practices; and finally "Conclusions" section presents the conclusions and points out directions for further work.

Background

This section presents an overview of methods and concepts on which the SoftCoDeR approach is based ("The foundations: CMD, DSR and ESE" section), and some related studies about these methods applied in the software area ("Related work" section).

The foundations: CMD, DSR and ESE

Cooperative Method Development (CMD) is a domain-specific adaptation of Action Research (AR) focused on software development practices, proposed by (Dittrich 2002). Choosing AR as the basic approach, CMD can be seen as an iterative process involving researchers and practitioners acting together on a particular cycle of activities, including problem diagnosis, action intervention, and reflective learning (Avison et al. 1999). 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* (qualitative empirical investigations), (2) *Deliberate Improvements* (aiming to solve the problems identified in the first phase) and (3) *Implement and Observe Improvements* (checking the effectiveness of the improvements). Afterwards, the results are evaluated together with the practitioners

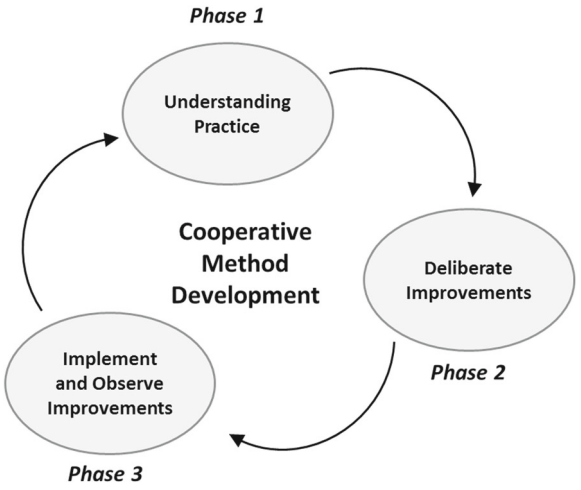

Fig. 1 CMD cycle

involved. The three phases can be repeatedly applied in the same context. Figure 1 shows the illustration of a CMD cycle.

Design Science Research (DSR) is a conceptual framework proposed by Hevner et al. (2004) to drive research in the field of Information Systems (IS). DSR aims 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. The artifacts typically produced by DSR are related to concepts, models, methods, and instantiations (March and Smith 1995). According Iivari and Venable (2009), DSR is an approach most commonly adopted in the area of IS, but it has been implicitly practiced in SE for decades.

The DSR framework consists of three research cycles: (1) *relevance cycle*, (2) *rigor cycle* and (3) *design cycle* (Hevner 2007). The *relevance cycle* connects the research environment with the DSR activities, focusing to capture real problems and opportunities to provide artifacts (design solutions) appropriate for the application domain. The *rigor cycle* connects the DSR activities with the *knowledge base* – composed by foundations and methodologies – to provide applicable knowledge. The *knowledge base* provides the raw materials from and through which research is accomplished. The *design cycle* consists of iterations to building and evaluating of artifacts and processes from the research. Figure 2 shows cyclical interaction between the areas environment, design science research and knowledge base.

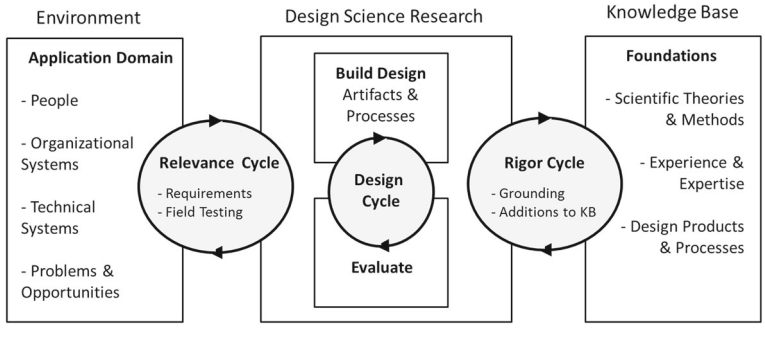

Fig. 2 Design science research cycles, adapted from Hevner (2007)

Experimental Software Engineering (ESE) is a sub-field of software engineering which aims at applying experimentation in the construction of new methods and technique for the measuring, understanding, and improvement of the software engineering phenomena in organizations (Shull et al. 2001). The ESE focuses on evaluating the tools and techniques used, developed, and intended for use in the Industry through empirical validation. According Sjøberg et al. (2005), research in ESE should aim to acquire general knowledge about which technology (process, method, technique, language, or tool) is useful for whom to conduct which (software engineering) tasks in which environments.

In the Evidence Based Software Engineering, the adoption of empirical strategies to validate studies depends on the degree of control and intended goals for research. Usually, the main adopted types of strategies are: surveys, case studies and controlled experiments (Easterbrook et al. 2008). The survey's goal may be descriptive, explanatory and/or exploratory; but it offers no control over execution or manipulation of variables. Case studies are mostly employed to tracking projects, activities or tasks through continuous observation of the correlated attributes. Regarding to the controlled experimental studies, there is a systematic control of the variables and the performed process.

The aim of the experimentation in the Software Engineering (SE) may be the confirmation of theories; confirmation of conventional knowledge; exploration of relationships; evaluation of models; and/or validation of measures and artifacts. Also, it is used to explore a new domain to see how and when they really work, to understand their limits, and to understand how to improve them (Basili 1996). Controlled experiments are encouraged when the researcher wants to have control over the situation, direct manipulation, precise and systematic phenomenon of behavior to be studied, as well as on the activities and distribution of people in the experimental groups (Shull et al. 2001).

A common attribute in all experiments is the control treatment, where one or more independent variables are manipulated to observe their effects on one or more dependent variables. Figure 3 shows the illustration of an experiment according to Wohlin et al. (2012). The independent variables are manipulated and controlled variables in an experiment, while the dependent variables are those who want to study to see the effect of changes in the independent variables. The effect observed in experimental results is measured and analyzed to validate or refute the previous hypotheses (Shadish et al. 2002).

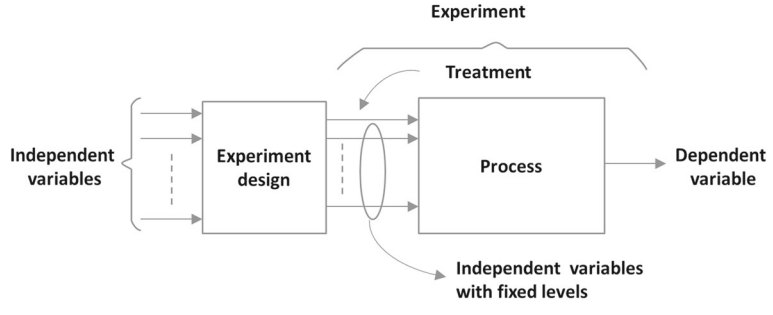

Fig. 3 Illustration of an experiment, adapted from Wohlin et al. (2012)

Related work

According to Dittrich et al. (2008), 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 (2014) have implemented CMD in an investigative 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. (2014) 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.

Regarding DSR, Adikari et al. 2009 adopted the Hevner et al.'s framework (Hevner et al. 2004) 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.

Rodríguez et al. (2014) 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.

ESE is a multi-disciplinary field that involving the cooperation between different groups of the Academia and Industry, such as software engineers, industrial actors and statisticians (Jaccheri and Osterlie 2005). According to a survey conducted by Sjøberg et al. (2005), experiments are most common in Academia, although the experimental subjects of these studies are usually students with little or no professional experience. On the other hand, it is difficult to find professionals for empirical studies, whereas students are more accessible, easier to organize, and represent lower costs (Rivero and Conte2012).

Dieste et al. (2013) identified that researchers have been facing to several obstacles to experimenting in Industry. The most important concerns are the time and cost demands that running an experiment places on the host company and the participating professionals. According them, experiments run in companies tend to be aligned with business goals/processes, but do not always match with researchers' interests. Usually, industrial environments have imposed many more constraints on the experimental setting than laboratory environments (Vegas et al. 2015). Additionally, due to very specific context in which experiments in industry are conducted, the external validity is very hard to be achieved, and oftentimes the results cannot be generalized.

The SoftCoDeR approach

In this section, we briefly discuss the reasons that motivated us to combine CMD and DSR in our research approach ("Methods" section). Afterwards, we provide an explanation of the SoftCoDeR approach extended with ESE ('SoftCoDeR: extended with ESE" section).

Methods

Iivari and Venable (2009) identified levels of overlapping for activities performed in the research methods matching AR and DSR. Some matched approaches have been proposed to drive research in the field of IS (Baskerville et al. 2009, Sein et al. 2011, Wieringa and Moralí 2012).

Baskerville et al. (2009) proposed Soft DSR approach that incorporates aspects of Soft Systems Methodology (that is a form of AR) into a DSR process. According to Baskerville (2008), 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.

Sein et al. (2011) 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 Moralı (2012) 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.

Taking into account that CMD is anchored in the AR, we have observed that it is also compatible with the DSR, once both are (1) interventionist methods (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 Cole et al. (2005). Furthermore, these methods are based on the argument that the research should ever contribute for both to Academia as well as to Industry.

We proposed the SoftCoDeR approach considering the aforementioned perspectives to match the two methods – CMD and DSR. We have chosen to follow a similar approach to the proposals of Baskerville et al. (2009) and Sein et al. (2011), in which the research activities are significantly overlapping. We adopt CMD because it is more directed towards the SE's field, focusing on shop floor software development practices. In addition, CMD provides a basic structured research process, which clearly defines three phases of research for understanding the practice, and for the deliberation, implementation, and evaluation of improvements. Also, CMD encourages short iterative cycles to put the improvements in practice. In particular, this is a highly appropriate characteristic for research in agile development software environment. Regarding to the DSR, its concepts were added to our approach in order to provide guidelines to produce artifacts of value that is an extremely relevant issue to the agile teams. Moreover, the DSR draws the attention to relevance and research's rigor that are aspects extremely important for the researchers.

SoftCoDeR: extended with ESE

In the previously published paper (Choma et al. 2015a), we mentioned that the ESE could be used as a mechanism for validation of proposed artifacts during the evaluation phase. An advantage of this practice is to evaluate the artifacts before introducing them directly in industrial practice, thereby minimizing the impact of an evaluation involving real projects. Furthermore, in situations where the

industrial environments impose constraints for an experimental validation such as recruitment of participants, the environment setting, and the reservation of needed equipment, it may be more appropriate to conduct experimentation in an academic environment.

Next, we provide an explanation of the SoftCoDeR approach as displayed in Fig. 4. It is important to emphasize that, the main difference of this SoftCoDeR version regarding to the version previously published in Choma et al. (2015a) is its extension with the use of ESE practices.

As shown in the Fig. 4, 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 insights to guide 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

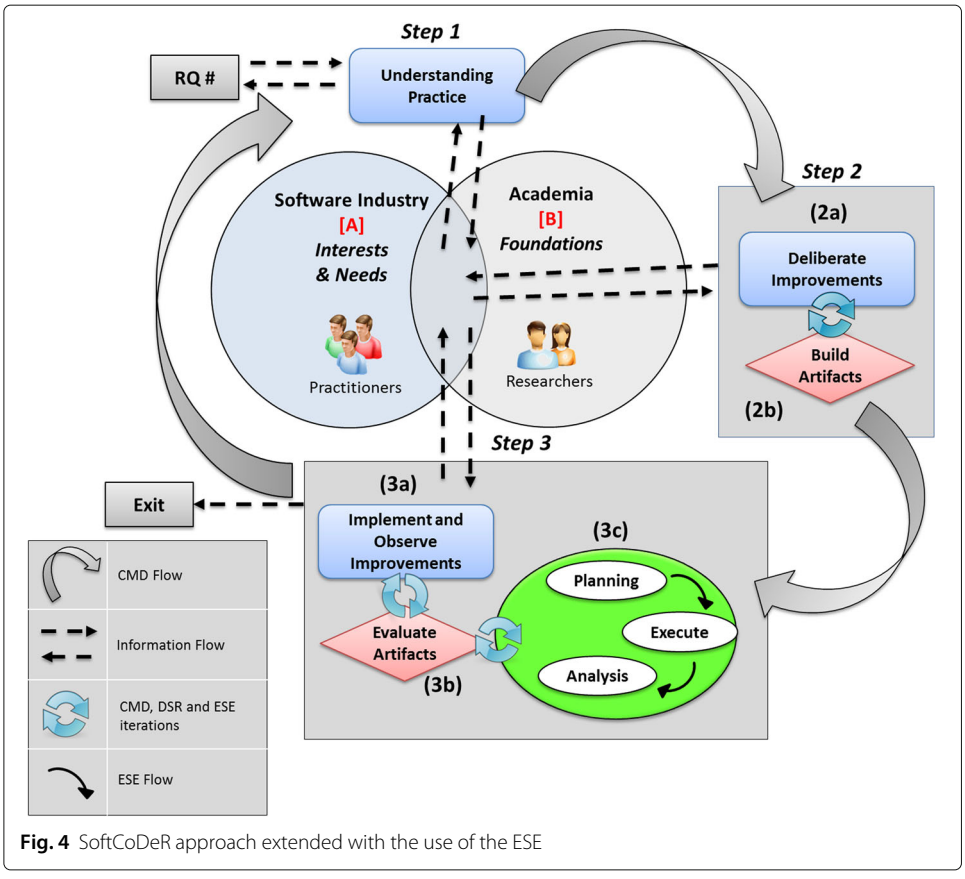

Fig. 4 SoftCoDeR approach extended with the use of the ESE

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. For this purpose, we refer to feasibility and observational studies to evaluate artifacts, for which experiments are usually the chosen empirical strategy. Therefore, at this stage of the research approach, we recommend the use of ESE practices to conduct experimental validations of artifacts before putting them in industrial practice. The ESE provides a set of guidelines for planning and execution of experiments, as well as for processing and analysis of collected data. Based on Wohlin et al. (2012), the three steps to carry out experimental studies are: planning, execution and analysis (3c).

Taking into account the investigation goals, the planning step involves: (i) context selection (e.g., in an academic or industrial environment; with students or professionals participants; involving a simulation or real problems); (ii) hypothesis formulation; (iii) variables selection (independents and dependents); (iv) selection of subjects (e.g., chosen based on convenience); (v) experiment design in order to describe how the tests will be organized and run; (vi) instrumentation (e.g., guidelines and others support materials); and (vii) validity evaluation. The experiment execution step involves: (i) preparation of the materials such as study objects, guidelines, and data collection instruments; (ii) execution of the experiment; and (iii) data validation in order to verify if the data were correctly collected. Finally, in the analysis step, the collected data should be analyzed and interpreted. Firstly, descriptive statistics should be used to visualize the data collected, and then, statistical tests should be run to evaluate the study hypotheses.

Thus, the findings of 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.

Results

This section presents, firstly, the general settings of the environment in which the SoftCoDeR has been applied. Then, we briefly describe the first two SoftCoDeR cycles wherein the artifacts were validated directly in the industrial environment. And after that, we describe in details a SoftCoDeR cycle using ESE to validate artifacts in academic environment before putting them into industrial practice.

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 2000 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 applied 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 – UX is a concept that encompasses a pleasant experience for users by means of simplicity, elegance, efficiency, effectiveness and emotional satisfaction of the people (Law et al. 2008).

Studies indicate that interaction design in ERP systems is too weak, which considerably affect the usability of the product (Singh and Wesson 2009). On the other hand, the integration of UX methods into software development processes based on agile methodologies is not an easy task, not even in ERP system domain. 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 (Isomursu 2012).

However, there are many issues that can complicate such integration (Salah et al. 2014), e.g., promoting synergy and establishing a good communication between the agile and the UX teams (Brown et al. 2011). 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.

SoftCoDeR pure: the first two cycles

Next, we describe the first two SoftCoDeR cycles applied in the aforementioned research project. Noteworthy that these cycles will be briefly described since were already published in details in Choma et al. (2015a).

First cycle

The first SoftCoDeR cycle arose from an issue that the industrial 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 (Cooper et al. 2014) and Nielsen's heuristics (Nielsen 2011) – as a possible solution. Considering the problem-situation, and the use of the mentioned concepts, we ha e defined two research questions to guide this 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?

Understanding practice

Aiming to answer the first research question (RQ1), firstly, we carried out a workshop entitled "Usability Heuristics for ERP system" where we introduced the Nielsen's heuristics and personas concepts for 59 practitioners (developers, testers, analysts, technical leaders and software architects). Secondly, about one month after, we interviewed 10 practitioners who had attended of this workshop, in order to investigate (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.

From the interviews, 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.

Deliberate improvements and build artifact

From the discussions between Academia and Industry (researchers, project leaders and the UX designer) about the outcomes achieved in the previous step, we (the researchers) have proposed building of two artifacts taking into account the second research question (RQ2): (1) a template for describe personas called Lean Personas, and (2) a protocol for communication of design solutions and usability recommendations.

Implementing and observing improvements and evaluating artifacts

We decided along with practitioners evaluate the two artifacts through two case studies. Both cases involved usability tests in an actual redesign project - a high fidelity prototype to the sub-module of Human Resources (Web-based) used for the registration of employees. In the first case study, the artifacts were not implemented, and we did not interfere in the teams' work; acting in observer's role. In the second case study, we participated along with the UX and Scrum teams during planning and execution of the usability test, in order to implement the use of the personas and the protocol to report the usability issues. For this intervention, we used the new version of the functional prototype that had been fixed by the Scrum team.

After the second case study, we met the Scrum team to gather 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 usability problem, so they do not had doubts on the adjustments they should fix. In addition, they demonstrated to recognize the importance and benefits of the personas to keep the focus in the typical users during the application development.

Second cycle

The second SoftCoDeR cycle arose when we observed that the protocol proposed in the first cycle did not meet the needs of the Product Owners (PO), who were more familiar to User Story (US) – a popular artifact used for conveying agile requirements. Hence, we suggested an adaptation of US incorporating the vision of the user experience, and this 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?

Understanding practices

We carried out a technical literature survey aiming to investigate the use of US in agile practices, including acceptance criteria. As a result, we also found some recommendations to deal with usability aspects and US together. In addition to the technical literature review, we carried out an ethnographic study in order to understand how US were being developed by PO. Then, we found that they were using one of the more traditional templates to develop the US, such as presented by Cohn (2004).

Deliberate improvements and build artifact

From the outcomes of the literature review and the ethnographic study, and taking into account the third research question (RQ3), we proposed to teams a new template to US called UserX Story. In the first version of artifact, a grammar - incorporating personas and Nielsen's heuristics - should be adopted to write the US, where the traditional grammar was modified, by replacing, 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).

From the discussions between Academia and Industry, the artifact was improved. Thus, the second version of UserX Story was outlined to expand the US from the user research findings. User research focuses on understanding user behaviors, needs, and motivations through observation techniques, task analysis, and

other feedback methodologies, such as usability testing that were performed in 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.

Before implementing the proposed artifact in real projects, we carried out a workshop entitled "Interaction Stories" in order to make a warm up with six POs at the writing of the UserX Stories. 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. At the end of the workshop activities we discussed with the POs the next steps for implementing the stories in real projects.

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. After this period, researchers carried out individuals' interviews with the POs to collect their experiences with the implementation of UserX Stories. Most of the POs approved the use of the interaction stories following 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. After this cycle, we met the Scrum team to collect their opinion about the new artifacts and other needs.

SoftCoDeR + ESE

The third SoftCoDeR cycle was applied due to the need for a more appropriate set of usability heuristics for ERP systems, and also an effective usability evaluation method for novice-inspectors. According to Singh and Wesson (2009), inspection methods for assessing the usability of ERP systems require more appropriate criteria to cover its most critical points, such as the usability issues related to navigation and learnability. Furthermore, during a meeting with Scrum team, we noticed that a more detailed guideline was essential to improve the use of the Nielsen's heuristics, because some practitioners – developers and testers – reported us difficulties and doubts on applying the heuristics, since none of them had experience on heuristic inspection. Taking into account these needs, we started a cycle from the following questions:

- RQ4: Can the proposed perspective-based ERP heuristics improve the performance of novice usability inspectors?
- RQ5: Can the proposed perspective-based ERP heuristics effectively identify the major usability issues in ERP systems?

Understanding practice

During the first meetings with Scrum teams, we seek to understand the difficulties of the practitioners in the use of Nielsen's heuristics, aiming to elaborate guidelines suitable to reality of the company and teams. Also, we observed that an intervention with ERP system's users would be necessary in order to understand the practice from their point of view. Therefore, together with practitioners, we decided to apply a survey with 23 issues to identify the degree of importance assigned by ERP systems' users for usability aspects. We prepared these issues based on (i) usability problems found in an usability inspection that we performed in different ERP systems available in the market; and (ii) the most common usability problems mentioned in the literature related to the domain of ERP systems. The 23 issues were related to the Nielsen's heuristics, and grouped into four categories: presentation; navigation; task support and feedback of messages. These categories were chosen based on Singh and Wesson (2009) and Conte et al. (2009).

The survey was answered by 37 users of ERP systems who attributed the degree of importance for each question using a 5-point Scale – very important, significantly important, important, somewhat important or not important. We observed that of the six questions related to the heuristic of error prevention (H5), three were assigned as very important by at least 62% of participants. Overall, the participants attributed greater importance to the usability aspects that are linked to error prevention (H5); and recover from errors (H9). Further details on survey can be seen in Choma et al. (2015b).

Deliberate improvements and build artifact

Discussing with practitioners about the two steps outcomes – the ERP systems inspection and the survey – and with purpose to avoid overlaps or redundancies on the definition of perspectives, we reviewed the usability categories concluding that two perspectives could be enough to represent the main usability problems: Presentation and Task Support.

Then, we proposed an artifact to guide the usability inspection in ERP systems. The proposed artifact consisting of (1) a set of perspective-based ERP heuristics mapping the Nielsen's heuristics into the ERP perspectives – Presentation (P) and Task Support (S), which led to eight heuristics in the perspective of presentation and five heuristics in perspective of task support; (2) three key-questions to guide the inspector through the perspective of Presentation – (i) *"Am I seeing?"*; (ii) *"Am I understanding?"*, and (iii) *"Is the message clear for me?"*, and through the perspective of task support – (iv) *"Can I complete the task without obstacles?"*; and (3) a set of tips to guiding the inspector during the inspection activity based on the users' survey issues.

Validating artifact by ESE

In this phase, we suggested to practitioners a validation of the proposed improvements and artifact into the academic environment, aiming minimize any impact of introducing the new artifact in practice. Considering the scope validation and with company's consent, a controlled experiment was carried out following the ESE steps (Fig. 4 - item 3c) in order to compare the efficiency and effectiveness of usability inspections between the perspective-based ERP heuristics (HPERP) and the traditional Nielsen's heuristics (HN).

Context, subjects and inspection procedures

The subjects of experiment were 11 undergraduates and 8 graduate students enrolled in Human-Computer Interaction (HCI) course, selected by convenience. Concerning the profile of the participants, only one participant (5%) had greater familiarity and experience with ERP systems, 11% of participants had some familiarity and the others either had no familiarity (42%) or had a very superficial knowledge of the subject (42%). As to the experience with heuristic evaluation, the majority of participants (84%) did not know the technique, 11% had some knowledge on the subject and only one participant (5%) had experienced the technique. The fact that most participants were novices in inspection technique has allowed us to verify the effectiveness of learning of technique.

For inspection activity, to the validation of the proposal, two sub-modules of an enterprise management Web-based system were provided by industrial partner. The first one was a medium-fidelity prototype of Holiday Planning of the Human Resources module; and the second one was a functional sub-module of Retail Sales of the Sales module.

The previous week the execution of the experiment, all participants received the same training during 4 hours which was split into: (i) the explanation of the concepts of heuristic inspection; and (ii) the warming up in which all participants, using the two sets of heuristics (HN and HPERP). Then, the experiment was undertaken in a controlled environment and a predetermined time-limit – not exceeding two hours – avoiding the tiredness of the participants and the misunderstanding of violations, factors that could affect negatively the inspection's outcomes (Nielsen, 1995b). Regarding inspection procedures, participants were divided into four inspection groups with 4 or 5 members: the first and the second group (G1 and G2, respectively) would work with a medium-fidelity prototype of Holiday Planning sub-module; while the third and the fourth group (G3 and G4, respectively) would inspect the functional sub-module of Retail Sales. Aiming to compare the two methods, we assigned to G1 and G3 the HN and to G2 and G4 HPERP proposal.

The groups were accommodated in two laboratories and the inspections were conducted individually. The participants received as support material: (1) inspection instructions task-based; (2) a spreadsheet to record usability problems found and the corresponding violated heuristics; and (3) the web links to medium-fidelity prototype and functional sub-module. In addition to the support material, the groups that used the HPERP (G2 and G4) received a summary table of the heuristics with the tips to guide the inspections through the perspectives of presentation and task support. The other groups (G1 and G3) followed the Nielsen's heuristics guidelines. Both HN and HPERP groups should accomplish the same tasks (task-based) in the system that they would inspect, avoiding that they had different degrees of difficulties concerning on the tasks of interaction. Further, upon concluding of the inspection activity, the participants were invited to answer a questionnaire based on model called TAM (Davis, 1989), aiming to identify factors involved in satisfaction of individuals regarding the acceptance and use of the inspection technique.

Results: analysis and interpretation

The results of the experiment were analyzed using statistical methods (descriptive and inferential statistics) in order to answer the two research questions (RQ4 and RQ5). It is noteworthy that we collected the severity level of usability problems identified by participants, but we did not consider this information in our analysis; taking into account

only the number of violations. The inspection results per participant and per inspection groups can be seen in Table 1.

Regarding the RQ4, in order to identify the novice-inspectors groups which had the best performance during the usability inspection activity, we found the efficiency indicators by calculating the ratio between the average of confirmed violations – real problems, no false positives – and the average time spent on inspection activity. As can be seen from Table 2, the best indicator of efficiency (approx. 23 violations per hour) was obtained by G3 who inspected the functional sub-module using HN; and the smallest indicator (approx. 7 violations per hour) was obtained by G1 that also had used the HN, but the inspection was on medium-fidelity prototype.

Regarding the RQ5, in order to identify the novice-inspectors groups which found the highest number of usability problems during the usability inspection activity, we found the effectiveness indicators by calculating the ratio of the average of confirmed violations and the number of known-problems. Known-problems refer to problems that were previously identified by two usability experts who inspected the same objects. One of the experts who used the HN identified 41 violations in the medium-fidelity prototype, and 43 violations in the functional sub-module. The second expert who used the HPERP pointed out 43 violations in the medium-fidelity prototype, and 50 violations in the functional sub-module. From Table 2, we can observe that the group G3 had the highest efficacy indicator, which detected 61.63% of the known-problems; while the group G1 had the lowest indicator of effectiveness, which detected only 24.19% of the known-problems.

Table 1 Results per participant and per inspection group

Group-Heuristic [a]	Part.	Time [b]	#V[c]	#FP[d]	#VC[e]	V/H [f]	V/KP[g]
G1-I	1	107	27	10	17	9.53	0.41
	2	112	19	5	14	7.50	0.34
	3	97	9	2	7	4.33	0.17
	4	50	11	4	7	8.40	0.17
	5	83	12	5	7	5.06	0.17
G2-II	6	70	20	5	15	12.86	0.35
	7	77	47	23	24	18.70	0.56
	8	107	36	9	27	15.14	0.63
	9	115	29	7	22	11.48	0.51
	10	111	21	7	14	7.57	0.33
G3-I	11	61	28	0	28	27.54	0.65
	12	71	41	6	35	29.58	0.81
	13	90	16	1	15	10.00	0.35
	14	60	41	13	28	28.00	0.65
G4-II	15	71	21	0	21	17.75	0.42
	16	66	37	3	34	30.91	0.68
	17	60	14	5	9	9.00	0.18
	18	88	22	2	20	13.64	0.40
	19	80	36	5	31	23.25	0.62

[a] HN = I; HPERP = II
[b] Number of minutes to complete the inspection
[c] Number of violations
[d] Number of false-positive
[e] Number of confirmed violations
[f] Violations/Hour
[g] Violations/Known Problems

Table 2 Efficiency and effectiveness indicators

Group	#VC[a]	Time (hours)[b]	#VC/Time[c]	#VC/Known Problems[d]
G1	10.40	1.50	6.95	24.19%
G2	20.40	1.60	12.75	47.44%
G3	26.50	1.18	22.55	61.63%
G4	23.00	1.22	18.90	46.00%

[a] Average number of confirmed violations
[b] Average time spent to complete the inspection
[c] Efficiency indicator
[d] Effectiveness indicator

In order to confirm significant differences between the efficiency and effectiveness of each group, we analyzed the results of inspection activity using the Mann-Whitney test (a non-parametric test), considering a confidence interval of 90% ($\alpha = 0.10$) due to the small sample size (Dybå et al. 2006). Also, we used boxplots to compare of the samples. Both, the statistical test and the boxplots were performed from Minitab17 software (Minitab 2000). We had formulated the following null and alternatives hypotheses:

- H_{01}: There is no difference between the efficiency of usability inspection technique in ERP systems performed with the HPERP or with the HN.
- H_{A1}: The usability inspection in ERP systems using HPERP is more efficient than usability inspection in ERP systems using HN.
- H_{02}: There is no difference between the effectiveness of usability inspection technique in ERP systems performed with the HPERP or with the HN.
- H_{A2}: The usability inspection technique in ERP systems using HPERP is more effective than usability inspection in ERP systems using HN.

Figure 5 shows the boxplots comparing the distribution of efficiency per group, and per inspected object. From boxplots, we can notice that the median of G2 is higher than the median of G1. When we compare the two samples with Mann-Whitney test, we found a significant difference between the two groups ($p = 0.037$). These results suggest that the HPERP was more efficient than HN when used to inspect a medium-fidelity prototype; hence, the null hypothesis (H_{01}) is rejected. Concerning to other two groups, we can notice in the related boxplots that the median of G3 is higher than the median of the G4. However, when we compared the two samples using the MannŰWhitney test, we found no significant differences between the two groups ($p = 0.54$). Thus, these results suggest

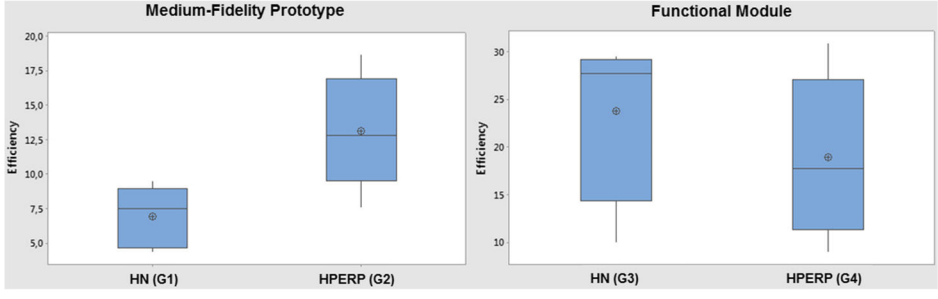

Fig. 5 Efficiency per groups, and per inspected objects

that both HN and HPERP provided similar efficiency when used to inspect a functional sub-module, and alternative hypothesis (H_{01}) is rejected.

Concerning to effectiveness, we can notice in Fig. 6 that the median of the G2 appears higher than the median of the G1. We compared the two samples with Mann-Whitney test, and we found again a significant difference between the two groups ($p = 0.044$). These results suggest that the HPERP was also more effective than HN when used to inspect a medium-fidelity prototype; thus, we can reject the null hypothesis (H_{02}). Likewise, the median of G3 appears slightly higher than the median of G4. However, when we compared the two groups using the Mann-Whitney test, we found no significant difference between them ($p = 0.39$). This suggests that both groups had a similar efficacy to inspect the functional sub-module. Hence, we can reject the alternative hypothesis (H_{A2}).

Findings and lessons learned

In summary, we found that the use of the perspective-based ERP heuristics (HPERP) to evaluate the usability of medium-fidelity prototypes can improve the performance of novice-inspectors (RQ4), and can identify effectively the major usability issues in ERP systems (RQ5). And, in the case of functional modules, both methods – HPERP and HN – can be considered similar in their efficiency and effectiveness. Further, upon concluding of the inspection activity, the participants were invited to answer a questionnaire based on model called TAM (Davis 1989), aiming to identify factors involved in satisfaction of individuals regarding the acceptance and use of the inspection technique. From this questionnaire, we found that no participant had doubts about the usefulness of the technique, and though some difficulties encountered, the majority of participants agrees that the technique was easy to use and understand.

There were some threats to validity that could affect our results. We controlled some threats related to internal validity, such as the training effects by preparing a same type of training – simultaneously applied – to the four inspection groups on the inspection technique about two set of heuristics. Regarding the participants' experience, any type of classification was not necessary, since the intention was to apply to people without experience, and most participants did not know the technique.

Regarding examining the possibility of the results being generalized beyond the academic environment (external validity), we should to consider two points: (i) the heuristics by perspectives have been proposed based on the results of an investigation of major usability problems seen in consolidated products in the ERP software market; and (ii) the inspected objects – functional module and medium-fidelity prototype – were developed

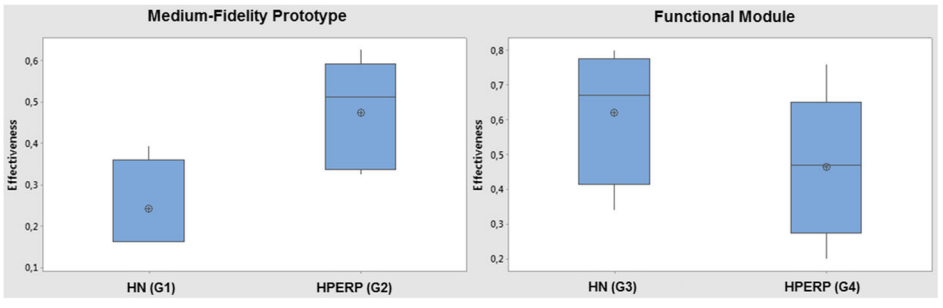

Fig. 6 Effectiveness per groups, and per inspected objects

by the development team and research partners from company. The experiment was set up based on the needs of the industrial partner. Nevertheless, is not possible to state that the results can be generalized to other companies.

After the experimental validation phase (Fig. 4 - item 3c), the results and the new artifact – perspective-based heuristics guideline – were discussed with the team in the company. The team highlighted the artifact could be an important guideline to support new professionals – with little or no practice in usability evaluation – in the team, mainly because there are people moving from one team to another and it can switch their roles. From the positive experiment results, together with practitioners, we concluded that other experiences in real environment would be needed prior to refine the process of usability evaluation. Therefore, a new workshop was carried out in order to teach the usability inspection method to the practitioners.

Discussion

So far, the practical applications of the SoftCoDeR approach were aimed at supporting research through Academia-Industry collaboration, and all research questions in each cycle were specifically geared to particular context of a company. In this particular case, the company sought us to help them improve their software development process by including new practices related to concerns about quality and satisfaction of its users. In our interaction with practitioners, we seek to make available our potential for research and scientific knowledge to meet the real needs of the industrial partner.

However, this can be an unusual situation, and very different from what commonly happens. In most cases, the research questions are motivated to answer specific interests of the researcher, e.g. when a gap is found in literature, and an empirical study in industrial context is necessary to verify theoretical assumptions related to research. However, we emphasize that the application of our approach implies in a research motivated and justified to meet practical concerns of the Industry. In this context, the gain of the researchers will be all experience gathered from practical applications of the approach.

Another crucial point is the collaboration between researchers and practitioners in order to develop and validate improvement solutions supported by artifacts. During the first two cycles, we had some difficulties to fit our research schedule to daily routine of the company. We have learned that it was not easy to deal with the lack of time of the professionals involved in their projects. During the third cycle, we had fewer problems like this, once the experimental validation was performed in the academic environment, and there was no need to interfere in company's activities. The experiment was planned and executed by the researchers. Practitioners provided the inspected objects, and two of them attended the experiment as observers. Thus, these practitioners could share a possibility of validation that was a novelty to the company.

When compared to the two previous cycles, we notice that the extended research cycle – using ESE practices –had required a greater scientific knowledge and a more active participation of researchers in order to drive the controlled experiment from planning to analysis. It is noteworthy that the experiment could have been conducted in the industrial environment. However, due to the issue type to be investigated, the academic environment could provide all resources – human and technical – and structure needed to meet

the experimental design. While in the company, we would have difficulties to recruit all participants, and to reserve space and the necessary equipment.

In the first two cycles, the project-based studies were performed in the industrial environment (in vivo), because the main objective was to validate the artifacts in real projects, and from the point of view of professionals - the planning and development teams. In the third cycle, the investigation purpose was to define the best usability inspection method from the point of view of novice inspectors. The investigation was a demand of the testers and developers who had difficulties in the adoption of Nielsen's heuristics caused by their little experience on usability inspection.

Hence, we chose to check beforehand which method would fit better to the novice inspector's knowledge. Focusing on checking the efficiency and effectiveness of the inspection method from the point of view of novice inspectors, it was relevant to involve a larger number of subjects through an experiment under controlled conditions (in vitro). Usually, a methodology based on experimentation is suggested in order to avoid wasted effort and to minimize problems when assessing the use of an artifact (concepts, tools, techniques, methods or methodologies) (Shull et al. 2001). And from these first results, a new study could be performed in the field under normal conditions – in industrial environment – by observational studies where there are no treatment or controlled variables (Basili 1996).

Conclusions

This paper presented the SoftCoDeR approach and its practical applications through Academia-Industry partnership, adding experimental software engineering practices to evaluate the artifacts before putting them in industrial practice. The core of the proposal has been to guide the academic knowledge to meet the real needs of the software Industry. From experience gathered from practical application, the researchers can be faced with problems that still have not been thoroughly investigated academically.

With regards to the limitations, we highlight the context in which our research approach has been applied. By means of this, we have been able to achieve our main research goals, even when we extend it using ESE practices. Notwithstanding, our experience with the SoftCoDeR approach has been in a single company until now, and within an agile software development context.

From point of view of the practitioners, this research approach allowed delivering valuable artifacts, adapting quickly to changes, and evolving from new research questions. For researchers, the approach allowed to develop partial theories since these were specifically generated from the context of the particular company. However, Wieringa and Daneva (2015) showed us that, in software engineering, it is much more useful to develop partial theories that can be applied to practice, than to develop a general theory that can be harder to be applied due to variability of the real world. Furthermore, in this field, engineers and researchers usually are interested in the interactions between an artifact and its context.

Based on the strategies to build theories identified by Wieringa and Daneva (2015), we can notice that the SoftCoDeR approach allows a hybrid strategy. In the first two cycles, the artifacts were validated during observational studies. In these cases, the strategy for generalization was a case-based research, in which artifacts were investigated in field with real projects. In this strategy, from the investigation of a single project,

we were able to hypothesize a generalization about all similar projects in the partner company.

When we used the ESE to validate the artifact, the strategy was to generalize from outcomes of the experiment, in which we could proposed a theory in order to claim that the inspection method (HPERP) that was experienced is more effective and efficient mainly for evaluate the usability of medium-fidelity prototypes. From outcomes of the controlled experiment, in case that a further study had been undertaken in the industrial environment with real projects, we would have a strategy of lab-to-field generalization. In lab-to-field strategy, researchers can start their investigations under ideal conditions in the lab and finish them under realistic conditions in the field.

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.

Acknowledgements
We would like to thank the company and the practitioners who were partners of this research project, and the students who participated in the phase of experimental validation. Also, we are very thankful to the anonymous reviewers for their excellent comments and suggestions. Grant 2014/25779-3, São Paulo Research Foundation (FAPESP).

Authors' contributions
LAMZ, JC and TSS worked on the research and design of the proposed model. JC and LAMZ defined the experiment design and carried out the experiment with the participants. TSS participated in the qualitative analysis of the first cycles of the study. JC carried out the experiment of the 3rd cycle about heuristics. JC and LAMZ analyzed the data collected in this cycle. All authors discussed the results and helped to write the manuscript. All authors revised and approved the final version of the manuscript.

Competing interests
The authors declare that they have no competing interests.

Received: 1 December 2015 Accepted: 25 November 2016
Published online: 15 December 2016

References

Adikari S, Mcdonald C, Campbell J (2009) Little design up-front: a design science approach to integrating usability into agile requirements engineering. In: International Conference on Human-Computer Interaction. Springer Berlin Heidelberg, pp 549–558
Al-Baik O, Miller J (2014) Waste identification and elimination in information technology organizations. Empir Softw Eng 19(6):2019–2061
Ardito C, Buono P, Caivano D, Costabile MF, Lanzilotti R (2014) Investigating and promoting ux practice in industry: An experimental study. Int J Human-Comput Stud 72(6):542–551
Avison DE, Lau F, Myers MD, Nielsen PA (1999) Action research. Commun ACM 42(1):94–97
Basili VR (1996) The role of experimentation in software engineering: past, current, and future. In: Proceedings of the 18th international conference on Software engineering. IEEE Computer Society, pp 442–449
Baskerville R (2008) What design science is not. Eur J Inform Syst 17(5):441–443
Baskerville R, Pries-Heje J, Venable J (2009) Soft design science methodology. In: proceedings of the 4th international conference on design science research in information systems and technology. ACM, p 9
Brown JM, Lindgaard G, Biddle R (2011) Collaborative events and shared artefacts: Agile interaction designers and developers working toward common aims. In: Agile Conference (AGILE). IEEE, pp 87–96
Choma J, Zaina LA, Da Silva TS (2015a) Towards an Approach Matching CMD and DSR to improve the Academia-Industry Software Development Partnership: A Case of Agile and UX Integration. In: Software Engineering (SBES), 2015 29th Brazilian Symposium on. IEEE, pp 51–60
Choma J, Quintale D, AM Zaina L, Beraldo D (2015b) A perspective-based usability inspection for ERP systems. In: Proceedings of the 17th International Conference on Enterprise Information Systems. Vol. 3. SCITEPRESS-Science and Technology Publications, Lda, pp 57–64
Cohn M (2004) User Stories Applied: For Agile Software Development. Addison-Wesley Professional, Indiana
Cole R, Purao S, Rossi M, Sein M (2005) Being Proactive: Where Action Research Meets Design Research. ICIS 2005 Proceedings, 27
Conte T, Massolar J, Mendes E, Travassos GH (2009) Web usability inspection technique based on design perspectives. IET Softw 3(2):106–123
Cooper A, Reimann R, Cronin D, Noessel C (2014) About Face: The Essentials of Interaction Design. John Wiley & Sons, Indiana
Davis FD (1989) Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Quarterly 13(3):319–340
Dieste O, Juristo N, Martínez MD (2013) Software industry experiments: A systematic literature review. In: Proceedings of the 1st International Workshop on Conducting Empirical Studies in Industry. IEEE Press, pp 2–8

Dittrich Y (2002) Doing empirical research on software development: finding a path between understanding, intervention, and method development. In: Social thinking. MIT Press, pp 243-262
Dittrich Y, Rönkkö K, Eriksson J, Hansson C, Lindeberg O (2008) Cooperative method development. Empir Softw Eng 13(3):231–260
Dybå T, Kampenes VB, Sjøberg DI (2006) A systematic review of statistical power in software engineering experiments. Inform Softw Technol 48(8):745–755
Easterbrook S, Singer J, Storey MA, Damian D (2008) Selecting empirical methods for software engineering research. In: Guide to advanced empirical software engineering. Springer, London, pp 285–311
Hevner AR (2007) A three cycle view of design science research. Scand J Inf Syst 19(2):4
Hevner AR, March ST, Park J, Ram S (2004) Design science in information systems research. MIS Q 28(1):75–105
Iivari J, Venable J (2009) Action research and design science researchŰseemingly similar but decisively dissimilar. In 17th European Conference on Information Systems 17:1–13
Isomursu M, Sirotkin A, Voltti P, Halonen M (2012) User experience design goes agile in lean transformation–a case study. In: Agile Conference (AGILE). IEEE, pp 1–10
Jaccheri L, Osterlie T (2005) Can we teach empirical software engineering? In: 11th IEEE International Software Metrics Symposium (METRICS'05). IEEE, pp 25–25
Law E, Roto V, Vermeeren AP, Kort J, Hassenzahl M (2008) Towards a shared definition of user experience. In: CHI'08 extended abstracts on Human factors in computing systems. ACM, pp 2395–2398
March ST, Smith GF (1995) Design and natural science research on information technology. Decis Supp Syst 15(4):251–266
Minitab I (2000) MINITAB statistical software. Minitab Release, 13
Nielsen J (2011) How to Conduct a Heuristic Evaluation. http://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/. Accessed 30 Sept 2014
Pearsall J, Hanks P (1998) The new Oxford dictionary of English. Clarendon Press, Oxford
Rivero L, Conte T (2012) Using an Empirical Study to Evaluate the Feasibility of a New Usability Inspection Technique for Paper Based Prototypes of Web Applications. In: Software Engineering (SBES), 2012 26th Brazilian Symposium on. IEEE, pp 81–90
Rodríguez P, Kuvaja P, Oivo M (2014) 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. ACM, pp 9–14
Rombach D, Achatz R (2007) Research collaborations between academia and industry. In: 2007 Future of Software Engineering. IEEE Computer Society, pp 29–36
Salah D, Paige RF, Cairns P (2014) 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. ACM, p 5
Santos G, Rocha AR, Conte T, Barcellos MP, Prikladnicki R (2012) Strategic alignment between academy and Industry: a Virtuous Cycle to Promote Innovation in Technology. In: Software Engineering (SBES), 2012 26th Brazilian Symposium on. IEEE, pp 196–200
Sein M, Henfridsson O, Purao S, Rossi M, Lindgren R (2011) Action Design Research. MIS Quarterly 35(1):37–56
Shadish WR, Cook TD, Campbell DT (2002) Experimental and Quasi-experimental Designs for Generalized Causal Inference. Houghton Mifflin, Boston, NY
Shull F, Carver J, Travassos GH (2001) An empirical methodology for introducing software processes. ACM SIGSOFT Softw Eng Notes 26(5):288–296
Singh A, Wesson J (2009) Evaluation criteria for assessing the usability of ERP systems. In: Proceedings of the 2009 annual research conference of the South African Institute of Computer Scientists and Information Technologists. ACM, pp 87–95
Sjøberg DI, Hannay JE, Hansen O, Kampenes VB, Karahasanovic A, Liborg NK, Rekdal AC (2005) A survey of controlled experiments in software engineering. Softw Eng IEEE Trans 31(9):733–753
Vegas S, Dieste Ó, Juristo N (2015) Difficulties in running experiments in the software industry: experiences from the trenches. In: Proceedings of the Third International Workshop on Conducting Empirical Studies in Industry. IEEE Press, pp 3–9
Wieringa R, Daneva M (2015) Six strategies for generalizing software engineering theories. Sci Comput Program 101:136–152
Wieringa R, Moralı A (2012) Technical action research as a validation method in information systems design science. In: Design Science Research in Information Systems. Advances in Theory and Practice, pp 220–238
Wohlin C, Runeson P, Höst M, Ohlsson MC, Regnell B, Wesslén A (2012) Experimentation in Software Engineering: An introduction (Vol. 6). Springer Science & Business Media