PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL

FACULDADE DE INFORMÁTICA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

A FRAMEWORK FOR INTEGRATING INTERACTION DESIGN AND AGILE METHODS

Tiago Silva da Silva

Thesis presented as partial requirement for obtaining a Ph.D. degree in Computer Science at Pontifical Catholic University of Rio Grande do Sul

Supervisor: Dr. Milene Selbach Silveira

Porto Alegre

2012

S586f da Silva, Tiago Silva
A framework for integrating interaction design and agile methods / Tiago Silva da Silva. – Porto Alegre, 2012.
110 f.

Tese (Doutorado) – Fac. de Informática, PUCRS.
Orientador: Prof. Dr. Milene Selbach Silveira.

1. Informática. 2. Engenharia de Software. 3. Interface com Usuário. I. Silveira, Milene Selbach. II. Título.

CDD 005.1

Ficha Catalográfica elaborada pelo
Setor de Tratamento da Informação da BC-PUCRS

Pontifícia Universidade Católica do Rio Grande do Sul
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

TERMO DE APRESENTAÇÃO DE TESE DE DOUTORADO

Tese intitulada "A Framework for Integrating Interaction Design and Agile Methods", apresentada por Tiago Silva da Silva, como parte dos requisitos para obtenção do grau de Doutor em Ciência da Computação, Sistemas de Informação, aprovada em 22/03/2012 pela Comissão Examinadora:

Profa. Dra. Milene Selbach Silveira – PPGCC/PUCRS
Orientadora

Prof. Dr. Rafael Prikladnicki - PPGCC/PUCRS

Prof. Dr. Marcelo Soares Pimenta - UFRGS

Prof. Dr. Frank Maurer – UofC

Homologada em 22/05/2012, conforme Ata No. 011 pela Comissão Coordenadora.

Prof. Dr. Paulo Henrique Lemelle Fernandes
Coordenador.

PUCRS
Campus Central
Av. Ipiranga, 6681 - P. 32 - sala 507 - CEP: 90619-900
Fone: (51) 3320-3611 - Fax (51) 3320-3621
E-mail: ppgcc@pucrs.br
www.pucrs.br/facin/pos

DEDICATION

To my dear family.

ACKNOWLEDGEMENTS

First, I would like to acknowledge and thank my family for all the support.

Thank to my fiancé Roberta for all the support during these four years of PhD research, for being by my side in the most difficult moments.

A special thank to Prof. Dr. Milene Silveira for being my supervisor and a real friend since my Masters degree till the end of the PhD.

A special thank to Prof. Dr. Frank Maurer for giving me the opportunity to work with you and being my mentor during the time that I spent at UofC.

I would like to thank Prof. Dr. Duncan Ruiz for being a friend and supporting me or even sponsoring me. A thank to Prof. Dr. Stanley Loh for giving me the first push, you'll always be my mentor.

A thank to Prof. Dr. Rafael Prikladnicki and Prof. Dr. Marcelo Pimenta for being of my committee and for being available everytime I needed.

To my friends Tiago Primo, Thyago Borges, Ricardo Piccoli, Ana Winck and Márcio Basgalupp. To Prof. Dr. Marcio Pinho, to my colleagues from GPIN and from the ASE Lab.

A thank to my "Canadian" friends, Karlson and Josy Oliveira, Alvaro Pereira, Anderson Macedo and Estela Costa.

A special thank to my brother Tulio Alcantara for all the support and the partnership and to my friend Tedd Hellmann.

A thank to Samuel Crescêncio and Luiz Parzianello, the men of Agile in Brazil. A thank Marco Migliavacca for opening the doors of the company allowing to finish my research. Also, I'd like to thank to Patricia, Diana, Cesar and Max for supporting with the studies performed in Brazil. A thank to Alan and Raul for giving all the support with my studies in Canada.

Last but not least, I'd like to thank to Grupo RBS, SMART Technolgies, PUCRS, HP Brazil R&D, HP Consulting, and CAPES.

FRAMEWORK PARA INTEGRAÇÃO DE DESIGN DE INTERAÇÃO E MÉTODOS ÁGEIS

RESUMO

Desenvolvimento Ágil tem estado em evidência no que diz respeito à processos de desenvolvimento de software. Juntamente com o crescimento da conscientização sobre a importância de uma boa Experiência de Usuário surgiu a necessidade de integrar estas duas áreas. Entretanto, o desenvolvimento Ágil possui uma cultura distinta que, num primeiro momento, parece entrar em conflito com o Design Centrado no Usuário. Assim, a integração destas duas áreas torna-se um desafio. Esta tese focaliza na integração destas duas áreas, fornecendo um conjunto de práticas e artefatos para apoiar equipes Ágeis e Designers de Interação a superar tal desafio. Uma Revisão Sistemática foi realizada a fim de identificar propostas de integração de métodos Ágeis e Design de Interação. Com base na reunião das práticas e artefatos mais comuns identificados na revisão, um framework foi proposto. A fim de verificar tal proposta, Pesquisas-Ação foram realizadas em duas organizações de dois diferentes países. Desta forma, o resultado desta pesquisa é a proposta de um framework para integrar Design de Interação e desenvolvimento Ágil.

Palavras-chave: interação humano-computador, experiência de usuário, engenharia de software, métodos ágeis, integração, framework.

A FRAMEWORK FOR INTEGRATING INTERACTION DESIGN AND AGILE METHODS

ABSTRACT

Agile development has become mainstream regarding software development processes. Along the increasing understanding of the importance of good User eXperience came the need to integrate these two areas. However, Agile development have a distinct culture that at first glance seems to conflict with User-Centered Design. Therefore, integrating these two areas becomes a challenging task. This thesis focuses on integrating these areas, providing a set of practices and artifacts to support Agile teams and Interaction Designers to overcome this challenge. A Systematic Literature Review was conducted in order to identify existing approaches regarding the integration of Agile and Interaction Design. A framework was proposed gathering the most common practices and artifacts identified in this review. Conducting Action Research in two companies from two different countries tried out this proposal. Thus, the result of this research is a framework proposal for integrating Interaction Design and Agile Development.

Keywords: human-computer interaction, user experience, software engineering, agile methods, integration, framework.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

1. INTRODUCTION

Due to their popularity, Agile software development methods are being adopted at an increasing rate in the industry and according to [BEY10], they have transformed how developers think about organizing the development project. [DYB08] presented a good review on the empirical studies in 2008, this review investigates what is currently known about the benefits and limitations of, and the strength of evidence for agile software development.

The agile approach promises to produce more useful and more reliable software, more quickly, and with better control than the traditional development, like in waterfall, for instance. Meanwhile Interaction Design is concerned with designing interactive products to support the way people communicate and interact in their everyday lives. Thus both Agile Development and Interaction Design aim at software quality, even though they are different concepts of quality, it is still software quality. However, it is known that the integration of Interaction Design[1] into Agile Methods is not adequately addressed [HUS09b].

Agile Methods have a distinct culture that at first glance seems, e.g. different approaches for requirements gathering, to conflict with User-Centered Design (UCD)[2] [MCI05]. However, according to these same authors, the use of agile methods can result in improved usability. Moreover, in their study, the authors did not find any interaction designers who preferred traditional approaches over agile processes.

One of the problems of integrating these two methodologies (Agile methods and UCD) is that traditionally they use different approaches on how resources are allocated in a project [FOX08]. Agile methods strive to deliver small sets of software features to customers as quickly as possible, in short iterations. On the other hand, UCD traditionally spend a considerable effort on research and analysis before development begins.

Interaction Design associated with non-agile teams has led to a combination of results [WIL07a]. For example, in non-agile projects, the Interaction Design group has written UI (User Interface) specifications that are Word documents ranging from 5 to 200

[1] There is no consensus regarding some terms in the literature studied. In this work, most of the times we use the term 'Interaction Design' as an umbrella term that includes Human-Computer Interaction (HCI)/usability/ user experience (UX)/user-centered design (UCD).

[2] However, sometimes we will use terms like UCD, UX Design, UX Designer because they were used in the studies analyzed, and we would like to keep the original terms used by their authors.

pages of description and images. It can take months to complete a UI specification, besides the need for meetings to review and provide answers to questions about it.

While the two methodologies have different approaches regarding requirements gathering and upfront design, they also have similarities. The main is that both approaches are user and customer focused. As the name suggests, UCD focuses on developing software with the user in mind. Agile usually involves a local representative of the client to shorten the feedback loop between the development team and the customer.

Also both Agile development and Interaction Design methods aim to build quality software. While Agile methods enable software development teams to create software that is valued by the customer, Interaction Design methods allow software development teams to create software that is usable for the user.

[PAE08] state that a key motivation behind agile and iterative development is the idea that software development is similar to the creation of new and inventive products and flexibility for research and creativity is required in the process.

Besides that, [PAT02] explains how the different perspectives support development in a complementary way: 'Agile development methods allowed us to deliver high quality software sooner, and interaction design concepts lent us the degree of end-user empathy we were missing to help increase confidence that we hit our target of end-user satisfaction'.

According to [FER11], claims about how these two approaches should work together based on an analytical appraisal, highlight similar points of focus and possible tensions. Still according to these authors, there is little guidance about integrating these two perspectives and still few detailed proposals of Agile software development and Interaction Design being combined in practice.

Thus the focus of this thesis is on the integration of Interaction Design and Agile methods, aiming to define a framework[3] encompassing common practices and artifacts, identified both in Academia and Industry, to help Agile teams regarding this topic.

1.1 Goals

The general goal of this thesis is to propose a framework for integrating Interaction Design and Agile Methods, providing a set of practices and artifacts to support Agile teams to overcome this challenge.

[3] In this work the term framework is used to describe a number of practices embraced within a defined process and supported by a set of tools.

Aiming to reach this general goal, we defined the following specific goals:

- Bring up an extensive literature review regarding Agile and Interaction Design.
- Identify existing approaches regarding the integration of Agile and Interaction Design.
- Define a framework for this integration.
- Analyze the framework proposed.

1.2 Relevance

Despite the integration of Agile and UX be a recent topic, there are some approaches for this integration. Some of them are based on interviews with practitioners - UX Designers and/or Developers, some are based on experience reports, others are based on Agile Methods by the book and some of them are just speculating approaches [daS11]. In their study, the authors did not observe science-based approaches with any kind of verification.

Thus, this research is relevant because it tries to bridge this gap. The framework proposed in this thesis is a theoretical-practice approach based on a systematic literature review that provides a greater integration between Academia and Industry. This framework proposal will be verified by the application of a qualitative approach - Action Research.

1.3 Research Design

This research can be classified as an applied research, qualitative and exploratory. This is a non-experimental research, carried out in the field.

In order to develop a framework for integrating Interaction Design and Agile Development, we organized the research methodology in two phases, as presented in Figure **1**.

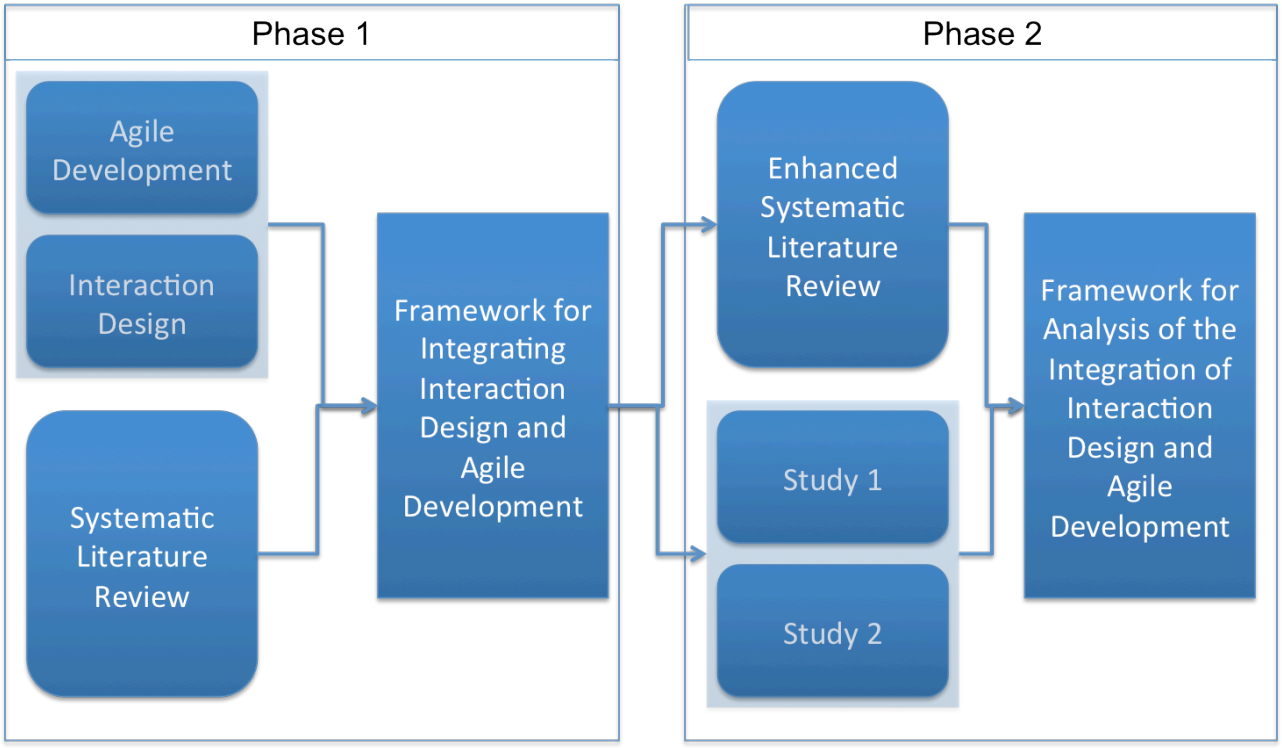

Figure 1. Research Design

At Phase 1 we defined the research problem as well as its strategy, the research design itself and we performed a Systematic Literature Review. At Phase 2, we refined this Systematic Literature Review and carried out two field studies at two companies that develop software products and aim to integrate Agile Development and Interaction Design.

Initially, we planned to perform two Action Research studies at the second phase. However, during the development of the studies we noticed that the implementation step of the Action Research was outside our control and we could not create organizational changes. Thus we performed two field studies using Grounded Theory techniques to collect and analyze the data at two companies, one in Canada and another one in Brazil, comprehending collecting, analyzing and presenting the results.

1.4 Thesis Structure

This document is divided in seven chapters, as follows:

Chapter 2 presents the background of the research field, including the basics of Human-Computer Interaction and Agile development needed to the reader's understanding of the proposed framework, and some challenges regarding the integration of these two areas.

In Chapter 3, a systematic literature review of related work is presented.

Chapter 4 brings up the proposed framework.

Chapter 5 presents the research methodology followed to analyze the proposed framework as well as the studies performed.

Chapter 6 presents the refined framework, after the completion of the studies.

Finally, Chapter 7 presents final considerations, including the main contributions of this research, limitations and potential future work.

2. BACKGROUND

In this Chapter, we provide an introduction to Agile development as well as to Interaction Design.

Section 2.1. presents the core idea of Agile methods, bringing up a brief description of two of the most common agile methods – XP and Scrum – focusing on the core elements shared by the Agile methods. Section 2.2. presents some concepts regarding Interaction Design and the components of this process.

2.1 Agile Methods

According to [LAR04], it is not possible to exactly define agile methods, as specific practices vary. However, these methods apply time boxed iterative and evolutionary development, adaptive planning, promote evolutionary delivery, and include other values and practices that encourage agility – rapid and flexible response to change. [LAR04] says that the slogan of Agile methods could be: embrace change; and its strategic point could be: maneuverability.

In addition, they promote practices and principles that reflect an agile sensibility of simplicity, lightness, communication, self-directed teams, programming over documenting, and more.

In 2001, in Salt Lake City, a group interested in iterative and agile methods met to find common ground. From this meeting resulted the Agile Manifesto[4] which says that:

- Individuals and interactions over processes and tools.
- Working software over comprehensive documentation.
- Customer collaboration over contract negotiation.
- Responding to change over following a plan.

This is the Agile Manifesto, which originated the agile principles[5] that have guided the agile development.

The agile principles have guided a number of agile methods, e.g. Scrum [SCH01], eXtreme Programming (XP) [BEC99] and Crystal [COC01].

Among the existing Agile methods, Scrum, followed by Scrum/XP Hybrid, were the most common agile methodologies used in 2010, according to the annual State of Agile

[4] http://agilemanifesto.org/

[5] They can be verified at [LAR04]

Development survey [VER10], and as can be noticed at Figure **2**. Next subsections briefly describe these two methodologies, Scrum [SCH01] and XP [BEC99].

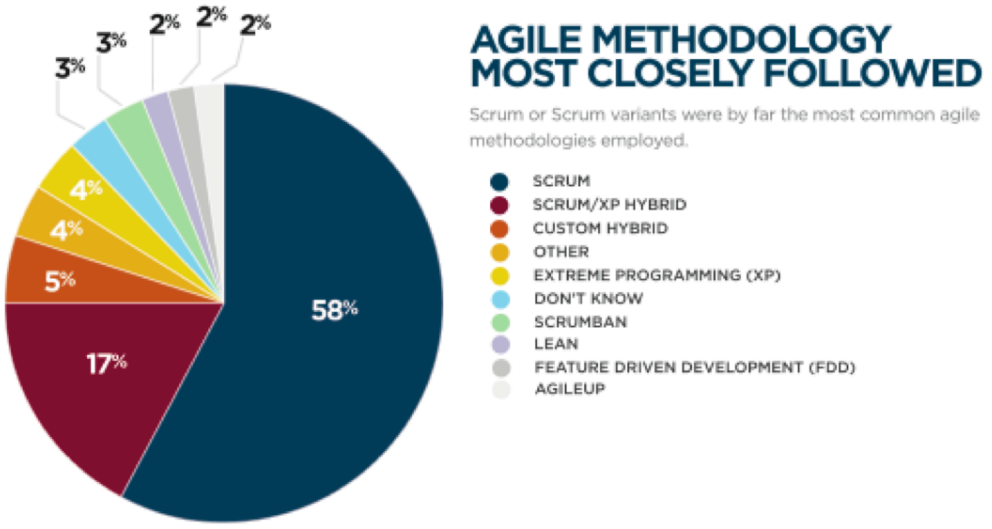

Figure 2. Agile Methodology Most Closely Followed in 2010 [VER10]

2.1.1 Scrum

Scrum was described as a software development process by [SCH01]. According to [BEY10], Scrum retains its roots as a product development framework, focusing more on project management and less on the specifics of coding procedures.

Having institutional support – Scrum Alliance[6], that train and certify people as Scrum Masters, for example – helped Scrum to be quite popular as a project management framework.

According to [LAR04], Scrum's distinctive emphasis among the Agile methods is its strong promotion of self-organizing teams, daily team measurement, and avoidance of following predefined steps. Some key practices include a daily stand-up meeting with special questions, and a demo to external stakeholders at the end of each iteration.

Figure **3** presents a visual introduction to Scrum.

[6] http://www.scrumalliance.org/

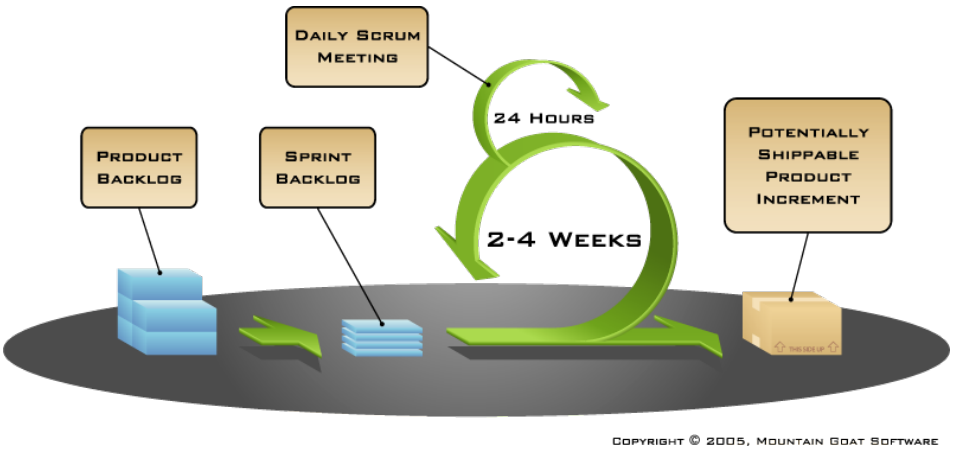

Figure 3. Scrum process from [KOH11]

According to [KOH11], a Scrum Team is typically made up of between five and nine people. The Team is multidisciplinary, and everyone on the project works together to complete the set of work they have together committed to complete within a sprint.

The Product Owner (PO) is the project's key stakeholder and representative of the user, customers and others in the process. The Product Owner is often someone from product management or marketing, a key stakeholder or a key user.

The Scrum Master is the facilitator, he is in charge of making sure the Team is as productive as possible. He removes impediments to progress, protects the Team from outside, and so on.

There is a Product Backlog, what is a prioritized list of features containing every desired feature or change to the product.

At the start of each Sprint, a Sprint Planning meeting is held, during which the Product Owner prioritizes the Product Backlog, and the Team selects the work they can complete during this coming sprint. That work is then moved from the Product Backlog to the Sprint Backlog, which is the list of tasks needed to complete the Product Backlog items the Team has committed to complete in the Sprint.

Each element of the desired solution is described in a User Story. The story is written on an index card or sticky note. The story is not intended to be a full description of the feature – rather, it captures enough about the feature to remind everyone what the feature is. User Stories must be small enough to be implemented within a single Sprint. Large, complex stories must be broken down into smaller stories, ideally, in such a way that each smaller story still makes sense when it is implemented on its own [BEY10].

One common format for writing stories is to write them in the form 'As a [user role], I want [a feature] so that I can [achieve some goal]'.

There is a Daily Scrum (stand-up meeting). This meeting helps set the context for each day's work and helps the team stay on track. All team members are required to attend the Daily Scrum.

At the end of each Sprint, the Team demonstrates the completed functionality at a Sprint Review meeting, during which, the team shows what they accomplished during the Sprint. In general, this takes the form of a demonstration of the new features.

2.1.2 eXtreme Programming

According to [LAR04], eXtreme Programming (XP) emphasizes collaboration, quick and early software creation, and skillful development practices. It is founded on four values: communication, simplicity, feedback and courage.

According to [BEY10], XP assumes that a real end-user can be a full team member. They refer to this person as the Customer, somewhat confusingly for Interaction Designers. The XP Customer can decide what is or is not useful in the product.

In the Release Planning game, the XP Customer arrives with User Story cards describing everything that they think is needed in the next release of the product. Developers then estimate the implementation time or complexity required for each story.

The result of the Release Planning game is a rough estimate of when the project will be completed. XP emphasizes tracking how much work a team can do in a sprint – the Team's Velocity – and use this measure to limit the work they commit to.

In XP, Sprints are referred to as Iterations. Each Iteration starts with an Iteration Planning session. This is a meeting in which the team selects the stories to be implemented during an Iteration. The XP Customer chooses the stories most important to provide value.

The Team uses their velocity to decide how many Stories to select. Once Stories are selected, the developers break down the Stories into tasks that can be done by different people on the team.

XP culture is strongly test-driven, and it values automated testing highly. Teams are expected to do nightly builds and run their entire suite of unit tests nightly; test-driven design, in which the tests are written first, is an approved practice. Teams want customer acceptance tests to be automatable and, ideally, defined along with the user stories. These automated acceptance tests can show whether the implementation meets the technical requirements defined by the user story.

Finally, XP defines a number of development practices intended to support rapid development with little documentation. Pair programming, test-driven development,

collective code ownership, and the nightly build are a few of these. Though these are important, they have little impact on how UX people fit into an agile team.

2.1.3 Scrum and XP

Because Scrum and XP address slightly different problems, they dovetail with each other fairly easily. Scrum provides the overall project management; XP provides more detailed guidance on running development. This is how many teams use them together.

However, XP practices are difficult and disruptive to the daily lives of developers. They may be valuable, but they require discipline to implement. Scrum, on the other hand, structures project management but makes fewer demands on developers. Therefore, when teams are new to agile development, they often adopt the backlog and sprints from Scrum but do not really change their development practices. UX professionals need to recognize how far along a development team has progressed in agile adoption in order to understand the best way to integrate with that team.

2.2 Interaction Design

There is no consensus regarding the classification of process in the HCI area, but according to [SHA07], Interaction Design is under the HCI area umbrella and it is concerned with designing interactive products to support the way people communicate and interact in their everyday and working lives. According to [DIX04], Interaction Design is not just about the artifact that is produced, whether a physical device or a computer program, but about understanding and choosing how that is going affect the way people work.

[SHA07] say that [SOM07] uses the term 'process' model to mean what they call a lifecycle model, and refers to the waterfall model as the software lifecycle, while [PRE92] talks about paradigms. In HCI the term 'lifecycle' is used more widely. For this reason, and because others use 'process model' to represent something that is more detailed than a lifecycle model, we have chose to use lifecycle model.

[SHA07] see the activities of Interaction Design as being related as shown in Figure **4**, and it has its roots in the software engineering and HCI lifecycle models, such as:

- Software Engineering: Waterfall, Spiral, Rapid Applications Development (RAD), Agile Development.
- HCI: Star, Usability Engineering, ISO 13407.

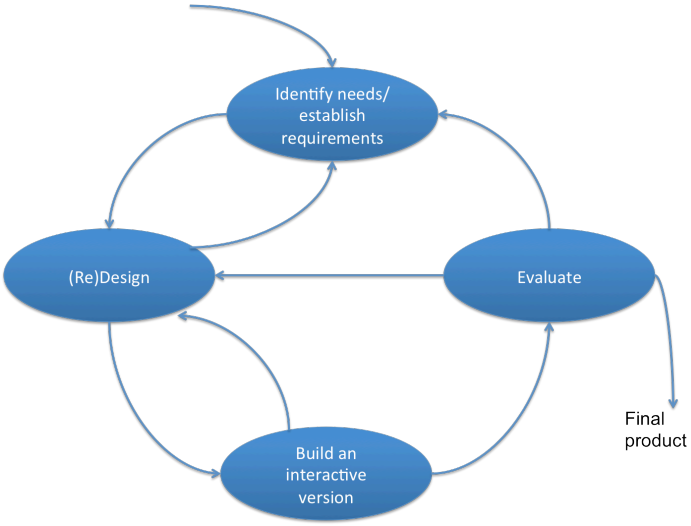

Figure 4. A simple interaction design lifecycle model [SHA07]

Due to the division/classification that we performed for the proposed framework we reorganized the Interaction Designers' practices and artifacts based on this lifecycle model proposed by [SHA07] (Figure **4**) as follows. We classified the activity 'Identify needs/establish requirements' as User Research, '(Re)Design' as Design and we kept 'Evaluate' as the same term.

Into the User Research activity are included requirements gathering and analysis.

The overall purpose of data gathering in the requirements activity is to collect sufficient, relevant, and appropriate data so that a set of stable requirements can be produced. Even if a set of initial requirements exists, data gathering will be required to expand, clarify, and confirm those initial requirements.

Observations and Interviews are probably the most popular techniques for gathering users' requirements, whereas Scenarios and Task Analysis are quite popular regarding users' requirements analysis.

According to [SHA07], Interviews can be thought as a 'conversation with a purpose', and like a conversation the interview can depend on the type of the interview method applied. Interviews can be unstructured, structured, semi-structured and group interviews (focus group). Observation is a useful data gathering technique at any stage during product development. Early in Design, observation helps Interaction Designers understand the user's context, tasks and goals. Whereas later in development, observation may be used to investigate how well the developing prototype supports these tasks and goals.

According to [DIX04], Task Analysis is the process of analyzing the way people perform their jobs: the things they do, the things they act and the things they need to know. It is used to investigate an existing situation, not to envision new product. It is used to analyze the underlying rationale and purpose of what people are doing. Whereas a

Scenario is a personalized, fictional story with characters, events, products and environments. They help the designer to explore ideas and the ramifications of design decisions in particular, concrete situations.

A popular technique for uncovering requirements related to the context of use is Contextual Inquiry [BEY99]. Contextual Inquiry is a technique that emerged from the ethnographic approach to data gathering. It is tailored to gather data that can be used in design and it follows an apprenticeship model, where the designer works as an apprentice to the user. According to [BEY99], Contextual Inquiry reveals the details and motivations implicit in people's work, makes the customer and their work needs real to the designers, introduces customer data as the basis for making decisions and creates a shared understanding of the data throughout the team.

Regarding Design, [PRE94] say that design activities begin once some requirements have been established. The design emerges iteratively, through repeated design-evaluation-redesign cycles involving users.

Prototyping is quite popular technique for Design even in the software engineering community, while task and interaction models are not that popular.

Prototypes are a useful aid when discussing ideas with stakeholders; they are a communication device among team members, and are an effective way to test out ideas. According to [SHA07], there are low-fidelity prototyping and high-fidelity prototyping. Some authors as [PRE94] still classify prototypes as horizontal - shows the user interface but has no functionality, and vertical prototype - contains all of the high level and low level functionality for a restricted part of a system. [DIX04] also classify approaches too prototyping as Throw-away, Incremental and Evolutionary

Regarding Evaluation, according to [PRE94], it is concerned with gathering data about the usability of a design or product by a specific group of users for a particular activity within a specified environment or work context. Also [DIX04] state that the Evaluation has three main goals: to assess the extent and accessibility of the system's functionality, to assess users' experience of the interaction, and to identify any specific problems with the system.

Evaluation is integral to the design process [SHA07]. It collects information about users' or potential users' experiences when interacting with a prototype, a computer system, a component of a computer system, or a design artifact. It focuses on both the usability of the system and on the users' experience when interacting with the system.

There are some different evaluation methods. Some of them involve users directly, such as Usability Evaluation [RUB94], Communicability Evaluation [deS09] and son on,

while others are performed by specialists, for instance, Cognitive Walkthrough [NIE94], Heuristic Evaluation [NIE94], and Guidelines Review [NIE94].

2.2.1 Interaction Design and Agile

XP, Scrum and Crystal Clear, for instance, according to [SHA07] differ, but they all exhibit certain characteristics, the agile principles. For example, [SHA07] state that they stress the importance of being able to handle emergent requirements, and of striking a good balance between flexibility and structure. They also emphasize collaboration, face-to-face communication, and streamlined process to avoid unnecessary activities, and the importance of practice over process.

[BEY10] state that all the Agile methods share core elements: short, well-defined iterations that deliver real user value; tight team processes for efficient development; minimal documentation of specifications; and continua feedback to validate progress. Agile methods also introduce a new development culture – values and attitudes that Agile teams are expected to adopt.

According to [SHA07], the agile approach is particularly interesting from the point of view of interaction design because it incorporates tight iterations and feedback, and collaboration with the customer. Although they emphasize this collaboration with the customer, it is important to mention that there are differences between Customer, User and Stakeholder.

[BEY10] states that Interaction Designers are used to making distinctions between these roles, but in the Agile community they are blurred. He classifies user as the person who interacts directly with the system to produce a desired result. Customers are the people who derive value from the system. And Stakeholders are people in the development organization who depend on the system being correct.

[SHA07] state that agile development methods do not favor one particular lifecycle model (although some have their own lifecycle, or development rhythm), but aim to be able to react to change quickly, and embed principles of iteration, communication and feedback, hence having characteristics sympathetic to user-centered design approaches.

3. SYSTEMATIC LITERATURE REVIEW

This chapter presents the entire process of a systematic literature review performed aiming at providing empirical support for a proposal of a framework for integrating Interaction Design and Agile, identifying most common practices and artifacts used. It also presents the results of this systematic review encompassing quantitative and qualitative results.

3.1 Review Methodology

A systematic review is a secondary study that identifies, evaluates and interprets all research available and relevant to a specific research question or phenomenon of interest [KIT07]. A systematic literature review is undertaken:

- 'to summarize the existing evidence concerning a treatment or technology.
- to identify any gaps in current research in order to suggest areas for further investigation.
- to provide a background in order to appropriately position new research activities.'

Additionally, systematic literature reviews can also be undertaken to examine the extent to which empirical evidence supports or contradicts theoretical hypotheses, or even to assist the generation of new hypotheses [KIT07].

As already mentioned, in this study a systematic review was conducted to provide empirical support for the framework to be proposed.

3.1.1 Terminology

HCI is heterogeneous, and frequently studies use different terms for quite similar concepts. Terms like UCD (User-Centered Design), UX (User eXperience), Usability and Interaction Design are used with a very similar meaning – specifically when we look at studies involving agile methods.

In this review, we use the term Interaction whenever an activity is related to the user, even it is related to design and/or evaluation of interaction and/or interface. Regarding Agile Methods, we use the term Agile as a superset of individual methods like XP, Scrum, Lean and others.

3.1.2 Protocol Development

For this systematic review, we used the recommendations of [BIO05] and [KIT07] in a complementary way. Our goal was to identify existing evidence regarding the integration of UCD and Agile. The research questions that guided this review were:

Q1: How are usability issues addressed in Agile projects?

Q2: What are common practices to address usability issues in Agile methods?

The research questions guided the selection of the search keywords for our systematic literature review. Initially, in addition to keywords from the Agile as well as Interaction Design fields, we used acronyms, e.g. XP for Extreme Programming. But an initial search including these acronyms identified a very large number of irrelevant papers and we decided to eliminate the acronyms from our search terms.

Table **1** presents the keywords used in the search.

Table 1. Keywords used in the review process

Category	Keywords
UCD	Usability
	Human-Computer Interaction
	Computer-Human Interaction
	Human Factor
	User Experience
	User-Centered Design
	User Interface
	Interaction Design
Agile	Agile
	Scrum
	Extreme Programming
	Lean Development
	Feature Driven Development
	Dynamic System Development
	Crystal Clear
	Agile Unified Process

3.1.3 Data Sources and Search Strategy

The search was a combination of UCD and Agile categories. Therefore, we have a search string as follows:

usability OR ”human-computer interaction” OR ”computer-human interaction” OR ”human factor” OR ”user experience” OR ”user-centered design” OR ”user interface” OR “interaction design“

AND

agile OR ”scrum” OR ”extreme programming” OR ”lean development” OR ”feature driven development” OR ”dynamic system development” OR ”crystal clear” OR “agile unified process“

The digital sources selected for the searches were:

- ACM Digital Library (http://www.acm.org/dl).
- IEEExplore (http://ieeexplore.ieee.org).
- Citeseer (citeseer.ist.psu.edu/).
- ISI Web of Science (apps.isiknowledge.com/).
- EI Compendex (http://www.engineeringvillage2.org/).
- Springer Link (http://www.springerlink.com/).
- Scopus (http://www.scopus.com).
- Google Scholar (http://scholar.google.com).

It is worthwhile to mention that each Digital Library (DL) has its own particularities concerning their search engines, therefore, the search strings required to be adapted for each source.

ACM Digital Library does not allow searching for papers by Title, Abstract and Keywords at the same time. Therefore we had to perform the searches by Title, then by Abstract and then by Keywords. Then we had to compare the results and cut out the repeated papers.

IEEExplore allows the search by Metadata. According to the IEEExplore website, Metadata includes Abstract (summary) and title text, and indexing terms (on both command and default advanced search).

Citeseer is like the ACM DL, we had to search for papers separately, by Title, by Abstract and by Keywords.

ISI Web of Science allows the search by title and by topic. Then we performed the searches by Title and by Topic. Afterwards we merged the results.

EI Compendex provides a feature to remove duplicates from their results. We used the Expert Search with this Remove Duplicates feature from the databases covered by Compendex to search for papers.

At Springer Link, we decided to use the Regular Search and use our entire string and search at the Full Text. We also filtered the results by Collection (Computer Science), Content Type (Journal Articles) and Language (English).

Scopus allowed us to search by Title, Abstract and Keywords in the same search. We could also use filters and export our results into different formats. We only filtered by area: Computer Science.

The searches at Google Scholar were returning around 9000 titles and we would not manage this amount of titles. So we decided to calculate the average of the papers by sources so far. The amount of papers was 1070, from 7 different sources. Then we calculated the average, 1070 / 7 = 152.85. So in the GoogleScholar search we selected the 150 most relevant papers, according to the relevance criteria[7] used by Google.

In addition, following the example of [DYB08], we hand-searched all the volumes of the following conference proceedings for research papers and experience reports on UCD and Agile:

- XP
- XP/Agile Universe
- Agile Development Conference

3.1.4 Inclusion/Exclusion and Quality Criteria

We classified the papers as Research Papers and Industry Reports. To do so, we defined different inclusion/exclusion and quality criteria for the papers as presented in Figure **5** and described in the following sections.

3.1.4.1 Research Papers

For Research Papers we defined the following criteria based on [DYB08]:

- Inclusion
    1. Studies that present empirical data on agile software development and pass the minimum quality threshold.
    2. Studies of both students and professional software developers.
    3. Studies not restricted to any specific type of intervention or outcome measure.
    4. Qualitative and quantitative research studies, published up to and including 2011.
    5. Studies written in English.

[7] Google Patents - Method for node ranking in a linked database [Goo01]

6. Papers peer reviewed.

- Exclusion
  1. If the focus, or main focus, is not on usability issues in agile software development.
  2. If the study does not present empirical data.
- Quality Assessment
  1. Is there a clear statement of the aims of the research?
  2. Is there an adequate description of the agile context in which the usability issues are addressed?
  3. Do they describe the agile approach as well as the usability approach?
  4. Apply detailed questions 4-11 appearing in Appendix B of [DYB08].

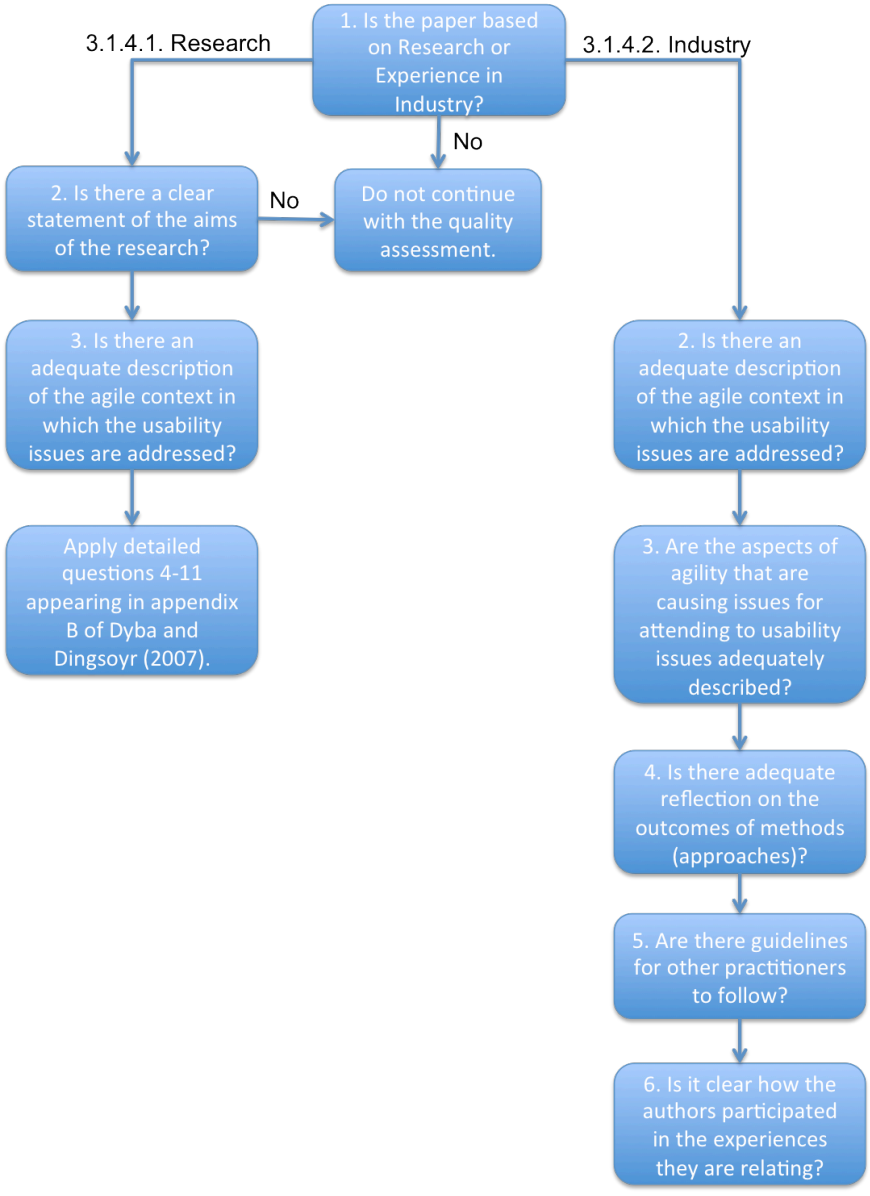

Figure 5. Inclusion/Exclusion and Quality Criteria

3.1.4.2 Industry Reports

For Industry Papers we defined the following criteria adapted from [DYB08,ARI09].

- Inclusion
    1. Reports that relate experiences of using agile development in industry and pass the minimum quality threshold.
    2. Reports from professional software developers.
    3. Reports published up to and including 2011.
    4. Reports written in English.
    5. Studies on agile methods in general, as well as specific methods: XP, Scrum, Crystal, DSDM, FDD, Lean.
- Exclusion
    1. If the focus, or main focus, is not on usability issues in agile software development.
- Quality Assessment
    1. Is there an adequate description of the agile context in which the usability issues are addressed?

       Consider whether the authors describe the agile approach as well as the usability approach.

       Consider whether the authors describe the business context, i.e. the company, team, and product details.

       Consider whether they are promoting a company/product.
    2. Are the aspects of agility that are causing issues for attending to usability issues adequately described?

       Are usability issues being related to agile issues?

       Do they describe what the usability issue is?
    3. Is there adequate reflection on the outcomes of methods (approaches)?

       Consider whether the authors mention benefits as well as drawbacks.

       Consider whether the authors give reasons for the outcomes of their approaches.
    4. Are there guidelines for other practitioners to follow? Are there key lessons/takeaways?

5. Is it clear how the authors participated in the experiences they are relating?

   Is it clear whether the authors were on the agile development team, the usability team, or other?

   Do the authors make their role clear in the events that are being related in the report (this could be in their affiliation information)?

First, based on the reading of papers titles, two researchers defined whether the papers were related to the research topic or not. Some papers were considered relevant based just on their titles, some papers were considered relevant because the researchers already knew them and some papers were kept for the full text reading because the researchers were not sure about their inclusion.

The papers were classified according the two general categories of information: research papers and industry reports, following the recommendations of [BIO05]:

- Focus on (user research, design, evaluation).
- Approach (specialist, generalist, specialist/generalist).
- Results (need of an approach, proposed approach, lessons learned, recommendations).
- Circumstances (in large teams, in small teams, in novel projects, in redesign projects).
- Perspective of (a UX designer, a developer, a business analyst, an academic).
- Conclusions.

3.1.5 Data Extraction

During this stage, data was extracted from each of the primary studies included in this systematic review according to a predefined extraction form that enabled us to record details of the articles under review and to be specific about how each of them addressed our research questions.

The papers were read and, as suggested by the protocol [BIO05], from this reading we derived objective and subjective information. For objective information, the following data were extracted: study identification, study methodology, study results and problems of the study. Regarding subjective information, which consists of those results that cannot be extracted directly from the study, the information was extracted as follows: additional

information through authors (if the reviewer contacted the study's authors to solve doubts or ask for more details about it) and general impressions and abstractions.

3.2 Results

The search in the digital libraries was conducted in June 2011. A total of 1220 papers were found, as presented in Table **2**. After a merge of the results, we have 525 repeated papers. Therefore, the final amount of papers to be analyzed was 695 papers.

Two researchers read the title and abstract of these 695 selected studies. Based on this reading, 127 papers were selected for a full text reading. After the full text reading of these 127 papers, 61[8] of them fit into the inclusion and quality assessment criteria.

Table 2. Sources used and the first amount of papers

Digital Library	Amount of papers	Percentage
ACM	146	11.97%
IEEExplore	79	6.48%
Scopus	164	13.44%
Citeseer	244	20.00%
ISI Web of Science	100	8.20%
EI Compendex	300	24.59%
Springer Link	244	20.00%
Google Scholar	150	12.30%
Total of papers	1220	100%
Repeated	525	43.03%
Set of papers to be analyzed	695	56.97%

Table **3** presents the results of each stage of the papers selection.

Table 3. Final amount of papers

Amount of papers	Selected based on Title and Abstract	Selected based on Full Text	Final amount of papers after the Quality Assessment
695	127	68	59
56.97%	10.41%	5.57%	4.83%

After collecting the information, we started a classification process. The first author suggested a classification for the papers selected, which was discussed with the other author. To increase internal validity, two researchers performed the classifications and then discussed differences to solve any possible disagreement.

[8] 59 papers were selected from this version of the Systematic Review and two of them were remaining papers from the previous version but were not collected in this version.

3.2.1 Quantitative Results

The findings of the quantitative analysis, as already mentioned, were divided in research-related information and content-related information.

Given the growing interest in agile methods and the concern with issues related to usability, it is interesting to note the number of articles published every year. This information is presented in Figure **6**.

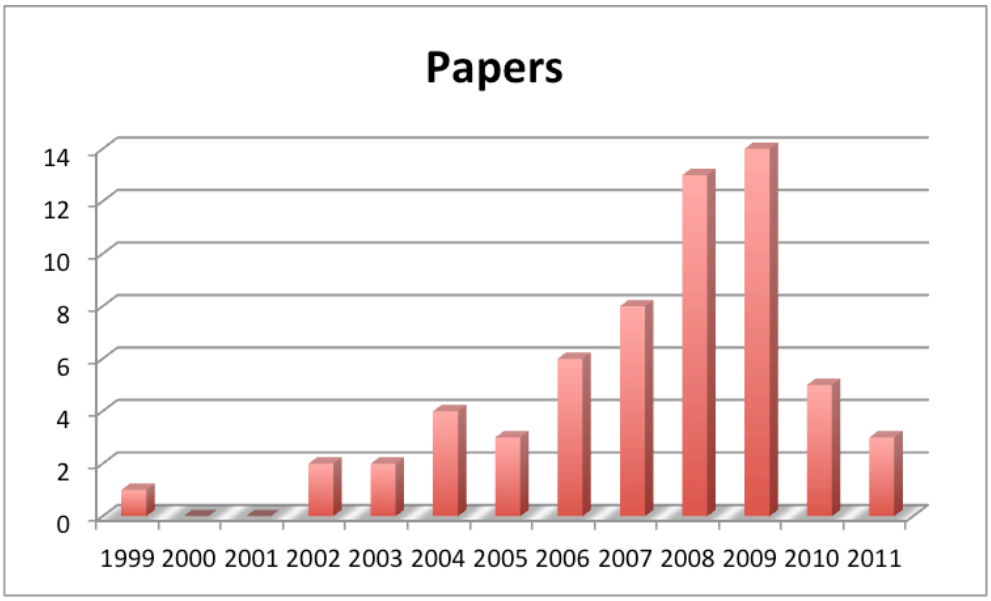

Figure 6. Papers by year

3.2.1.1 Descriptive Information

From the 61 selected papers, 25 were industry papers papers and 36 were research papers. As we can notice at Figure **7**, most of the papers are industry reports. We believe that it happens due to the number of UX Designers reporting their experience facing the challenge of adapting themselves to an Agile project or culture.

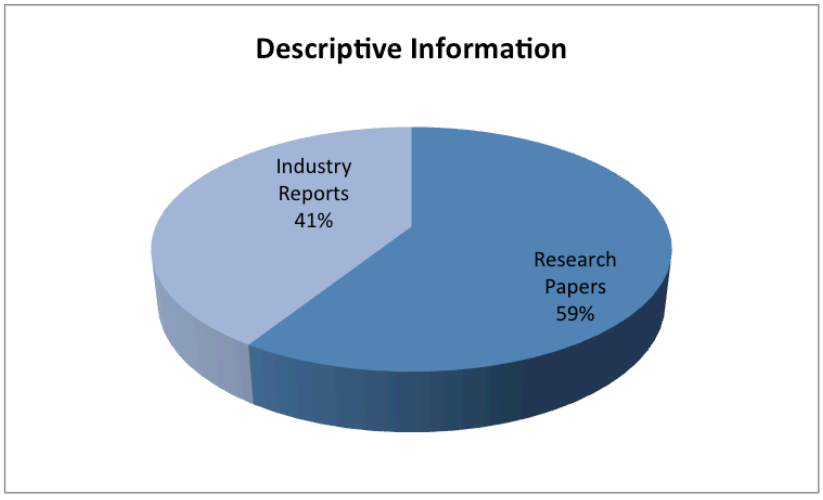

Figure 7. Descriptive information

3.2.1.2 Content-related Information

Concerning the content-related information, the papers were classified according to their focus, approach, circumstances, perspective, results and conclusions.

Regarding the focus, the papers were classified according to the activities of the Interaction Design process, as follows. As User Research if activities such as requirements analysis, interviews, contextual inquiries, and/or observations were considered in the integration of Interaction Design and Agile. They were classified as Design if the focus of the integration was also on activities such as prototyping, sketching, or storyboarding. Finally, the papers were classified as Evaluation if the focus was on the UI evaluation. As we can notice on Figure **8**, 45 studies were focused on User Research, 56 studies were focused on Design, and 41 studies were focused on Evaluation. It is worthwhile to mention that the classification is not mutually exclusive, 45 papers focused both on User Research and Design, and 31 papers focused on the three categories, User Research, Design and Evaluation.

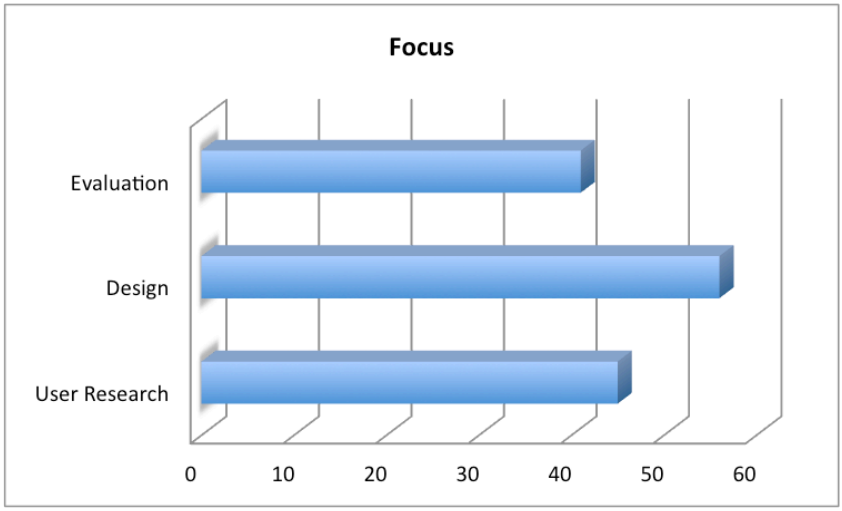

Figure 8. Content-related information: focus

Regarding the approach used, we classified the work as a Specialist, Generalist or Generalist/Specialist approach, as proposed by [FOX08]. A specialist approach means that the team used specialists for UCD work. A generalist has all team members fulfilling both roles. And a Generalist/Specialist is a hybrid approach in which some development team members fulfill both roles but not all. Figure **9** clearly indicates that most studies used specialists; 49 studies used the specialist approach and only one used the Generalist approach.

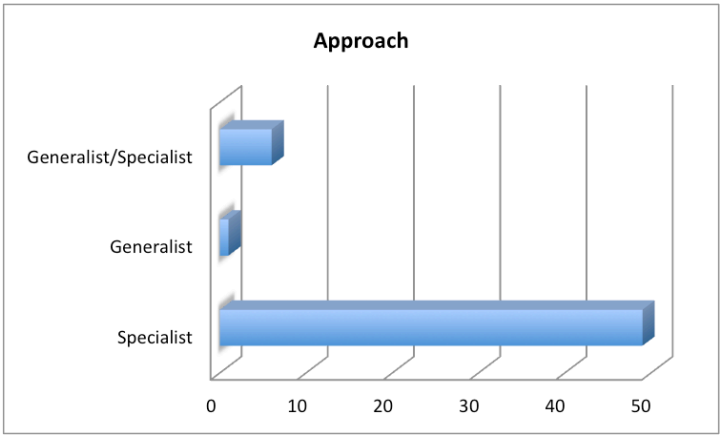

Figure 9. Content-related information: approach

We also classified the studies according to the circumstances under which they were conducted. Regarding this topic, the papers were classified as follows: in large teams, in small teams, in novel projects and in redesign projects. It is worthwhile to mention that this classification is also not mutually exclusive. If we could not identify the circumstances, we classified them as “Not Applicable/Don’t Know”. As we may notice at the Figure **10**, most of the papers do not mention under which circumstances the studies were performed, and 14 of them were performed in small teams – less than 10 members – and 18 of them were performed in novel projects.

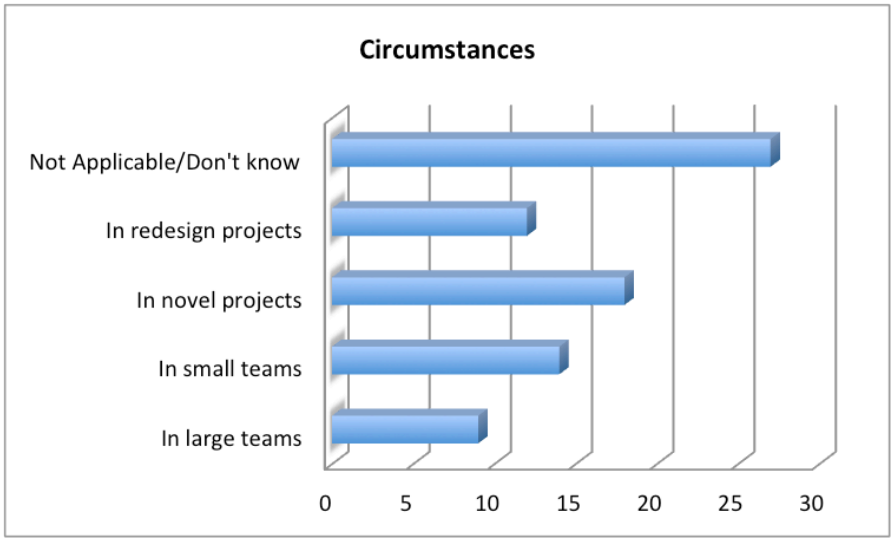

Figure 10. Content-related information: circumstances

Regarding the perspective from which the studies were described, we classified the papers as of: a UX Designer, a Developer, a BA (Business Analyst) person or an Academic perspective. If the paper did not mention from which perspective the study was described, it was classified into the category “Don’t know”.

As can be noticed in Figure **11**, 31 studies were described from the perspective of an Academic person, and 20 of them were described by a UX Designer. It is worthwhile to

mention that 4 papers were described from the perspective of a BA person and a UX Designer, and 2 papers were described from the point of view of a UX Designer and a Developer.

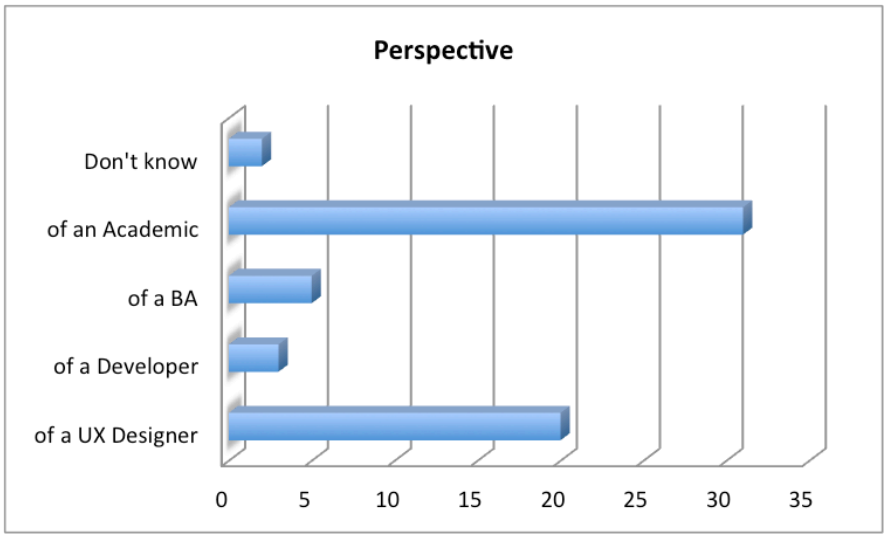

Figure 11. Content-related information: perspective

Concerning the results of each paper, they were classified into four categories: Need of Initiative (those papers that concluded that there is a need for a proposal to the integration of Agile and Interaction Design), Initiative Proposal (those that propose an integration of Interaction Design and Agile), Lessons Learned (those that present lessons learned from some experience with the integration) and Recommendations (papers that make recommendations based on previous experience or literature review). These results are presented in Figure **12**. We may observe that 2 studies are more notional presenting just a need of an initiative, while 12 studies proposed recommendations, 17 presented a initiative proposal of integration of Interaction Design and Agile and, 34 studies presented lessons learned from attempts of integration.

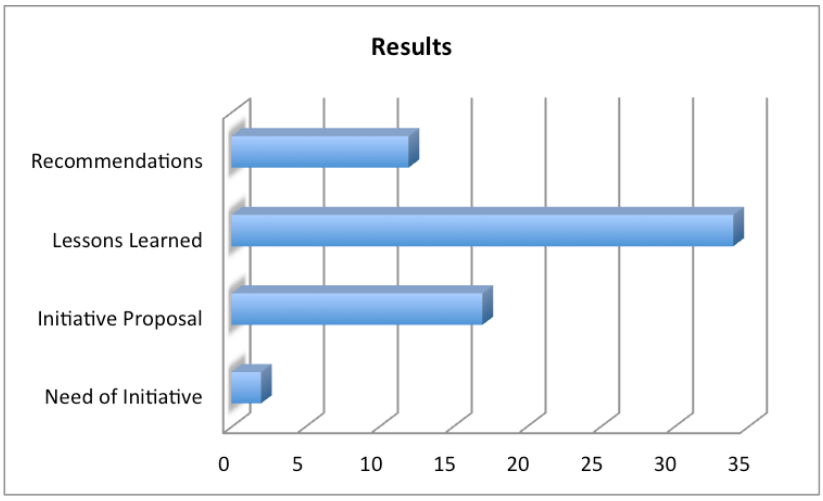

Figure 12. Content-related information: results

Some papers had multiple classifications. [HOD05] was classified as Lessons Learned and also Recommendations; [AMB06] was classified as the Need of an Initiative, Lessons Learned and Recommendations, and [FER07b] was classified as Initiative Proposal and Lessons Learned, for instance.

We also identified the most common practices, artifacts and needs in the papers analyzed. We used the term Conclusions to list these topics as follows:

- LDUF (SDUF): Little Design Up Front or Some Design Up Front, in other words, some work must be performed before the start of development, but sparingly.
- Close Collaboration: the indication that working with Agile improved collaboration and communication between UX teams and development teams. In other words, developers can better understand what designers are trying to say.
- Low Fidelity Prototypes: the use of low fidelity prototypes.
- Users Testing: the use of users testing for usability evaluation.
- User Stories: the use of User Stories by the UX team, creating them or enriching them.
- Inspection Methods: the use of usability inspection methods.
- One Sprint Ahead: the indication that the UX team must work at least one iteration ahead of the development team.
- Big Picture: the recommendation to not lose the holistic view of the project.
- Scenarios: the use of scenarios in the software development process.
- Personas: the use of personas.
- BDUF: Big Design Up Front, use plenty of time to research issues related to users before the start of development.
- Parallel Sprint: the indication that the UX team must work in parallel to the development team, in the same iteration.
- Interaction Models: the use of interaction models;
- Guidelines: the use of design guidelines.
- Essential Use Cases: the use of Essential Use Cases proposed by [CON99].

From 15 topics identified in the papers, only 2 are seen as problematic by the authors, in other words, topics that should be avoided. Teams losing sight of Big Picture of a project is sometimes perceived as a problem with agile methods generally, not just when

integrating Interaction Design and Agile. The authors comment when working in a piecemeal fashion as in any iterative approach, it is easy to lose the big picture of the project. BDUF is suggested by one paper [BEN10]. [PAT05] suggests collaborative modeling sessions and the use of Garret's elements of User Experience [GAR03] to the maintenance of the Big Picture. All other papers suggest LDUF because according to them, BDUF goes against agile principles.

The quantitative results of this classification are presented in Figure **13**.

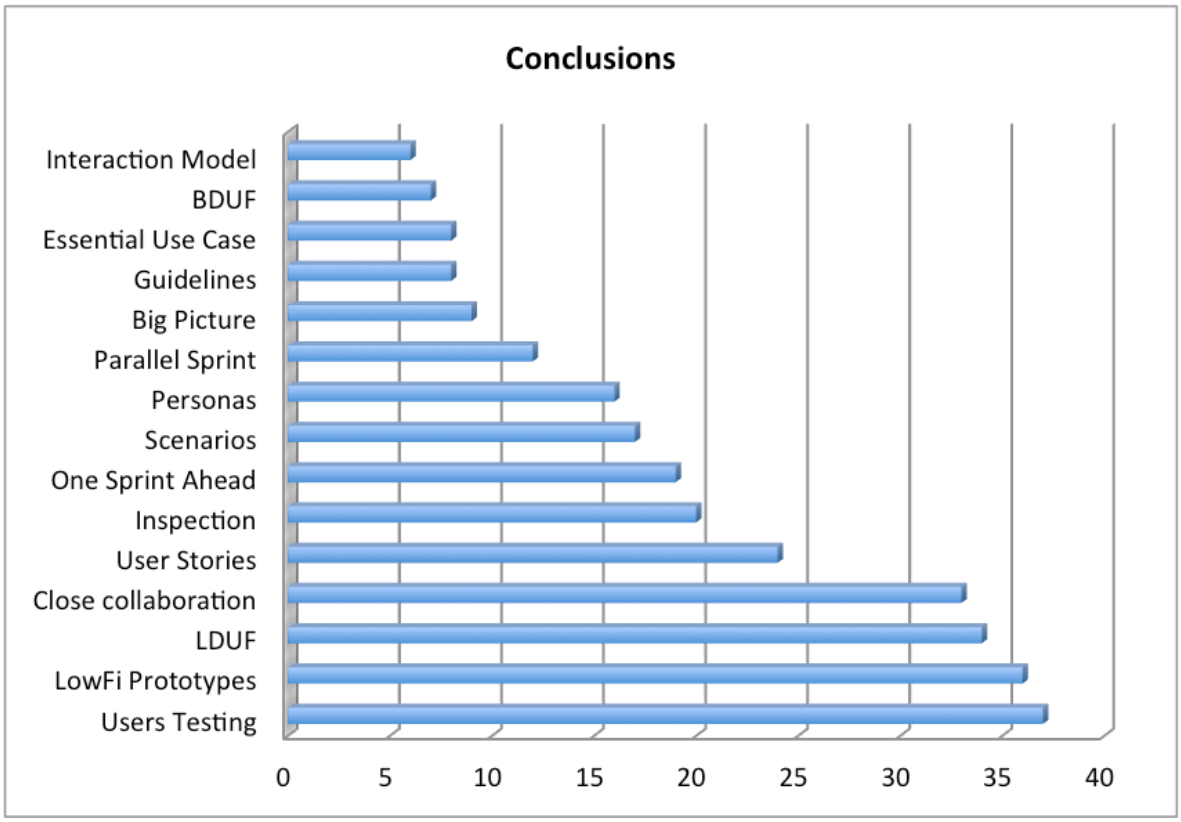

Figure 13. Content-related information: conclusions

3.2.2 Qualitative Analysis

At this point we are aiming at identifying some key aspects, as highlighted by the primary literature, concerning the integration of Interaction Design and Agile:

- Little Design Up Front
- Prototyping
- User stories
- User testing
- Inspection evaluation
- One sprint ahead

According to this classification, next subsections will present some qualitative data extracted from the final set of papers of the systematic review.

3.2.2.1 Little Design Up Front

Concerning Little Design Up Front, [JOK04], [CON02], [ADI09], [FER07], [KRO09], [ARI09], [SYD08], [LEE07], [UNG08], [CHO09] and [FER10] just mention that BDUF is not an option regarding agile methods, but they do not comment which artifacts or practices should be used. [DET07] and [WIL07] suggest that activities related to the UI design should be performed before the official kickoff of the project.

Additionally, [WIL07] suggest the use of story cards and that there are at least two roles in the UCDS (User Centered Design Specialist) team, a UCD Researcher and a UCD Prototyper. [AMB06] suggests doing some UI modeling up front by using modeling tools, which reflect agile practices, such as index cards, whiteboard sketches, or paper-based low fidelity prototypes, because according to him, these artifacts enable quick iteration when gathering user information. [HOD05], [KOL09], [CHA06], [FOX08], [NAJ08] and [HUD03] suggest the use of Sprint 0 for contextual inquiry and user interviews. [COA11] and [ARM04] suggest the use of Contextual Inquiry. Besides Contextual Inquiry, [PAT05] suggests the application of Contextual Observation. [CHA06] also suggests that this should be done before the Planning Game, then usability aspects can be discussed during the course of the Planning Game. [HOL05] suggest that usability aspects should be discussed during the Planning Game without previous discussions.

[NAJ08] suggest the creation of personas in Sprint 0. [HUD03], [HAK03], [HUS09] and [HUS09b] suggest the definition of personas as well. [WOL08] also suggest the creation of personas, but in this case Extreme Personas, which according to the authors would be an extension of XP’s User Stories. [MES06] advocates LDUF and reports the use of paper prototypes for early usability testing. Test results lead to new User Stories that are included in the Backlog for prioritization. In her U-Scrum, [SIN08] proposes the creation of a specific product owner for usability aspects and also User Stories that contemplate usability criteria.

[CHO09] mentions that less time should be spent creating high fidelity designs in isolation and focus more on problem definition and facilitating collaborative problem solving. Prior to kicking off the three-week Sprint cycle, the Interaction Designer prepare for the upcoming sprint by conducting a series of problem definition and framework development sessions. According to the author, this happens during the final week of the

previous Sprint. The Interaction Designer gathers a list of epics[9] and features that will require UI designs for the upcoming Sprint from product leads.

[CON02a] states that we draw diagrams only when we have to or when drawing them is faster than not drawing them and that the focus must be on user intentions and system responsibilities in order to help distinguish genuine needs of users from mere wants and wishes. Accordingly, [BEY10] advocates the use of Contextual Inquiry because what the users want is different than what the users need.

[FER07a] provide some of the main concepts that emerged in their interviews that relate to the issue of up-front interaction design, such as: do most, though not all, interaction design up front; much of the interaction design involves study of the clients and users; interaction design is informed by software implementation and; cost and time are the issues. They also suggest the design involve close work with business analysts, market clients, and end users.

[BEY04] proposes the integration of Rapid Contextual Design and Agile. The authors assume that customer representatives working with a team of at least two Interaction Designers will play the customer role. [FER07a] conducted semi-structured in-depth one-on-one interviews with interaction designers and developers and found out that up-front interaction design is commonplace in agile development, and indeed there is agreement that most interaction design be done up front.

3.2.2.2 Prototyping

Concerning prototyping, [KOL09], [KRO09] and [SYD08] comment that it is important to prototype. [COA11], [SOH10], [FOX08], [MES06], [HOL05], [DET07], [WIL07], [CHA06] and [MIL05] propose or mention that the prototyping stages occur at the initial stages of the development process. They also comment on the benefits of using prototypes regarding to the communication between developers and Interaction Designers, and the use of such prototypes for usability evaluations both by inspection and by user testing.

[CHA06] and [WIL07] suggest that the prototype evolves into a high-fidelity prototype. Accordingly, [HAK03] suggest that low fidelity prototypes should form interactive specifications. [HUS08] comment that prototypes can be derived from the User Stories, and [HAI07] suggests the construction of prototypes from personas created in his approach. All previous approaches, including [UNG08] and [BEN10] suggest that

[9] Sometimes large stories are called epics.

Interaction Designers teams should develop UI prototypes one sprint ahead of the development team. While [FED08] suggests that teams work in parallel. Although, [SYD07] reports that Interaction Designers teams should design one sprint ahead and evaluate one sprint behind. [BROW08] comments that sketches in addition to User Stories can be used as means of revealing errors, temporal information such task sequence, contextual information etc. [CON02a] and [AMB06] just mention the use of sketches while [PAE08], besides the sketches suggest the use of storyboards to represent user's workflows.

In [BRO08], UI prototypes are used to bring the known customer requests into the discussion as quickly as possible and to possibly serve as a template for the development. [HUS09] and [HUS08] and [HUD03] use low fidelity paper prototypes and high fidelity prototypes to perform inspection evaluations and usability tests with the customer.

3.2.2.3 User Testing

Concerning Users Testing, [HUD03], [HUS09], [MES06], [LEE07], [FOX08], [OBE08], [HUS08a] and [HOL05] mention or suggest to perform Users Testing on paper prototypes, however, [HOL05] recommends the use of Thinking Aloud whereas [FIS09] propose the use of RITE (Rapid Iterative Testing and Evaluation).

[OBE08] recommend the use of scenarios to guide the user testing. [WIL07] and [FED08] recommend the execution of users testing on interactive prototypes. All of them aimed at refining the UI prototype for the next iteration. [NAJ08] and [WOL08] recommend users testing whenever possible, but they do not comment whether the tests are performed on prototypes or on working software. Only [NAJ08] points out that user testing is performed with the customer.

[BEN10], [LEE07], [SOH10], [FER07], [DUC07], [DET07] and [KAN03] recommend user testing on the working software. Whereas [BEN10] and [LEE07] suggest to perform user testing to validate the UI, [BEN10] and [KAN03] comment that usability testing should be integrated into the acceptance tests. [CAR04] suggest the inclusion of usability criteria on story cards to perform acceptance and usability evaluation together. [AMB06] suggests that user testing encompasses both acceptance and usability testing. [DUC07] suggests that user testing should be performed during the Sprint Reviews and [DET07] recommends user testing with remote users at the end of the release, because he considers code generated within an iteration too unstable to perform user testing.

[MIL05] reports that they conduct usability tests on low-fi and high-fi prototypes and [ILL09] mention that usability tests can be conducted, but in a lightweight form and not

inside a usability laboratory. [BEY04] suggests that the UI should be tested with the users using paper, with mock-ups and interviews because User Stories are fairly fine-grained definition of system functionality and they can be covered in a single paper prototype test. They also suggest tests with a more detailed UI if there was time and resources.

3.2.2.4 User Stories

Concerning User Stories, [BAR09] and [LEE07] comment that User Stories should be originated from Usability Scenarios, while [SOH10] suggest that User Stories should be integrated with scenario-based design.

[JOK04] commented that activities such as Task Analysis of users should contribute to the development of User Stories. Whereas [MES06] suggest that User Stories should be originated from usability tests on paper prototypes. [HUS08a] and [FOX08] comment that User Stories could be defined for the construction of prototypes. [HOL05] also comment that User Stories could be used as tasks to be performed by users on user testing using these prototypes.

[BRO08] reports the integration of prototypes and User Stories. [DUC07] comments that Product Backlog and User Stories are the best places to capture usability requirements, while [SIN08] mentions that User Stories should contain the usability issues in their acceptance criteria. [BEY04] suggest that UI mockups should be part of the User Story definition and acceptance testing criteria. As already mentioned, [CAR04] recommend that usability criteria should be on story cards, then acceptance and usability evaluation could be performed together.

[BUD09] suggests the existence of a specific product owner for usability issues, and they also suggest a specific product backlog for usability aspects. [WIL07] considers if you have a backlog containing detailed UI specifications it would be a waste of time, because you could end up specifying something that will not be implemented. [FER07] suggests that User Stories should always be fed with the results of user tests performed at the end of each sprint.

[HUS08a] mentions that User Studies should be used to develop User Stories and that Interaction Designer should be trained in XP-story writing to be able to deliver User Stories in a technical-aware manner, giving report in the form of checkpoints which then be converted into user stories quickly instead of a big report of a formal usability test.

3.2.2.5 Usability Inspection

[CON02a], [HUD03], [HUS09], [WIL07], [FOX08], [HUS08a], [UNG08] and [MIL05] suggest or mention the use of usability evaluation on paper prototypes, always with the goal of refining the UI for the next iteration. Besides that, [HUS09] propose a testing framework integrating HCI instruments into XP, presented in Figure **14**.

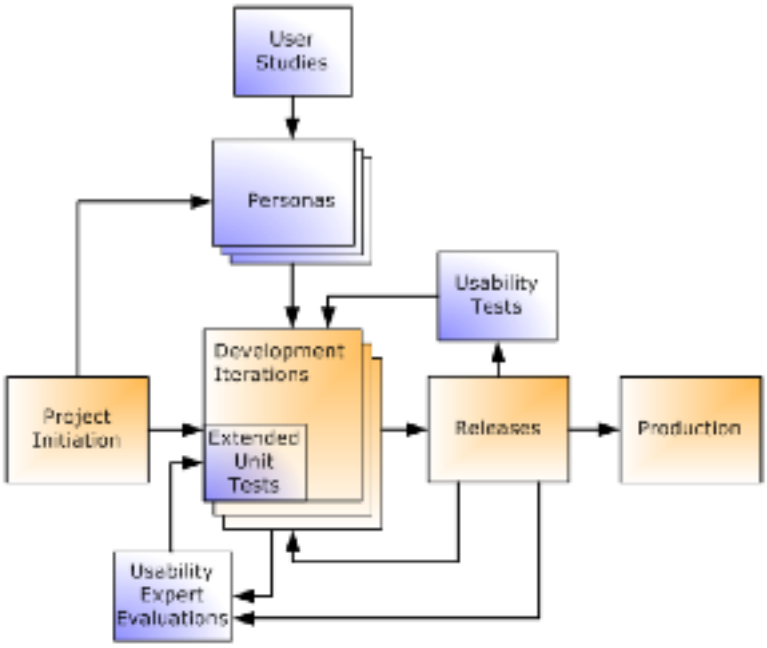

Figure 14. Testing framework proposed by [HUS09]

[OBE08] also suggests an evaluation of paper prototypes, but guided by scenarios. [FED08] suggest inspection evaluations on prototypes, but focusing on interactive prototypes, instead of on paper prototypes.

[DET07], [WOL08], [SYD08], [NAJ08] and [BEN10] suggest evaluations on UIs already implemented aiming at their validation. [BEN10] and [KAN03] suggest the use of Heuristic Evaluation, and [DUC07] comments that Sprint Reviews are good opportunities to conduct usability evaluations. [HUS08a] execute inspection evaluations on low-fi and high-fi prototypes to write UI related stories.

Finally, [ALB10] reports that developers did UI reviews, and that UI reviews had completely changed the way developers saw the Interaction Designer's work. Seeing the work of others from the perspective of somebody who does not care how simple and professional the code is, but rather what has being used by people, seemed to have a profound impact on developers.

3.2.2.6 One Sprint Ahead

Concerning One Sprint Ahead, [CHA06], [NAJ08], [SYD08], [WIL07] and [UNG08] suggest that Interaction Designers teams work one sprint ahead of the development team. [CHA06], [NAJ08] and [SYD08] also suggest that this practice has already started in Sprint 0. But [BUD09], with their approach of Product Owner, Product Backlog and User Stories specifics for UI, suggest that the entire UCDSs team work at least one or two sprints ahead.

[ILL09] suggests that Interaction Designers have to work two or three iterations ahead of the rest of the team while paying close attention to the current iteration and the opportunities to include research findings effectively. [CHO09a] commented that user experience is part of the business strategy, it needs to be aligned with the business and product owner team. Still according to the authors, Interaction Designers need to understand business objectives and should be able to compromise user experience objectives and this enables the team to agree on prioritization tactics and success metrics. Thus, Interaction Designers should be aligned with the business strategies, participating even before any iteration.

[SYD07] suggests that the Interaction Designers team should work one sprint ahead of the development team in terms of designing, but they should work one sprint behind in terms of evaluating the code implemented in the previous sprint, as presented in Figure **15**.

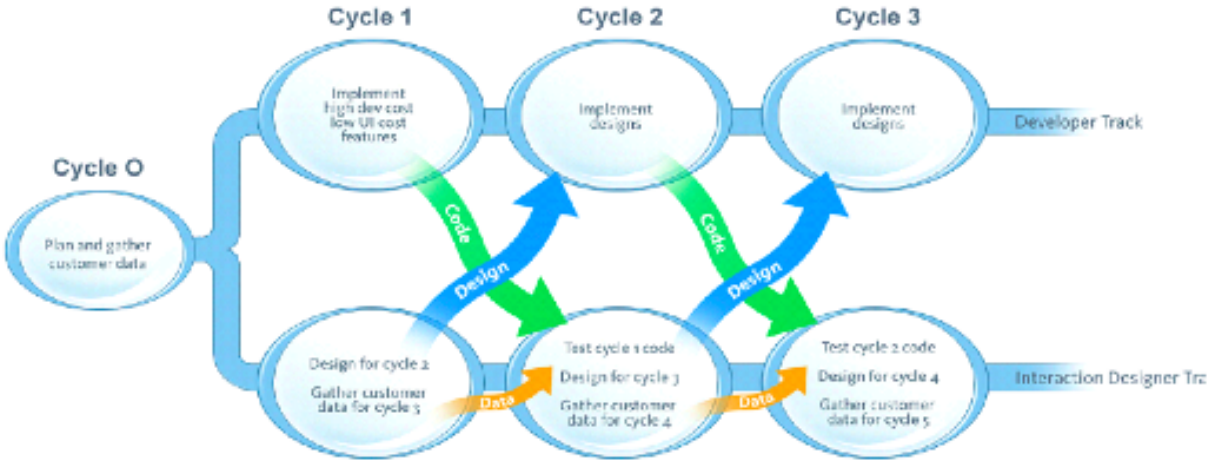

Figure 15. Workflow proposed by [SYD07]

3.2.3 Discussion

This systematic review has a number of implications for research and practice. For research, the review shows a clear need for more empirical and/or experimental studies regarding Interaction Design and Agile Methods. As we could notice, 69% of the studies retrieved in this systematic review were industry reports. Another important point is that is more common to have an Interaction Designer (or in some cases, a team) directly involved in the project.

The systematic review has identified recurring themes and patterns of the most common activities and artifacts used by teams integrating agile methods and Interaction Design.

We identified 1220 studies by searching 8 digital libraries and hand-searching conference proceedings, of which 61 were found to be research studies or industry reports of acceptable rigor and relevance to our study.

The studies were classified regarding their content and research method. Regarding content, they were classified considering their approach, results, focus, circumstances and perspective.

The studies were also classified considering the practices and artifacts used by the teams, such as: LDUF, use of personas, use of low-fi prototypes, use of inspection methods, use of user testing, use of scenarios, use of Use Stories, use of guidelines, if the Interaction Designer team work one sprint ahead of the development team or they work in parallel etc.

These issues were used as the basis for a proposal of a framework combining Interaction Design and Agile principles.

At least, two conclusions can be drawn from the quantitative and qualitative analysis in this systematic review.

Conclusion 1: The focus of integrating agile methods and Interaction Design should be on user research, on design, as well as on usability evaluation. Regarding user research, most of the studies suggest just 'enough' user research up-front. For design, most of the times low fidelity prototypes are used. Regarding evaluation, low-fi prototypes are often tested aiming at improving design.

Conclusion 2: Although there is a reasonable number of papers on the integration of Interaction Design and Agile, none of them is verified. Evidence exists in form of lessons learned and experience reports. Further empirical research is needed.

If we related our findings, we can answer our research questions proposed in the review protocol, as follows:

Q1: How are usability issues addressed into Agile projects?

There are addressed in various ways. For example, approaches with Interaction Designers in the development team, approaches without UCD specialists in the development team etc.

Q2: What are the most common practices to address usability issues in Agile methods?

We believe they are presented in the previous section, which refers to the conclusions of the papers.

According to all the experience reports identified and according to [BEY10], Agile development and user-centered design are a natural fit. Agile development assumes a close connection to users, and user-centered design assumes rapidly iterating designs with users.

We identified the following as the main limitations of this systematic review.

The reliability of the method to classify the papers, because we did not use an already established classification, proposing a new categorization.

Another issue already commented is that Agile Methods and Interaction Design are not standardized and our choice of keywords and string searches could missed relevant studies. For example, it is possible that Generalist UI practitioner reports may have been missed, as these authors may have used specific technique keywords like paper prototyping rather than the generic UCD keywords we utilized in our searches.

Despite the limitations, we believe the results were satisfactory regarding identification of the state of the art of the area as well as providing a good theoretical basis concerning common practices used in this area.

A framework proposal and the used artifacts derived from the findings of this systematic review are presented at Chapter 4.

4. PROPOSED FRAMEWORK

[JOK04] and [SOH10] commented that Interaction Design and agile methods fit well, and that the challenge is not to make Agile less agile, but in adapting the methods of UCD so they can be "light" and efficient at the same time.

[HUS08] pointed out some beneficial similarities between UCD and Agile, e.g., having the client on-site, continued testing and iterative development. Moreover, as noted by [WIL07], the two methods have much to offer when they share iterations, because the iterations of Agile facilitate usability testing, enable developers to incorporate results of these tests in subsequent iterations. However, [CON02] commented that improve the usability of a product does not come without costs or risks, even when the methods are rationalized.

In order to integrate Agile and Interaction Design, trying to minimize these costs and risks, we propose a framework with usability artifacts and practices in a condensed form, as suggested by the agile principles, trying to positively impacts on the improvement of the products usability developed and at the same time trying to minimally impact on the activities of agile development.

The framework structure is similar to the processes described by [SYD07], [FOX08], and [FER07]. The difference is a combination of the most common practices and processes identified in the systematic review. This framework derived from the findings of the systematic literature review.

We know the characteristics of flexibility and adaptability of agile methods, so the intent of this proposal is not to tense these methods, but, as mentioned by [SOH10], adapting usability practices to improve the quality of use of products developed using these methods. [SYD07] says that to achieve such integration, we should pay attention to three key aspects: timing, granularity and reporting.

Timing is related to choosing the best time to perform some activity, for example, the reduced time that we have to design up front. According to [JOK04], Interaction Design teams need to adapt their working speed to the Agile teams.

Granularity is related to the extent to which something is divided into small pieces, for example, the extension of a usability problem, if there is time for resolution in the current iteration or not.

Reporting is related both to report problems and to present designs. This information is only useful for agile teams if they are on the product, and not "lost" in the 200-page Word documents with details, as quoted by [WIL07].

The proposal will be discussed according to these three key aspects for now.

4.1 Timing

Regarding Timing, and specifically to LDUF previously mentioned, it is proposed the adoption of Sprint 0, in other words, one iteration takes place before the start of implementation, or even before the official kickoff of the project. It is important to share this stage with Interaction Designers due to their skills in gathering and analyzing data from UX because, according to [PEI09], developers tend to listen to what customers want instead of looking at what they do.

In this iteration, there should be activities such as context research (Contextual Inquiry), Observations, Task Analysis and Interviews. Such activities should result in paper prototypes and/or design cards that should help to define the User Stories.

With User Stories defined, Interaction Design team should then design it and validate such designs, and thus add usability aspects as acceptance criteria for the User Stories. The Interaction Design team should deliver these Feature Cards to the development team. These activities should be done before the panning game, so they can be discussed during it.

To help also with LDUF, activities related to requirements gathering and analysis must be distributed throughout the process, avoiding concentration of these only in the beginning of the project. This leads to just-in-time design, in other words, as the agile methods themselves, the Interaction Designers should focus on a few tasks at a time, without the need to design the entire release earlier.

Thus, contextual investigation, task analysis and interviews, as well as prototyping and validation of them could be performed in each iteration. This makes the Interaction Design team work one sprint ahead of the development team regarding user research and design.

4.2 Granularity

Concerning Granularity, the Interaction Design team should work closely with the development team to support them in terms of designing and conducting inspection evaluations on the implementation of the current sprint and provide feedback, without blocking the development team. However, the Interaction Design team must analyze the problem identified and determine if there is time to fix it in the same sprint or if it will be reported to be fixed only in the next sprint, feeding the User Stories.

One option for defining the problem, and whether this will be fixed in the current iteration or only in the next one, is the use of complexity points per task as used by Agile teams to perform some estimates.

Regarding evaluation, it is proposed to evaluate the entire implementations one sprint behind. Because evaluating the implementation of the current sprint could make the development team to deliver before the end of the sprint. It would reduce staff time for development and let them idle while the Interaction Design team conducts evaluations.

It is suggested the implementation of inspection evaluations regularly. Performing user testing during the sprints is hard, given the time spent on preparation, scheduling, conducting and analysis concerning this type of test. So, it is suggested to achieve these, at least, before the delivery of the release. It is worthwhile to mention that these tests should be conducted with real users. An alternative is the inclusion of activities related to data capture and usability when performing acceptance tests.

4.3 Reporting

Prototyping, Design Cards, Issue Cards[10] and Feature Cards could be used to report both evaluations results and designs.

A lean way of reporting such issues is very important in the Agile context. Using simple artifacts helps in communication between all stakeholders, adding value to the development process. It is suggested also the use of a User Experience Board for maintaining a shared vision, if possible.

We can use prototypes and/or Design Cards for communication among UCDSs team members. When delivering designs to the development team, you can also make use of prototypes and Feature Cards.

To report problems and/or modifications can be used in daily meetings Oral Storytelling and Issue Cards then the development team can incorporate the design improvements.

It is worthwhile to mention that our intention is not turn the process rigid, we are trying to provide a set of tools and artifacts for that this integration can be adapted for the reality of each company.

[10] Physical cards used to communicate information from observations and interviews with users.

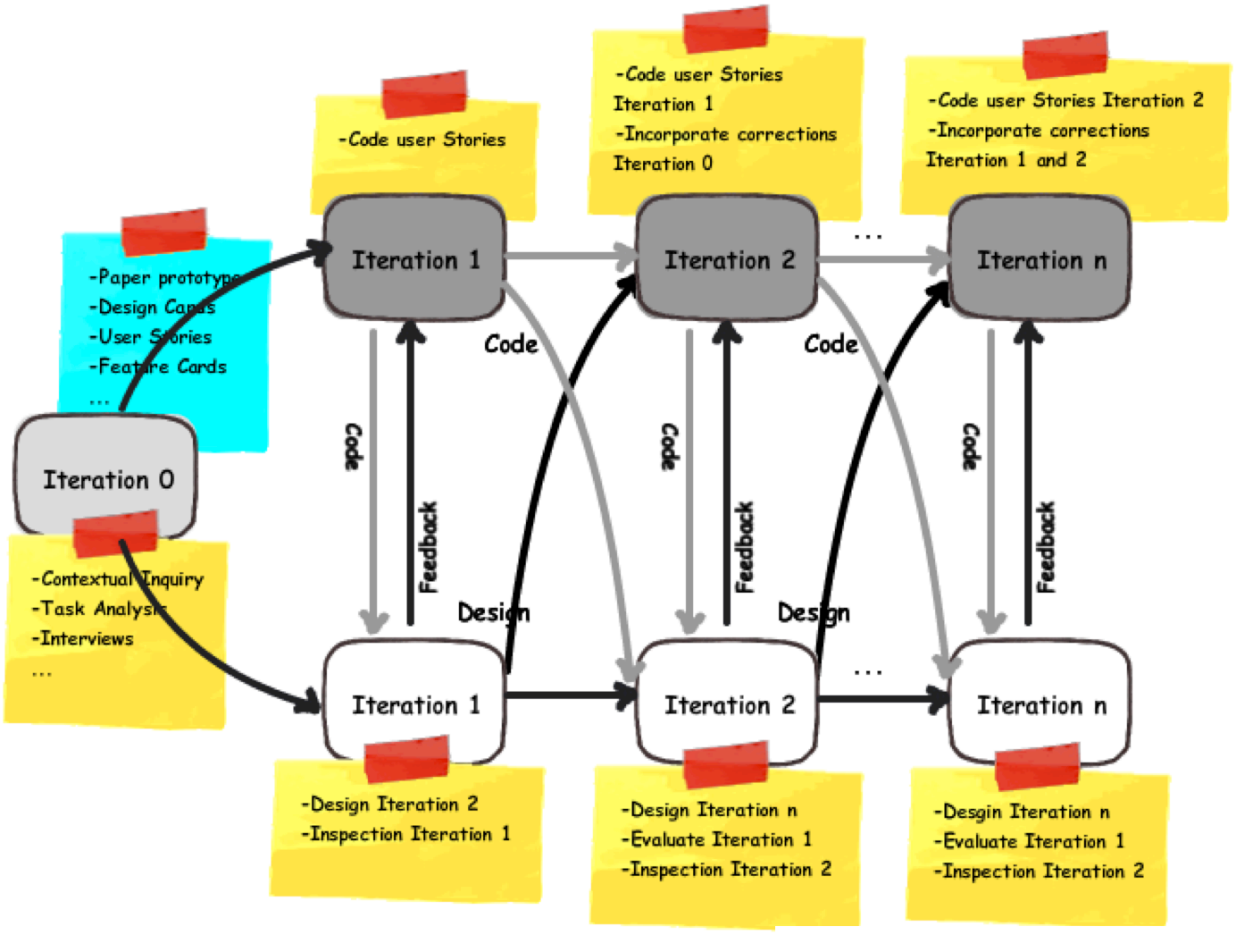

Figure 16. Proposed framework

Following, we present a more detailed description of the proposed framework presented at a high level at Figure **16**:

- During the *Sprint 0*, Interaction Design and Development Team could perform the following activities: Contextual Inquiry, Task Analysis and Interviews using these Artifacts: Paper prototypes, Design Cards, User Stories with acceptance criteria with usability issues and Feature Cards.
- During the *Sprint 1*, Interaction Designers could Design by performing Contextual Inquiry, Task Analysis, Interviews for Sprint 2 and Evaluate by performing Inspection Evaluation on the code of the current Sprint and provide feedback still in this Sprint. And using the following Artifacts: Oral Storytelling for the feedback in the current sprint; Prototypes, Design Cards and User Stories for Sprint 2. While the Development team could Code the User Stories designed in Sprint 0.
- During the *Sprint 2*, Interaction Designers could Design by performing Contextual Inquiry, Task Analysis, Interviews for Sprint n and Evaluate by

using Inspection Evaluation on the code of the current Sprint providing feedback still in this Sprint and perform Inspection Evaluation and User Testing on the Sprint 0 design that was coded in Sprint 1. Also using the following Artifacts: Oral Storytelling for the feedback in the current sprint and Issue Cards to report problems of the code implemented in Sprint 1 (designed in Sprint 0). The Development team could Code User Stories designed in Sprint 1 and Incorporate corrections reported from Interaction Designers on what was coded in Sprint 1 (designed in Sprint 0).

- During the *Sprint n*, the Interaction Design could Evaluate by performing Inspection Evaluation on the code of the current sprint providing feedback still in this Sprint; perform Inspection Evaluation and User Testing on the code of Sprint n-1 (designed on Sprint n-2) and perform Inspection Evaluation and User Testing on the code of Sprint n (designed on Sprint n-1). They could use the following Artifacts as already mentioned: Oral Storytelling for the feedback in the current Sprint; Prototype, Design Cards, User Stories for Sprint n; Issue Cards to report problems of the code implemented in Sprint n-1 (designed in Sprint n-2) to be incorporated in Sprint n; Issue Cards to report problems on the code implemented in Sprint n (designed in Sprint n-1) to be incorporated before the release. While the Development team could Code User Stories designed in Sprint n-1 and Incorporate corrections reported from the UCDSs about what was coded in Sprint n-1 (designed in Sprint n-2).

A very important point is to maintain the Big Picture, which is difficult given the characteristic of iterative and piecemeal development in agile projects.

In order to maintain the Big Picture and to stimulate this collaboration [BEY10] suggest the sharing of documents, artifacts, and especially of knowledge between the teams. The use of prototypes, Design Cards for stand-up meetings and the use of Issue Cards to report usability issues would be good choices.

We believe that the main issues are those not addressed but the papers in the review, for example the Little Design Up Front. Despite a lot of papers mention that just Some or Little design up front is necessary, they do not have a conclusion about which techniques or artifacts to use in this design up front.

Another important point is how to communicate your design decisions to the stakeholders. Most of the papers address how to improve the communication between the

UCD team and the development team but do not how to improve the communication with the stakeholders of a project.

5. PROPOSAL'S ANALYSIS

According to [JOH11], Quantitative Research uses numbers to reach a state of knowing. Controlled environments are created to isolate one particular aspect of reality. The questions are stated up front, and only data related to the research questions are observed and recorded.

[JOH11] states that Qualitative Research, on the other hand, uses systematic observations to reach understanding. Researchers take the world as they find it instead of trying to manipulate conditions to isolate variables. The questions are more open ended and less defined, with plenty of room to collect a variety of data through collateral observations.

According to [AVI99], when used appropriately, Qualitative approaches have been accepted as equal in value to Quantitative approaches, and a particular strength of qualitative methods is their value in explaining what goes on in organizations. This statement led us to use a qualitative research method.

There are different types of qualitative research, such as: grounded theory, ethnography, life stories, conversational analysis, action research - which will carried out in this study - and so on [STR90].

According to [BAS99], Action Research (AR) produces highly relevant research results, because it is grounded in practical action, aimed at solving an immediate problem situation while carefully informing theory. According to the same author, Action Research has been described as a technique characterized by intervention experiments that operate on problems or questions perceived by practitioners within a particular context.

Action Research is one of the few valid research approaches that we can legitimately employ to study the effects to specific alterations in systems development methodologies in human organizations [BAS99].

[AVI01] state that unlike the case study researcher, who seeks to study organizational phenomena but not to change them, the action researcher is concerned to create organizational change and simultaneously to study the process.

According to [BAS99], the collaborative structure of an Action Research diminishes the researcher's control of the process and the outcomes of the research. The qualitative and interpretive foundations lead to a lack of generally agreed criteria for evaluating Action Research, and these are some of the limitations of this method.

Despite these limitations, [BAS99] states that Action Research responds directly to the pronounced needs for relevance in information systems research, and provides a

rewarding experience for researchers who want to work closely with the practitioner community.

According to [DAV04], Action Research as a method of inquiry is founded on the assumption that theory and practice can be closely integrated by learning from the results of interventions that are planned after a thorough diagnosis of the problem context. In Action Research, the researcher wants to try out a theory with practitioners in real situations, gain feedback from this experience, modify the theory as a result of this feedback, and try it again [DAV04]. [AVI99] state: 'In Action Research, the emphasis is more on what practitioners do than on what they say they do.'

Besides, there is the Cooperative Method Development (CMD) method [DIT05], which is understood as a domain-specific adaptation of Action Research and emerged from the discontent of some researchers with how existing research approaches both in software engineering and information systems addressed use-oriented software development. According to [DIT08], the existing approaches did not address the following questions: 'How do software development practitioners tackle their everyday work, especially the cooperation with users around the design of software?', and 'How can methods, processes and tools be improved to address the problems experienced by practitioners?'.

According to [DIT08], CMD combines qualitative social science fieldwork, within problem-oriented method, technique and process improvement. The action research based approach focusing on shop floor software development practices allows an understanding of how contextual contingencies influence the deployment and applicability of methods, processes and techniques.

[DIT05] state that CMD takes the existing practice of software development in concrete industrial settings as a starting point. This enables the researchers to address the actual problems that had been encountered, and it allows them to generalize technical and methodological recommendations that are rooted in successful practices. The research is implemented as evolutionary cycles consisting of qualitative empirical research, technical and methodological innovation in co-operation with the involved practitioners and implementation of these innovations evaluated by accompanying empirical research. As already mentioned, CMD is a domain-specific adaptation of action research, consisting of three phases where they can be applied repeatedly in an evolutionary cycle, as presented in Figure **17**.

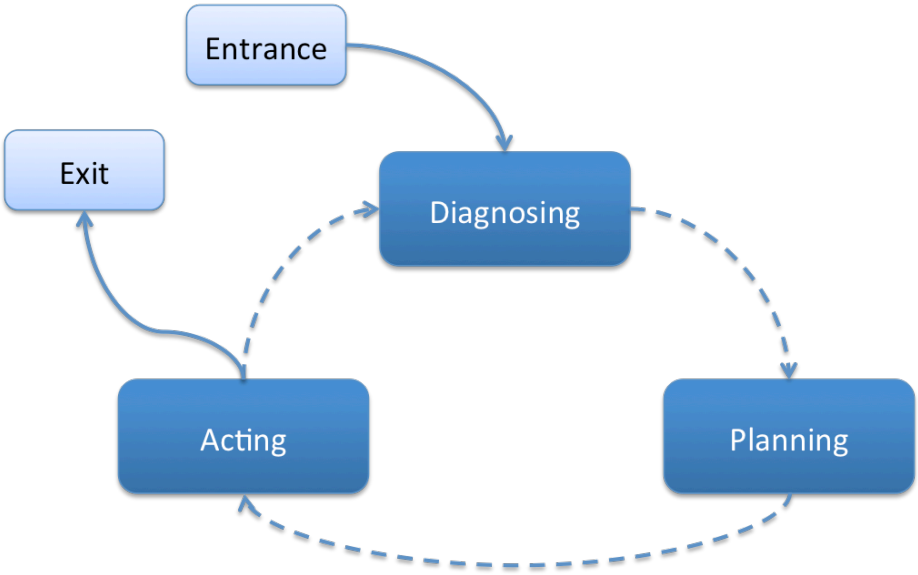

Figure 17. CMD process (adapted from the Action Research proposed by [DAV04])

In the **Diagnosing** phase, also called Phase 1 by [DIT05], the research starts with qualitative empirical investigations into the problem domain. The empirical research aims at understanding and explaining the practices and designs from a practitioner's point of view. The intention is to understand existing practices out of their historical and situation context, and to identify problematic aspects.

In the **Planning** phase, or Phase 2, according to [DIT05], the results from the first phase are then used as an input for the identification of problematic aspects of the situation at hand and the design of possible improvements. This is done in co-operation between researchers and practitioners involved. The result of this phase is the design of measures that can be expected to improve the situation at hand and address some of the problems identified together.

The improvements will be implemented in the **Acting** phase, or Phase 3, by [DIT05]. The researchers will accompany these method improvements as participatory observers. The results are evaluated together with the practitioners involved. The result of this evaluation will both summarize concrete results for companies involved and build the base for scientific evaluation of the proposed improvement measures for the researchers involved.

[DIT08] mention that in their projects, they used participant observation to develop a basic understanding of the organization, but often they complemented their field material with semi-structured interviews, document analysis, or additional workshops if it is necessary. Research publications are then based on an analysis in a grounded theory fashion.

In the next sections, we present in details the three stages of the research methodology.

5.1 Diagnosing

In the following sections we will provide a summary of the techniques for data collection and analysis that were used in our studies.

5.1.1 Data collection procedures

According to [LET05], when conducting field studies it is important to obtain accurate and reliable information about the phenomenon under study. These authors also state that there is important to use multiple data collection methods in order to learn about different aspects of a phenomenon.

In our studies we used interviews and observations. There are considered as first-degree techniques because they require the direct[11] access to a participant population, and they will be detailed in the following subsections.

5.1.1.1 Observations

According to [LET05], observational first-degree techniques provide a real-time portrayal of the studied phenomena. However, it is more difficult to analyze the data, both because it is dense and because it requires considerable knowledge to interpret correctly.

Observation occurs when the researcher observes participants engaged in their work, or specific experiment-related tasks, such as meetings or programming. The researcher can observe many participants at one time or can shadow[12] only one participant at a time. The main advantages from shadowing and observation are that they are easy to implement, give fast results and require no special equipment.

[LET05] state that the first-degree contact generally involves videotape, audiotape or manual record keeping. Videotape captures the most complete record, while manual record keeping captures the least complete record. On the other hand, videotaping invokes the greatest amount of interference in the work environment, while manual record keeping invokes the least amount of interference. Besides that, videotape is the most time-

[11] Direct methods means that the researcher is in direct contact with the subjects and collect data in real time [Run09].

[12] It is a specific technique of observation.

intensive data to use and interpret, while manual record keeping is the least time-intensive data to use and interpret.

Despite the fact that manual record keeping is the most data sparse method and hence captures the least complete data record, manual record keeping is the quickest, easiest, and least expensive method to implement.

Some authors suggest that the researcher should be well trained to identify certain behaviors, thoughts or concepts during the collection process. Aiming to satisfy these request, the researcher followed some recommendations according to [EME11].

5.1.1.2 Interviews

According to [LET05], interviews are also considered as inquisitive first-degree techniques that allow the researcher to obtain a general understanding of the software engineering process. Such techniques are probably the only way to gauge how enjoyable or motivating certain tools are to use or certain activities to perform. However, they are often subjective, and additionally do not allow for accurate time measurements.

We choose to conduct semi-structured interviews. According to [RUN09], in semi-structured interviews, questions are planned, but they are not necessarily asked in the same order as they were listed. The development of the conversation in the interview decide in which order the different questions are handled, and the researcher can use the list of questions to be certain that all questions are covered.

According to [LET05], interviews are highly interactive; in interviews researchers can clarify questions for respondents and probe unexpected responses; and interviewers can also build rapport with a respondent to improve the quality of responses.

[HOV05] suggest that special care must be taken. In this situation it is important that to the interviewee is ensured confidentiality and that the interviewee trusts the interviewer when the interview contains personal and sensitive questions, e.g. concerning opinions about colleagues, why things went wrong, or questions related to the interviewees own competence. In our studies, the interviews were not recorded following the requests of some interviewees. According to [LET05], if the data from interviews do not consist of audio or videotapes, careful note-taking may often be an adequate substitute for audio or video recording. During the interviews, notes were carefully taken instead of audio or video recording, also following the recommendations by [EME11].

5.1.2 Analysis procedures

According to [LET05], field studies techniques produce enormous amounts of data and the purpose of this data is to provide insight into the phenomenon being studied. To reach this goal, this data must be reduced to a comprehensible format. Traditionally, this is done through a process of coding, in other words, developing a scheme to categorize the data using the goals of the research as a guide.

Once the data is categorized, it can be subject to a quantitative or qualitative analysis. Qualitative analyzes provide a general characterization based on the researchers' coding schemes, instead of quantitative analyzes which rely on quantitative measures to describe the data.

In summary, the way the data is coded will affect its interpretation and the possible courses for its evaluation. Therefore it is important to ensure that coding schemes reflect the research goals.

[DAV04] state that triangulation of data from different sources (interviewing multiple participants) and different approaches (quantitative and qualitative methods) can help to address validity concerns. To satisfy this need, more than one data source was used. Observations and interviews were performed in different projects with people with different roles in the company.

According to [RUN09], analysis of qualitative data is conducted in a series of steps [ROB11]. First the data is coded, which means that parts of the text can be given a code representing a certain theme, area etc. One code is usually assigned to many pieces of text, and to one piece of text can be assigned more than one code. Codes can form a hierarchy of codes and sub-codes. The coded material can be combined with comments and reflections by the researcher, e.g. memos. When this has been done, the researcher can go through the material to identify a first set of hypotheses.

This can, for example, be phrases that are similar in different parts of the material, patterns in data, differences between sub-groups or subjects etc. The identified hypotheses can then be used when further data collection is conducted in the field, e.g. resulting in an iterative approach where data collection and analysis is conducted in parallel as described above. During the iterative process a small set of generalizations can be formulated, eventually resulting in a formalized body of knowledge, which is the final result of the research attempt.

According to [EME11], qualitative analytic coding can proceeds in two different phases. In open coding the researcher reads fieldnotes line-by-line to identify and formulate any and all ideas, themes, or issues they suggest, no matter how varied and

disparate. In focused coding the researcher subjects fieldnotes to fine-grained, line-by-line analysis on the basis of topics that have been identified as of particular interest. Here, the researcher uses a smaller set of promising ideas and categories to provide the major topic and themes.

5.2 Planning

Elements and categories that have emerged from the interpretive processes described in the previous chapter suggest key areas or aspects of the situation that need to be dealt with in any plan for taking action.

According to [STR07], at this phase, we need to work creatively to formulate actions that lead to a resolution of the problem(s). What can we do, we should ask, that will enable us to achieve better results, or a more positive outcome? What steps can we take to ensure that we accomplish the outcomes we desire?

[STR07] states that in the planning phase, research facilitators meet with major stakeholders to devise actions to be taken. As stakeholders devise a course of action that "makes sense" to them and engage in activities that they see as purposeful and productive, they are likely to invest considerable time and energy in research activities, developing a sense of ownership that maximizes the likelihood of success.

Still according to [STR07], often there are multiple related issues or a number of subsidiary issues requiring action, so participants will need to make decisions about the issue on which they will first focus and some order or priority for other issues. To accomplish this, participants should:

- Identify the major issue(s) on which their investigation focused
- Review other concerns and issues and that emerged from their analysis
- Organize issues in order of importance
- Rate the issues according to degree of difficulty (it is often best to commence with activities that ate likely to be successful)
- Choose the issue(s) they will work on first
- Rank the rest in order or priority

Participants then plan a series of steps that will enable them to achieve a resolution of the issues investigated. Each issue is first restated as a goal. Teams of relevant stakeholders should develop a plan for each issue and bring them to plenary sessions for discussion, modification and endorsement.

5.3 Acting

Following the CMD [DIT05] approach, this phase comprehends Implementing and Evaluating the results of the study.

Regarding the implementation, [STR07] says that collaborative processes often start with a flourish. Much enthusiasm and energy are generated as plans are articulated and people set off to perform their designated tasks. The best of intentions, however, often run up against the realities of the everyday life. Participants in the research process reenter family work, and community contexts, where responsibilities and crises crowd out new activities. As participants attempt to implement the tasks that have been set, research facilitators should (a) provide the emotional and organizational support they need to keep them on track and to maintain their energy, (b) model sound community-based processes, and (c) link the participants to a supportive network.

Regarding evaluation, at some stage, the need for a formal evaluation of the project may become evident. People who contributed funds and/or personal or political support will probably appreciate some statement or report that provides information about the extent to which progress has been made or desired ends have been achieved.

Tasks and activities that have resulted in a satisfactory resolution are delineated, and those that are unresolved become subject to continued action.

However, according to [AVI01], in Action Researches there are difficulties of generalization and validation. Action Research is highly situational, each project is unique. It makes it difficult for Action Research to be assessed and, perhaps, impossible for them to suggest general laws for the conduct of AR projects. They have only been able to suggest guidelines for controlling AR projects.

Since we are aiming at combining theory and practice and verifying a proposal framework that emerged from the Systematic Literature Review, we attempted to perform two studies following the CMD research approach. These studies will be described in the next chapter.

In the next sections, we will explain the studies carried out in order to analyze the proposed framework. We performed two studies, one in a company in Canada and another one in a company in Brazil. It is worthwhile to mention that these studies have added much more than just check our proposal.

5.4 Study in Canada

This study consists of identifying how a specific company works regarding the integration of agile methods and interaction design, finding out possible gaps and identifying what work well and can improve our framework.

The field study was carried out in a world leading technology company that develops and manufactures collaboration products[13]. This company uses Agile Methods and has usability as one of its main focus.

Some projects were observed and some interviews were performed in order to understand the process used in this company to make Agile and UX work together.

The research questions that drove this study were:

Q1: How does the company integrate Agile Methods and UX?

Q2: What works in this process?

Q3: What does not work in this process?

According to [RUN09], the case is referred to as the object of the study and it contains one or more units of analysis. In this study, the case is the software development process model and the units of analysis are the projects selected. Two projects were selected to be observed inside the company. The subjects are the members of the projects that were interviewed and the meetings of the projects that were observed.

We describe each study in terms of the people, the project, the research site, and the research methodology (how we collected the data and performed the data analysis). Next, we then relate our findings in terms of the key aspects we observed in our Systematic Review.

5.4.1 The People

Our study involved a team of seven individuals and one UX designer and was carried out over three months iteratively. The developers were part of the ‘Development Team‘ and the designers part of the ‘UX Team‘. The developers had been developing software using Scrum for approximately two years. Although they are called developers, individuals in the team have their own role according to their area and skills. The roles were Project Manager/Scrum Master, Product Owner, Technical Leader, Developer and Tester as can be seen in Table **4**.

[13] This is the description provided by the company's research facilitator. The name of the company and the projects were omitted due to confidentiality constraints.

Table 4. The roles and number of individuals for each role at the Development Team

Role	Individuals
Project Manager/Scrum Master	1
Product Owner	1
Technical Leader	1
Developer	2
Tester	2

Information architects, graphic designers and interaction designers compose the UX team/division. Each project has one UX designer, but a UX designer usually work with more than one development team. The same goes for Project Managers, and they are also known as Scrum Masters in the teams.

5.4.2 The Project(s)

Due to confidentiality constraints, we cannot provide much information about the projects. As already mentioned, we accompanied two projects, and we will call them here as Project X and Project Y.

All we can say about them is:

- Project X: consists of the development of new features for an existing product of the company.
- Project Y: consists of the development of an existing product of the company for a mobile/tablet device.

The UX member's role in Project X was to help software engineers to envision new features for this product. In Project Y, the UX member's role was to prototype and design the User Interface and the User Interaction flow for the product.

5.4.3 Research site

The team of developers was one of several Scrum teams in the company working on software development[14]. The developers and designers were seated in an open-plan office space located in the same building. However, they were co-located, in other words, they were not seated together. They were spread in the building, but the UX team members were seated close to each other. The researcher was seated with the UX member that was working in these projects.

[14] The company also develops hardware.

5.4.4 Research Methodology

In the next sections, we present in details the three stages of the research methodology, applied to this study.

5.4.4.1 Diagnosing

As mentioned in the previous chapter, we used two first-degree techniques, interviews and observations.

Regarding observations, due to the characteristics of invoking the least amount of interference in the work environment and the least expensive method to implement and still because the company did not permit video or audio recording of the meetings, we choose to manual record the observations of the meetings. We shadowed a UX person during his activities for a couple days and observed some meetings that he was involved, such as meetings of the UX Team of the company and some meetings of two different projects, as follows:

- Project X: 2 requirements meetings, 1 retrospective meeting.
- Project Y: 1 demo meeting, 3 planning meetings, 3 retrospective meetings and 2 user testing sessions.
- UX group meetings: 4 meetings.

Regarding interviews, we interviewed three members of the UX group that work in different projects and one project manager.

The Project Manager was interviewed aiming to define which Agile Method the company uses and how this integration of UX and Agile works or not through his point of view. And the UX people were interviewed aiming to understand UX people work on the different projects of the company.

Regarding the analysis, as already described, we performed Open and Focused Coding.

Initial memos were extracted by the researcher from the fieldnotes produced during the observations and from the interviews performed with members of the teams.

Having the memos produced, Open Coding was performed aiming to generate new insights and themes. Focused Coding was also performed and this coding consisted of linking the memos generated to the key aspects identified in the Systematic Review. Also some new aspects emerged from the analysis of the observations and interviews.

Later, some integrative memos were also written in order to relate the fieldnotes, the key aspects and the new codes emerged from the open coding.

We classified the findings according to the key points used to the Focused Coding and presented them to the company in order to validate them. These insights are presented as follows.

Regarding **LDUF**, we could notice that the company does not perform any design up front. Another thing that we could notice is that there is no collaboration between the UX Team and the Marketing Team, based on comments like:

'*Some User Research is performed by the Marketing Team. In general, the Marketing Team knows what they say they need, not what they really need. It's a not a target effort to gather what the user need… It's a sell visit.* – UX1[15]

'*We don't need to design everything up front* '– UX3

Since there is no User Research before the start of the project, UX Team has to try to think from the user point of view. But we could observe that in some projects there is an effort from the UX Team to participate of the Requirements Meetings to understand the needs of the Customer, however it is worthwhile to remember that is not always that the Customer is the User.

Regarding **Prototyping**, we could notice that the UX Team makes a really good use of prototypes. Depending on the situation they use paper prototypes, low-fi prototypes, high-fi prototypes to perform inspection evaluation or user testing to validate their ideas. For example:

'*We used to use Paper prototype, high fidelity prototype, product… some prototyping tools, sometimes high-fi prototypes, sometimes low-fi* ' – UX1

We also noticed that since the members of the UX Team have different backgrounds, some of them can code some functional – high-fidelity – prototypes and some of them just cannot do it, as we can see at:

'*It's tricky to UX people to code*' – UX2

We could notice that **User Testing** with real users are rarely performed, even when the project is in its final stages.

User Testing used to be performed with internal users, justifying that there is always new and old employees with different profiles and backgrounds. Some examples that based this statement are presented below:

'*Internally studies… new people and old people from inside the Company (...) With real users just at the final stages of the project*' – UX2

[15] UXx is the UX Team member interviewed and PM is the Project Manager.

Regarding **User Stories**, we noticed that there are not specific User Stories for UX issues. Sometimes when UX User Stories are written, these stories are then broken in smaller stories with technical development criteria. Hence, in general, User Stories do not have usability issues as acceptance criteria. We could also notice that sometimes User Stories are used to report usability issues identified during usability tests, as in the following quotes.

'*Some user stories we start with UX criteria and we brake them down in six stories with development criteria*' – UX2

'*Sometimes we add new user stories based on the results of the User Testing. But it depends on the problem. We also can put as a bug*' – UX2

These different ways that teams and UX members use User Stories lead to another observation, a lack of standard on how to report UX issues.

We also observed that the UX Team performs some **inspection evaluations**, e.g., a member of the UX Team designs a prototype and then another member perform some evaluations. Even they do not follow a specific method of inspection evaluation, it seems to work very well for them. Response examples:

'*We perform some experts evaluations, peer review*' – UX1

'*We perform some inspection evaluations, peer review with some UX member*' – UX2

Regarding the **One Sprint Ahead** topic, it is totally clear that the UX Team do not work one sprint ahead from the development team. Although the UX Team knows the benefits of designing one sprint ahead, it is not possible for now ('*We should work at least one sprint ahead the development team*'– UX3).

One of the problems is that the UX Team members do not have time to design one sprint ahead of the development team, probably because they are busy with other projects. It is consistent with the fact that there is not only One Team as Agile principles suggest. We could observe that the UX is almost outsourced in the company. Even if there is a UX Team within the company, UX members are not full team members, they are always working at many projects at the same time. Response examples:

'*We used to work on multiple projects, but it's really easier to work on only one product. It is so much easier to get involved, you know what's going on. Your attention is right directed*' – UX2

'*I'm working on 7 to 10 projects at the same time. With different levels of inclusion.*' – UX3

We observed that there is no collaboration between the UX and the Marketing Team ('*We absolutely are not close to the Marketing Team*'– UX3), but there is a good communication between the UX Team and the Development Team. However, sometimes UX members block the Development Team because they are working on other projects. Sometimes even the daily meeting block the UX member, because since they are working on too many projects, they have a lot of meetings to attend. Response examples:

'*The information is not shared with the UX Team. Sometimes we need to ask them: What do you know?*' – UX3

'*UX people should be Pigs.*[16]' – PM

Regarding the Big Picture, we did not observe problems about the maintenance of the Big Picture of the company's projects ('*We don't have any standard. We're trying to modify that. We're trying to implement some templates.*' – UX3).

Regarding the agile process used by the company, we could observe that the company uses an adapted Scrum. In a Scrum by the book, there are: Product Owner, Scrum Master and Team. In the company observed there are: Project Manager, Product Manager, Function Manager and Team.

The Product Manager acts more like the Product Owner. The Project Manager acts more like the Scrum Master in terms of being a facilitator, because he is not a developer and he also manages more than one project.

The Function Manager is in charge of the teams' members. For example, the UX Team has a Function Manager who defines which project should be prioritized. That is why we said that UX is almost outsourced, the UX member is not always available to the rest of the team. We may notice this misunderstanding by the following quote.

'*Sometimes, a member of the team doesn't know who to answer, if to the Project Manager or to the Function Manager* ' – PM

It is extremely important to note that all of the observations and findings are related to new projects or new products. We notice that for products that are being developed for a long time, the UX members that have worked for a long time in the same project already have some standards that they follow. But these standards are too specific for a product or specific from a specific member of the UX Team.

Finally, according to [LET05], it is useful to go back to the original participant population to discuss the findings. Based on this statement, we presented to and

[16] This is a fable told by Scrum practitioners about a pig and a chicken who considered starting a restaurant. "We could serve ham and eggs," said the chicken. "I don't think that would work," said the pig. "I'd be committed, but you'd only be involved." [BEY10]

discussed our findings with the company aiming at a better planning for the next phase of the study.

5.4.4.2 Planning

As suggested by [STR07], we met with the UX Team and its Function Manager in order to validate our findings and plan the action step for this research.

We had a really good feedback regarding our findings. The team seemed pleased and some members of the UX Team seemed surprised whereas other members looked like just confirming their thoughts.

This presentation of the findings to the company was really important and useful, because they provided us useful feedbacks. Both the case study and the validation of its findings provided us a lot of important results, which were used to refine and improve our framework.

After this presentation we met with one of the UX member that we were shadowing and the UX Team Function Manager to define the plan and what would be feasible to be implemented.

5.4.4.3 Acting

We discussed a plan for the Actions with the UX Team Function Manager, but most of the changes involve changes at the company structure. It demanded the involvement of higher levels of management inside the company, and according to the Function Manager, it is not that simple. So, unfortunately, we could not carry out the Action step.

Despite these constraints, we would start the Action stage in a new project of the company. However, the time was not appropriate because a lot of employees were on vacation and those who were at the company were overworked. Thus, Function Manager told us that it would be really complicated try to change the process during that period.

But the plan is there and we are still in touch with the company in order to complete this research. Also, we are working together with another researcher who will proceed with the research at the company.

5.5 Study in Brazil

Different from the first study, which was trying to identify how a company works regarding the integration of agile methods and interaction design, this one consists of identifying how the teams of a company work regarding this topic. We notice that each

team works in a different way, even only a different person in a team changes the way the team works.

The field study was carried out in a company that is present in digital media through seven major portals aimed at providing information, services and opportunities to the public of the major Brazilian cities and also through a mobile marketing organization. The company is undergoing a change in its business model. The company started this change by adopting Agile software development. As the first study, we followed the guidelines proposed by [RUN09] to carry out this study.

We observed two different teams that work on different products and we interviewed two UX designers and two Product Leaders.

As in the first study, the research questions that drove this study were:

Q1: How does a team integrate Agile Methods and UX?

Q2: What works in this process?

Q3: What does not work in this process?

As described in the first study, in this study, the case is the software development process model and the units of analysis are the teams selected. Two teams were selected to be observed inside the company. The subjects are the members of the projects that were interviewed and the meetings of the projects that were observed.

The structure for this section is the same used in the previous section, we describe our observational study in terms of the people, the project, the research site, and the research methodology (how we collected the data and performed the data analysis).

5.5.1 The People

A study of UX designers and their interactions with an Agile team working on the same product was carried out over two iterations – 25 working days. The length of the sprints varies from project to project, but for the two teams observed they have two weeks sprints with a week between the sprints.

As already mentioned, our study involved two teams. The teams are composed by Product Leader/Product Owner, UX Designer, Developer, Tester and Search Engine Optimization (SEO), with little differences as can been noticed in Table **5** and Table **6**.

Table 5. The roles and number of individuals for each role in Team A

Role	Individuals
Business Owner/Director	1
Product Leader/Product Owner	1
Scrum Master	1
UX Designer	1
Graphic Designer	1
Developer	4
Tester	1
SEO	1

One team - Team A - has two individuals focused on UX, a UX Designer and a Graphic Designer, whereas the other team - Team B - has just a UX Designer who performs the job as a Graphic Designer as well.

Table 6. The roles and number of individuals for each role in Team B

Role	Individuals
Business Owner/Director	1
Product Leader/Product Owner	1
Scrum Master	1
UX Designer	1
Developer	6
Tester	1
SEO	1

5.5.2 The Project(s)

The company is not structured by projects, but by digital products. It is a digital product-driven business. Each product has a team, as follows.

Team A works on a product that is a web portal about agribusiness in the country. Team B works on a product that is a web portal of services and opportunities in which there are addresses and data from companies and services from the Southern Brazil.

The UX designer's role in Team A was to perform user research, benchmarking and interaction design. The Graphic designer's role was to design the UI based on the wireframes provided by the UX designer.

Whereas in Team B, UX designer used to perform both roles, performing user research, benchmarking, interaction design and UI design.

5.5.3 Research site

Unlike the first study, there is no separated UX Team and Developers Team. Each team has its own individuals, a team does not share a UX person, for instance.

These teams were selected because they were the most senior Agile teams in the company. According to their own definition, they do not follow Scrum or XP, they follow their own Agile methodology.

The developers and designers were seated in an open-plan office space located in the same building and in the same floor. Each team is seated together, co-located. The researcher was not in the building all the time, he used to go to the company to perform observations and interview, thus being free to observe any activity of the projects.

5.5.4 Research Methodology

In the next sections, we present in details the three stages of the research methodology, applied to this study.

5.5.4.1 Diagnosing

As in the first study, we conducted interviews and observations, manual recording our observations. We observed mainly Daily Meetings of the selected teams.

Regarding interviews, we interviewed the UX Designer and the Product Leader of the two selected teams.

Regarding the analysis, we extracted memos from the fieldnotes, and then we performed Open and Focused Coding as in the previous study. We also classified the findings according to the key points used for the Focused Coding.

Regarding **LDUF**, we could notice that the teams do not usually perform design up front. They said they do not have time do design up front, for instance:

'*Sometimes we organize some Focus Groups to gather some users' needs*' - UX B

'*I performed a benchmarking. Then I'm consuming this material. I can keep myself up-to-date.*' - UX B

'We don't have much time to work up front. Then we try to do at least something up front, and it seems to be working' - UX B

'*We perform some speculative research, analysis of competitors*' - UX A

What they do is some research up front; as we can notice from their comments, benchmarking and Focus Groups. If we consider User Research as design, we would say

that they design up front. However, if we consider that User Research is another stage separated from design, we would say that the teams do not perform design up front.

The next excerpts confirm this:

'*We prepare the next sprint to show the way, clarify doubts*' - PL[17] A

'*We have something that we call Discovering that happens before the planning*' - PL A

They mentioned that the material collected up front is used to keep a holistic view of the product, as follows:

'*Although the development is in pieces, with the studies before the planning, I can consume this material (from the Focus Groups and benchmarking) and keep a view of the whole product. It facilitates the things for me. - UX B*

This stretch is also related to another key aspect, the maintenance of the Big Picture. We could notice that this research up front help the Interaction Designer, and consequently the team, to keep this Big Picture.

If we look more deeply, we notice that the concept of Design Up Front may have two variations: Design Up Front the Release and Design Up Front the Sprint.

So, with these concepts, we would say that they perform some researches up front the Release and use the results to design up front the sprints.

Regarding **Prototyping**, sometimes they use high-fidelity and sometimes low-fidelity prototypes. Concerning low-fidelity prototypes:

'*Once the product is defined, I prototype it in two or three weeks. Paper prototype to communicate between us and some HTML to present to directors.*' - UX B

'*User Stories are visually represented by wireframes.*' - UX A

User Experience designers used to low-fidelity prototype to represent and validate some ideas with the Product Leaders. Whereas the high-fidelity prototypes are used to test issues related to graphic design and to verify workflows:

'*As we work close (UX Designer and Business Analyst), we validate our ideas together by prototyping and then specifying User Stories*' - UX B

'*We put an effort to build an HTML version, then the effort to add some links and build a functional prototype is minimal, then we can present them to* ' - UX A

Regarding **User Testing**, it is totally clear that the teams do not perform any:

'*We did not perform any User Testing*' - UX A

'*We don't perform User Testing yet.*' - UX B

[17] PL means Product Leader and UX means User eXperience Designer

Although they mention that they know the importance of this kind of evaluation, they say they could not fit them into their process yet.

They mention the use of some tracking tools. But this is a common practice mainly for web-based systems:

'*We used some tracking tools. But did not perform any User Testing yet*' - UX A

'*We used some tracking tools. This is our only initiative so far.*' - PL A

The team B has a database of users that they used to call to carry out focus groups.

'*As we have a set of users (database of volunteers), we can call them and carry out some focus groups. We have 4 different personas with them*' - UX B

Concerning **User Stories**, we could notice that UX issues are used as acceptance criteria in the User Stories.

'We set up UX criteria as acceptance criteria. But the specification is more visual than textual. Moreover, our communication is essentially visual'- UX B

'*PL define the acceptance criteria. I define some criteria for the wireframe and the PL transfer them to the acceptance criteria*' - UX A

'*We put UX criteria as acceptance criteria at the User Stories, or we reference the behavior of the interface in a sequence of wireframes*' - UX A

We also notice that the User Stories are defined by the Product Leader and the UX Designer together ('*This definition is done by me and the UX*' - PL B). UX Designer visually represents the Story specified by the Product Leader. Also, when identified some usability problem, it may become a new User Story according to the judgment of the UX Designer and the Product Leader. They state that for a UX problem becomes a new story depends on how complex and how important it is. As follows:

'Once we have a problem, we insert a new Story into the Backlog and then we perform a prioritization' - UX A

'*Depending on how important a usability problem is, it might become a new User Story or just go back to the Product Backlog'* - UX B

Regarding **Inspection**, we may notice that the teams used to perform some peer review, sometimes with the Graphic Designer and sometimes with the Product Leader. They do not perform any of the traditional inspection methods as Heuristic Evaluation or Guideline Review, according to the following quotes:

'*Me and UX peer review the UIs*' - PL A

'*We perform a lot of informal evaluations. Me and the GD*' - UX A

We also observed that the fact of the entire team being co-located have facilitated the interaction between them, improving collaboration and communication based on their reports as follows:

'*I seat beside him.*' - UX A

'*I would say that our design is participatory/collaborative. Everybody in team collaborate.*' - UX A

'*Some UX issues don't go to the User Stories. They are addressed face-to-face.*' - UX B

When asked about **User Testing** or **Inspection evaluation**, the use of tools for 'automatic evaluations' has emerged.

'*We monitor the system via Google Analytics etc.*' - UX B

'*If we don't have time. We deliver without testing.*' - UX B

Regarding **One Sprint Ahead**, there is an attempt of working one sprint ahead of the developers, but it does not work all the time. As can be notice in the following quotes, it works for the Team B but not for the Team A.

'*We have a Sprint 0, pre-production*' - PL A

'*Me and UX work one sprint ahead of the rest of the team*' - PL A

'*I gotta provide something to the developers. Once the developers start, I can work on stuff for the next Sprint or even Release*' - UX B

'*I don't know what's coming in the next release. But at least I have an idea*' - UX B

According to our observations and we notice that to be able to work one sprint ahead of the developers, UX Designer and the Product Leader should have worked on some design up front.

We observed an environment that enables a really high level of collaboration and communication among the members of the team. We also noticed that there is a Close Collaboration mainly between the UX Designer and the Product Leader regarding UX design, at least at the Team B as can be observed in the quotes:

'*UX is my right-hand man*' - PL B

'*I'm a Business Analyst*' - PL B

'*Everybody votes in the Planning meetings. UX votes because he's extremely important in our process*' - PL B

The communication between the UX Designer and the Developers is very good, it is usually face-to-face. This happens because there is only one team, UX Designers are full-time team members, as the quotes present:

'*I'm a full team member.*' - UX B

'*The participation of the UX during the Sprint is very important, to provide feedback and validate ideas.*' - PL A

'*I'm a full-time team member of the team. And I work really close to the BA.*' - UX A

'*Developers request our help a lot during the sprint.*' - UX A

'*In the past, when we weren't exclusive for a specific team there was a conflict of interest. I had to divide myself between two or more teams.*'

Of course, many of the snippets presented for each topic are related, and they will be discussed following.

5.5.4.2 Planning

As in the first study and suggested by [STR07], we met with the UX Designer of the Team B, and with the Product Leader and the UX Designer of the team A, in order to validate our findings and plan the action step for this research.

Unfortunately, Team A was blocked waiting for some bureaucracy contractual to have content to feed the Product under development. Their development sprints will begin just in the next month.

But fortunately, we could carry out the Action stage with the Team B and it will be described in the next section. As in the first study and suggested by the literature, we presented our findings to the people involved in the project. Then together with them we defined which actions could be possible to be taken and the way it could work.

As previously described, most of the practices suggested by us are already being performed by the Team B. Nevertheless, we could support the team to go through at least one practice, **User Testing**. It is described as follows.

5.5.4.3 Acting

Supported by the UX Designer, we started the Action stage. This team never had performed any User Testing because according to their report, they never had time to do that. So, our challenge was to identify the timing for it.

As already mentioned, the team works with a release of three sprints, and each sprint consists of two weeks. After the last sprint of the release there is a break of a week to review everything before the release of the product.

The team was exactly at the last week of the last sprint before the release of the product. We then defined that this 'break' week would be an opportunity to perform User Testing. Although we advocate that this week should be used to implement eventual

corrections, at this point of the project this week is a good opportunity to carry out some user testing. Therefore, eventual complex corrections would go to the product backlog to be addressed in the next release.

The portion of the system that was under development for this release was directed to internal employees. Therefore, was easier and cheaper to recruit users for the tests.

Some user testing was performed, but so far we do not have their results. So, we do not know yet how useful these user testing sessions were. We assume that they could provide a useful feedback to the team, but we do not have evidence so far.

5.6 Discussion

As already mentioned, initially we were aiming at performing Action Research studies in real companies. Action Research is an excellent approach because it allows a great integration between theory and practice. Also this kind of research is not widely used in the HCI and Software Engineering fields.

However, during the studies' development we noticed that the Action step was outside our control and we could not perform the Action stage indeed. Consequently, we could not create organizational changes. Thus we consider the performed studies as two field studies in the real world by the use of Grounded Theory techniques to collect and analyze the data, one carried out in Canada and another one in Brazil, comprehending collecting, analyzing and presenting the results.

6. REFINED FRAMEWORK

In this section we present some considerations about the studies performed in order to compare the results, how they helped to improve the proposed framework and the enhanced version of the framework.

Although we know that most of the companies do not follow a specific agile methodology “by the book” and each company adapt or combine some methodologies to their reality, we believe that the roles involved in the process must be well defined.

For example, in the first study, in Canada, there is the Function Manager role. Sometimes it causes a confusion in which the UX Team member does not know who to respond, because the Project Manager wants him working on a specific task of his projects but the UX member has to work on other project prioritized by the Function Manager.

Although there are some peculiarities in the names to define roles used by the teams observed at the study performed in Brazil, the roles are well defined. The company's structure makes it easier, because all members of the team have only one team to work with.

Thus, having a UX Designer dedicated to only one project/product at a time becomes extremely important. As we could notice in the first study, UX Designers work on to many projects at the same time causing blocks to the Development Team. In the second study, we did not observe any kind of block caused by UX Designers, what led us to conclude that having a UX Designer as a full-time team member is extremely important.

Another topic related is the co-location of the UX Designers. We could observe that the communication and collaboration between UX Designers and Development Team is immensely better when they are co-located. This co-location topic may generate a broad discussion about distributed software development, but it is not the topic to be addressed for now.

Still about this topic, we could notice through the Systematic Review that most of the papers analyzed suggest the use of a UX Specialist in the development team. But just one paper comments about the benefits of having the UX Specialist co-located with the development Team [FER10]. We think that this practice of having the UX member co-located with the development team should be more explored.

Also, we believe that in addition to this non-collaboration between the Marketing Team and the UX Team observed in the first study, the fact of the UX not working on just on project contributes to the non-performing of some design up front. Even at different stages of the project, there is always something to be done by the UX Designer.

Assuming that 'there is always something'[18] and based on our observations, we suggest that the UX Designer should work on four sprints at the same time looking at four different directions, as follows.

As presented at Figure **18**, assuming that the current sprint is Sprint n, the UX Designer should work to:

1. Evaluate and review what has been built at Sprint n-1.
2. Provide feedback and clarify design in current Sprint n.
3. Design for the next sprint (Sprint n+1).
4. Explore and research for the subsequent sprint (Sprint n+2).

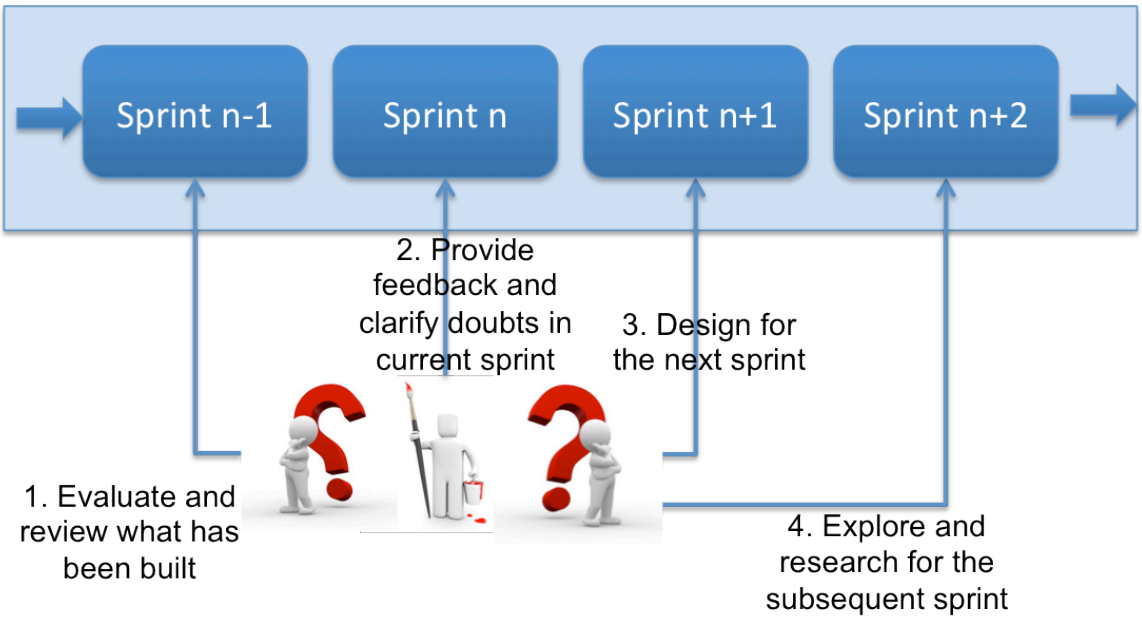

Figure 18. Interaction Designer looking at four directions

Consequently, having no design or research up front, UX Designer cannot work one sprint ahead. Without this work up front, is almost impossible to construct a holistic view, the big picture.

Another aspect that was already addressed by the literature was the inclusion of UX issues as acceptance criteria of the User Stories. A couple of authors suggested that, but we have not found any evidence about that. We noticed in the first study that sometimes, simple UX problems showed up just at the end of the sprint, when the UX Designer was inspected the UI. This problem could be avoided by the addition of a simple criteria regarding this UX issue into the User Story. We also observed that this practice has been used by the team in the second study. And the Designers use low-fidelity prototypes attached to the User Stories in order to communicate these UX criteria.

[18] Comment from Jeff Patton at http://agileproductdesign.com/blog/emerging_best_agile_ux_practice.html

Regarding prototyping, in our first study, we could observe that the teams make a really good use of low-fi prototypes. Sometimes using paper sketches, or Visio[19] prototypes or even functional prototypes, but still low-fidelity ones. The UX Designers use these prototypes to perform peer reviews with other Designers and validate ideas with the Development Team; this peer review practice is going to be added to the proposed framework.

Another important issue that emerged from the first study is regarding the communication between Interaction Designers and some stakeholders. The UX team has some questions about how to communicate their design decisions to an audience that do not have the same background. They argue that the use of just low-fidelity prototypes is not enough to effectively communicate their ideas.

However, most of the papers analyzed suggest the use of prototypes, sometimes low-fi and sometimes high-fidelity prototypes. In the second study, we noticed that this communication from the Designers to the directors, for instance, was by the use of high-fidelity prototypes of some portions of the product supported by a presentation of the concepts given by the UX Designer and the Business Analyst.

This presentation by the UX Designer and the Business Analyst just happens because of the really good and close collaboration that happens in our second study. They said that they take all decisions together. This happens because the UX Designer is close to the Business Analyst since the beginning of the project/product, and the UX Designer can work since the beginning because he is not swamped with other projects in parallel.

We then get into a cycle, as in the Figure **19**. We observed that whenever a Interaction Designer is working on too many projects, he cannot close collaborate with the Business Analysts or with the Development Team, and it may not allow the Designer to design up front or work one sprint ahead of the Development Team, and it can lead to the missing of the Big Picture or even to blocking the team.

[19] http://office.microsoft.com/en-us/visio/

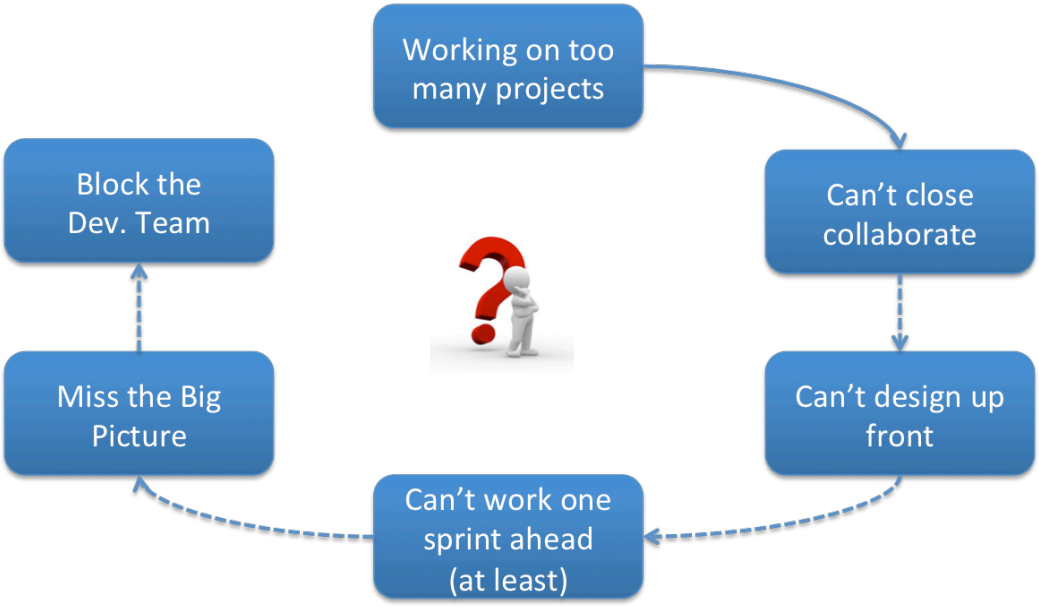

Figure 19. Cycle of problems that the UX Designer **may** face or lead

Finally, we believe that both studies contributed a lot for our knowledge and for our framework proposal. Although we could not experiment most of our ideas of the proposed framework, we could observe that when most of the practices of the framework are not implemented, we notice a lot of complaints and problems. And when the practices are used, the problems decrease significantly.

Therefore, after analysis and refinements, our framework proposal based on theory and practice, is presented at Figure **20** and general guidelines related to those key aspects are described as follows:

- Little Design Up Front.
  - Research, model and design, but only enough, not everything.
  - Use the Iteration 0 to define the system scope and structure.
- Close Collaboration.
  - Interaction Designers should be full-time team members.
  - Interaction Designers should work co-located with Developers.
  - Interaction Designers and Developers should share documents.
- Low-Fidelity Prototypes.
  - Prototype in low fidelity, whatever it is, in order to test and validate ideas as fast as possible.
  - Treat prototypes as specification.
- User Testing.

- Interaction Designers should cultivate a user validation group for continuous user validation (design partners, according to Jeff Patton[20]).
- Get user feedback in context whenever possible.

- User Stories.
  - Interaction Designer should chunk his design work.
  - Add UX issues into the acceptance criteria.
- Inspection.
  - Peer review ideas and designs.
  - Provide feedback to Developers during the current Iteration. Collaborate and support.
- One Sprint Ahead (at least).
  - Research two Iterations ahead.
  - Design one Iteration ahead.
  - Collaborate and support at the current Iteration.
  - Validate working software one Iteration behind.

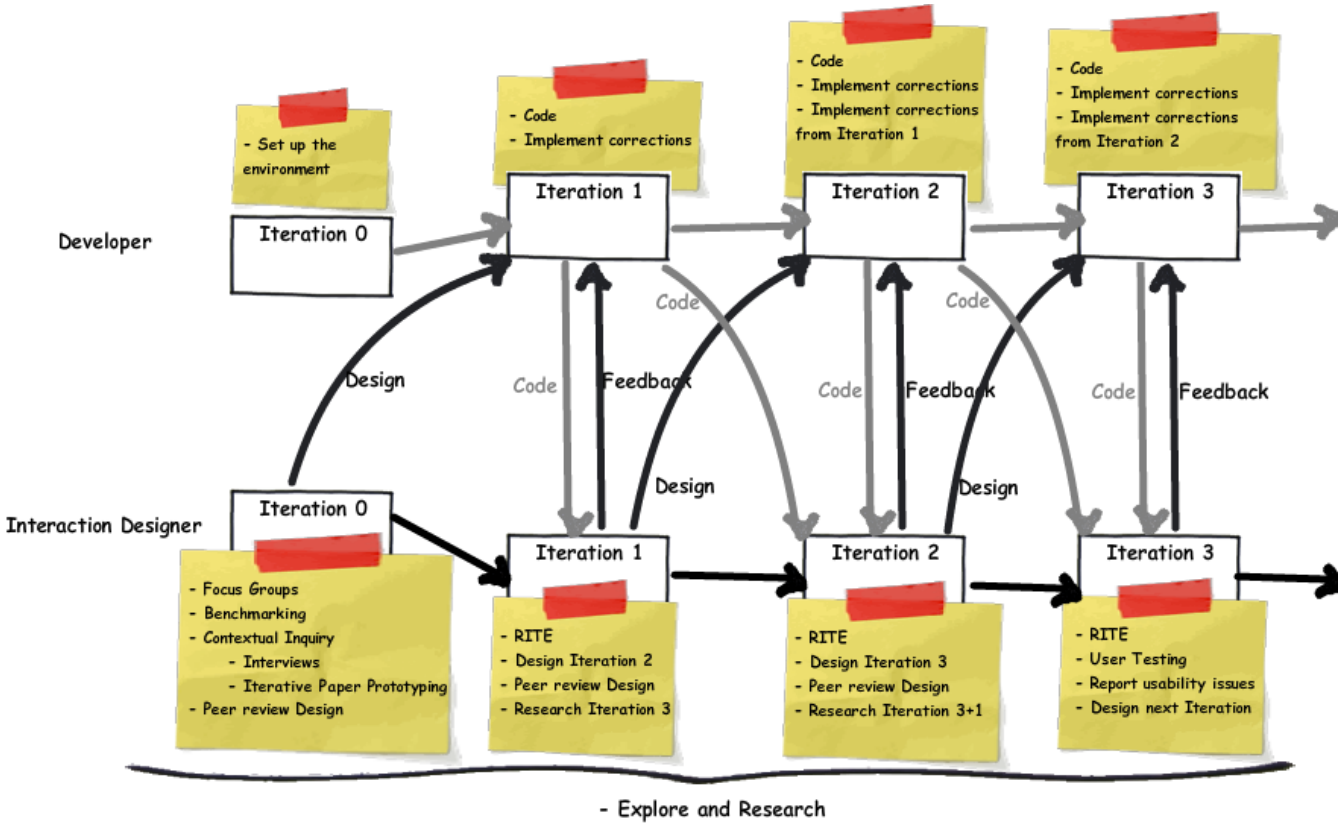

Figure 20. Framework for integrating Interaction Design and Agile Development

[20] http://agileproductdesign.com/blog/emerging_best_agile_ux_practice.html

In order to detail the framework presented at Figure **20**, we present which practices might be used by Developers at Table **7** and which practices and artifacts might be used by Interaction Designers at Table **8**.

Table 7. Developers' practices

Iteration	Practice
Zero (0)	Set up the environment
One (1)	Code features designed by the Interaction Designer at Iteration 0 Implement corrections suggested by the Interaction Designer during the current Iteration
Two (2)	Code features designed by the Interaction Designer at Iteration 1 Implement corrections suggested by the Interaction Designer during the current Iteration Implement corrections added by the Interaction Designer to the code developed at Iteration 1
Three (3)	Code features designed by the Interaction Designer at Iteration 2 Implement corrections suggested by the Interaction Designer during the current Iteration Implement corrections added by the Interaction Designer to the code developed at Iteration 2

However, for these practices work, there are some constraints that should be respected, such as:

- The team **must** follow the Scrum basic structure;
- The roles in the team **must** be well-defined;
- The team **must** have at least one UX Specialist;
- The team and the UX Specialist **should** be co-located.

As previously mentioned, we also tried to organize the guidelines according to the activities that compose the Interaction Design process. It is worthwhile to mention that splitting the practices into these three activities is a difficulty task, because most of them are interconnected. Still, we have tried to do this division as follows.

Table 8. Interaction Designers' practices and artifacts

Iteration	Practice	Artifact
	Research	
	Carry out Focus Groups	
	Benchmarking	
	Perform Contextual Inquiry	
Zero (0)	**Design**	
	Prototype Iteration 1	Prototypes (low or high fidelity)
	Peer review design	User Stories
	Provide feedback during the current Iteration	Acceptance criteria
		Face-to-face communication
	Research	
	Explore and Research	
	Design	
	Prototype Iteration 2	Prototypes (low or high fidelity)
One (1)	Peer review design	User Stories
	Provide feedback during the current Iteration	Face-to-face communication
	Evaluation	
	Perform RITE	Acceptance Criteria
	Peer review (inspection) evaluation	Face-to-face communication
	Report usability issues	User Stories
	Research	
	Explore and Research	
	Peer review design	
	Design	
	Prototype Iteration 3	
Two (2)	Peer review design	
	Provide feedback during the current Iteration	
	Evaluation	
	Perform RITE	Face-to-face communication
	User Testing	User Stories
	Report usability issues	Presentations

6.1.1 Research

Regarding Research practices, we suggest the use of Contextual Inquiry, proposed by [BEY99]. We suggest the use of Focus Groups since we notice good results in one of our studies. Or we suggest the practice of benchmarking because this analysis of competitors helped a lot the Interaction Designer and the Business Analyst in our second study. Also, it is important to mention that the user research should be a continuous and iterative process, distributed throughout the entire development process.

6.1.2 Design

Concerning Design, we suggest the use of low-fidelity prototypes to consume, in other words, to test and validate ideas and to communicate design decisions to the Development Team. We also suggest the use of high-fidelity prototypes to deliver, in other words, to communicate design decisions to the stakeholders, because with this audience sometimes there are some misconceptions. Therefore, there is a need of designs with a different level of abstraction.

6.1.3 Evaluation

Regarding Evaluation practices, we suggest RITE, because according to [FIS09], this method allows the team to incorporate several feedback loops in a short period of time, fixing obvious UI problems in real-time during testing.

If unable to perform any RITE session, we suggest the use of usability peer reviews, which consists of a review of the UI by Interaction Designers. This practice has proved that allows the identification of usability problems earlier in both the studies performed.

We also suggest User Testing, although we know that fitting this practice into an Agile process is a difficulty task. However, we believe that if some practices are followed, perform User Testing in an Agile context could be easier.

7. FINAL CONSIDERATIONS AND FUTURE WORK

Agile development has become mainstream regarding software development processes. Along the increasing understanding of the importance of good User eXperience came the need to integrate these two areas.

However, Agile development have a distinct culture that at first glance seems to conflict with User-Centered Design. Therefore, integrating these two areas becomes a challenging task.

This thesis focused on defining a framework for integrating these areas, providing a set of practices and artifacts to support Agile teams and Interaction Designers to overcome this challenge. We believe that this goal was achieved, although we understand that there is a need for more practical applications.

Regarding the specific goals of this thesis, we believe that all of them were achieved. We could bring up an extensive literature review regarding Agile and Interaction Design by the conduction of a Systematic Review. This Systematic Review helped us to identify existing approaches regarding the integration of Agile and Interaction Design. These results based the definition of a framework for this integration, which was verified through practical applications.

7.1 Contributions

We believe that this thesis contributed significantly to further studies regarding the integration of Software Engineering and Human-Computer Interaction, in the specific context of Agile development and Interaction Design.

From the theoretical point of view, it contributed for researchers working on this field by the Systematic Review published at the Agile Conference'11 [daS11]. This Systematic Review was conducted in June, 2010 and then conducted again in July, 2011. Also, the methodology adopted to conduct the studies facilitated the entry of the academy in the industry, allowing and increasing the collaboration between these two 'worlds' and contributing for theory and practice. Even just the interviews and observations helped the members of the teams to do a self-analysis of their own work.

Regarding the practical perspective, this research contributed a lot, mainly for the first study carried out. The research highlighted the interest by the industry and will be continued in the companies that participated in the studies. Still regarding the practice, based on the lessons learned from performing these studies in the industry context, we

presented a tutorial [daS111] at IHC+CLIHC'11 (Brazilian Symposium on Human Factors in Computer Systems and Latin American Conference on Human Computer Interaction).

7.2 Limitations

Regarding the methodology adopted, as already forewarned by the literature, this methodology does not allow generalizations since it is focused on a specific environment without isolation of variables. Besides, since we could not perform the Action stage, we could not state that this framework works. Thus, we could say that this framework can be applied to analyze and evaluate if Interaction Design is integrated with Agile Development harmoniously.

Also, we believe that even not creating organizational changes in terms of the company's process yet, we believe that is going to happen by the sequence of the studies.

According to [FER11], the difficulty with focusing on process, or methods, is that processes are rarely mechanically followed in practice. We faced this problem because different companies, teams, or even people use processes or methods in different ways. However, both of the companies studied follow an adaptation of Scrum. Since our proposal is based on the Scrum's structure, we would say that this framework could be used for those who follow Scrum and its adaptations.

The projects/products/teams analyzed were developing internal products. The Product Owner was an internal costumer, a customer from inside the company. So, we could not analyze how this communication between the team and external stakeholders works. This communication can be a problem, because this was a question asked by UX people during the first study. They have some projects in which they work with external customers. This kind of question did not emerge from the second study, because even though their product consumers are external, the customer is from inside the company. So, we would say that this framework is recommended for teams that work with internal costumers.

Finally, we studied two extremely different realities. In the first study, they do not use most of the practices that we suggest and they report a lot of problems. In the second study, they follow a lot of the practices that we suggest and we notice just a couple of problems. This might be another limitation, because we could not find a balance between them.

7.3 Future work

Some specific questions emerged from this research, as follows.

We have encountered a question still unanswered, ´How much is Little Design Up Front?´. This is a research question that we are working on, trying to answer how much is enough.

Another important question is ‘How to prototype in low-fidelity when working with distributed teams?’. We are researching about prototyping tools to overcome this challenge.

Also, we affirm that the Interaction Designer should work co-located with the Development Team, but ‘What about when this co-location is not possible? How could we improve collaboration and communication?’.

We also noticed a huge growth potential in this research, however it depends a lot on partnerships between academy and industry.

In Canada, we notice this integration of academy and industry with applied research really close, but it is not a common practice in Brazil yet. Although we have TECNOPUC[21] at PUCRS, what was really helpful in this work.

The methodology used allows us to perform future works with the industry mainly, since it is a methodology that contributes both for the theory and for practice. We are already in touch with some companies to go ahead with this work.

[21] TECNOPUC is a technology in partnership with PUCRS that encourages research and innovation through the integration between academia, government and private institutions.
http://www.pucrs.br/agt/integrating_university_companies_society.pdf

REFERENCES

[ADI09] ADIKARI, S.; MCDONALD, C.; CAMPBELL, J. "Little Design Up-Front: A Design Science Approach to Integrating Usability into Agile Requirements Engineering". In: Human-Computer Interaction - HCII2009. Berlin, Heidelberg: Springer-Verlag. 2009, pp. 549-558.

[ALB10] ALBISETTI, M. "Launchpad's Quest for a Better and Agile User Interface". In: XP. 2010, pp. 244-250.

[AMB06] AMBLER, S. "Bridging the gap - Agile software development and usability". *Dr.Dobb's - The World of Software Development*, 2006, pp. 1-4.

[ARI09] ARIAS, C. S.; ANTUNES, A. E. "A conceptual knowledge base representation for agile design of human-computer interface". In: International Symposium on Intelligent Information Technology Application. Piscataway, NJ, USA: IEEE Press. 2009, pp. 156-160.

[ARM04] ARMITAGE, J. "Are agile methods good for design?" *interactions*, New York, NY, USA, vol. 11, January 2004, pp. 14-23.

[AVI01] AVISON, D.; BASKERVILLE, R.; MYERS, M. "Controlling action research projects". *Information Technology People*, vol. 14-1, 2001, pp. 28-45.

[AVI99] AVISON, D. E. *et al.* "Action Research". *Communications ACM*, New York, vol. 42-1, January 1999, pp. 94-97.

[BAR09] BARKSDALE, J. T.; RAGAN, E. D.; MCCRICKARD, D. S. "Easing Team Politics in Agile Usability: A Concept Mapping Approach". In: Agile Conference. Washington, DC, USA: IEEE Computer Society. 2009, pp. 19-25.

[BAS99] BASKERVILLE, R. "Investigating Information Systems with Action Research". *Commun. AIS*, Atlanta, vol. 2-3, 1999,

[BEC99] BECK, K. "eXtreme Programming eXplained: embrace change". [S.l.]: Addison-Wesley, 1999.

[BEN10] BENIGNI, G. *et al.* "USABAGILE: A Web Agile Usability Approach for Web Site Design". *Computational Science and Its Applications--ICCSA 2010*, 2010, pp. 422-431.

[BEY04] BEYER, H.; HOLTZBLATT, K.; BAKER, L. "An Agile Customer-Centered Method: Rapid Contextual Design". In: XP/Agile. 2004, pp. 527-554.

[BEY10] BEYER, H. "User-Centered Agile Methods". [S.l.]: Morgan & Claypool Publishers, 2010.

[BEY99] BEYER, H.; HOLTZBLATT, K. "Contextual Design". *Interactions*, New York, vol. 6-1, January 1999, pp. 32-42.

[BIO05] BIOLCHINI, J. *et al.* "Systematic Review in Software Engineering", Tech. rep., COPPE/UFRJ, Rio de Janeiro, 2005

[BRO08] BROSCHINSKY, D.; BAKER, L. "Using Persona with XP at LANDesk Software, an Avocent Company". In: Agile Conference. Washington, DC, USA: IEEE Computer Society. 2008, pp. 543-548.

[BROW08] BROWN, J.; LINDGAARD, G.; BIDDLE, R. "Stories, Sketches, and Lists: Developers and Interaction Designers Interacting Through Artefacts". In: Agile Conference. Washington, DC, USA: IEEE Computer Society. 2008, pp. 39-50.

[BUD09] BUDWIG, M.; JEONG, S.; KELKAR, K. "When user experience met agile: a case study". In: CHI. New York, NY, USA: ACM. 2009, pp. 3075-3084.

[CAR04] CARBON, R.; DÖRR, J.; TRAPP, M. "Focusing Extreme Programming on Usability." 2004, pp. 147-152.

[CHA06] CHAMBERLAIN, S.; SHARP, H.; MAIDEN, N. "Towards a framework for

integrating agile development and user-centred design". In: XP. 2006, pp. 143-153.

[CHO09] CHO, L. "Adopting an Agile Culture". In: Agile Conference. Washington, DC, USA: IEEE Computer Society. 2009, pp. 400-403.

[CHO09a] CHO, L. "Adopting an Agile Culture A User Experience Team's Journey". In: Agile Conference. Washington, DC, USA: IEEE Computer Society. 2009, pp. 416-421.

[COA11] COATTA, T.; GOSPER, J. "UX design and agile: a natural fit?" *Commun. ACM*, New York, NY, USA, vol. 54, January 2011, pp. 54-60. Capturado em: <http://doi.acm.org/10.1145/1866739.1866753>,

[COC01] COCKBURN, A. "Agile Software Development". 1st Edition. ed., [S.l.]: Addison-Wesley Professional, 2001.

[CON02] CONSTANTINE, L. L.; D., L. A. "Usage-Centered Engineering for Web Applications". *IEEE Softw.*, Los Alamitos, CA, USA, vol. 19, March 2002, pp. 42-50.

[CON02a] CONSTANTINE, L. L. "Process Agility and Software Usability: Toward Lightweight Usage-Centered Design". *Information Age*, vol. 1, 2002, pp. 1-10.

[CON99] CONSTANTINE, L. L.; D., L. A. "Software for use: a practical guide to the models and methods of usage-centered design". [S.l.]: ACM Press/Addison-Wesley Publishing Co., 1999.

[daS11] DA SILVA, T. S. *et al.* "User-Centered Design and Agile Methods: A Systematic Review". In: Agile COnference. 2011, pp. 77-86.

[daS111] DA SILVA, T. S.; SILVEIRA, M. S.; MAURER, F. "Best Practices for Integrating User-Centered Design and Agile Development". Porto de Galinhas: ACM, 2011, 43-45p.

[DAV04] DAVISON, R. M.; MARTINSONS, M. G.; KOCK, N. "Principles of canonical action research". *Information Systems Journal*, vol. 14, 2004, pp. 65-86.

[deS09] DE SOUZA, C. S.; LEITÃO, C. F. "Semiotic Engineering Methods for Scientific Research in HCI". [S.l.]: Morgan & Claypool, 2009.

[DET07] DETWEILER, M. "Managing UCD within agile projects". *Interactions*, New York, NY, USA, vol. 14, May 2007, pp. 40-42.

[DIT05] DITTRICH, Y. *et al.* "Co-operative Method Development Revisited". In: Workshop on Human and social factors of software engineering (HSSE '05). New York: ACM. 2005, pp. 1-3.

[DIT08] DITTRICH, Y. *et al.* "Cooperative method development: Combining qualitative empirical research with method, technique and process improvement". *Journal of Empirical Software Engineering*, vol. 13-3, 2008, pp. 231-260.

[DIX04] DIX, A. *et al.* "Human-Computer Interaction". 3rd Edition. ed., Harlow: Pearson Education, 2004.

[DUC07] DÜCHTING, M.; ZIMMERMANN, D.; NEBE, K. "Incorporating user centered requirement engineering into agile software development". In: Human-Computer Interaction - HCII. Berlin, Heidelberg: Springer-Verlag. 2007, pp. 58-67.

[DYB08] DYBA, T.; DINGSOYR, T. "Empirical studies of agile software development: A systematic review". *Inf. Softw. Technol.*, Newton, MA, USA, vol. 50, August 2008, pp. 833-859.

[EME11] EMERSON, R. M.; FRETZ, R. I.; SHAW, L. L. "Writing Ethnographic Fieldnotes". 2nd. ed., Chicago: The University of Chicago Press, 2011.

[FED08] FEDEROFF, M. *et al.* "Extreme usability: adapting research approaches for agile development". New York, NY, USA: ACM. 2008, pp. 2269-2272.

[FER07] FERREIRA, J.; NOBLE, J.; BIDDLE, R. "Agile Development Iterations and UI Design". In: Agile Conference. Washington, DC, USA: IEEE Computer Society. 2007,

pp. 50-58.

[FER07a] FERREIRA, J.; NOBLE, J.; BIDDLE, R. "Up-front interaction design in agile development". In: XP. Berlin, Heidelberg: Springer-Verlag. 2007, pp. 9-16.

[FER07b] FERREIRA, J.; NOBLE, J.; BIDDLE, R. "Interaction designers on extreme programming teams: Case studies from the real world". *IN NZCSRSC '07: 5TH NEW ZEALAND COMPUTER SCIENCE RESEARCH STUDENT CONFERENCE*, 2007, pp. 1-10.

[FER10] FERREIRA, J.; SHARP, H.; ROBINSON, H. "Values and Assumptions Shaping Agile Development and User Experience Design in Practice". In: XP. 2010, pp. 178-183.

[FER11] FERREIRA, J.; SHARP, H.; ROBINSON, H. "User experience design and agile development: managing cooperation through articulation work". *Softw. Pract. Exper.*, New York, NY, USA, vol. 41, August 2011, pp. 963-974.

[FIS09] FISHER, K. G.; BANKSTON, A. "From Cradle to Sprint: Creating a Full-Lifecycle Request Pipeline at Nationwide Insurance". In: Agile. Washington, DC, USA: IEEE Computer Society. 2009, pp. 223-228.

[FOX08] FOX, D.; SILLITO, J.; MAURER, F. "Agile Methods and User-Centered Design: How These Two Methodologies are Being Successfully Integrated in Industry". In: Agile. Washington: IEEE Computer Society. 2008, pp. 63-72.

[GAR03] GARRET, J. J. "The Elements of User Experience". [S.l.]: AIGA/New Riders, 2003.

[HAI07] HAIKARA, J. "Usability in agile software development: extending the interaction design process with personas approach". In: XP. Berlin, Heidelberg: Springer-Verlag. 2007, pp. 153-156.

[HAK03] HAKIM, J.; SPITZER, T.; ARMITAGE, J. "Sprint: Agile specifications in Shockwave and Flash". New York, NY, USA: ACM. 2003, pp. 1-14.

[HOD05] HODGETTS, P. "Experiences Integrating Sophisticated User Experience Design Practices into Agile Processes". In: Agile Development Conference - ADC. Washington, DC, USA: IEEE Computer Society. 2005, pp. 235-242.

[HOL05] HOLZINGER, A. *et al.* "From extreme programming and usability engineering to extreme usability in software engineering education (XP+UE->XU))". In: International Computer Software and Applications Conference - COMPSAC. Washington, DC, USA: IEEE Computer Society. 2005, pp. 169-172.

[HOV05] HOVE, S. E.; ANDA, B. "Experiences from Conducting Semi-structured Interviews in Empirical Software Engineering Research". Como: IEEE Computer Society. 2005.

[HUD03] HUDSON, W. "Adopting user-centered design within an agile process: a conversation". *Cutter IT Journal*, 2003,

[HUS08] HUSSAIN, Z. *et al.* "Agile User-Centered Design Applied to a Mobile Multimedia Streaming Application". In: USAB. Berlin, Heidelberg: Springer-Verlag. 2008, pp. 313-330.

[HUS08a] HUSSAIN, Z. *et al.* "User Interface Design for a Mobile Multimedia Application: An Iterative Approach". In: International Conference on Advances in Computer-Human Interaction. Washington, DC, USA: IEEE Computer Society. 2008, pp. 189-194.

[HUS09] HUSSAIN, Z. *et al.* "Integration of Extreme Programming and User-Centered Design: Lessons Learned". *Agile Processes in Software Engineering and Extreme Programming*, vol. 31, 2009, pp. 174-179.

[HUS09b] HUSSAIN, Z.; SLANY, W.; HOLZINGER, A. "Current State of Agile User-Centered Design: A Survey". In: USAB. Berlin, Heidelberg: Springer-Verlag. 2009, pp. 416-427.

[ILL09] ILLMENSEE, T.; MUFF, A. "5 Users Every Friday: A Case Study in Applied Research". In: Agile Conference. Washington, DC, USA: IEEE Computer Society.

2009, pp. 404-409.

[JOH11] JOHNSON, A. P. "A Short Guide to Action Research". 4th Edition. ed., New Jersey: Pearson Education, 2011.

[JOK04] JOKELA, T.; ABRAHAMSSON, P. "Usability Assessment of an Extreme Programming Project: Close Co-operation with the Customer Does Not Equal to Good Usability". *Lecture Notes in Computer Science - Product Focused Software Process Improvement*, vol. 3009, 2004, pp. 393-407.

[KAN03] KANE, D. "Finding a Place for Discount Usability Engineering in Agile Development: Throwing Down the Gauntlet". In: Agile Development Conference. Washington, DC, USA: IEEE Computer Society. 2003, pp. 40-46.

[KIM08] KIM, Y. Y. *et al.* "The Application of Biosignal Feedback for Reducing Cybersickness from Exposure to a Virtual Environment". *Presence: Teleoperators and Virtual Environments*, Massachusetts, vol. 17-1, February 2008, pp. 1-16.

[KIT07] KITCHENHAM, B.; CHARTERS, S. "Guidelines for performing Systematic Literature Reviews in Software Engineering", Keele University and Durham University Joint Report, 2007, 57p.

[KOH11] KOHN, M. "Mountain Goat Software". An Overview of Scrum for Agile Software Development, 2011. Capturado em: <http://www.mountaingoatsoftware.com/scrum/overview>, December 2011.

[KOL09] KOLLMANN, J.; SHARP, H.; BLANDFORD, A. "The Importance of Identity and Vision to User Experience Designers on Agile Projects". In: Agile Conference. Washington, DC, USA: IEEE Computer Society. 2009, pp. 11-18.

[KRO09] KROHN, T.; KNDSMÜLLER, M. C.; HERCZEG, M. "User-Centered Design Meets Feature-Driven Development: An Integrating Approach for Developing the Web Application myPIM". In: Human Centered Design - HCII. Berlin, Heidelberg: Springer-Verlag. 2009, pp. 739-748.

[LAR04] LARMAN, C. "Agile & Iterative Development: A Manager's Guide". [S.l.]: Addison-Wesley, 2004.

[LEE07] LEE, J. C.; MCCRICKARD, D. S. "Towards Extreme(ly) Usable Software: Exploring Tensions Between Usability and Agile Software Development". In: Agile Conference. Washington, DC, USA: IEEE Computer Society. 2007, pp. 59-71.

[LET05] LETHBRIDGE, T. C.; SIM, S. E.; SINGER, J. "Studying Software Engineers: Data Collection Techniques for Software Field Studies". *Empirical Softw. Eng.*, 2005, pp. 311-341.

[MCI05] MCINERNEY, P.; MAURER, F. "UCD in Agile Projects: Dream Team or Odd Couple?" *Interactions*, 2005, pp. 19-23.

[MES06] MESZAROS, G.; ASTON, J. "Adding Usability Testing to an Agile Project". In: Agile Conference. Washington, DC, USA: IEEE Computer Society. 2006, pp. 289-294.

[MIL05] MILLER, L. "Case Study of Customer Input For a Successful Product". Washington, DC, USA: IEEE Computer Society. 2005, pp. 225-234.

[NAJ08] NAJAFI, M.; TOYOSHIBA, L. "Two Case Studies of User Experience Design and Agile Development". In: Agile Conference. Washington, DC, USA: IEEE Computer Society. 2008, pp. 531-536.

[NIE94] NIELSEN, J.; MACK, R. L. "Usability Inspection Methods". New York: John Wiley & Sons, 1994.

[OBE08] OBENDORF, H.; FINCK, M. "Scenario-based usability engineering techniques in agile development processes". In: CHI. New York, NY, USA: ACM. 2008, pp. 2159-2166.

[PAE08] PAELKE, V.; NEBE, K. "Integrating agile methods for mixed reality design space

exploration". 2008, pp. 240-249.

[PAT02] PATTON, J. "Hitting the Target: Adding Interaction Design to Agile Software Development". In: OOPSLA 2002 Practitioners Reports (OOPSLA '02). New York: ACM. 2002.

[PAT05] PATTON, J. "Finding the forest in the trees". In: OOPSLA. New York, NY, USA: ACM. 2005, pp. 266-274.

[PEI09] PEIXOTO, C. S. A.; DA SILVA, A. E. A. "A Conceptual Knowledge Base Representation for Agile Design of Human-Computer Interface". In: IITA'09: Proceedings of the 3rd international conference on Intelligent information technology application. Nanchang: IEEE Press. 2009, pp. 156-160.

[PRE92] PRESSMANN, R. S. "Software Engineering: A Practitioner's Approach". New York: McGraw-Hill, 1992.

[PRE94] PREECE, J. *et al.* "Human-Computer Interaction". Harlow: Addison-Wesley, 1994.

[ROB11] ROBSON, C. "Real World Research: a resource for users of social research methods in appled settings". 3rd. ed., Chichester: John Wiley & Sons Ltd, 2011.

[RUB94] RUBIN, J. "Handbook of Usability Testing: how to plan, design and conduct effective tests". New York: John Wiley & Sons, 1994.

[RUN09] RUNESON, P.; HÖST, M. "Guidelines for conducting and reporting case study research in software engineering". *Empirical Softw. Engg.*, Hingham, vol. 14-2, April 2009, pp. 131-164. ISSN 1382-3256.

[SCH01] SCHWABER, K.; BEEDLE, M. "Agile Software Development with Scrum". [S.l.]: Prentice Hall, vol. 18, 2001.

[SHA07] SHARP, H.; ROGER, Y.; PREECE, J. "Interaction Design: beyond human-computer interaction". 2nd Edition. ed., [S.l.]: John Wiley & Sons, 2007.

[SIN08] SINGH, M. "U-SCRUM: An Agile Methodology for Promoting Usability". In: Agile Conference. Washington, DC, USA: IEEE Computer Society. 2008, pp. 555-560.

[SOH10] SOHAIB, O.; KHAN, K. "Integrating Usability Engineering and Agile Software Development: A Literature Review". *Computer Design and Applications ICCDA 2010 International Conference on*, vol. 2, 2010,

[SOM07] SOMMERVILLE, I. "Software Engineering". 8th Edition. ed., Harlow: Addison Wesley, 2007.

[STR07] STRINGER, E. T. "Action Research". 3rd Edition. ed., Thousand Oaks: Sage Publications, 2007.

[STR90] STRAUSS, A.; CORBIN, J. "Basics of Qualitative Research: Grounded Theory Procedures and Techniques". London: SAGE Publications Inc., 1990.

[SYD07] SY, D. "Adapting Usability Investigations for Agile User-centered Design". *Journal of Usability Studies*, vol. 2-3, May 2007, pp. 112-132.

[SYD08] SY, D.; MILLER, L. "Optimizing agile user-centred design". In: CHI. New York, NY, USA: ACM. 2008, pp. 3897-3900.

[UNG08] UNGAR, J. "The Design Studio: Interface Design for Agile Teams". In: Agile. Washington, DC, USA: IEEE Computer Society. 2008, pp. 519-524.

[VER10] VERSION ONE. "State of Agile Survey 2010", Version One Simplifying Software Delivery, 2010, 4p.

[WIL07] WILLIAMS, H.; FERGUSON, A. "The UCD Perspective: Before and After Agile". In: Agile Conference. 2007, pp. 285-290.

[WIL07a] WILLIAMS, H.; FERGUSON, A. "The UCD Perspective: Before and After Agile". In: Agile Conference. Washington D.C.: IEEE Computer Society. 2007, pp. 285-290.

[WOL08] WOLKERSTORFER, P. *et al.* "Probing an agile usability process". In: CHI. New York, NY, USA: ACM. 2008, pp. 2151-2158.

APPENDIX A – List of papers of the Systematic Review

The column Sources indicates by which source the paper was returned:

1. ACM
2. IEEExplore
3. Scopus
4. Springer
5. Web of Science
6. Citeseer
7. Compendex
8. Google Scholar
9. SLR = paper from the previous version of the Systematic Review

ID	Title	Authors	Year	Sources
P1	Building agility for developing agile design information systems	Reich, S Konda, E Subrah- manian	1999	8
P2	Process agility and software usability: Toward lightweight usage-centered design	Constantine	2002	6-8
P3	Designing Requirements: Incorporating Usage- Centered Design into an Agile SW Development Process	Patton	2002	SLR
P4	Adopting user-centered design within an agile process: a conversation	Hudson, W.	2003	3-7
P5	Sprint: Agile specifications in Shockwave and Flash	Hakim, Jack; Spitzer, Tom; Armitage, John	2003	1-3-7
P6	Are agile methods good for design?	J Armitag	2004	7-8
P7	Focusing Extreme Programming on Usability	R Carbon, J Drr	2004	8
P8	Integrating User-Centered Design and Software Engineering: A Role for Extreme Programming?	H Sharp, H Robinson	2004	8
P9	An agile customer-centered method: rapid contextual design	H Beyer, K Holtzblatt	2004	3-5-7-8
P10	Finding the forest in the trees	Jeff Patton	2005	1
P11	Case study of customer input for a successful product	Miller, Lynn	2005	1-2-5-6-7
P12	Experiences integrating sophisticated user experience design practices into Agile processes	Hodgetts, Paul	2005	1-2-5-7
P13	Bridging the gap - Agile software development and usability	Ambler SW	2006	5
P14	Interaction Designers on eXtreme Programming Teams: Two Case Studies from the Real World	J Ferreira, J Noble	2006	6-8
P15	The role of the interaction designer in an agile software development process	MA Lievesley	2006	1-8
P16	XPnUEDefining and Teaching a Fusion of eXtreme Programming and Usability Engineering	H Obendorf, A Schmolitzky	2006	8
P17	Adding usability testing to an agile project	G Meszaros, J Aston	2006	1-2-3-5-6- 7-8
P18	Towards a framework for integrating agile development and user-centred design	S Chamberlain, H Sharp	2006	3-5-7-8
P19	Adapting usability investigations for agile user- centered design	D Sy	2007	8
P20	Software Development Methodologies, Agile Development and Usability Engineering	D Parsons, R Lal, H Ryu	2007	1-8
P21	Agile development iterations and UI design	Ferreira, J. Noble, J.; Biddle, R.	2007	1-2-3-5-7
P22	Managing UCD within agile projects	Detweiler, Mark	2007	3-7
P23	Towards extreme(ly) usable software:	Lee, J.C.;	2007	1-2-3-5-6-7

	exploring tensions between usability and agile software development	McCrickard, D.S.		
P24	Usability in agile software development: extending the interaction design process with personas approach	J Haikara	2007	1-3-5-7-8
P25	When user experience met agile: a case study	M Budwig, S Jeong	2007	1-3-7-8
P26	Up-Front Interaction Design in Agile Development	Ferreira, Noble, Biddle	2007	SLR
P27	Integrating agile methods for mixed reality design space exploration	V Paelke	2008	1-3-7-8
P28	Integrating Extreme Programming and User-Centered Design	Hussain et al	2008	8
P29	On designing a usable interactive system to support transplant nursing	Narasimhadevara, A.; Rad- hakrishnan, T.; Leung, B.; Jayakumar, R.	2008	1-3-5-7
P30	Agile methods and user-centered design: How these two methodologies are being successfully integrated in industry	Fox, David; Sillito, Jonathan; Maurer, Frank	2008	1-2-3-5-7
P31	Agile user centered design: enter the design studio-a case study	J Ungar	2008	1-3-7-8
P32	Agile user-centered design applied to a mobile multimedia streaming application	Z Hussain, M Lechner, H Milchrahm	2008	1-3-5-7-8
P33	Are you sure? Really? A contextual approach to agile user research	Evnin, J.; Pries, M.	2008	1-2-3-7
P34	Probing an agile usability process	P Wolkerstorfer, M Tsche- ligi, R Sefelin	2008	1-3-7-8
P35	Scenario-based usability engineering techniques in agile development processes	H Obendorf	2008	1-3-7-8
P36	The design studio: Interface design for agile teams	J Ungar	2008	1-2-3-5-7-8
P37	Two case studies of user experience design and agile development	Najafi, Maryam; Toyoshiba, Len	2008	1-2-3-5-7
P38	U-SCRUM: An agile methodology for promoting usability	Singh, Mona	2008	1-2-3-5-7
P39	Using persona with XP at LANDesk software, an avow- cent company	Broschinsky, David; Baker, Lisa	2008	2-3-7
P40	Current state of agile user- centered design: A survey	Z Hussain, W Slany	2009	1-3-5-7-8
P41	From cradle to sprint: Creating a full-lifecycle request pipeline at nationwide insurance	Fisher, Kevin G.; Bankston, Arlen	2009	1-2-3-5-7
P42	Investigating agile user- centered design in practice: A grounded theory perspective	Hussain, Zahid; Slany, Wolfgang; Holzinger, Andreas	2009	1-3-5-7
P43	Satisfying Stakeholders' Needs Balancing Agile and Formal Usability Test Results	J Winter	2009	6-8
P44	5 users every Friday: A case study in applied research	Illmensee, Tom; Muff, Alyson	2009	1-2-3-5-7
P45	Adopting an agile culture: A user experience team's journey	Cho, Lily	2009	1-2-3-7-5
P46	Easing team politics in agile usability: A concept mapping approach	Barksdale, Jeremy T.; Ragan, Eric D.; McCrickard, D. Scott	2009	5-6-1-2-3-7
P47	Examining the foundations of agile usability with extreme scenario-based design	Lee, Jason Chong; M.C.- Crickard, D. Scott; Stevens, K. Todd	2009	1-2-3-5-6-7
P48	Integration of extreme programming and user- centered design: Lessons learned	Z Hussain, H Milchrahm, S Shahzad, W Slany	2009	5-7-8
P49	Little Design Up-Front: A Design Science Approach to Integrating Usability into Agile Requirements Engineering	S Adikari, C McDonald	2009	1-3-5-7-8
P50	Successful User Experience in an Agile Enterprise Environment	M Federoff	2009	1-3-5-7-8
P51	The importance of identity and vision to user experience designers on agile projects	Kollmann, Johanna; Sharp, Helen;	2009	1-2-3-5-7

		Blandford, Ann		
P52	User-Centered Design Meets Feature-Driven Development: An Integrating Approach for Developing the Web Application myPIM	T Krohn, M Kindsmller	2009	8-1-3-5-7
P53	Concept and Design of a Contextual Mobile Multi- media Content Usability Study*	Hussain et al	2009	SLR
P54	AFFINE for Enforcing Earlier Consideration of NFRs and Human Factors When Building Socio-Technical Systems Following Agile Methodologies	Bourimi, M.; Barth, T.; Haake, J.M.; Ueberschar, B.; Kesdogan, D.	2010	7
P55	An incremental approach for building accessible and usable web applications	Medina, Nuria Medina; Bu- rella, Juan; Rossi, Gustavo; Grigera, Julin; Luna, Este- ban Robles	2010	5-7
P56	Design and Development in the ”Agile Room”: Trialing Scrum at a Digital Agency	Tzanidou K, Ferreira J	2010	5
P57	The Focus on Usability in Testing Practices in Indus- try	M Larusdottir, E Bjar- nadottir	2010	8
P58	Values and Assumptions Shaping Agile Development and User Experience Design in Practice	J Ferreira, H Sharp	2010	5-8
P59	Evaluating eXtreme scenario-based design in a distributed agile team	Lee, Jason Chong; Judge, Tejinder K.; Scott Mc- Crickard, D.	2010	1-7
P60	User experience design and agile development: managing cooperation through articulation work	J Ferreira, H Sharp	2011	8
P61	UX Design and Agile: A Natural Fit?	Gosper, Julian; Agathos, Jean-Luc; Rutter, Richard; Coatta, Terry	2011	7