

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/224256376>

User-Centered Design and Agile Methods: A Systematic Review

Conference Paper · September 2011

DOI: 10.1109/AGILE.2011.24 · Source: IEEE Xplore

CITATIONS

147

READS

2,064

4 authors, including:



Tiago Silva Da Silva

Universidade Federal de São Paulo

37 PUBLICATIONS 337 CITATIONS

[SEE PROFILE](#)



Frank Maurer

The University of Calgary

293 PUBLICATIONS 4,493 CITATIONS

[SEE PROFILE](#)



Milene Silveira

Pontifícia Universidade Católica do Rio Grande do Sul

159 PUBLICATIONS 565 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Collaboration Meets Interactive Spaces: A Springer Book [View project](#)



Data Visualization of Social Networks [View project](#)

User-Centered Design and Agile Methods: A Systematic Review

Tiago Silva da Silva¹, Angela Martin², Frank Maurer³, Milene Silveira¹

¹PUCRS, Porto Alegre, RS, Brazil. tiago.silva@pucrs.br, milene.silveira@pucrs.br

²Waikato University, Hamilton, New Zealand. angela@cs.waikato.ac.nz

³University of Calgary, Calgary, AB, Canada. frank.maurer@ucalgary.ca

Abstract—This paper presents the results of a systematic review of existing literature on the integration of agile software development with user-centered design approaches. It shows that a common process model underlies such approaches and discusses which artifacts are used to support the collaboration between designers and developers.

Keywords – *systematic review; agile methods; user-centered design; integration.*

I. INTRODUCTION

Agile Methods have a distinct culture that at first glance seems to conflict with User-Centered Design (UCD) [1]. However, according to these same authors, the use of agile methods can result in improved usability. Moreover, the authors did not find any interaction designers who preferred traditional approaches than agile processes.

One of the problems of integrating these two methodologies is that traditionally they use different approaches on how resources are allocated in a project [2]. Agile methods strive to deliver small sets of software features to customers as quickly as possible in short iterations. On the other hand, UCD spend a considerable effort on research and analysis before development begins.

For example, UCD associated with non-agile teams has led to a combination of results [3]. In non-agile projects, the UCD group has written UI (User Interface) specifications that are Word documents ranging from 5 to 200 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 involve a local representative of the client to shorten the feedback loop between the development team and the customer.

This paper presents the results of a systematic review about existing literature on the integration of agile methods and user-centered design approaches.

It is organized as follows: Section 2 discusses the review method used in this study, Section 3 presents the results of the review and the interpretation of these results, and finally

Section 4 presents our conclusions and suggestions for future work.

II. 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 [4]. 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 [4].

In this work, the systematic review was conducted to provide empirical support for a proposal of a methodology for integration of UCD and Agile, identifying most common practices and artifacts used.

A. 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 paper, we use the term UCD 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.

B. Protocol development

For this systematic review, we used the recommendations of [5] and [4] in a complementary way. Our goal is to identify existing evidence regarding the integration of UCD and Agile. The research questions that guided this review are:

- 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 UCD 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 I. 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
Agile	Agile
	Agile Method
	Agile Development
	Agile Practice
	Agile Project
	Agile Lifecycle
	Scrum
	Extreme Programming
	Lean Development
	Feature Driven Development
	Dynamic System Development
	Agile Unified Process

C. 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"
AND

agile OR "agile method" OR "agile development" OR "agile practice" OR "agile project" OR "agile lifecycle" OR scrum OR "extreme programming" OR "lean development" OR "feature driven development" OR "dynamic system development" OR "agile unified process"

The sources selected for the search were:

- IEEExplore Digital Library
- Elsevier ScienceDirect
- CiteSeer
- Scopus¹

It is worthwhile to mention that each digital library has its own particularities concerning their search engines, therefore, the search strings required to be adapted for each source.

In addition, following the example of [6], we hand-searched all volumes of the following conference proceedings for research papers and experience reports on UCD and Agile:

- XP
- XP/Agile Universe

- Agile Development Conference

D. Inclusion and exclusion criteria

We searched for industrial experience reports, theoretical, empirical papers and experimental papers, in other words, for papers that had peer reviewed.

To include a paper in the analysis (inclusion criteria), the paper must have been peer reviewed, must have been available online, must have been written in English, and must have reported on the integration of UCD and Agile. The papers were classified following a two-step approach. First, based on the reading of papers title, and abstract, the papers were classified according to the inclusion criteria.

All the papers that clearly did not match the inclusion criteria were excluded, while the others were analyzed more carefully based on the reading of introduction and conclusion. We kept only those papers discussing the integration of UCD and Agile.

One researcher applied the search strategy to identify the primary papers, and filtered the identified papers, by reading the abstract. This was followed by a reading of the full text, and a second classification step was executed, checking whether the inclusion/exclusion criteria were satisfied. In case of any conflict, a second researcher made the verification. The papers were classified according to two general categories of information, following the recommendations of [5]:

- Descriptive information: theoretical, experience reports, empirical or experimental.
- Content-related information: focus on (evaluation, design or both), approach (specialist, generalist or specialist/generalist), results (need of an approach, proposed approach, lessons learned or recommendations) and conclusions.

E. 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 [5], 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, it consists of those results that cannot be extracted directly from the study. These information were extracted as follows: additional information through authors (if the reviewer contacted the study's author to solve doubts or ask for more details about it) and general impressions and abstractions.

III. RESULTS

The search in the digital libraries was conducted in August 2010. A total of 309 papers were found, as presented in Table 2.

Inclusion: indicating the papers collected and possibly related to integration of UCD and Agile.

¹ As Scopus seems to cover ACM and Springer, we chose to use this digital library.

Exclusion: indicating the papers collected but not related to integration of UCD and Agile.

TABLE II. SEARCH EXECUTION, FIRST RESULTS

Digital Library	Amount of Papers	First Classification	
		Inclusion	Exclusion
IEEEExplore	59	24	35
Elsevier Science Direct	4	0	4
CiteSeerX	59	0	59
Scopus	154	24	130
Agile	21	21	0
XP	12	12	0
Total	309	81	228
Percentage	100%	26%	74%

After the initial filtering, 81 of 309 papers were selected for further evaluation. Several papers were returned in multiple searchers. They are listed as “repeated” in the Table 3. Some were excluded from further evaluations after a more thorough reading because we noticed that they were not relevant for our research questions.

A lot of papers were retrieved from the search engines of the digital libraries but they did not describe integration of UCD and Agile Methods. For example, as the term “human factor” was used, a paper on how human factor poses new challenges for organizations that intent on adopting agile methods was retrieved. But this paper does not fit our inclusion criteria.

Therefore, we had a set of 58 papers for analysis. These papers are listed at the following link: <http://www.silvadasilva.com/home/systematicreview>.

TABLE III. FINAL SET OF PAPERS TO BE ANALYZED

Digital Library	Papers	Second Classification		Total Selected
		Repeated	Not Relevant	
Total	81	16	7	58
Percentage	100%	20%	9%	72%

We read 58 articles and derived objective and subjective information.

It is worthwhile to mention that four papers had no proposals for integration, but still were kept because we considered them important background information: [7] presents a summary of a tutorial held at ICSE 2003, [8] presents a summary of a workshop held at ICSE 2005, [9] presents a brief report of a panel held at CHI 2008 and [10] presents the description of a special interest group (SIG) on Agile UCD.

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 authors. To increase internal validity, two

researchers performed the classifications and then discussed differences to solve any possible disagreement.

A. Quantitative analysis

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 concern with issues related to usability, it is interesting to note the number of articles published every year. This information is presented in Fig. 1.

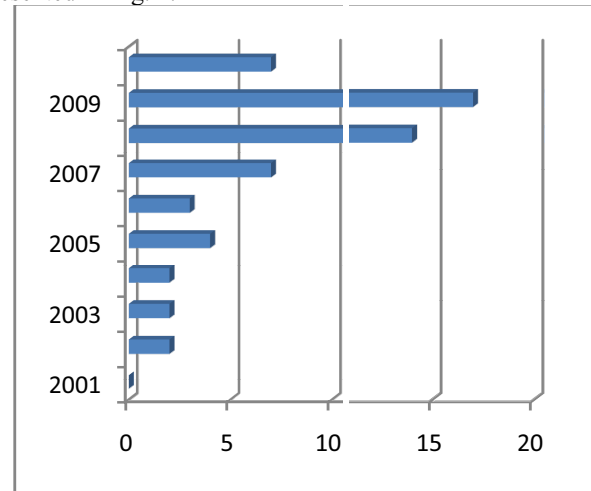


FIGURE 1 - PAPERS PUBLISHED BY YEAR

The fact that the year 2010 presents only seven published papers does not necessarily mean a drop down in interest in the topic, because the literature search was conducted in August 2010.

Descriptive information: We considered as theoretical those papers which presented reflections about the integration of the two methods but without any validation. We considered as experience reports those papers which presented some practical experience of an integrated approach. Papers which attempted to insert some aspect of UCD in Agile, but which were not a controlled experiments are classified as experimental. Fig. 2 presents the number of papers published by year in relation to the type of the research.

21 of the papers were theoretical and 20 were empirical. 19 papers were also classified as experience reports. Only one paper was classified as experimental. Additionally some papers were classified under multiple types as follows. Paper [11] performed usability evaluations in a controlled process of development using XP "by the book" just for this purpose, to check how the inclusion of usability evaluation in an XP project works. Three papers were classified as theoretical and empirical [12], [13] and [14], and one paper was classified as empirical and experience report [15].

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

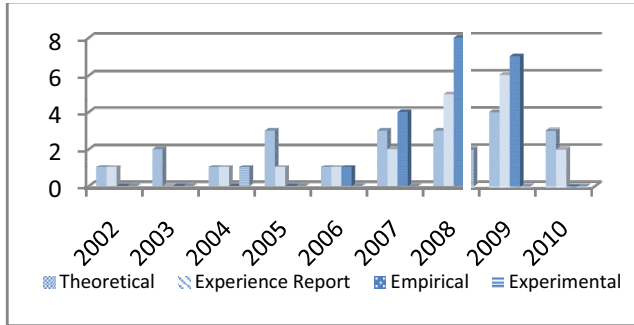


FIGURE 2 - DESCRIPTIVE INFORMATION

Regarding the focus, the papers were classified as “Design” if the focus of the integration of UCD and Agile of the paper was only on the design, including User Research. “Evaluation” if the focus was only on the UI evaluation or “Both” if the focus was on design and evaluation. Fig. 3 presents the results.

We considered that our classification would not apply to paper [16], because the focus of this work was neither evaluation nor design.

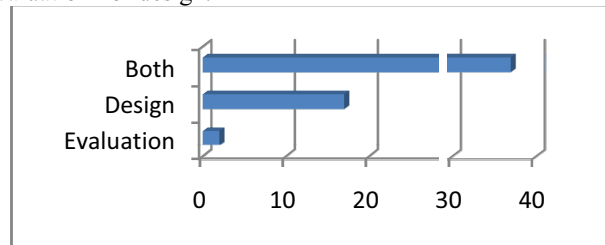


FIGURE 3 - CONTENT-RELATED INFORMATION: FOCUS

Regarding the approach used, we classified the work as a specialist approach, or Generalist Generalist/Specialist, as proposed by [2]. Specialist means that the team used specialists for UCD work. A generalist approach has all team members fulfilling both roles. And Generalist/Specialist is a hybrid approach in which some development team members fulfill both roles but not all. In 11 studies, we found that this type of classification did not apply. Figure 4 clearly indicates that most studies used specialists.

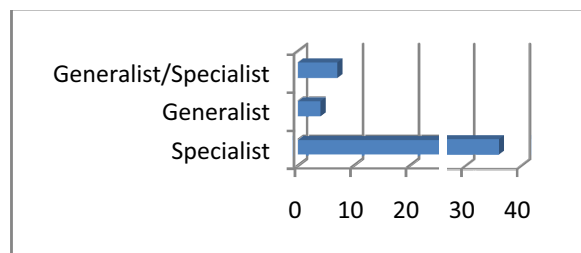


FIGURE 4 - CONTENT-RELATED INFORMATION: APPROACH

Concerning the results of each paper, they were classified into four categories: the need of initiative (those papers that concluded that there is a need for a proposal to integrate with Agile UCD), initiative proposal (those that propose an integration of UCD 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 Fig. 5.

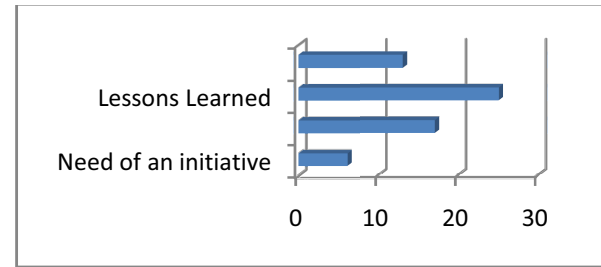


FIGURE 5 - CONTENT-RELATED INFORMATION: RESULTS

Most papers (25) present lessons learned. 17 present initiative proposals and 13 present recommendations. Additionally some papers had multiple classifications. [17] was classified as lessons learned and also recommendations; [18] was classified as the need of an initiative, lessons learned and recommendations, and [19] was classified as initiative proposal and lessons learned.

We also identified common artifacts, practices and needs that were used in the studies:

- 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: indication that working with Agile improved collaboration and communication between UCD 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: use of User Stories by the UCD team, creating them or enriching them.
- Inspection Methods: the use of usability inspection methods.
- One Sprint Ahead: indication that the UCD team must work at least one iteration ahead of the development team.
- Big Picture: recommendation to not lose the holistic view of the project (Big Picture).
- Scenarios: 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: indication that the UCD team must work in parallel to the development team, in the same iteration.
- Interaction Models: use of interaction models;
- Guidelines: use of design guidelines.

- Essential Use Cases: use of Essential Use Cases proposed by [20].

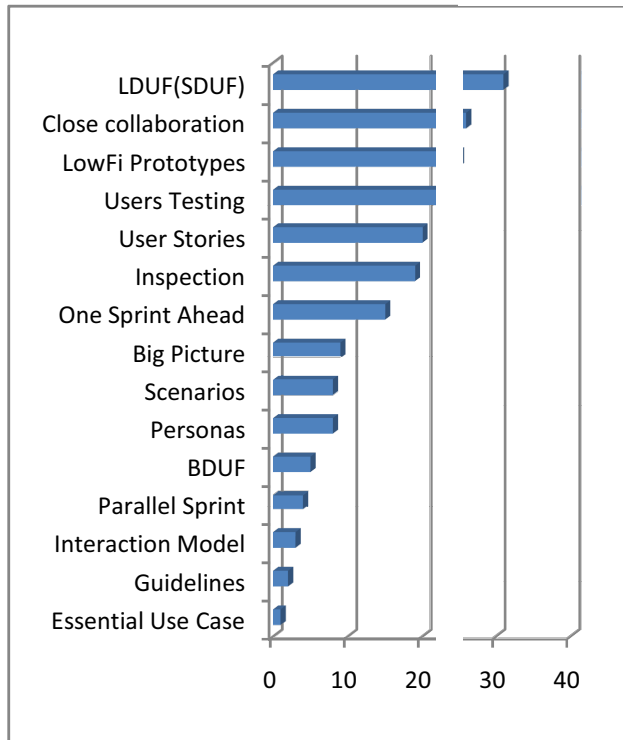


FIGURE 6 - CONTENT-RELATED INFORMATION: RESULTS

From 15 topics identified in the papers, only 2 are seen as problematic by the authors. Teams losing sight of “Big Picture” of a project is sometimes perceived as a problem with agile methods generally, not just when integrating UCD 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 [21]. All other papers suggest LDUF because according to them, BDUF goes against agile principles.

The quantitative results of this classification are presented in Fig. 6. 31 papers recommend the use of LDUF but strictly limit the upfront effort. 25 studies indicate the use of low-fidelity prototypes in the agile development process. 19 papers comment on usability inspection methods and 22 on users testing. 15 studies indicate that the UCD team should work one iteration ahead of the development team. 20 studies indicate that User Stories should capture usability issues. 26 papers report that the (attempted) integration of UCD and Agile has improved communication and collaboration between the development and design teams.

B. Qualitative Analysis

We now want to identify some key aspects (as highlighted by the primary literature) concerning the integration of UCD and Agile:

- Little Design Up Front;
- Prototyping;
- User Stories;

- User testing;
- Inspection evaluation;
- One sprint ahead.

Little Design Up Front: Concerning Little Design Up Front), [11], [7], [22], [23], [24], [25], [18], [12], [26], [27] and [28] just mention that BDUF is not an option regarding agile methods, but they do not comment which artifacts or practices should be used. [17] and [3] suggest that activities related to the UI design should be performed before the official kickoff of the project. Additionally, [3] 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. [29], [30], [31], [2] and [32] suggest the use of Sprint 0 for contextual inquiry and user interviews. [31] also suggests that this should be done before the Planning Game, then usability aspects can be discussed during the course of the Planning Game. [33] suggest that usability aspects should be discussed during the Planning Game without previous discussions. [32] suggest the creation of personas in Sprint 0. [34] 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. [35] 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, [14] proposes the creation of a specific product owner for usability aspects and also User Stories that contemplate usability criteria. [36] reports the use of Contextual Inquiry as one-on-one interviews conducted at the user's workplace that focus on observations of ongoing work. [37] 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 UCDS 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 UCDS gather a list of epics and features that will require UI designs for the upcoming Sprint from product leads. [38] proposes the integration of Rapid Contextual Design and Agile. The authors assume that customer representatives working with a team of at least two UCDS will play the customer role. [39] 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.

Prototyping: Concerning prototyping, [30], [24] and [18] comment that it is important to prototype. [40], [2], [35], [33], [17], [3], [31] and [41] 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 UCDSs, and the use of such prototypes for usability evaluations both by inspection and by user testing. [31] and [3] suggest that the prototype evolves into a high-fidelity prototype. [42] and [13] comment that prototypes can be derived from the User Stories, and [43] suggests the construction of prototypes from personas created in his approach. All previous approaches, including [26] and [21] suggest that UCDSs teams should develop UI prototypes one sprint ahead of the development team. While [9] suggests that teams work in parallel. [44] comments that sketches in addition to User Stories can be used as means of revealing errors, temporal information such task sequence, contextual information etc. In [45], 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. [46] and [47] use low-fi paper prototypes and high-fi prototypes to perform inspection evaluations and usability tests with the customer.

User Testing: Concerning Users Testing, [35], [12], [2], [19], [47] and [33] mention or suggest to perform Users Testing on paper prototypes, however, only [33] comments which kind of test to use, recommending the use of Thinking Aloud. [19] recommend the use of scenarios to guide the user testing. [3] and [9] recommend the execution of users testing on interactive prototypes. All of them aimed at refining the UI prototype for the next iteration. [32] and [34] recommend users testing whenever possible, but they do not comment whether the tests are performed on prototypes or on working software. Only [32] points out that user testing is performed with the customer. [36] reports that they use lightweight usability testing which consists of the capture of screens and mouse movements, the use of Thinking Aloud method and recording users' comments. [21], [12], [40], [23], [48], [17] and [49] recommend user testing on the working software. Whereas [21] and [12] suggest to perform user testing to validate the UI, [21] and [49] comment that usability testing should be integrated into the acceptance tests. [48] suggests that user testing should be performed during the Sprint Reviews and [17] 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. [41] reports that they conduct usability tests on low-fi and high-fi prototypes and [50] mention that usability tests can be conducted, but in a lightweight form and not inside a usability laboratory. [38] suggests that the UI should be tested with the users using paper, with mock-up 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 you have time and resources.

User Stories: Concerning User Stories, [51] and [12] comment that User Stories should be originated from Usability Scenarios, while [40] suggest that User Stories

should be integrated with scenario-based design. [11] commented that activities such as Task Analysis of users should contribute to the development of User Stories. Whereas [35] suggest that User Stories should be originated from usability tests on paper prototypes. [47] and [2] comment that User Stories could be defined for the construction of prototypes. [33] also comment that User Stories could be used as tasks to be performed by users on user testing using these prototypes. [42] suggests that User Stories should only provide details for the construction of prototypes while [45] reports the integration of prototypes and User Stories. [48] comments that Product Backlog and User Stories are the best places to capture usability requirements. While [14] mentions that User Stories should contain the usability issues in their acceptance criteria. [38] suggest that UI mockups should be part of the User Story definition and acceptance testing criteria. [52] suggests the existence of a specific product owner for usability issues, and they also suggest a specific product backlog for usability aspects. [3] 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. [23] suggests that User Stories should always be fed with the results of user tests performed at the end of each sprint. [46] mentions that User Studies should be used to develop User Stories and that UCDS 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.

Usability Inspection: [3], [2], [47], [26] and [41] suggest or mention the use of usability evaluation on paper prototypes, always with the goal of refining the UI for the next iteration. [42] suggest that such evaluations should be carried out until the prototype is stable to serve as a basis for the UI implementation. [19] also suggests an evaluation of paper prototypes, but guided by scenarios. [9] suggest inspection evaluations on prototypes, but focusing on interactive prototypes, instead of on paper prototypes. [17], [34], [18], [32] and [21] suggest evaluations on UIs already implemented aiming at their validation. [21] and [49] suggest the use of Heuristic Evaluation, and [48] comments that Sprint Reviews are a good opportunities to conduct usability evaluations. [46] execute inspection evaluations on low-fi and high-fi prototypes to write UI related stories. Finally, [53] reports that developers did UI reviews, and that UI reviews had completely changed the way developers saw the UCDS work. Seeing the work of others from the perspective of somebody who does not care how brilliant the code is, but rather what was being used by people, seemed to have a profound impact on developers.

One Sprint Ahead: Concerning One Sprint Ahead, [31], [32], [18], [3] and [26] suggest that UCDSs teams work one sprint ahead of the development team. [31], [32] and [18]

also suggest that this practice has already started in Sprint 0. But [52], 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. [50] suggests that UCDS 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. [37] 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, UCDSs 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. This way, UCDS should be aligned with the business strategies, participating even before any iteration.

IV. CONCLUSION

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 UCD and Agile Methods. Another important point is that is more common to have a UCDS (or in some cases, 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 UCD.

At a high level, an integrated process and the used artifacts is presented in the Fig. 7. The process is derived from the findings of this systematic review. It is similar to processes described by [54], [2], and [23] in their respective work but it is not the same, because it is a combination of the most common practices and processes identified in the review.

During the Sprint 0, UCDSs 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, UCDSs could Design performing Contextual Inquiry, Task Analysis, Interviews for Sprint 2 and Evaluate performing Inspection Evaluation on the code of the current Sprint and provide feedback still in this Sprint. And using these 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, UCDSs could Design performing Contextual Inquiry, Task Analysis, Interviews for Sprint n and Evaluate 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 these 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 UCDSs on what was coded in Sprint 1 (designed in Sprint 0).

During the Sprint n, UCDSs could Evaluate 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). And they could use the following Artifacts: 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 development in agile projects.

Both in maintaining the Big Picture as the stimulus for this collaboration [55] 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.

A number of conclusions can be drawn from the quantitative and qualitative analysis in this systematic review.

Conclusion 1: The focus of integrating agile methods and UCD should be on design as well as on usability evaluation. For design, personas and low fidelity prototypes are used. Evaluation often happens using low-fi prototypes with the goal to improve design.

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

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

Q1: How are usability issues addressed into Agile projects?

These are addressed in various ways. For example, approaches with UCDS in the development team, approaches without specialists 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.

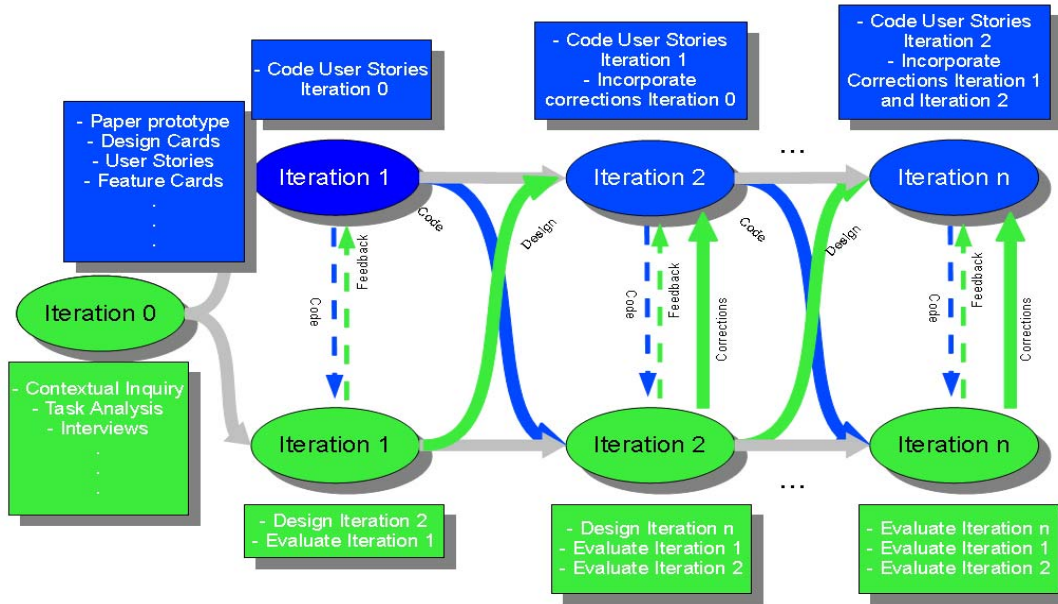


FIGURE 7 - FRAMEWORK

A. Limitations of this review

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

- The number of selected sources, considering that only 4 digital libraries met all the requirements for research. And from these 4, only 2 had papers in the final process of revision. However, we are not aware of any papers that were missed by the search.
- The reliability of the classification method for the papers. Because we did not use an already established classification, proposing a new categorization.
- The primary work underlying this systematic review lacks sound controlled studies.
- Finally, only one researcher had fully read the final set of papers. It may lead the inclusion of bias in the analysis and classifications of the papers.

Another issue already commented that agile methods and user-centered design keywords 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.

B. Final considerations and next steps

We identified 309 studies by searching 4 digital libraries, of which 58 were found to be studies of acceptable rigor and relevance to our study.

The studies were classified regarding their content and research method. Concerning the research, they were classified as experimental, empirical, experience report and theoretical. Regarding content, they were classified considering their approach (Specialist, Generalist or Generalist/Specialist), results (need of an initiative, initiative proposal, lessons learned and recommendations), focus (evaluation, design, both).

The studies were also classified considering the techniques used by the teams that were studies, such as: LDUF, use of personas, use of low-fi prototypes, use of inspection methods, use of user testing, the use of scenarios, use of Use Stories, use of guidelines, if the UCDS 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 process model of software development combining UCD with Agile principles.

According to all the experience reports identified and according to [55], 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.

To validate the process model that resulted from this systematic review, we intend to perform an observational study in an organization to determine if the model captures their actual work process. We are planning to perform action research to help this company to improve its Agile UCD process.

REFERENCES

- [1] P. McInerney and F. Maurer, "UCD in Agile Projects: Dream Team or Odd Couple?," *Interactions*, pp. 19-23, 2005.
- [2] D. Fox, J. Sillito, and F. Maurer, "Agile Methods and User-Centered Design: How These Two Methodologies are Being Successfully Integrated in Industry," in *Proceedings of the Agile 2008*, Washington, 2008, pp. 63-72.
- [3] H. Williams and A. Ferguson, "The UCD Perspective: Before and After Agile," in *Proceedings of the AGILE 2007*, Washington D.C., 2007, pp. 285-290.
- [4] B. Kitchenham and S. Charters, "Guidelines for performing Systematic Literature Reviews in Software Engineering," Keele University and Durham University Joint Report EBSE 2007-001, 2007.
- [5] J. Biolchini, P. G. Mian, C. C. C. Natali, and G. H. Travassos, "Systematic Review in Software Engineering," COPPE / UFRJ, 2005.
- [6] T. Dyba and T. Dingsøyr, "Empirical studies of agile software development: A systematic review," *Inf. Softw. Technol.*, vol. 50, no. 9-10, pp. 833-859, Aug. 2008.
- [7] L. L. Constantine and L. A. D. Lockwood, "Usage-Centered Engineering for Web Applications," *IEEE Softw.*, vol. 19, pp. 42-50, 2002.
- [8] M. John, F. Maurer, and B. Tessem, "Human and Social Factors of Software Engineering - Workshop Summary," *SIGSOFT Softw. Eng. Notes*, vol. 30, no. 4, pp. 1-6, 2005.
- [9] M. Federoff, et al., "Extreme Usability: Adapting Research Approaches for Agile Development," in *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, Florence, 2008, pp. 2269-2272.
- [10] L. Miller and D. Sy, "Agile User Experience SIG," in *CHI '09: Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, Boston, 2009, pp. 2751-2754.
- [11] T. Jokela and P. Abrahamsson, "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, pp. 393-407, 2004.
- [12] J. C. Lee and D. S. McCrickard, "Towards Extreme(ly) Usable Software: Exploring Tensions Between Usability and Agile Software Development," in *Proceedings of the the AGILE 2007*, Washington D.C., 2007, pp. 59-71.
- [13] Z. Hussain, et al., "Agile User-Centered Design Applied to a Mobile Multimedia Streaming Application," in *USAB '08: Proceedings of the 4th Symposium of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society on HCI and Usability for Education and Work*, Graz, 2008, pp. 313-33-.
- [14] M. Singh, "U-SCRUM: An Agile Methodology for Promoting Usability," in *Proceedings of the Agile 2008*, Toronto, 2008, pp. 555-560.
- [15] J. Ungar, "The Design Studio: Interface Design for Agile Teams," in *AGILE Conference Agile 2008*, 2008, pp. 519-524.
- [16] S. Meckem and J. L. Carlson, "Using "Rapid Experimentation" to Inform Customer Service Experience Design," in *CHI EA '10: Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems*, Atlanta, 2010, pp. 4553-4566.
- [17] M. Detweiler, "Managing UCD Within Agile Projects," *Interactions*, vol. 14, pp. 40-42, 2007.
- [18] D. Sy and L. Miller, "Optimizing Agile User-Centred Design," in *CHI '08: CHI '08 extended abstracts on Human factors in computing system*, Florence, 2008, pp. 3987-3900.
- [19] H. Obendorf and M. Finck, "Scenario-Based Usability Engineering Techniques in Agile Development Processes," in *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, Florence, 2008, pp. 2159-2166.
- [20] L. L. Constantine and L. A. D. Lockwood, *Software for use: a practical guide to the models and methods of usage-centered design*. New York: ACM Press/Addison-Wesley Publishing Co., 1999.
- [21] G. Benigni, O. Gervasi, F. L. Passeri, and T.-H. Kim, "USABAGILE Web: A Web Agile Usability Approach for Web Site Design," in *ICCSA (2) - Lecture Notes in Computer Science (LNCS)*, Berlin, 2010, pp. 422-431.
- [22] S. Adikari, C. McDonald, and J. Campbell, "Little Design Up-Front: A Design Science Approach to Integrating Usability into Agile Requirements Engineering," in *Proceedings of the 13th International Conference on Human-Computer Interaction. Part I*, San Diego, 2009, pp. 549-558.
- [23] J. Ferreira, J. Noble, and R. Biddle, "Agile Development Iterations and UI Design," in *Proceedings of the AGILE 2007*, Washington D.C., 2007, pp. 50-58.
- [24] T. Krohn, C. Kindsmüller, and M. Herczeg, "User-Centered Design Meets Feature-Driven Development: An Integrating Approach for Developing the Web Application myPIM," in *Human Centered Design, HCI 2009, LNCS 5619*, San Diego, 2009, pp. 739-748.
- [25] C. S. A. Peixoto and A. E. A. da Silva, "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, 2009, pp. 156-160.
- [26] J. Ungar and J. White, "Agile User Centered Design: Enter the Design Studio - A Case Study," in *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, Florence, 2008, pp. 2167-2178.
- [27] L. Cho, "Adopting an Agile Culture," in *AGILE '09: Proceedings of the Agile 2009*, Chicago, 2009, pp. 400-403.
- [28] J. Ferreira, H. Sharp, and H. Robinson, "Values and Assumptions Shaping Agile Development and User Experience Design in Practice," in *Agile Processes in Software Engineering and Extreme Programming*, Trondheim, 2010, pp. 178-183.
- [29] P. Hodgetts, "Experiences Integrating Sophisticated User Experience Design Practices into Agile Processes," in *Proceedings of the Agile Development Conference*, Denver, 2005, pp. 235-242.
- [30] J. Kollmann, H. Sharp, and A. Blandford, "The Importance of Identity and Vision to User Experience Designers on Agile Projects," in *Proceedings of the 2009 Agile Conference*, Chicago, 2009, pp. 11-18.
- [31] S. Chamberlain, H. Sharp, and N. Maiden, "Towards a

- Framework for Integrating Agile Development and User-Centred Design," in *Extreme Programming and Agile Processes in Software Engineering*, Oulu, 2006, pp. 143-153.
- [32] M. Najafi and L. Toyoshiba, "Two Case Studies of User Experience Design and Agile Development," in *Proceedings of the Agile 2008*, Toronto, 2008, pp. 531-536.
- [33] A. Holzinger, M. Errath, G. Searle, B. Thurnher, and W. Slany, "From Extreme Programming and Usability Engineering to Extreme Usability in Software Engineering Education (XP+UE->XU)," in *COMPSAC '05: Proceedings of the 29th Annual International Computer Software and Applications Conference*, Washington, 2005, pp. 169-172.
- [34] P. Wolkerstorfer, et al., "Probing an Agile Usability Process," in *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, Florence, 2008, pp. 2151-2158.
- [35] G. Meszaros and J. Aston, "Adding Usability Testing to an Agile Project," in *AGILE Conference*, Minneapolis, 2006, pp. 289-294.
- [36] B. Belchev and P. Baker, "Improving Obama Campaign Software: Learning from Users," in *Agile 2009 Conference*, Chicago, 2009, pp. 395-399.
- [37] L. Cho, "Adopting an Agile Culture: A User Experience Team's Journey," in *Agile 2009 Conference*, Chicago, 2009, pp. 416-421.
- [38] H. Beyer, K. Holtzblatt, and L. Baker, "An Agile Customer-Centered Method: Rapid Contextual Design," in *Extreme Programming and Agile Methods - XP/Agile Universe 2004*, Calgary, 2004, pp. 527-554.
- [39] J. Ferreira, J. Noble, and R. Biddle, "Up-Front Interaction Design in Agile Development," in *Agile Processes in Software Engineering and Extreme Programming 2007*, Como, 2007, pp. 9-16.
- [40] O. Sohaib and K. Khan, "Integrating Usability Engineering and Agile Software Development: A Literature Review," in *Computer Design and Applications (ICDDA), 2010 International Conference on*, Qinhuangdao, 2010, pp. V2-32-V2-38.
- [41] L. Miller, "Case Study of Customer Input For a Successful Product," in *Agile 2005*, Denver, 2005, pp. 225-234.
- [42] D. T. Hellmann, A. Hosseini-Khayat, and F. Maurer, "Supporting Test-Driven Development of Graphical User Interfaces Using Agile Interaction Design," in *Software Testing Verification and Validation Workshop, IEEE International Conference on*, Los Alamitos, 2010, pp. 444-447.
- [43] J. Haikara, "Usability in Agile Software Development: Extending the Interaction Design Process with Personas Approach," in *XP'07: Proceedings of the 8th international conference on Agile processes in software engineering and extreme programming*, Como, 2007, pp. 153-156.
- [44] J. Brown, G. Lindgaard, and R. Biddle, "Stories, Sketches, and Lists: Developers and Interaction Designers Interacting Through Artefacts," in *Agile 2008 Conference*, Toronto, 2008, pp. 39-50.
- [45] D. Broschinsky and L. Baker, "Using Persona with XP at LANDesk Software, an Avocent Company," in *Agile 2008 Conference*, Toronto, 2008, pp. 543-548.
- [46] Z. Hussain, et al., "Integration of Extreme Programming and User-Centered Design: Lessons Learned," in *Agile Processes in Software Engineering and Extreme Programming 2009*, Pula, 2009, pp. 174-179.
- [47] Z. Hussain, et al., "User Interface Design for a Mobile Multimedia Application: An Iterative Approach," in *ACHI '08: Proceedings of the First International Conference on Advances in Computer-Human Interaction*, Washington, 2008, pp. 189-194.
- [48] M. DÜchting, D. Zimmermann, and K. Nebe, "Incorporating User Centered Requirement Engineering into Agile Software Development," in *HCI'07: Proceedings of the 12th international conference on Human-computer interaction*, Beijing, 2007, pp. 58-67.
- [49] D. Kane, "Finding a Place for Discount Usability Engineering in Agile Development: Throwing Down the Gauntlet," in *Proceedings of the Conference on Agile Development*, Salt Lake City, 2003, pp. 40-.
- [50] T. Illmensee and A. Muff, "5 Users Every Friday: A Case Study in Applied Research," in *Agile 2009 Conference*, Chicago, 2009, pp. 404-409.
- [51] J. T. Barksdale and D. S. McCrickard, "Concept Mapping in Agile Usability: A Case Study," in *CHI EA '10: Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems*, Atlanta, 2010, pp. 4691-4694.
- [52] M. Budwig, S. Jeong, and K. Kelkar, "When User Experience Met Agile: A Case Study," in *CHI '09: Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, Boston, 2009, pp. 3075-3084.
- [53] M. Albisetti, "Launchpad's Quest for a Better and Agile User Interface," in *Agile Processes in Software Engineering and Extreme Programming*, Trondheim, 2010, pp. 244-250.
- [54] D. Sy, "Adapting Usability Investigations for Agile User-centered Design," *Journal of Usability Studies*, vol. 2, no. 3, pp. 112-132, 2007.
- [55] H. Beyer, *User-Centered Agile Methods*, J. M. Carrol, Ed. Morgan & Claypool Publishers, 2010.