

Agile and UCD Integration Based on Pre-development Usability Evaluations: An Experience Report

Jade Mendes Inácio de Carvalho¹, Tiago Silva da Silva^{1(✉)},
and Milene Selbach Silveira²

¹ ICT-UNIFESP, Universidade Federal de São Paulo, São Paulo, Brazil
jade.sj2@gmail.com, silvadasilva@unifesp.br

² PUCRS, Faculdade de Informática, Porto Alegre, Brazil
milene.silveira@pucrs.br

Abstract. Agile Methods and User-Centered Design have radically transformed the way people develop software, mostly by taking more focus on people than on processes. This study is focused on understanding the integration of Agile and User-Centered Design through usability evaluations before the development cycles. Based on a pilot study carried out in a Scrum web-based software development team, we observed great results regarding to rework decrease as well as the usability improvement of the developed system throughout its development cycle. In the context of this study, we concluded that, besides reducing rework and improving usability, pre-development usability evaluations resulted in a mindset change, affecting both the development team and stakeholders, helping them to understand the importance of this early integration of Agile and User-Centered Design.

Keywords: Agile · User-Centered Design · Evaluation · Pre-development

1 Introduction

Agile Methods have been transforming the way people work with software development mostly by taking more focus on people than processes. Similarly, the techniques that compose the User-Centered Design (UCD) methodology have been contributing to enable a simpler and more intuitive human-computer interaction, focusing on user experience and accessibility.

There are several studies around the integration of both methodologies, Agile and UCD, that aim to apply them together in a software development environment [11]. Basically because it is expected that once applied together they can bring great improvements to interactive computing systems. However, there is still a lot to be discussed about how to perform this integration.

One of them is applying usability evaluations on mockups before the development starts in order to anticipate possible defects and avoid rework. Based on

this statement, the main goal of this study is to understand if the integration of UCD and Agile through usability evaluations before development cycles is useful to eliminate rework.

The study was divided into two phases, the first dedicated to a systematic review based in the work of da Silva et al. [12] and the second dedicated to the implementation of a study using a set of methods identified in the review in order to collect results and lessons learned.

The review of da Silva et al. [12] concentrate papers about methods to integrate Agile Methodology and UCD from the most famous HCI conferences around the world. In order to select the studies that should be analyzed in the systematic review we filtered the papers according to the timing of usability evaluation execution. We considered only the papers related to methods that propose usability evaluations before the development cycles during the quantitative and qualitative analysis.

In the second phase, we ran a pilot study using a set of methods from the systematic review applied according to project's context. The lessons learned gathered during this experience transcended the expected benefits and brought not only rework reduction but also a new and collaborative mindset to the teams involved.

The remaining of this paper is organized into four sections as follows: the second section is regarding to the systematic review results, the third one is a description about the pilot study and the application of the methods, the fourth one is related to the results and lessons learned and the last section presents the conclusions of this study.

2 Systematic Review

As aforementioned, the first phase of this study was the implementation of a systematic review based on the study by da Silva et al. [12] in order to map usability evaluation methods applied in a pre-development stage in an agile environment.

Starting from a total of 46 papers about the integration of Agile and UCD, only 17 were identified by containing evaluation methods in the timing expected. The next two subsections present the quantitative and qualitative analyzes about the methods identified.

2.1 Quantitative Analysis

Regarding to the agile framework used in experiences reported in the papers, 4 of the 17 registered the use of XP, 3 registered Scrum, 2 used both and 4 did not mention which agile framework was used. The other 4 papers were not about a real experience but only a method proposal that did not mention any specific framework. This means that the most part of the analyzed methods that applies pre-development usability evaluations uses XP.

Regarding to the artifacts used in these evaluations, evolutive prototypes are the most used, in other words, prototypes that starts simple and evolves according to user tests results and requirements grooming. The second one is the low fidelity prototypes once the evaluations are being executed during a phase when nothing was developed yet. These results are presented in Fig. 1.

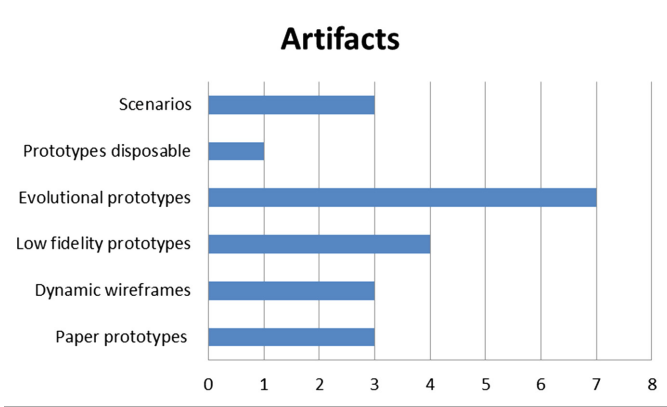


Fig. 1. Artifacts used per paper.

In relation to the distribution of papers based on the usability evaluation method applied, user tests are still the most used one, even before the development. The user tests are combined to the other methods (Figs. 2 and 3).

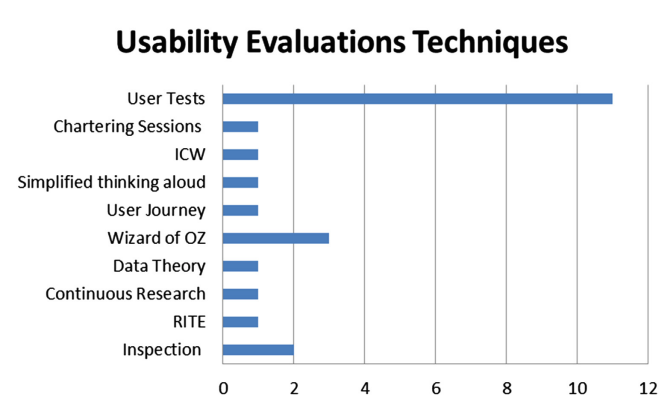


Fig. 2. Usability evaluation methods.

2.2 Qualitative Analysis

In this subsection we present the usability evaluation methods identified and the results presented by using each one of them.

User Tests. User Tests are mentioned as pre-development usability evaluation method in 11 – [1, 2, 4, 6–8, 10, 13, 14] – of the 17 papers.

The artifacts used are mainly different levels of prototypes used alone or combined with other artifacts such as scenarios and storyboards. About the prototype level, each study presents different advantages, for example, in low fidelity prototypes the users feel more comfortable to suggest changes, otherwise high fidelity prototypes allows a better feedback about what is working well or not.

The results presented by the papers that experienced user tests as main method, combined or not with other methods, were positive regarding to rework elimination which reflects directly in the reduce of project duration and cost [1, 2, 4, 6, 7, 14]. Besides that, this method not only showed results in UCD but also in requirements grooming, setting users expectations about the project and allowing the development team to focus on implement a more valuable software.

Chartering Sessions. This method was quoted only in one – [14] – paper and combined with user tests based on low fidelity prototypes. The Chartering Sessions are sessions conducted after user tests focused on the discussion of feedback from user tests and on the analysis if the system meets users real needs.

Decrease of rework hours and improvement on requirements prioritization are the two most valuable results reached by using this method.

ICW - Informal Cognitive WalkThrough. This method is mentioned only in one paper – [5] – and it is used alone and not combined to user tests. The ICW main objective is to identify and correct usability issues easily and quickly. It is an inspection that is based on potential usability issues and the discussions around its solutions. The main results registered by using this are the decrease of user participation on the process besides the decrease of rework hours once the usability issues are fixed before the implementation.

Simplified Thinking Aloud. In the same way as the last method, this one is mentioned only in one paper – [7] – that describes it as a method of interviews, on which the users execute usability tests out loud exposing their objectives, feelings and worries during all evaluation execution in order to allow the observer to capture pain points, improvements and issues.

Once it promotes an easy language of iterative feedback, this method can or not count with the participation of a UCD specialist, what is a great benefit compared to the others that requires this role.

User Journey. This method is mentioned in a superficial way in only one paper – [2] – and it is composed by scenarios and its execution by different personas in order to obtain what are the paths taken when executing the software main tasks and to identify the issues that may be faced by users. The paper does not present any information about the results obtained when using it.

Wizard of Oz. Mentioned in three papers, this is the second most used method, one of these papers talks about its use and the others propose tools to use it [9].

The users are divided to play the roles of key users, system and help system, the last one offering instructions about how the system should answer based on each action and helping with key users' doubts.

Besides the reduction of rework hours, there is the identification of important features that should be absorbed in project scope and the assurance of users acceptance in Users/Usability Acceptance Tests (UAT).

Data Theory. It is assumed that the iterative planning affects user interface, also, this interactions guide usability tests that impacts on development cycle [4]. Taking this in mind, in order to prevent possible changes, the method proposes that UCD specialists must design prototypes and conduct usability tests in order to refine requirements two cycles ahead the first development cycle. The user requirements were better defined and comprehended guiding the prioritization what brings as results the decrease of rework hours.

RITE - Rapid Iterative Testing and Evaluation. The main problem that this method aim to solve is the identification of usability issues before the software implementation giving to the team a timebox to solve and improve it [3]. It is a user story maturity model based on usability tests that are watched by the development team that capture easily users needs and problems faced by them.

Usability Inspection. The usability inspection is usually applied within user tests. These inspections are not executed by users but by UCD specialists. The UCD specialists navigates in the prototypes searching by possibles usability problems based on this methodologies principles [6]. The main advantage of this model is the reduction of user tests cycle time once the specialists capture and solve usability problems during user tests.

3 Experience Report

This section presents the pilot study developed based on the application of pre-development evaluation methods identified during the first phase – systematic review.

3.1 The Research Site

The company where the study was carried out is a north American pharmaceutical multinational. More specifically in the sub-area responsible for web implementation that was transitioning from waterfall to Agile methods using Scrum as the main framework.

It is easy to understand that the team was not mature yet concerning agile implementation and they were institutionalizing a more people-centered mindset by taking the Agile Manifesto as basis.

About the User-Centered Design, the Latin America department does not have any focus on this neither specialists with background that covers this topic.

3.2 The Project

The software deliver area aimed to improve data collection to generate software engineering metrics. This collection has already been generated manually. A global team is in charge of this collection through excel files filled by all software development areas of the company.

The process of collecting these metrics presented a lot of problems to generate metrics in addition to the overtaxed process to consolidate all the gathered information, keeping the consistency and data standardization.

In order to optimize this process, the global team responsible by collect and generate these metrics requested an internal project to develop a web portal that allows project managers to input project information used to generate engineering metrics.

Besides that, the project scope was also composed by additional features as reporting calendar configuration, roles delegation for vacation or other out of office activities and alerts setting. The web portal should be accessed by people from around the world with different profiles and this implies in the importance and relevance of a UCD approach.

3.3 The People

The company implemented a new software development program composed by technicians hired through a process based on software development marathons. That way, the team was composed by young people with a low level of experience lead by a senior analyst.

The development team was composed by 5 people from the above cited program. On one hand, the team did not have maturity on software development, and some technical problems were faced. On the other hand, this lack of knowledge and experience provided a mindset change to this team and this fact facilitated the utilization of agile methods and UCD methods.

Regarding to the Scrum Master, he did not played this role before and did not know about UCD despite some trainings related to that. Also, the development team and Scrum Masters were allocated in the same room during the entire project.

The Product Owner also did not have any knowledge related to agile or UCD. He was a member of the global team accountable by software engineering metrics definition and consolidation. Also, this person was not even allocated in the same country as the rest of the team and participated remote in all meetings.

3.4 Methods Definition

During the Sprint 0, user stories and high fidelity prototypes were created besides the identification of the key users. Based on these artifacts and information available in this preparation phase, that occurs before the beginning of development cycle, the following methods identified during systematic review were chosen to be applied: user tests with high fidelity prototypes and usability inspections. Besides that, guidelines based on Nielsen's Heuristics were created to drive team's mindset implementation and UCD application.

About the other methods identified during the systematic review:

- Chartering Sessions could not be applied because the team did not count with any UCD specialist;
- ICW has as main motivator for its applicability decrease the hours of user participation, but it was not a problem in this context, contrariwise, the user engagement was really good once it was an internal project.
- The Wizard of Oz requires users together in some sessions, and it was discarded because once the users works in different projects, they had some difficult to participate in the same sessions.
- Once the project already had a set of high fidelity prototypes and user stories defined, Data Theory was also discarded.
- RITE was not used once it takes a lot of time from development team in usability evaluations and this fact goes against the conditions presented by leadership team – not use development teams time to usability methods in order to not impact project's implementation.

3.5 Methods Application

When presenting the proposal of applicability of UCD evaluation methods pre-development for the agile project stated here, the company allows the pilot execution using the two methods mentioned before, if the following conditions were respected:

- Development Team should not be allocated for too much time to work in the execution of UCD methods.
- Not all usability issues identified during the UCD evaluations would be corrected but only the prioritized ones.
- The usability evaluations should not generate rework to development team.

In order to start the application of evaluation methods, the first step to be performed was the definition of personas based on the stakeholders already

identified by the product owner. Once the personas were defined, at least one person representing each persona was called to execute the user tests based on the high fidelity prototypes.

After the execution of user tests, the identified defects were organized in an artefact to be presented to the Product Owner in order to be prioritized. The artefact used was organized by dividing the defects by the total number of people who reported it, starting from those ones reported by the most part of the personas until those reported by only one.

Once the defects were prioritized, they were fixed according to the Product Owner definition. After the defects fixing, the prototypes were updated in order to reflect the changes applied and the first sprint development was based on them.

Once UCD was a method never used before by the development team, the absorption of its application importance took more time than expected. To aware the team and help them to apply this methods, a workshop was presented explaining UCD methodology, focusing mainly, in the Nielsen's Heuristics and how to use it in usability inspections. Examples using the interface of this project itself were used in the workshop as a way to facilitate team's engagement and learnability.

After the workshop, usability inspections were performed in the updated high fidelity prototypes and other versions were created to fix the issues identified. Also, as an alternative to help the team to develop the expected mindset, a guideline was developed contemplating Nielsen's heuristics and usability best practices (e.g. error messages standardization, buttons design, etc.).

After the guidelines implementation, every sprint from the second onwards followed the same rules. The team kept developing user stories until the product backlog be completed and the deploy was executed. During all the development cycle the team worked on the continuous improvement of the guidelines in order to cover all the rules that could help them to increase the use of usability methods and provide a portal that allows a better and easier experience to the user.

In order to gather conclusive information, after the deployment, the user tests were executed again and the results generated were analyzed and compared to the results obtained in the user tests executed in the prototypes in the preparation phase (Sprint 0).

4 Findings and Discussion

We observed great results with regards to the main goal of this experiment that is rework decreasing. Also, the integration of User-Centered Design and Agile Methodology worked very well even with this change in the way usability evaluations used to be applied.

Besides the expected results already mentioned, it was verified mindset changes from the stakeholders, Product Owner and the development teams regarding to the importance of usability matter.

Those results will be detailed in the next sections.

Rework Decrease. The results obtained regarding to the decrease of rework hours and the usability issues identified on each sprint were recorded and compared.

In the first sprint, one user test was already executed in the preparation phase but the workshop was not been executed yet. Therefore the team was still immature and the usability issues were bigger than in other phases. From the moment that the workshop ran and the team was presented to UCD concepts and methods, the usability issues number were decreasing, sprint by sprint, according to the team experience and the guidelines were being improved.

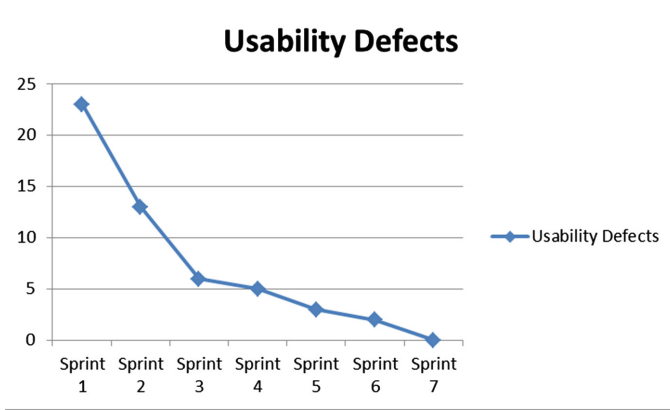


Fig. 3. Usability Defects per sprint

As expected, the usability issues were anticipated once the evaluations were been executed pre-development cycles and, this way, the most part of the user stories were fixed even before its implementation.

Usability Improvement. The results generated in the first user test – that occurred before the usability methods – were compared to results of the user tests that occurred after the pilot implementation and evaluation methods application.

It is important to mention the measure used to compare those two user tests. During the user tests, the key users representing personas executed 9 tasks. For each task it was registered if the key user had success on completing this or not. Also, for each task executed it was count the quantity of steps needed to finish the task. Comparing the results from both user tests, in the second round of tests, a hundred percent of the key users executed the tasks with less steps than in the first round.

Regarding the results, it can be observed in the Figs. 4 and 5 that the quantity of issues decreased substantially once the usability methods were applied and the improvements were made based on it.

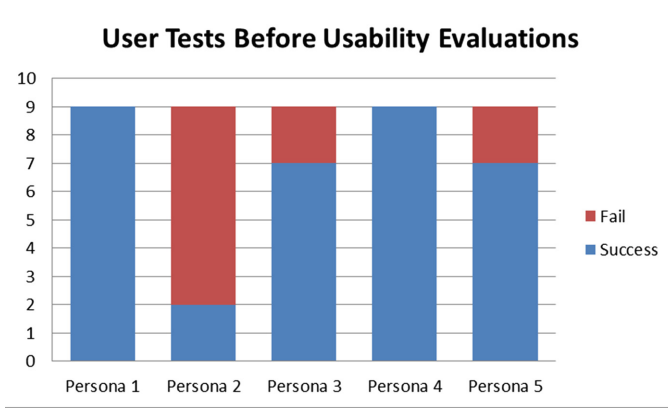


Fig. 4. User Tests before evaluations.

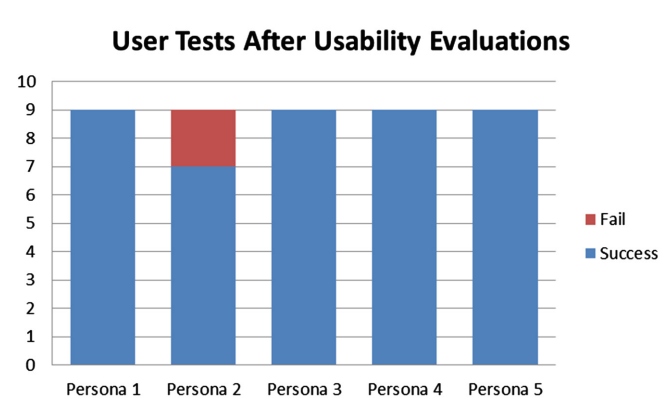


Fig. 5. User tests after evaluations.

Besides the counting of steps mentioned before, another used measure was to analyze if the user experience was improved or not. After the user tests, each key user provided feedback about the system usability. In the first user test, the team received a bad feedback from key users once they reported a lot of difficulties to complete the tasks and understand the system. In the second user tests, the feedback received were really positive and some of them mentioned how great was execute those test before development cycles.

5 Conclusion and Lessons Learned

We faced a lot of barriers to implement this method. Aware the team about the importance of usability quality and avoid rework by doing usability evaluations

before the development begins was not a problem once the team was young and had already the mindset of continuous improvement pretty ingrained.

Still, one of the challenges faced was the defects report to development team. In the beginning there was a little discomfort with the quantity of usability defects reported. However, after the workshop presentation, the team learnt about usability evaluations, defects, and the importance of a good system usability. This was essential to change their behaviour with regards to the defects reported. After this workshop they created a critical opinion about system usability quality and the team itself started to report usability defects and fix them.

The main challenge faced to implement this model was the part of convincing the leadership board that it was an important matter and it has value including to spend time from development team. In order to go through this and receive the free pass to implement the proposed model, it was presented to them the results from user tests A, the comments added by key users in feedback sessions and mainly, all of this organized in the way to exposure how many defects were identified and tasks that were not completed.

Another interesting issue that should be mentioned is regarding to the project team profile: the team were not composed by any UX specialist and even this way, the team could execute usability evaluations and identify usability defects and, also, correct them before development what avoid rework, as mentioned before.

Another important issue is that, besides the team mindset change, the usability evaluations started to be appreciated also by the PO team and indirect stakeholders. We shared with them all work executed based on Usability heuristics and user tests and how the system was improved by that. This impacted the way they evaluate a system quality and even in UAT tests, they requested some changes regarding to the usability. In other words, their mindset changed and they also recognized the importance of this matter.

About the rework reduction, it is clear, based on the discussed results, that it was successful. Also, the goal to integrate Agile methods and UCD worked well even executing the usability evaluations before the development begins and the results exceeded the expectations affecting also the sphere of team and stakeholders mindset.

Acknowledgments. This research was supported by FAPESP, Brazil, proc. 2014/25779-3.

References

1. Beyer, H., Holtzblatt, K., Baker, L.: An agile customer-centered method: rapid contextual design. In: Zannier, C., Erdogmus, H., Lindstrom, L. (eds.) *XP/Agile Universe 2004*. LNCS, vol. 3134, pp. 50–59. Springer, Heidelberg (2004)
2. Chamberlain, S., Sharp, H., Maiden, N.A.M.D.: Towards a framework for integrating agile development and user-centred design. In: Abrahamsson, P., Marchesi, M., Succi, G. (eds.) *XP 2006*. LNCS, vol. 4044, pp. 143–153. Springer, Heidelberg (2006)

3. Fisher, K.G., Bankston, A.: From cradle to sprint: creating a full-lifecycle request pipeline at nationwide insurance. In: Proceedings of the 2009 Agile Conference, AGILE 2009, pp. 223–228. IEEE Computer Society, Washington, DC (2009)
4. Fox, D., Sillito, J., Maurer, F.: Agile methods and user-centered design: how these two methodologies are being successfully integrated in industry. In: Proceedings of the AGILE 2008, pp. 63–72. IEEE Computer Society, Washington, DC (2008). <http://dx.doi.org/10.1109/Agile.2008.78>
5. Grigoreanu, V., Mohanna, M.: Informal cognitive walkthroughs (icw): paring down and pairing up for an agile world. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2013, pp. 3093–3096. ACM, New York (2013). <http://dx.doi.acm.org/10.1145/2470654.2466421>
6. Hussain, Z., Milchrahm, H., Shahzad, S., Slany, W., Tscheligi, M., Wolkerstorfer, P.: Integration of extreme programming and user-centered design: lessons learned. In: Abrahamsson, P., Marchesi, M., Maurer, F. (eds.) Agile Processes in Software Engineering and Extreme Programming. LNBP, vol. 31, pp. 174–179. Springer, Heidelberg (2009)
7. Kane, D.: Finding a place for discount usability engineering in agile development: throwing down the gauntlet. In: Proceedings of the Agile Development Conference, ADC 2003, pp. 40–46, June 2003
8. Kollmann, J., Sharp, H., Blandford, A.: The importance of identity and vision to user experience designers on agile projects. In: Agile Conference, AGILE 2009, pp. 11–18, August 2009
9. Meszaros, G., Aston, J.: Adding usability testing to an agile project. In: Proceedings of the Conference on AGILE 2006, pp. 289–294. IEEE Computer Society, Washington, DC (2006). <http://dx.doi.org/10.1109/AGILE.2006.5>
10. Najafi, M., Toyoshiba, L.: Two case studies of user experience design and agile development. In: Proceedings of the AGILE 2008, pp. 531–536. IEEE Computer Society, Washington, DC (2008). <http://dx.doi.org/10.1109/Agile.2008.67>
11. Silva, T., Martin, A., Maurer, F., Silveira, M.: User-centered design and agile methods: a systematic review. In: Agile Conference, AGILE 2011, pp. 77–86 (2011)
12. Silva da Silva, T., Silveira, F.F., Silveira, M.S., Hellmann, T., Maurer, F.: A systematic mapping on agile UCD across the major agile and HCI conferences. In: Gervasi, O., Murgante, B., Misra, S., Gavrilova, M.L., Rocha, A.M.A.C., Torre, C., Taniar, D., Apduhan, B.O. (eds.) ICCSA 2015. LNCS, vol. 9159, pp. 86–100. Springer, Heidelberg (2015)
13. Ungar, J.: The design studio: interface design for agile teams. In: Proceedings of the AGILE 2008, pp. 519–524. IEEE Computer Society, Washington, DC (2008). <http://dx.doi.org/10.1109/Agile.2008.51>
14. Williams, H., Ferguson, A.: The ucd perspective: before and after agile. In: Proceedings of the AGILE 2007, pp. 285–290. IEEE Computer Society, Washington, DC (2007). <http://dx.doi.org/10.1109/AGILE.2007.61>