

Coordination Support for Integrating User Centered Design in Distributed Agile Projects

Ohoud Almugham, Sultan Alyahya

Information Systems Department
College of Computer & Information Sciences
King Saud University, Riyadh, Saudi Arabia
ohoudhasan08@gmail.com, sualyahya@ksu.edu.sa

Abstract— Agile process and User Centered Design (UCD) approach are two fields with high beneficial opportunities when integrated together. Both are based on minimal design up front and frequent feedback from users, and both are human-centered. However, one of the main concerns in integrating them is how to coordinate the work properly specially in distributed projects. The traditional methods used (e.g. video conferencing) do not provide full support to involve the UX-designers during the daily project activities. UX-designers may struggle to know if their design is technically applicable which makes it necessary for developers to review the designs from their perspective. Similarly, developers need instant answers from UX-designers during and after developing the work items. Furthermore, the review of thirty agile management tools that support the integration of UCD into agile projects, reveals a lack of computer-based support for the development teams and UX-design teams. This paper aims to identify the situations in which UX-designers and developers require proper coordination mechanisms. This research identified 14 UCD-related activities that require further coordination support. A questionnaire was conducted for the evaluation. The results show that the integration of UCD activities in distributed agile environments still needs coordination especially between the UX-design team and the development team. The result is important because it can help in overcoming the limitations of the informal and formal approaches, support managing the UCD activities in distributed agile environments and facilitating the coordination of the work between developers and UX-designers.

Keywords— User Centered Design; Geographically Distributed Agile; User eXperience, process, coordination.

I. INTRODUCTION

In the last few years, agile methods have played a significant role in the area of software development. Agile methods aim to produce high-quality software through the use of frequent communication among team members, continuous delivery of working software to the customer, and fast response to the rapid changes requested by the customer [5].

Although the agile approach is iterative and focuses on real users during testing, in practice, it does not focus on creating an experience for the end users. Software practitioners have recognized the lack of consideration for user age, geographical location, ethnicity, gender, and education in this approach and other software development approaches. Different disciplines have arisen to fill this gap, including User-Centered Design (UCD) [2][20]. UCD's main objective is to create an optimal product based on the user's needs rather than forcing users to engage with the new features of the product [1].

Many researchers (e.g. [2],[8],[9], and [10]) have discussed the importance of integrating UCD activities and the agile approach to increase the usability of a product. They have addressed the similarities between the two processes, which encourage integration. Both processes are based on minimal design up front and frequent feedback from users, and both are human-centered.

Nowadays, it is common that developers, designers, and users are spread all over the world during developing software projects. This has encouraged software organizations to adapt the agile approach with UCD to gain the benefits from both sides: while the agile approach is driven by developers with their technical perspective, the UCD approach is driven by the UX-designers with their usability perspective [18]. In addition, the integration of UCD activities into the agile approach enables developers and users to be both involved during the early stages of design, which will save time and costs [8].

Despite the fact that there are many ways of describing how the merging of UCD activities into agile development is to be carried out (e.g., just in time, parallel design...etc), it has been widely recognized that coordinating design activities with development activities is one of the most common problems of integrating UCD activities into agile development projects [19]. This problem can be more difficult if an agile software project is distributed over multiple sites, as has been described in a tremendous number of case studies (e.g.[2], [5] and [45]).

Currently, distributed agile software organizations use two main approaches to help coordinate UCD activities with agile development activities: informal and formal methods. The informal methods can be characterized into synchronous (e.g., video conference, telephone conference, or IM) and asynchronous (e.g., emails and forums) communication methods [45]. In addition to the temporal, geographical, and socio-cultural barriers associated with the informal method, a main limitation is the possibility of a lack of awareness for how the technical activities and the UCD activities affect each other. Even if they are aware, team members may forget to inform the affected team members or there may be a long time between the need to coordinate a change and the next possible meeting with the corresponding team members [2].

The second approach is the formal methods applied through using agile project management tools that are integrated with pure design tools. Jira [11], Rally [12], and many other agile management tools provide such integration, which increases the involvement of the UX-designers, but which also still need more coordination and transparency with the development teams. The development teams who use such tools might find it difficult to have instant guidance from the UX-designers about the design deliverables. These tools only link the development activities and the design deliverables but without providing the necessary coordination that helps team members understand how the design deliverables affect their daily work.

In order to overcome the limitations of the informal and formal methods, this research aims to support the effective management of integrating the UCD activities into distributed agile projects. A computer-based approach will be used to allow synchronization of the activities between UX-design teams and development teams. It is expected that it will provide more transparency between both and raise awareness on how design changes affect the other development activities and vice versa.

The remainder of the paper is organized as follows: Section II presents background information. Section III reviews the current coordination support for the UCD activities in distributed agile projects. Section IV describes the coordination needed to integrate UCD activities in distributed agile projects, section V presents the evaluation results of those mechanisms. Finally, section VII is the conclusion of the paper.

II. BACKGROUND

A. User-Centered Design (UCD)

The term User-Centered Design originated in Donald Norman's research laboratory at the University of California San Diego (UCSD) in the 1980s [22]. UCD aims to include the users from the early stages of design. It explores who will use the system and design based on the user's age, culture, and geographical location. During design activities, the users will be included in User Experience (UX) activities to experiment how they deal with the user interfaces (UIs).

In order to apply UCD activities, the following three steps are included [21]:

1- Research: this step consists of activities such as interviews, surveys, focus groups, workshops, brainstorming, task analysis, content analysis, mental modeling, user profiles/personas, user scenarios, cataloging of existing branding, and competitor evaluation.

2-Concept: creating initial high-level architectures, storyboards, sketching, card sorting, wireframes, page layout, navigation modeling, metadata design, or taxonomy design.

3- Design: this step includes developing graphic design, logo design, branding, icon design, screen mock-ups, low/high fidelity prototyping, usability evaluations and testing, mobile testing, lab testing, web analytics, and performance testing.

B. UCD Integration into Agile approach

Agile methods have principles and practices that appear inconsistent with UCD methodology. However, many studies encourage to improve agile and UCD by integrating them [2] [3] [4] and [6]. According to Barksdale et al [15], that using agile helps to maximize the usability and UX-designers prefer to work in agile projects rather than traditional projects.

Agile methods and UCD process are both user and customer focused which encourage the integration between them. The difference is in the approaches of gathering the requirements. Agile do not support big design up front (BDUF) while the acceptance testing conducted after developing the features at the end of the iteration (release cycle). Whereas, UCD gathers all the requirements at earliest phases including the usability test before the feature is being developed. To gain benefits from both methodologies, Silva et al [43] and Beyer et al [44] suggest to focus the requirement gathered initially on understanding the fundamentals then use the UCD activities for the features being developed for each iteration which have implication to guide for rapid iterations.

The following are the current techniques used to integrate UCD into the development process of agile projects:

1) *Little Design Up Front LDUF (or Sprint 0)*: This technique is used to create the basic structure for a project so that future sprints can be truly incremented. It includes building a flexible enough framework that facilitates the refactoring process. The UX-designers usually design two sprints ahead to be ready for development team in the right sprint [23] [46].

Jennifer et al [24] suggest to use a task level for the design requirements in case of completing the UI design before the development began.

2) *Design in Parallel*: The UX specialist Miller [25] explains that, in her experience, the UX activities need to be a part of the whole project, from idea to testing, and the UX designers need to work in parallel with the developers. The authors of [28] suggest using two parallel tracks: iterating the design and implementation separately, but simultaneously. They state that, "*A key principle of the User Experience Team's UCD process is design iteration; we need to be able to catch design failures early, change designs as many times as needed, and then incorporate the design fixes.*" Design in parallel can be implemented through:

a) *UX-friendly requirements artifacts*: UX artifacts (e.g. wireframes, mockups, prototypes...etc) will be included as a user story in the backlog. Alan Cline in [26] describes UX

requirement as the detailed user story or use case which is associated with a screen mockup, picture, navigational map, or any other document describing the user interactivity. This artifact is part of a functional requirement used as a bridge between the design and the requirements.

b) *Just In Time (JIT)*: If the goal of design is limited to a single feature or a small set of features, it is effective to do UI development on a JIT basis and perform the UX activities as part of the development during the iteration [27][29]. With this approach, the team identifies an issue that needs to be resolved, some members will be responsible to explore the issue, and then everyone will continue on as before. EXtreme Programmers (XPers) call it a stand-up design session or a customer Q&A session, and it is clearly common on traditional projects as well [13].

c) *Design Spikes*: Development teams sometimes use the spikes as a solution to work on some obstacles. Spikes are the time-boxed periods that are used for exploration, research, and sometimes prototyping. The purpose of a spike is to explore the information necessary to understand a development approach or a requirement or to make a story estimate more reliable [14][7]. With a few modifications, the spike can be adjusted to serve the needs of design teams while maintaining the overall structure of scrum. It is called a design spike, which is a method used by the design teams to address complex UX issues within scrum methodology.

III. Current Coordination Support for the UCD Activities in Distributed Agile Projects

In distributed agile environments, two approaches are used to help coordinate UCD activities with agile development activities: informal and formal methods [16]. These are discussed below.

1) Informal Methods

Agile teams can coordinate their design and development tasks through the use of informal communication, which can be either synchronous or asynchronous. The synchronous methods allow teams to communicate via ways such as:

- Videoconferencing (VC): a means of communication that allows teams in separate locations to meet using live video stream.
- Teleconference call: a telephone call in which distributed teams can talk and listen to each other.
- Instant Messaging (IM): a real-time chat service that allows exchanging the information in text messages.

Asynchronous communications can be via:

- Email: The teams can communicate using emails to share the information and attach the design deliverables.
- Forums or discussion boards: This method allows the different team members to discuss a certain issue, design artifact, or a new idea.

Different studies [17][42] and [41] show that the informal communications methods are not beneficial enough because of the difficulties that appear when depending on such methods, for instance:

- Different time zones make it difficult for some members to attend meetings.

- Diversity in languages and culture may lead to misunderstanding and lack of discussion if some members keep silent.
- Difficulties in scheduling the meetings especially when meetings are conducted regularly.
- Technical problems that can interrupt meetings.

In addition to these problems, there is a main limitation that almost all distributed agile teams face when using informal methods to support the coordination of UCD activities; they may not be aware of how their technical work affects the UCD activities and how the UCD activities affect the technical work. Even if they know, they may forget to inform the affected team members or there may be a long time between the need to coordinate a change and the next possible meeting with the corresponding team members.

2) Formal Methods

Table 1 shows all the current UCD features supported by thirty agile project management tools. The review of the selected tools include both commercial and open source tools and covers the most popular agile tools according to the survey in [77]. The bold circle indicates that the tool fully supports the feature. The empty circle indicates that the tool partially supports the feature. Here, these features are described below:

a) Central Design Deliverables Repository

Once the design deliverables (e.g. mockups, prototypes...etc) are ready to be developed, they will be shared with the development teams. The deliverables will be available in any of the following repositories:

- *On the Cloud*: Tools such as Easy Redmine, Blossom, GitHub, TargetProcess, and PivotalTracker use Cloud services such as OneDrive, GoogleDrive, Dropbox, or Box to share the design deliverables as PNG, JPG, PDF or even video. Jira has its own Cloud service called JiraCloud that is used for the same purpose.
- *On dedicated servers*: All the tools are associated with servers for more deployment flexibility. The servers can be used by UX-designers to share the designs, e.g. Confluence Server of Jira can be used by UX-designers to share work in progress and gather feedback from the stakeholders. A typical page will include some context about the problem being solved associated with photos of sketches, high fidelity mockups, or links to the prototypes. People will add comments and questions, and the UX- designer will post updated designs as the project progresses.
- *As an attachment*: Tools like easyBacklog and AgileWrap attach the design deliverables with the related user story. The AgileAgenda tool uses emails attachment for submitting those deliverables.

b) Integration with User-Centered Design (UCD) Tools

Some tracking tools realize the importance of bringing UCD design during the agile development process by integrating with UX/UI designing tools.

Rally and VersionOne integrate with the customer and user experience tool UX360[31]. Rally has a feature not existing in VerionOne, which is linking each user story with its design to synchronize the changes between Rally and UX360.

MagicDraw provides a service of UI prototyping with a Cameo simulation toolkit to create a working UI prototype directly from the model, also it is possible to generate mockups from the available UI.

Jira supports UCD by integrating with many UX design tools such as CUTE for JIRA [32] and RefinedWiki UI Toolkit [33] to create and manage UIs, Balsamiq Mockups [34] and TrackDuck [35] to create UX deliverables and share them among other teams, Comala Canvas boards [36] and PSD Preview for JIRA [37] to review the designs and share the ideas, and Draw.io Diagrams for Confluence [38] and Gliffy [39] to create basic diagrams such as flowcharts and UML.

Trello and PivotalTracker integrate with InVision [30] which is a collaborative design tool, where the designs are submitted to the related user story using a drag and drop approach to the tracking tool.

c) *Separate Design Task Board*

EasyRedmine, Blossom, Trello, and Jira provide a way to track the tasks of designers and developers from the same tool. For example, in Trello, the designer’s taskboard has four columns: ‘In progress’, ‘Needs review’, ‘Approved’, and ‘Push to developers’.

TABLE 1. A REVIEW OF THE CURRENT SUPPORT OF UCD ACTIVITIES IN AGILE PROJECT MANAGEMENT TOOLS.

Agile Tool	Central Design Deliverables Repository	Integration with UCD Tool	Separate Designer Task Board	Separate Design Backlog	Design User Story (or Task) Label
1- EasyRedmine [47]	●		●		
2- Blossom [48]	●		●		●
3- Trello [49]	●	●	●		○
4- Jira [51]	●	●	●	○	●
5- GitHub [52]	●	●		○	
6- Rally [53]		●			
7- Mingle [56]	●			●	●
8- VersionOne [55]	●	●			○
9- TargetProcess [57]	●	○	○	●	●
10- PivotalTracker[58]	●	●			
11- BaseCamp [59]	●	●		●	
12- ScrumPad [60]	●	●		●	
13- Asana [61]	●	●		○	○
14- Aha! [62]	●				○
15- ScrumDesk [63]	●				○
16- Sprint.ly [64]	●	○			
17- easyBacklog [65]	●				
18- Xplanner plus [66]	●				
19- ScrumWork [67]	●			○	○
20- Scrum VSTS [68]	●				
21- Agilefant [69]	●				
22- IceScrum [70]	●				
23- Taiga.io [71]	●	○			
24- Agile Forest [72]	●			○	○
25- AgileAgenda [50]	●			●	
26- Feature Map [73]	●	●	●		
27- Agilo for scrum[74]	●				
28- Wrike [75]	●				
29- AgileWrap [76]	●			●	
30- ScrumDo [54]	●	●			

d) *Separate Design Backlog*

Some tools have a separate design backlog because it is essential in cases like when a product is developed on multiple platforms (e.g., iOS and Android) with design concerns that are independent of the platform, or to estimate the design work without affecting development velocity [40]. Trello, VersionOne, PivotalTracker, ScrumPad, and Asana support this mechanism.

e) *Design User Story (or Design Task) Labeling*

Some tools allow for labeling the design user stories (or tasks). This labeling makes it easier to distinguish the design work items among other tasks, and makes it easier to group the design work items in one theme if needed. Blossom labels them as ‘Blocked by Design’ so the user story cannot be developed until the design is finished, whereas Trello and GitHub use the coloring mechanism to classify the user stories according to the type as either design, bug, test, or defect.

The review of the agile project management tools revealed that there is a lack of support for UCD activities with agile tools. The backlog is not managed properly to reflect the changes of the UX designers to the developers. The development teams who use such tools might find difficulty in having instant guidance and updates from the UX-designers about the design deliverables. These tools provide a linkage between the development activities and the design deliverables without providing the necessary coordination that helps team members understand how the design deliverables affect their daily work.

IV. *Coordination needed to integrate UCD activities in distributed agile projects*

This research proposes a set of coordination mechanisms that can help to overcome the limitations of the informal and formal approaches mentioned in the previous section. The proposed mechanisms will support managing the UCD activities in distributed agile development environments and will facilitate the coordination of the work of developers and UX-designers.

The review of the current tools, reading the forums of UX-designers working in agile environments and the literature reviews [78] [79] [80] help to identify a set of UCD activities need to be managed and coordinated to effectively integrate UCD in distributed agile development.

Table 2 presents the activities with the proposed support to properly coordinate UCD activities in distributed agile environments and facilitate communication between the UX design teams and the development teams.

The proposed mechanisms of this research can be used for any of the UCD integration approaches described in Section II-B. The mechanisms will target the coordination of UCD activities between the UX design team and the development team regardless of how they do schedule their work (i.e. either working in parallel or preparing the design prior of the development cycles).

TABLE 2 THE ACTIVITIES WITH THE PROPOSED SUPPORT.

UCD Activity	Current Tool Support	Proposed Support
1- Adding/Modifying business logic code	The current tools provide the basic activities of adding/modifying or deleting the codes	These activities can be supported by checking for dependent UI artifact. In case of dependency exists and the developer chooses to proceed, the affected developers/UX-designers will be notified.
2- Deleting business logic code		
3- Adding/Modifying event handler		
4- Deleting event handler		
5- Adding/Modifying UI artifact		
6- Deleting UI artifact		
7- Write/modify UX-task	The current tools provide the basic activities of adding/modifying or deleting UX-task without reflecting the change to design backlog	It can be supported by automatically adding the UX-task from the product backlog to the design backlog with higher priority for the UX-task of the earliest next two sprints. In case of adding/modifying UX-task in product backlog, affected UX-designer must be notified.
8- Delete UX-task		
9- Prioritize/reprioritize design backlog	Prioritize/reprioritize only in product backlog	It can be supported by reflecting any prioritization/reprioritization for UX-task by the product owner to the design backlog. Reprioritization which affects the UX-task of the next two sprints in product backlog can be managed by notifying the affected UX-designer.
10- Start UX-task designing	Not Supported	It can be supported by automatically block the user story and the tasks related to the UX-task.
11- Complete design	Not supported	It can be supported by notifying the product owner and developer to check the design.
12- Accept/Reject UX-design	Only one tool (Rally) allows the developers to check the design and either accept or reject it.	It can be supported by providing necessary coordination. if the UX-design is accepted: - Automatically change the status of the UX-task into “design done”. - Automatically unblock the user story/task. If the UX-design is rejected: - Notifying the owner of the UX-task with the corrections/improvements needed.
13- Add a discussion among Developer/UX-designer	Formal and informal approaches	Almost all the tools provide the basic process. It can be supported by associating the UX-designer’s name with each UX-task in the specifications along with an instant way of communication (e.g. sharing the desktop through Skype).
14- Review UX-artifact	Not supported	It can be supported by: - Adding UX-design criteria to an acceptance test. - notifying the UX-designer of the user story/task to review the developed user story/task before transferring them into DONE column.

V. Evaluation

To evaluate the previous findings, a quantitative data collection method had been used to help understanding the real-world problems that may occur when integrating UCD activities in distributed agile environment and the challenges faced by the UX-designers and developers in such situation. Evaluation of the proposed support of the UCD activities was conducted based on questionnaire helped in collecting the data from different countries, sharing their concerns, and getting some suggestions to overcome certain issues.

The questionnaire was created to evaluate the proposed coordination mechanisms for the UCD activities. It was distributed through a link to domain experts and companies on Twitter, LinkedIn, Facebook as well as emails.

The questionnaire consists of two parts. The first part inquires about the background information including the job title, how much the experience, the agile method used, the artifact shared between UX-designer team and development team, the way of communication and the distribution of team members.

The second part provides the list of UCD activities for which the developed proposed support is suggested (identified in Table 2) and respondents was asked to rate the extent to which they agree that those are really need to be coordinated.

1. Respondents' Background Information

The participants reached 32 persons including executives, project managers, developers, UX/UI designers and scrum masters. Figure 1 shows that most of the participants were developers/software engineer/programmer with 37%.

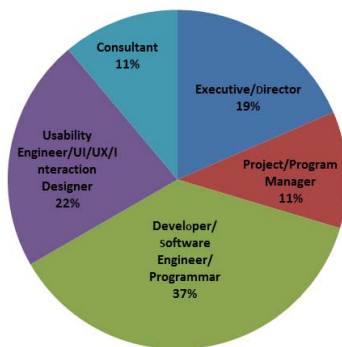


Figure 1. Respondents' Job Title

Their experience varies from 1 year to more than 10 years as shown in Figure 2.

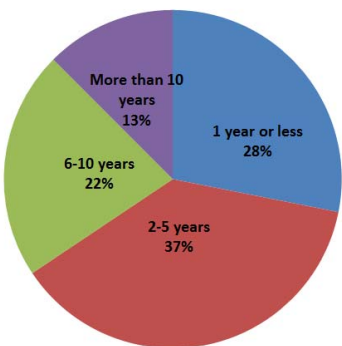


Figure 2. Respondents' Experience.

More than two third of the participants use scrum whereas 9% for lean development, 15% for hybrid scrum/XP, 6% for Agile Unified Process and other methodologies such as Kanban as shown in Figure 3.

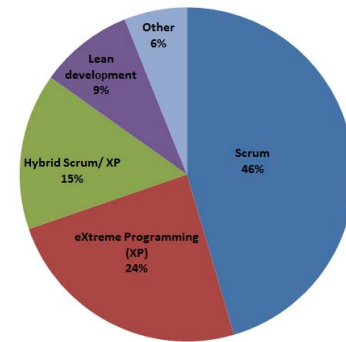


Figure 3. The Methodologies Used Among the Respondents.

The deliverables from UX-designer team to the development team was 21% high fidelity prototype, 17% low fidelity prototype, 24% wireframes, 14% mockup, 6% personas and 14% conceptual designs as shown in Figure 4.

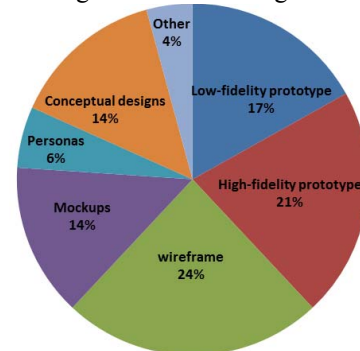


Figure 4. The Design Deliverables Used Among the Respondents.

The questionnaire shows that the team members including the UX-designers and developers usually communicate using more than one. Figure 5 shows that the formal methods were the most popular with 36%, followed by synchronize informal methods with 31%, followed by asynchronize informal methods with 26%.

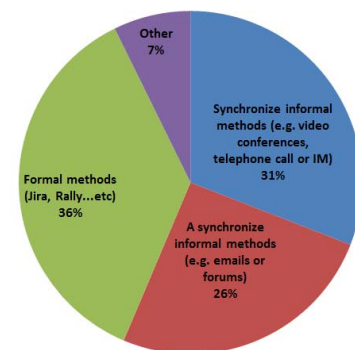


Figure 5. Ways of Communications Used Among the Respondents

Regarding the distribution of the team members. The analysis shows that most of the respondents (68%) were either: mostly distributed or fully distributed as shown in Figure 6.

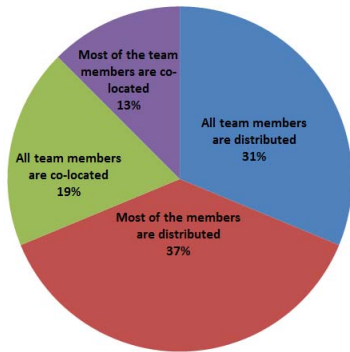


Figure 6. Team Distribution of the Respondents.

2. Evaluation of the Identified UCD Activities That Need to be Coordinated

Each UCD activity on Table 2 was tested against a defined question. The questions defined with a rating scale from 5 to 1 where 5 means very important and 1 means not important. The following describes the results in details:

Activity of managing changes made on business logic code: 44% of the respondents find that the changes made on the business logic code (addition, modification or deletion) by the developers; must be coordinated properly, whereas 28% of the respondents find it less important or not important to coordinate such an activity as shown in Figure 7. Therefore, according to the majority of the respondents, we can conclude that there is a need to provide support for this activity.

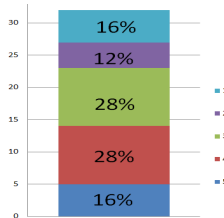


Figure 7. Changes made to the business logic code.

Activity of managing changes made on event handler code: 41% of the respondents find that the changes made on the event handler code (addition, modification or deletion) by any team member; must be coordinated properly, whereas 34% of the respondents believe it is not important to coordinate this activity as shown in Figure 8. Therefore, based on the opinions of the majority of respondents, we can conclude here also that there is a need to provide support for this activity.

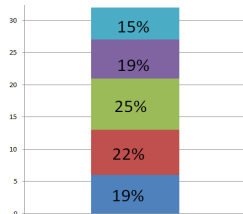


Figure 8. Changes made to the event handlers code.

Activity of managing changes made on UI artifact: Most of the respondents (53%) find that the changes made on the UI artifact (addition, modification or deletion) by any team member; must be coordinated properly. However, 22% find it less important or not important to coordinate such activity as shown in Figure 9. Therefore, based on the majority of the respondents, we can conclude that there is a clear need to provide support for this activity.

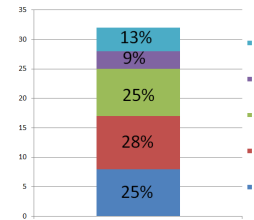


Figure 9. Changes made to the UI artifacts.

Activity of synchronizing the product backlog with the design backlog: 29% of the respondents find that the product backlog and the design backlog should be synchronized to reflect any changes made on the UX user story/task prioritizing or reprioritizing by any team member, whereas 19% of the respondents believe it is less important to coordinate such activity and 28% believe it is not important, because their team is using the same product backlog for UX-designers as shown in Figure 10. Therefore, for those who use two different backlogs for product and design backlog, they can benefit from a synchronization mechanisms between them.

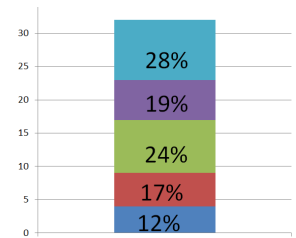


Figure 10. Synchronizing the product backlog and the design backlog.

Activity of managing user stories that contain UX-task being designed: Only 28% of the respondents find that the user story with UX task being designed must be managed. However, 19% of the respondents find it less important to coordinate this activity and 28% find it not important as shown in Figure 11.

As responses show, this activity is the least important activity among those we propose, depending on the way of managing the tasks of product backlogs among the scrum teams.

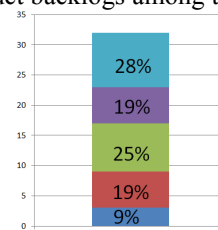


Figure 11. Managing the user stories that containing UX-task being designed.

Activity of accept/reject the design deliverable from the technical perspective: 43% of the respondents agree that the design deliverables should be reviewed from technical perspective by the development team. However, 16% of the respondents believe it is less important to coordinate this activity and the other 16% find it not important as shown in Figure 12. Based on the majority of the respondents, this activity needs to be properly coordinated.

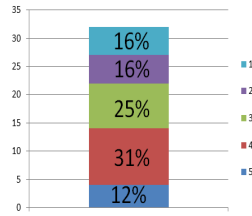


Figure12. Accept/Reject design deliverable from technical perspective

Activity of adding a discussion among developer/UX-designer: Most of the respondents (59%) find it either very important or important, that the communication between developers and UX-designers be enhanced. Only 13% of the respondents find it not important to coordinate this activity as shown in Figure 13. Therefore, based on the majority of the respondents, this activity is the most required one.

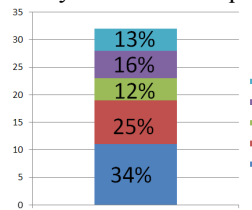


Figure 13. Discussion among developers/UX-designers.

Activity of checking the developed product from usability perspective: Most of the respondents (56%) find it either very important or important, that the developed product checked by UX-designers from usability perspective before the releasing. Only 16% of the respondents find it not important to coordinate such activity as shown in Figure 14. Therefore, it is clear that there is a need to provide support for this activity.

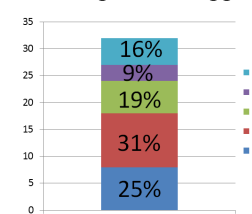


Figure14. Checking the developed product from usability perspective

Finally, statistical analysis applied on the whole responses of the second part of the questionnaire to know if each UCD activity actually needs to be coordinated. Figure 15 illustrates the summary for the UCD activities that the participants gave them a high rating considering them as very important to be coordinated.

Most of the respondents agreed on the need to coordinate the communication between the developers and the UX-designers with 23%, followed by 17% for managing the changes made

to UI artifacts, 16% for checking the developed product from UCD perspective, 12% for managing the changes made to the event handler code, 10% for managing the changes made to the business logic code, 8% for both synchronizing the product backlog with the design backlog and accept/reject the developed product from usability perspective and finally 6% for managing the user story that containing UX-task being designed.

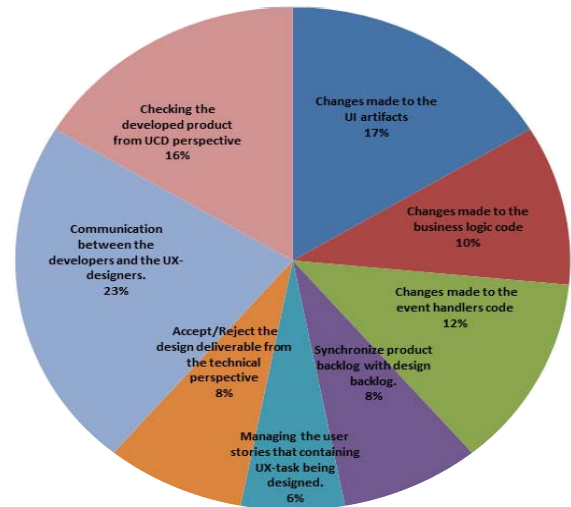


Figure 15. Statistical results of UCD

VI. CONCLUSIONS

This research is about improving the integration of UCD activities in distributed agile teams by coordinating the work between developers and UX-designers. The existing approaches for managing the process of the integration have been reviewed in this paper. The review revealed that there is a main limitation that almost all the distributed agile teams face when using informal methods to support the coordination of UCD activities; they may not be aware of how their technical work affects the UCD activities and how the UCD activities affect the technical work in the project. In addition, the review of the tools indicates that UCD integration with agile development is either entirely not supported or limited.

This paper has identified a set of UX-related processes that require further coordination support.. A questionnaire was used for the evaluation. The results show that the integration of UCD activities in distributed agile environments still needs further coordination especially between UX-design team and the development team. The result is important because it helps in overcoming the limitations of the informal and formal approaches, support managing the UCD activities in distributed agile environments and facilitating the coordination of the work between developers and UX-designers.

This research will be used for proposing a set of computer-based functionalities that can help integrate the UCD activities in distributed agile projects. The proposed functionalities will support the coordination between developers and UX-designers.

REFERENCES

- [1] Frank E., Gordon D. Baxter, and Elizabeth F. Churchill. "User-centered systems design: a brief history." Foundations for designing user-centered systems. Springer London, 2014. 33-54. Ritter.
- [2] Gabriela, Theodore D. Hellmann, and Frank Maurer. "Integrating agile and user-centered design: a systematic mapping and review of evaluation and validation studies of agile-UX." Jurca. Agile Conference (AGILE), 2014. IEEE, 2014.
- [3] Sohaib, Osama, and Khalid Khan. "Integrating usability engineering and agile software development: A literature review." Computer design and applications (ICCD), 2010 international conference on. Vol. 2. IEEE, 2010.
- [4] Stefan.e Blomkvist "Towards a model for bridging agile development and user-centered design." Human-centered software engineering—integrating usability in the software development lifecycle," , Springer Netherlands, 2005, pp. 219-244.
- [5] S. Alyahya, W. K. Ivins, and W. A. Gray, "A Holistic Approach to Developing a Progress Tracking System for Distributed Agile Teams," pp. 503-512, 2012.
- [6] Stephanie, Helen Sharp, and Neil Maiden. Chamberlain, "Towards a framework for integrating agile development and user-centred design," in International Conference on Extreme Programming and Agile Processes in Software Engineering, Springer Berlin Heidelberg.
- [7] Pieter. "Get Agile: Scrum for UX, design & development" Jongerius. BIS Publishers, 2014.
- [8] Schwartz, Lou. "Agile-user experience design: with or without a usability expert in the team?." ICSEA 2013, The Eighth International Conference on Software Engineering Advances. 2013.
- [9] Scott W. Ambler, "Tailoring usability into agile software development projects," in *Maturing Usability*, Springer London, 2008, pp. 75-95.
- [10] Sy, Desiree, and Lynn Miller. "Optimizing agile user-centred design." In CHI'08 extended abstracts on Human factors in computing systems, pp. 3897-3900. ACM, 2008.
- [11] atlassian.[Online].Available:https://www.atlassian.com/software/jira/.Accessed:March 2017.
- [12] Rally.[Online].Available:https://www.rallydev.com/. Accessed: March 2017.
- [13] M. A."A comparison between agile and traditional software development methodologies." Awad. (2005) University of Western Australia .
- [14] Lou. "Agile-user experience design: with or without a usability expert in the team?" Schwartz. (2013) ICSEA 2013, The Eighth International Conference on Software Engineering Advances.
- [15] Barksdale, Jeremy T., and D. Scott McCrickard. "Software product innovation in agile usability teams: an analytical framework of social capital, network governance, and usability knowledge management." International Journal of Agile and Extreme Software Development 1, no. 1 (2012): 52-77.
- [16] Alyahya, Sultan, Wendy K. Ivins, and W. A. Gray. "Raising the Awareness of Development Progress in Distributed Agile Projects." JSW 8, no. 12 (2013): 3066-3081.
- [17] J. D. Herbsleb and R. E. Grinter, "Splitting the organization and integrating the code: Conway's law revisited," in Proceedings of the 21st international conference on Software engineering, 1999, vol. Los Angeles, no. 5, pp. 85-95..
- [18] Danny Bluestone. (2015, Nov) [Online] .Available:http://www.uxbooth.com/ articles/overcoming-the-agile-ux-divide/. Accessed: March 2017.
- [19] Kati, Tommi Mikkonen. Santtu, Pakarinen."Agile user experience development in a large software organization: good expertise but limited impact" Kuusinen. Human-Centered Software Engineering. Springer Berlin Heidelberg, 2012. 94-111
- [20] P. McInerney and F. Maurer, "UCD in Agile Projects: DreamTeam or Odd Couple?" Interactions, pp. 19-23, 2005.
- [21] E., Lahiri, A."The Institutionalization of UX: A step-by-step guide to a user experience practice." Schaffer. (2014)
- [22] Frank E., Gordon D. Baxter, and Elizabeth F. Churchill."User-centered systems design: a brief history."Foundations for designing user-centered systems. Springer London, 2014. 33-54. Ritter.
- [23] Veli-Pekka, and Kai Koskimies."Aligning architecture knowledge management with Scrum."Eloranta. Proceedings of the WICSA/ECSA 2012 Companion Volume. ACM, 2012.
- [24] Jennifer, James Noble, and Robert Biddle. Ferreira,"Agile development iterations and UI design." in *Agile Conference (AGILE)*, 2007. IEEE, 2007.
- [25] L Miller, "Case Study of Customer Input For a Successful Product" in *Agile*, 2005, pp. 225-234.
- [26] Alan Cline, *Agile Development in the Real World*.: Apress, 2016.
- [27] Andrew, and D. W. He."Design for agile assembly: an operational perspective." Kusiak. International Journal of Production Research 35.1 (1997): 157-178.
- [28] Desirée. Sy."Adapting usability investigations for agile user-centered design." Studies 2.3. (2007)
- [29] Sisira, Craig Mcdonald, and John Campbell. Adikari. Human-computer interaction. New trends. Springer Berlin Heidelberg, 2009. 549-558.
- [30] InVision[Online]Available:https://www.invisionapp.com. Accessed: March 2017.
- [31] tandemseven.[Online].Available:http://www.tandemseven.com/ux360-customer-user-experience-software-tools. Accessed: March 2017.
- [32] atlassian.[Online].Available:https://marketplace.atlassian.com/plugins/de.communardo.atlassian.plugins.cute/server/overview. Accessed: March 2017.
- [33] atlassian.[Online].Available:https://marketplace.atlassian.com/plugins/com.refinedwiki.confluence.plugins.uitookit/server/overview. Accessed: March 2017.
- [34] Balsamiq. (2016, May)[Online] Available: https://balsamiq.com. . Accessed: March 2017.
- [35] atlassian.[Online].Available:https://marketplace.atlassian.com/plugins/com.rackduck.jira-plugin/cloud/overview. Accessed: March 2017.
- [36] atlassian.[Online].Available:https://marketplace.atlassian.com/plugins/com.comalatech.canvas/cloud/overview. Accessed: March 2017.
- [37] atlassian.[Online].Available:https://marketplace.atlassian.com/plugins/com.almworks.jira.psd-preview/server/overview.Accessed:March 2017.
- [38] atlassian.[Online].Available:https://marketplace.atlassian.com/plugins/com.mxgraph.confluence.plugins.diagramly/cloud/overview. Accessed: March 2017.
- [39] atlassian.[Online]Available:https://marketplace.atlassian.com/plugins/com.glify.integration.jira/cloud/overview. Accessed: March 2017.
- [40] Jonathan Berger. (2014, April) PivotalTracke. [Online] .Available: https://blog.pivotal.io/labs/labs/separate-design-backlog. Accessed: March 2017.
- [41] E. Carmel "Global software teams: collaborating across borders and time zones." vol. ISBN:0-13-. Prentice Hall PTR, 1999, p. 269.
- [42] J. D. Herbsleb and D. Moitra, "Global software development," IEEE Software, vol. 18, no. 2, pp. 16-20, 2001.
- [43] Da Silva, Tiago Silva, Milene Selbach Silveira, Frank Maurer, and Theodore Hellmann. "User experience design and agile development: From theory to practice." Journal of Software Engineering and Applications 5, no. 10 (2012): 743.
- [44] Bayer Hugh, Karen Holtzblatt, and Lisa Baker. Beyer "An agile customer-centered method: rapid contextual design" in *Conference on Extreme Programming and Agile Methods.Springer Berlin Heidelberg, 2004.*
- [45] Alyahya, Sultan, Wendy K. Ivins, and W. A. Gray. "Managing versioning activities to support tracking progress of distributed agile teams." International Journal of Computer Applications 39, no. 16 (2012): 12-19.
- [46] M. Rizwan Jameel, Ahmed Barnawi, and Aiesha Ahmad. Qureshi, "Proposal of Implicit Coordination Model for Performance Enhancement Using Sprint Zero." in International Journal of Information Technology and Computer Science (IJITCS), 2012.
- [47] easyredmine.[Online].Available:https://www.easyredmine.com/Accessed:March 2017.

- [48] blossom. [Online]. Available:<https://www.blossom.co/> Accessed: March 2017.
- [49] Trello. [Online]. Available: <https://trello.com/> Accessed: March 2017.
- [50] AgileAgenda.[Online].Available:<http://www.agile.org.uk/tag/agile-agenda> Accessed:March 2017
- [51] atlassian.[Online].Available:<https://www.atlassian.com/software/jira>.Access ed:March 2017.
- [52] GitHub. [Online]. Available:<https://guides.github.com/> Accessed: March 2017.
- [53] Rally. [Online]. Available:<https://www.rallydev.com/> Accessed: March 2017.
- [54] ScrumDo. [Online]. Available:www.Scrumdo.com/. Accessed: March 2017.
- [55] VersionOne.[Online]. Available: <https://www.versionone.com/> Accessed: March 2017.
- [56] Userstories.[Online]Available:<http://www.userstories.com/products/10-mingle> Accessed: March 2017.
- [57] TargetProcess.[Online].Available:<https://www.targetprocess.com/>Accessed: March 2017.
- [58] pivotaltracker.[Online].Available:<https://www.pivotaltracker.com/>Accessed :March 2017.
- [59] BaseCamp. [Online]. Available:<https://basecamp.com/> Accessed: March 2017.
- [60] Blog. [Online]. Available:<http://blog.scrumpad.com/> Accessed: March 2017.
- [61] Asana. [Online]. Available:<https://asana.com/> Accessed: March 2017.
- [62] Aha! [Online]. Available:<http://www.aha.io/> Accessed: March 2017.
- [63] ScrumDesk. [Online]. Available:<http://www.scrumdesk.com/> Accessed: March 2017.
- [64] sprint.ly. [Online]. Available:<https://sprint.ly/> Accessed: March 2017.
- [65] easyBacklog. [Online]. Available:<https://easybacklog.com/> Accessed: March 2017.
- [66] Xplanner. [Online]. Available:<http://ww1.xplanner.org/> Accessed: March 2017.
- [67] collabNet.[Online].Available:<http://www.collab.net/products/scrumworks/A> ccessed: March 2017.
- [68] visualstudio.[Online].Available:<https://www.visualstudio.com/en-us/get-started/work/work-in-sprints-vs/> Accessed: March 2017.
- [69] agilefant. [Online]. Available: <http://www.agilefant.com/> Accessed: March 2017.
- [70] IceScrum. [Online]. Available: <https://www.icescrum.com/> Accessed: March 2017.
- [71] taiga. [Online]. Available:<https://taiga.io/> Accessed: March 2017.
- [72] Agile Forest. [Online]. Available: <http://agileforest.com/> Accessed: March 2017.
- [73] FeatureMap.[Online]. Available:<https://www.featuremap.co/en> Accessed: March 2017.
- [74] AgiloForScrum.[Online].Available:<http://www.agilosoftware.com/>Accessed:March 2017.
- [75] Wrike. [Online]. Available:www.Wrike.com/ Accessed: March 2017.
- [76] AgileWrap. [Online]. Available:www.AgileWrap.com/. Accessed: March 2017.
- [77] VersionOne.[Online].Available:<https://versionone.com/pdf/VersionOne-10th-Annual-State-of-Agile-Report.pdf>. Accessed: March 2017.
- [78] Law, Effie Lai-Chong, Virpi Roto, Marc Hassenzahl, Arnold POS Vermeeren, and Joke Kort. "Understanding, scoping and defining user experience: a survey approach." In Proceedings of the SIGCHI conference on human factors in computing systems, pp. 719-728. ACM, 2009.
- [79] Vermeeren, Arnold POS, Effie Lai-Chong Law, Virpi Roto, Marianna Obrist, Jettie Hoonhout, and Kaisa Väänänen-Vainio-Mattila. "User experience evaluation methods: current state and development needs." In Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries, pp. 521-530. ACM, 2010.
- [80] Jurca, Gabriela, Theodore D. Hellmann, and Frank Maurer. "Integrating Agile and user-centered design: a systematic mapping and review of evaluation and validation studies of Agile-UX." In Agile Conference (AGILE), 2014, pp. 24-32. IEEE, 2014.