

A Practitioner Perspective on Integrating Agile and User Centred Design

Dina Salah, Richard Paige, Paul Cairns
The University of York
Department of Computer Science
York, UK.
{dm560, richard.paige, paul.cairns}@york.ac.uk

This paper discusses an empirical study that identified challenges exhibited by industrial practitioners in integrating Agile development processes and User Centred Design (UCD). The study also identified practices utilised in order to achieve the integration. Fourteen in-depth, one-to-one interviews were conducted with 14 participants from 11 companies of varying sizes in five different countries. Those interviews covered a broad range of issues ranging from usability and user experience goals to incorporating user feedback in Agile teams. The study revealed the existence of a set of challenges including: lack of management support to UCD activities, lack of allocated time for upfront activities, communication between the development team and UCD practitioners, conducting usability testing, absence of UCD practitioners or their increased work load, and lack of documentation. The study also revealed methods utilised by industrial practitioners to tackle some of these challenges via iteration 0, parallel tracks and lightweight documentation. Industrial practitioners can utilise the description of integration challenges and corresponding practices in identifying potential challenges and proposed practices to deal with these anticipated challenges.

Agile Software Development Processes; User Centred Design; Agile User Centred Design Integration

1. INTRODUCTION

Agile methods are lightweight software development methods that aim to deal with volatile requirements via discarding upfront, precisely defined plans. They are iterative and are used to develop software incrementally. Different Agile processes implement these ideas in different ways. All Agile processes share common values and principles, defined in the Agile Manifesto Beck (2000).

User Centred Design (UCD) is a set of techniques, methods, procedures and processes as well as a philosophy that places the user at the centre of the development process in a meaningful, appropriate and rigorous ways Gould and Lewis (1985); Detweiler (2007). The goal of applying UCD is to attempt to satisfy users via producing usable and understandable products that meet their needs and interests Detweiler (2007). The usability of a product is the consequence of systematic UCD work that occurs throughout the development process and continues even after product release in order to enhance subsequent versions Detweiler (2007).

In the past decade there has been an increased industrial and research interest in Agile and User

Centred Design Integration (AUCDI) in the Agile and UCD community. This interest in AUCDI is arguably due to three reasons: first, the reported advantages of UCD on the developed software as it enables developers to understand the needs of the potential users of their software, and how their goals and activities can be best supported by the software thus leading to improved usability and user satisfaction. Second, the Agile community hardly discusses users or user interfaces, thus implying either a negligence of UX or focus on less sophisticated UX projects Armitage (2004). Moreover, none of the major Agile processes explicitly include guidance for how to develop usable software Lee, McCrickard and Stevens (2009). In addition, the interaction design role, usability, and user interface design in an Agile team is unclear and largely overlooked Constantine (2001). Furthermore, principles and practices for understanding and eliciting usability and user requirements and evaluating Agile systems for usability and UX are generally considerably deficient Lee, McCrickard and Stevens (2009). Third, there appear to be philosophical and principled differences between Agile methods and UCD that suggest that their integration will be fundamentally challenging.

This paper provides details of an empirical study that was conducted on Agile and UCD integration. This study aimed to identify various challenging factors that restrict Agile and User Centred Design Integration and to explore the proposed practices to deal with these challenges.

The rest of this paper is structured as follows: section 2 discusses the research method and details the data collection and data analysis method. Section 3, discusses the participants' and projects' profiles. Section 4 provides details of the interview results in regards to AUCDI challenges and practices. Section 5 discusses the conclusion.

2. RESEARCH METHOD

This section provides the details of the research method used in conducting the interview study including details on the data collection method and data analysis method used.

2.1. Data Collection

The objective of this research was to generate *themes* built on the experience and perspective of industrial practitioners, whether UCD practitioners or developers. Thus a qualitative approach was chosen as the appropriate method for data collection. In this study semi-structured interviews were utilised for data collection since they are flexible, adaptable, offer the possibility of following up interesting responses, and provide rich and highly illuminating material Robson (2011).

A number of interview questions were set in advance. The first version of the interview questions were used in conducting a pilot interview. This initial interview highlighted several drawbacks with the interview questions and the interview guide. Most of the drawbacks were related to the length of the interview or the wording of the questions. As a result the number of interview questions were decreased and edited for more clarity.

Networking was utilized to reach the initial sampling that involved Agile and UCD practitioners working on projects that utilise Agile methods and developing software that places significant importance on usability and UX. Fourteen in-depth, one-to-one interviews (either face-to-face or via Skype) were conducted with 14 participants from 11 companies of varying sizes in five different countries including the United Kingdom, Canada, Netherlands, Portugal, and Egypt. Interviews lasted between 66 and 180 minutes (approximately one to three hours). The interviews covered a broad range of issues ranging from usability and UX goals to incorporating user feedback in Agile teams.

Hand written notes were utilised by the interviewer during the pilot interview. However, after conducting the pilot study it was concluded that recording the interviews and then later transcribing it would make the interviewer more efficient and focused in the interview. Thus all interviews, except the pilot interview, were audio recorded after acquiring the informed consent of the interviewees. This was followed by transcribing all interviews.

2.2. Data Analysis-Thematic Analysis

This section discusses the data analysis method. Thematic analysis is a method for identifying, analysing and reporting patterns (themes) within data Boyatzis (1998); Braun and Clarke (2006). A theme captures something important about the data related to the research question, and represents patterned meaning within the data set Braun and Clarke (2006). Thematic analysis minimally describes and organises the data set in rich detail and interprets various research topic aspects Boyatzis (1998).

Phases of Conducting Thematic Analysis

Phases of conducting thematic analysis were adopted as follows Boyatzis (1998):

1. Familiarisation With Data

This phase involved transcribing data, reading and re-reading the data, noting down initial ideas. It resulted in familiarity with the depth and breadth of the content.

2. Generating Initial Codes

Codes refer to "the most basic segment, or element, of the raw data or information that can be assessed in a meaningful way regarding the phenomenon" Boyatzis (1998). This phase involved coding interesting data features in a systematic fashion across the entire data set and collating data relevant to each code. This phase resulted in the generation of 237 initial codes that identify interesting data features that are meaningful to the studied phenomena. However, after removing redundant codes and merging codes with similar meaning, 179 codes were reached.

Examples of those codes are requirements elicitation techniques, usability goals, user experience goals, UCD funds, parallel tracks, upfront requirements gathering, low fidelity prototypes.

3. Searching For Themes

This phase involved collating codes into potential themes, gathering all data relevant to

each potential theme and collating all the relevant coded data extracts within the identified themes. Examples of themes that were found included: AUCDI challenges, usability testing methods, and AUCDI practices.

An access database was created for the results of this phase in which a number of fields were kept including a short extract, category, participant, and company.

4. Reviewing Themes

This phase involved the refinement of candidate themes to ensure that data within themes cohere meaningfully together and that clear and identifiable distinctions exist between themes. This occurred via checking whether the theme is relevant to the coded extracts (Level 1) and the entire data set (Level 2) and ended by generating a thematic map of the analysis. Thematic maps provide a visual representation of relevant themes. In this phase some themes were excluded due to a variety of reasons, for example, lack of enough supportive data while other themes were merged into each other. Other themes were broken down into separate themes.

This phase ended by understanding how the different themes fit together and the overall story they tell about the data.

5. Defining and Naming Themes

This phase involved refining the specifics of each theme and the overall story that the analysis tells in relation to the research questions. This phase resulted in generating clear definitions and names for each theme that reflect the essence of each theme and overall themes.

2.3. Participants' and Projects' Profiles

The following sections provide details on participants' and projects' profiles that were involved in this empirical study.

2.4. Participants' Profiles

This section will introduce participants' profiles and their selection process.

Since the research was focused on exploring the space of different environment, thus through networking and recommendations, 14 participants were purposefully recruited. The study aimed to interview industrial practitioners from a variety of countries, varying size companies, and who develop different types of software at different stages of development. Moreover, participants were included from projects that developed new

software and new user interface designs, and other participants from projects that developed new versions of existing software and existing user interface designs. Participants who worked in successful and unsuccessful projects were also included.

Table 1 provides an overview on participants profiles.

No.	Job	Work Experience Years	Agile Experience Years	Education
PT1	Technical team leader	7	1	Bachelor in CS
PT2	Team leader and a system analyst	3.5	1.5	Bachelor in CS
PT3	Technical architect	10	2	Bachelor in IS and CS
PT4	Principal technical consultant	14	7	Bachelor in IS and CS and masters and PhD in IS
PT5	Independent technical consultant	10	2	Bachelor in CS
PT6	Lead software engineer	8	2	Bachelor in structural engineering and masters in CS
PT7	Senior developer	5	4	Bachelor in CS
PT8	Interaction design group leader	2	1.5	Postgraduate studies in user system interaction
PT9	Software engineer	1.5	1.5	–
PT10	Product manager	17	13	Bachelor in CS and masters in BA
PT11	Product manager	12	6	Bachelor in CS and masters in BA
PT12	Vice president of technology and scrum master	5	1	Bachelor in CS
PT13	Business analyst	15	8	Bachelor degree and PhD in computing and maths
PT14	Independent technical consultant	12	6	Bachelor in CS and management

Table 1: Participants Overview

Participants were resident in five different countries including United Kingdom, Canada, Netherlands, Portugal, and Egypt. Participants will be referred to as PT1..PT14. All of the interviewees were on different development teams working for different companies with the exception of PT6 and PT11 who worked for the same company and in the same project but played different roles, and PT2 and PT12

who worked for the same company and in the same project but played different roles in the development team. Participants PT1, PT6, and PT11 all worked for the same company.

2.5. Project Profiles

This section will introduce the profile of the projects covered in the interviews. The study tried to cover a wider variety of projects so some of the projects involved completely new software and new user interface designs, while others were new versions of existing software and existing user interface designs. Some projects are finished and others are still ongoing. Projects will be referred to as P1..P12.

Table 2 provides an overview on project profiles.

No.	Agile Method	Domain	Duration
P1	Scrum	School educational portal	4 Months
P2	Scrum	Business intelligence application for marketers	4 Years
P3	Scrum	Enterprise resource planning system	1 Year, then aborted
P4	Microsoft Solutions Framework Agile	Governmental data warehouse portal and dash board	10 Months
P5	Mixture of Scrum and Kanban Lean Method (Scrumban)	Project management tool for Agile projects	6 Months
P6	Mixture of Scrum and XP	Educational authoring tool	4-5 Months
P7	Mixture of Scrum and XP	Content management	9 Months
P8	Scrum	Portable global positioning	4-5 Months
P9	Mixture of Scrum and XP	TV guide	1.5 Year
P10	Mixture of Scrum and XP	Internet service provider software	Ongoing
P11	Scrum	Extranet portal	2.5 Years
P12	Microsoft Solutions Framework Agile	Enterprise resource planning system	1 Year

Table 2: Projects Overview

3. INTERVIEW RESULTS

This section discusses the interview findings via discussing the main themes that emerged and linking each theme with the relevant interview data in order to illustrate and clarify the results. Some conventions were utilised in reporting quotes; in case one of the quotes include words such as they, he, etc. the meaning was explained between square brackets []. In case the quote includes a sentence and then few irrelevant sentences and then another relevant sentence, the convention sentence one [...] sentence two was used, where [...] signifies the irrelevant sentence.

3.1. Theme one: Agile and User Centred Design Integration Challenges

The first theme that emerged is related to the reported challenges that were faced by participants in their industrial attempt to integrate Agile development processes and UCD.

3.1.1. AUCDI Challenge One: Lack of Management Support to UCD Activities

Participants reported the lack of management support to UCD efforts. This was attributed to a variety of reasons including lack of management awareness of UCD impact on the overall quality of the product, lack of awareness on the importance of UCD practitioner role, tight schedules, and lack of funds. This lack of management support resulted in management reluctance to allocate time, priority or funds for hiring a usability engineer and conducting UCD related activities including usability studies, usability testing, etc.

Participant PT3 discussed management reluctance to allocate a usability engineer for the project as a result of management's lack of awareness of the importance of UCD and UCD practitioner role. Participant PT3 stated that

I tried to get a usability engineer on board and [...] there was a new project manager and she thought there was no use for a usability engineer and that it was not useful [...] or important for the project [...] she thought it was not worth it and that we are wasting time, effort, and money.

Participant PT10 noted his failure to convince management to allocate funds for conducting usability studies. Participant PT10 also stated that

At one stage I tried to have a discussion with the company on having some usability studies done on the product and they were not willing to fund.

Participant PT14 did not conduct usability testing on users but rather on customers due to budget constraints forced by management. He stated that

What prevented me [from testing on users] was not having full time or more than one usability engineer in order to do more research and some limitation with the budget.

It can be observed from participants PT3, PT10, PT14, PT11, and PT6 quotes that management support to UCD activities is essential and can either enhance or hinder UCD activities' inclusion in team based planning and scheduling, and execution.

3.1.2. AUCDI Challenge Two: Lack of Allocated Time for Upfront Activities

Participants reported the lack of allocated time for conducting upfront activities due to the focus

on frequent delivery of software within shorter timescale. As a result some participants struggled to allocate time for upfront activities including: analyzing competitive products (PT11), conducting user research and gathering requirements (PT1, PT2, PT3, PT4, PT5, PT6, PT8, PT13), and design activities.

Participant PT8, who worked as an interaction design group leader within the UX team, expressed the difficulty faced by the UX team to conduct upfront activities due to the tight Agile time scales and lack of management support to UCD activities by stating that

We are pushing as priority, UX takes less priority compared to functionality and this is something that we struggle with right now and also because in this particular project that I am referring to, we had a head start [iteration 0] [...] so we are challenged to have this head start before sprint one and this is something that is hard to convince the scrum master and the project team that this is needed.

3.1.3. AUCDI Challenge Three: Communication between the Development Team and UCD Practitioners

Another reported challenge was the communication problems between the development team and UCD practitioners. This had multiple origins including the following:

- Lack of awareness of development team members of the role of UCD practitioner and its importance.
- The absence of a clear road map for communicating between UCD practitioners and the development team to clarify how UCD activities will fit in the Agile iterative development life cycle.
- Presence of UCD practitioner as a part time rather than a full time team member.

Communication problems were maximized in distributed teams where UCD practitioners and developers were not collocated.

The Agile development process changed the relationship between UCD practitioner (or the team member playing their role) and the rest of the development team Ferreira, Noble and Biddle (2007). The incremental and iterative nature of Agile development processes requires continuous communication between both parties Detweiler (2007). This continuous communication involves UCD practitioners sharing design vision, rationale and designs with developers at regular intervals, clarifying design related questions and accepting developers feedback

on design issues. The interviews conducted revealed that the absence of continuous communication resulted in frustration among developers, lack of synchronization, delays and bottle necks in the Agile development process.

Participant PT9 was a developer who worked in a distributed team where developers and UX team were located in different countries. Participant PT9 shows an example that reflected the frustration raised among the development team as a result of failure of UX team to share with the development team the user needs and goals collected from requirements elicitation phase. When participant PT9 was asked about the technique used by the UX team for user requirement elicitation he replied that

No [PT9 does not know] and I kind of wish I did but I do not know [...] I wish I knew more about it because our team is entirely frustrated [...] I am totally blind on that.

This resulted in lack of developers' understanding or approval of the design vision as noted by participant PT9 who stated that

We are trying to understand some of the decisions that were made or have been made because we cannot imagine how this is a good UX but we are not involved in that and at best we can provide some feedback but the feedback we provide is that it [the user interface] does not do what is required but we do not get to affect the requirements very much if at all.

Although participant PT9 faced difficulties that resulted from lack of shared product and design vision, participant PT11, who worked as a product manager, reported a different perspective that shows the importance of involvement of developers in the design vision process. Participant PT11 was a product manager in the same project as participant PT6 and tried to raise awareness among the development team to the importance of usability issues. This was achieved via sharing the product vision, team vision, results of user studies, usability tests and product feedback. Moreover, the development team was encouraged to attend meetings with users and jointly deciding on high level product goals with the development team.

Participant PT11 efforts resulted in creating a shared team ownership of usability and a shared vision that focuses on users and usability related issues as is evident in participant PT6 quote. Participant PT6, who worked as lead software engineer in the same project as participant PT11, stated that

The ownership of quality and usability was shared, and it was a spirit the team had as a clear objective,

because at the beginning of the project when we were making release planning, we put team value and we put ourselves in a state on what is the objective of what we are doing, so [...] [PT11] had a great concentration on usability so this was something totally on our [team] mind.

Another form of communication problems were revealed in power struggles between developers and UCD practitioners [or the team member playing their role] and was reported by participant PT1, PT9, and PT13. Participant PT1 discussed how the part time usability engineer faces power struggles with developers or customers. When asked about what could be improved in his development process he stated that

Having another usability engineer for consultancy because one usability engineer can sometimes get challenged from developers or customers.

Participant PT13, who worked as a business analyst but acted as the usability engineer, also reported issues related to power struggles between developers and UCD practitioners, she stated that

The guy who worked in the corporate for I do not know five years prior to that was really upset because he kind of went from being the go-to-guy to not being the go-to-guy and the business people went from complaining all the time to praising all the time and that did not work out too well [...] he felt that he did not need anyone that he knew best and he got really upset that he could not ignore me and go ask the user what they want and ignore the exploring stuff.

Participant PT9 also discussed how the UX team ignored the development team’s feedback by stating that

Most of the feedback provided by the development team on UX was ultimately not taken into account, I mean we could say all what we wanted but ultimately no one was actually listening.

The issues related to power struggles reported by participant PT1 and PT13 reflect the importance of acceptance of UCD practitioners by the development team. This acceptance can occur by increasing UCD awareness among team members in order to understand the importance of UCD practitioner’s role and what is involved in his work and as a result have realistic expectations from him. However, participant PT9 comment indicates the importance of acceptance of development team comments by UCD practitioners since this can lead to shared UI ownership.

3.1.4. AUCDI Challenge Four: Conducting Usability Testing

The iterative, rapid Agile time lines, budget constraints, difficulty of accessing users, and lack of full time usability engineers resulted in lack of user testing. This resulted in testing on customers, developers, Quality Assurance (QA) team, conducting hallway testing, or delaying usability testing (sometimes after product is released).

Table 3 provides details on the involved parties in usability testing in the different projects. Only 6 projects out of 12 (P2, P3, P6, P10, P11, P12) conducted usability testing on users whereas the rest of the development teams resolved to testing on development team members, quality assurance team (testers), UCD practitioners, members from other development teams and / or customers.

Project No.	Testing on Users	Others Involved in Testing
P1	No	Customers, quality assurance team, product manager, usability engineer
P2	Yes	Users, team leader, customers, testers
P3	Yes	Users, developers, customers, testers
P4	No	Developers, customers
P5	No	Developers, testers, product manager
P6	Yes	Users, quality engineers, developers, product manager, internal project manager, customers
P7	No	Customers
P8	No	Developers, testers, UX practitioners
P9	No	Customers, quality assurance team, developers
P10	Yes	Users, customers
P11	Yes	Users, customers
P12	Yes	Users, customers

Table 3: Usability Testing

A number of reasons were provided for not testing on users including: the iterative, rapid Agile time lines, difficulty of accessing users, lack of funds, the software being a new version of an existing software or being developed to be used by developers in case of developing project management tools.

Participant PT1 did not test on users due to difficulty of accessing users and lack of funds as stated in the following quote

Due to difficulty to get teachers and students when is suggested to get them.

When participant PT1 was asked for reasons of testing on customers rather than users, he stated that this was because of

Time [lack of it] and difficulty of testing on users.

Participant PT8 did not test on users since his company develops one type of software and the

software developed was a new version of an existing software. Participant PT8 conducted peer testing via hallway testing with developers and field test via developers, participant PT8 stated that

We have a research team that does usability studies and it depends on the goal basically but it ranges from hallway usability testing so that is peer to peer with people on the floor as the most Agile way of testing things [...] we invite people from outside the office to help us in more in depth usability studies and sometimes it is in the office but often we use the car to drive around as in ourselves or we invite people to join us, so that is another really quick way to test our products.

Participant PT5 developed a Project Management (PM) tool for Agile projects, a visual dash board for monitoring and tracking Agile teams' project status. He did not test on users since the software developed was for developers and thus he thought that the development team will be sufficient to test the software. Participant PT5 stated that

The advantage that we have is that we can relate to the user easily because the users resemble us because they are technical people and we wanted to make an application that we ourselves will like to use, so this was our advantage.

A number of projects (P1, P2, P3, P5, P6, P8, P9) involved the quality assurance team (testers) in testing the software functionality and usability.

Participant PT1 declared that QA team tested both functionality and usability every two days and provided the usability engineer with feedback. Participant PT1 stated that

QA tested every two days since they tested with every build which is every two days and could give feedback to usability engineer so were checked and validated and corrected, feedback was right away as defects [reported right away].

Participant PT14 conducted usability testing regularly on customers where they were observed while using the low fidelity or high fidelity prototypes. He stated that

Every two weeks we made a demo to customer this occurred all along until end of four months.

3.1.5. AUCDI Challenge Five: Communication between the Customer and the Development Team

A number of reported challenges for integrating Agile development processes and UCD were related to customers. These included misrepresentation of customers in Agile teams and lack of customer awareness on Agile methods.

Misrepresentation of Customers in Agile Teams

Participants reported that customers were misrepresented in Agile teams: customers did not present the wider user population and were not aware of user needs, problems or goals. Moreover, some participants remarked that customers did not have the authority to take decisions when needed. Misrepresentation of customers in Agile teams led to erroneous decision making in regards to user needs, problems or goals.

Participant PT3 expressed the lack of customer authority and knowledge in the domain by stating that

I think what would have helped is to educate the customer, [...] he just had some practices which he learned throughout the years, [...] he did not have in my opinion the best inventory practices [...] he did not have the authority or the knowledge.

Participant PT14 also expressed the lack of customer domain knowledge which led to erroneous decision making in regards to user needs, problems and goals by declaring that

Sometimes the customer did not have the real knowledge about how things really work [...] Yes that created some problems because in some cases we had to redo some things.

Lack of Customer Awareness on Agile Methods

Lack of customer awareness on Agile methods was shown in customer reluctance to provide continuous and frequent communication with the development team. This led to delayed customer feedback and made it harder for the development team to achieve customer and subsequently user satisfaction. Some participants reported that customers lack of frequent and continuous communication at the requirements gathering phase and usability testing phase led to delayed feedback that hindered Agile team's productivity.

Participant PT1 expressed the importance of customers awareness on Agile methods to achieve continuous involvement and frequent communication

Agile is not suitable for projects in which customer does not understand what is Agile, customers should be educated and know that they should be more involved and give frequent feedback.

Participant PT9 expressed the effect of customer latency in providing feedback to the development team by declaring that

We would ship something to partners [customers] every month and they would not start using it until

another month so it will be two months [...] it was frustrating to us as developers but we could not convince them [customers] that if this was just a shorter cycle it would be much better.

Participant PT2 discussed the failure of customers in abiding by the tight agile time lines in requirements gathering by stating that

Requirement gathering is painful for customer because he is a business customer so he has a lot of work to do. So always asks about can you send this to me by mail and I will get back to you.

3.1.6. AUCDI Challenge Six: UCD Practitioners

Another set of challenges that impacted the integration efforts were related to UCD practitioners. These included the absence of UCD practitioners in Agile teams and their presence as a shared resource.

Absence of UCD Practitioner

Some of the participants' teams included a UCD practitioner whereas others did not. As it can be shown from table 4, 8 out of 12 projects did not include a usability engineer. As a result this role was played by another team member including: developers, business analysts, technical consultants or designers. However, this raises concerns on whether those team members were qualified to play this role. Moreover, this resulted in delays in the development process, difficulty of inclusion and prioritization of usability or UX features in planning and scheduling activities, discarding some UCD activities such as user testing, and compromising the usability or UX of the software.

Project No.	Presence of Usability Engineer (UE)	Team Member Performing UE Role	UE Status
P1	Yes	–	Part Time
P2	No	Developers	–
P3	No	Graphics Designer	–
P4	No	Principal Technical Consultant	–
P5	No	Developers	–
P6	Yes	–	Part Time
P7	No	User Interface/User Experience Designer	–
P8	Yes	–	Part Time
P9	Yes	–	Part Time
P10	No	Business Analyst	–
P11	No	Independent Technical Consultant	–
P12	No	Graphics Designer	–

Table 4: Role and Responsibility for Usability and User Experience

Participant PT12 was discussing the implications of the lack of UCD practitioner and how this role was

played by developers and the impact that this had on the project progress, he stated that

Most of my time is wasted in UI [...] people [developers] who ask for time extension always say due to UI problems because they are developers and I have no UX designer so they take time, sometimes due to inconsistencies someone may overwrite on the other [UI work].

Participant PT5 stated that the lack of UCD practitioner led to lack of formal usability testing, he stated that

The problem with implementation up till now is that we do not have formal usability methods, we need to have a usability engineer who works with some formalization for usability methodologies up till now everything [usability issues] was implicit.

UCD Practitioner Workload

Table 4 shows that only four teams (P1, P6, P8, P9) included a usability engineer. Although Project P7 had a UI/ UX designer and projects (P3, P12) had a graphics designer yet those team players were only responsible for interface design activities and not usability engineering activities. Moreover, those four teams that included a usability engineer had him as a shared resource and acted as a part timer. Although this is a rather common situation, arguably an Agile development process adds more burden to UCD practitioners due to the Agile nature that requires team members to attend a number of meetings for all the teams they are involved in. Moreover, usability engineers do not have enough time to finish their work due to the rapid and iterative Agile time lines. The consequences of this included discarding some UCD activities including gathering requirements from users and instead gathering it from customers (PT14). Moreover, UCD practitioner workload led to slow response from usability engineers that eventually led developers to play the role of UCD practitioner in case of his absence (PT6).

Participant PT14 had a part time usability engineer who was contracted to work 40 hours per month, when asked about feedback on this model of work he stated that it led to lack of requirement gathering on user needs and subsequently failure of the software to satisfy users

[...] the funds for design were cut out because the customer wanted everything a lot cheaper and my administration thought that that was the person to get rid off and I could not really do anything about it, just 40 hours per month was the best thing I had [...] The disadvantages because we had to drop something like the usability engineer [as a full timer] and like some usability requirement gathering and

some other parts we lacked some more knowledge about the user [...] we were replacing an existing one [software] and we failed and we were trying to make something similar to the previous one [software] and at the same time innovative and that was really not very easy to do and not very interesting to do at the end result we should really have a usability requirement gathering in order to address the project because in some cases the customer was saying to us no no they really enjoy how this application is built and the end user would tell us no no this sucks.

Participant PT6, lead software engineer, expressed the impact of not having a full time usability engineer by stating that

The people who work in graphics and usability are shared resources [...] They are actually allocated for a while but we need to arrange for more time to sit with him [...] the problem in this is that he [usability engineer] can leave us before the end of the project.

Thus it can be concluded from theme one that AUCDI challenges have various causes related to three dimensions: UCD infrastructure, AUCDI process, and people involved in the integration process. First, UCD infrastructure is exemplified in challenge one; lack of management support to UCD activities. UCD infrastructure is embodied in the allocation and utilisation of dedicated funds for conducting UCD related activities as well as management support to UCD activities. Second, AUCDI process is exemplified in challenge two; lack of allocated time for upfront activities and challenge four; conducting usability testing. AUCDI process is related to a development life cycle that embraces the planning and inclusion of UCD activities within the iterative and incremental Agile development process. Third, people is related to the communication and attributes of those involved in the AUCDI process, for example, customers, users, developers, and UCD practitioners. It is exemplified in challenge three; communication between the development team and UCD practitioners, challenge five; communication between the customer and the development team, and challenge six; UCD practitioners.

Those three dimensions affect AUCDI endeavors and accordingly any challenges related to them need to be tackled for successful integration of Agile and UCD.

3.2. Theme Two: Agile and User Centred Design Integration Methods

The second aim of this empirical study is to identify the proposed integration methods. This section includes the emerged themes regarding integration methods for Agile development processes and user centred design. Those include methods for

upfront design, synchronizing the activities of developers and UCD practitioners, and lightweight documentation. Those methods were listed below according to the order of their utilisation in the Agile development life cycle in order to achieve Agile and UCD integration.

Iteration 0

The different Agile teams faced challenge two; lack of allocated time for upfront activities via allocating an iteration 0 for UCD practitioners or the Agile team member playing their role to gather requirements, understand users, user goals and context of use, prioritize features and conduct upfront design in order to achieve holistic design vision. Iteration 0 was utilised by developers in working on back end features such as selecting system architecture and development environment.

Iteration 0 was used by participants for analyzing competitive products (PT11), requirements gathering (PT1, PT2, PT3, PT4, PT5, PT6, PT8, PT13), and upfront design. The length of iteration 0 varied among the different teams, since some teams, for example, PT6, had more than iteration 0.

Participant PT8 discussed activities performed in iteration 0 by stating that

The first sprint for development was focused on non UI features basically there was no team work needed for it but later on there was and that was good because the designer had time to design and write special cases [...] developers were working on back end [...] mostly about architecture and server side complexity that they need to solve. It [iteration 0] was about optimizing code and preparation mostly for the functionality.

Participant PT6 noted that his team utilised more than iteration 0.

Actually we had more than one iteration 0, we took a while to collect UI requirements and functional requirements and comparing with other products.

Parallel Tracks

Parallel tracks Sy (2007) were utilised by participants (PT1, PT2, PT3, PT5, PT7, PT8) in order to coordinate the work between UCD practitioners and developers via organizing work in two parallel and interrelated tracks. Parallel tracks organizes work around a number of cycles: cycle n, involves developers working on back end features and UCD practitioners working on design of features that will be implemented by developers in cycle n+1 via building prototypes. These prototypes are used to test the design, conduct design test,

and fix prototype errors. Cycle n+1 involves UCD practitioners presenting the designs from cycle 1 to developers in order to implement them and UCD practitioners working on prototyping and usability testing for cycle n+2 features. These cycles continue until design goals are achieved. Parallel tracks Sy (2007) were utilised by a number of participants including participants (PT1, PT2, PT3, PT5, PT7, PT8) in order to coordinate the work between UCD practitioners or the team member playing their role and developers via organizing work in two parallel and interrelated tracks.

Documentation Methods

A number of issues needs to be documented including user requirements, design rationale, prior designs, usability testing procedures, designs to be implemented, and their associated delivery schedule and usability testing results Sy (2007). A number of different simple, easy, and lightweight methods were used by the different teams for documentation purposes including: Wiki (PT3, PT6, PT7, PT10, PT14), Excel (PT3, PT6, PT7, PT10, PT13), and information wall (PT1, PT8, PT9, PT10, PT6, PT13). All these methods are simple, easy, and lightweight.

4. CONCLUSION

This paper investigated industrial practices for integrating Agile development processes and UCD. The study revealed the absence of a road map for AUCDI. This was aggravated by facing a set of challenges including lack of allocated time for upfront activities, difficulty of chunking, communication between the development team and UCD practitioners, conducting usability testing, absence of UCD practitioners or their increased work load in case of their presence, and lack of documentation.

The interview study also revealed that in addition to those challenges, Agile teams suffer from the additional challenge of lack of management support to UCD activities that occurred to participants (PT3, PT10, PT14). This was attributed to a variety of reasons including lack of management awareness on UCD impact on the overall quality of the product, lack of awareness on the importance of UCD practitioner role, tight schedules, and lack of funds. This lack of management support resulted in management reluctance to allocate time, priority or funds for hiring a usability engineer and conducting UCD related activities including usability studies, usability testing, etc.

The study revealed methods utilised by industrial practitioners to tackle some of these challenges

via iteration 0, parallel tracks and lightweight documentation. Industrial practitioners can utilise the description of integration challenges and corresponding practices in identifying potential challenges and proposed practices to deal with these anticipated challenges.

Industrial practitioners can benefit from a clear road for integrating Agile development processes and UCD that provides roles, activities, and responsibilities involved in the integration. As long as such a road map is absent then industrial practitioners will continue to come up with their own integration approaches or solutions.

REFERENCES

- Armitage, J. (2004). Are Agile Methods Good for Design? *Interactions* 11(1), 14–23.
- Beck, K. (2000). Manifesto for Agile Software Development.
- Boyatzis, R. (1998). *Transforming Qualitative Information: Thematic Analysis and Code Development*. Thousand Oaks, CA: SAGE.
- Braun, V. and V. Clarke (2006). Using Thematic Analysis in Psychology. *Qualitative Research in Psychology* 3 (2)(ISSN 1478-0887), pp. 77–101.
- Constantine, L. (2001). Process Agility and Software Usability: Towards Lightweight Usage Centred Design.
- Detweiler, M. (2007). Managing UCD within Agile Projects. *Interactions* 14(3), 40–42.
- Ferreira, J., J. Noble, and R. Biddle (2007). Agile Development Iterations and UI Design. In *Proceedings of the AGILE 2007, AGILE '07*, Washington, DC, USA, pp. 50–58. IEEE Computer Society.
- Gould, J. and C. Lewis (1985). Designing for Usability: Key Principles and What Designers Think. *Communications of ACM* 28(3), 300–311.
- Lee, J. C., S. McCrickard, and T. Stevens (2009). Examining the Foundations of Agile Usability with eXtreme Scenario-Based Design. In *Agile Conference*, pp. 3–10.
- Robson, C. (2011). *Real World Research: A Resource for Social Scientists and Practitioner-Researchers*. Blackwell Publishers Ltd.
- Sy, D. (2007). Adapting Usability Investigations for Agile User-Centred Design. *Journal of Usability Studies* 2(3), 112–132.