

# USABILITY EVALUATION PRACTICES WITHIN AGILE DEVELOPMENT: ENGAGING WITH USABILITY AGILE PRACTITIONERS' CONCERNS

AZIZ BIN DERAMAN<sup>1</sup>, FOUAD ABDULAMEER SALMAN<sup>2</sup>

<sup>1</sup>Faculty of Ocean Engineering Technology and Informatics, University Malaysia Terengganu (UMT),  
Malaysia

<sup>2</sup>Engineering Technical College, Al-Bayan University, Baghdad, Iraq

E-mail: <sup>1</sup>a.d@umt.edu.my, <sup>2</sup>fouad.a@albayan.edu.iq

## ABSTRACT

The success of developing software systems largely depends on the effectiveness of coordination between software developers and usability engineers. Yet its integration has shown not to be straightforward because of various priorities and approaches which make it further complicated. This paper proposes an approach for improving the coordination between agile developers and usability engineers. The contribution of this paper is to investigate the challenges that prevent such integration. The investigation involved a combined questionnaire survey and interviews with participants from software development companies. The result reveals that during the development process, there is a lack of collaboration between usability practitioners and agile developers. This lack is classified into three main aspects: activities management, artifacts management and communication management. Also, the authors describe an integration model to be used as a solid basis to mitigate the challenges and propose a policy agenda for future work toward improving the coordination in multidisciplinary agile usability software teams.

**Keywords:** *Usability, Usability Evaluation, Software Development Process, Usability Management*

## 1. INTRODUCTION

Agile methods have revolutionized how software development teams are organized. Agile approaches handle software development in short iterations with continuous testing and a flexible delivery date [1] as opposed to months of development followed by months of field tests preceding a release. The agile approach promises to produce more valuable and reliable software with better control than traditional development methods. It leans toward engaging all developers, who do not only have a shared understanding of the design but are equally qualified to work on any part of the system [1][2]. This approach may lead to better communication between agile team members and increased flexibility regarding who does what work. But, it can be difficult for usability practitioners to work with an agile team, given their different focus areas, backgrounds, and concerns.

Usability engineering is characterized by the active involvement of end users and a clear

understanding of their needs and expectations; proper distribution of work between users and technology; the iteration of design solutions; and multidisciplinary design [3]. In addition, the usability practitioners strive to develop products tightly aligned with the user's needs instead of technology-centred products.

The sprint iterations environment seems complicated for usability practitioners to work in, given their different focus areas, backgrounds, and concerns. This is especially true with the quantity and variety of techniques and methodologies in both fields, further complicating implementation. Dissimilarities in backgrounds, concerns, and practices lead to a lack of collaboration between agile development members and usability engineers [3]. The agile software development process has to deal with two sorts of complexity: the complexity of the artifacts being produced or involved, e.g. requirement, prototype or code, and the complexity of the activities concerning those artifacts (the distributed software development process).

While much work have been discussed on integrating usability work into the agile development process, limited articles tried to develop a deeper understanding of the situations and challenges that hamper engaging usability practitioners with the agile development team. Also, these articles usually tell positive stories of problems solved rather than describing persistent difficulties, worsening situations or failures. As a result, they provide snapshots of successful practice but almost certainly do not represent the state of the practice. Indeed, few articles describe major unresolved problems or failures, resulting in a general publication bias towards only reporting success. This paper presents a study that investigates these obstacles to address this limitation. Based on the obtained results, a unified approach is proposed for reconciling usability work with the agile development environment. The author's strategy contributes to a set of combination information about the structure of artifacts and activities.

## 2. AGILE SOFTWARE AND USABILITY

There are much research have been done by researcher to simplify cooperation between software engineering and usability practitioners. According to Lee and McCrickard [4], tension between usability and software development approaches stems from differing aims and motivations of software engineering (SE) and Human Computer Interaction (HCI) practitioners, and is further complicated by the variety of techniques existing in both fields. The differences of aims between SE and HCI practitioners [4] negatively affect the aforementioned cooperation. SE practitioners focus on designing, implementing and maintaining software, tending to minimize the relevance of human-computer interfaces. In contrast, HCI practitioners focus on developing highly user oriented software to allow effective use of software. This dissimilarity of goals could lead to a lack of collaboration during the development stages. Jerome and Kazman [5] found that SE and HCI practitioners do not closely collaborate with other professionals outside their knowledge areas. Furthermore, their limited collaboration tends to occur too late in the software development process, which reduces its effectiveness.

Seffah and Andreevskaia [6] proposed educating software engineers on certain usability concepts with an economic educational framework. Conversely, Faulkner and Culwin [7] proposed the

adoption of human-computer interaction (HCI) principles to guide the development of computer systems. Evidently, these approaches do not emphasize cooperation, as they both draw heavily on concepts from either software engineering or usability engineering. These strategies tilt more towards either field. This is disadvantageous to the problem of incorporating usability evaluation approaches within agile software development, which is increasingly acknowledged and addressed [8]. In fact, processes in both fields have many similar foundational concepts, such as iterative development and being user-focused. Singh [9] presented a U-SCRUM methodology that incorporates usability evaluation concepts within agile methods. Unlike typical SCRUM, USCRUM has two product owners. One emphasizes usability, while the other emphasizes more typical functionality. Despite the fact that diverse integration approaches are continuously being introduced and developed utilizing varied methodologies and techniques, how to manage this integration is either undocumented or inadequately described in the literature. Determining how to bridge this gap would enhance the exchange of shared knowledge, hence increasing the success of the collaboration.

## 3. INVESTIGATION OF THE OBSTACLES AGAINST THE INTEGRATION

To determine the management obstacles that hinder the coordination between usability and agile practitioners, the authors conducted a survey involving two stages: (i) An online questionnaire survey. The questionnaire was distributed to Agile development teams who are experienced in using agile methods in their software development process and use usability techniques. The survey was constructed in Google Form, an online survey tool. The survey included 20 questions. The questions were developed, reviewed, and edited based on experts from academic and industry fields. The questions in the questionnaire were grouped into four sections: information on the companies and respondents' experience, activities management, artefact management, and communication management. (ii) A series of interviews with software development practitioners are conducted. According to [11], semi-structured interviews and interview aids were developed. The core questions of the interviews were about to which extent they are familiar with using different usability techniques and how they manage the activities, tasks, and communication, as well as

what kind of obstacles they encountered when working on a project as a multidisciplinary team.

### 3.1 Participants

The respondents of this study were chosen from different specializations based on several common characteristics [10] such as:

- Developing software with a graphical user interface (e.g. web applications, mobile applications, PC software, games, etc).
- Working with the agile methods.
- Working with usability evaluation techniques.
- Geographically located within Kuala Lumpur (Malaysia) or nearby areas.
- Belong to a business with several members, not a hobby firm.

With the resources available, the authors decided that it would not be possible to conduct a survey, not to mention an interview series, across all of Malaysia. Therefore, the authors bounded the study to a well-defined geographical area, namely the region of Kuala Lumpur.

### 3.2 Data collection

By telephone, the authors reached out to possible respondents and determined their willingness to engage in the online questionnaire survey, resulting in a total of 44 participants. The authors employed face-to-face data collecting to guarantee that respondents understood all questions and provided high-quality responses.

To elucidate, clarify, and obtain a deeper understanding of the outcomes of the online questionnaire survey, the authors chose to conduct interviews with the respondents who can answer the questionnaire. All respondents were contacted and 12 agreed to participate in an interview.

The 12 interviews with respondents were conducted on-site. According to [11], semi-structured interviews and interview aids were developed.

In each interview, a respondent and one of the authors served as interviewer and interviewee, respectively. The author posed clarifying questions when necessary. To record the interviews, permission was acquired to audio record them.

### 3.3 Data analysis

There are both open and closed questions on the form. Due to the semi-structured nature of the interviews, the authors had to employ various analytic methodologies. A quantitative analysis is conducted due to the significant amount of data produced from the questionnaire and the objective of gaining an overview of usability evaluation perspectives. Nonetheless, the outcomes of the open-ended questions have to be quantified. Nonetheless, the outcomes of the open-ended questions have to be quantified. This method was conducted using open coding, as outlined in Strauss and Corbin's [12] grounded theory technique. The authors Encoding sentences, we assessed the data from each of the open questions. Then, the code for each sentence was discussed, and a consensus was reached on a single code. In addition, the authors classified the codes into categories, which were then utilized to obtain a simplified summary of the questionnaire answers.

Using the approach of meaning condensation, the recordings from the interview sessions were analyzed on the same or the following day [11]. This method minimizes interpretation and allows empirical data to substitute audio recordings.

### 3.4 Demographic Data

This section discusses the respondents' experiences and practices relevant to software development. The majority of responders were developers (42%), testers (18%), usability engineers (21%), whilst only 4% were project managers and 15% worked in other areas.

The majority of respondents (51%) were employed by their current firm for between one and three years, while (36%) indicated four to ten years. (8%) of respondents reported working for more than ten years. Only (5%) of respondents have been employed at their current firm for less than a year.

In terms of experience in agile software development, 56% of respondents reported being involved between 1 and 3 years, whereas 32% of the respondents reported less than 1 years of involvement. 12% have been involved for more than 3 years of experience.

Finally, all respondents worked either in small or medium companies, of which 68% worked in companies with less than 20 employees. 32% of

the respondents work in medium sized companies, defined as companies with more than 21 employees (see Figure 1 and Figure 2).

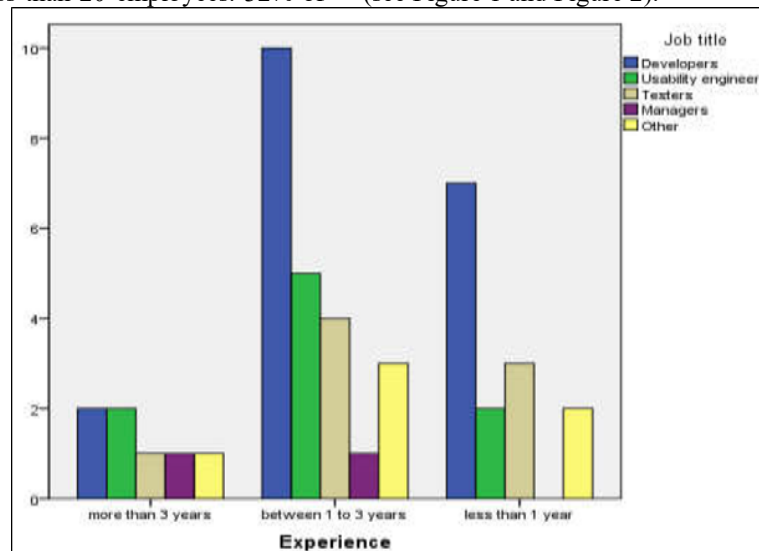


Figure1: Respondents profile

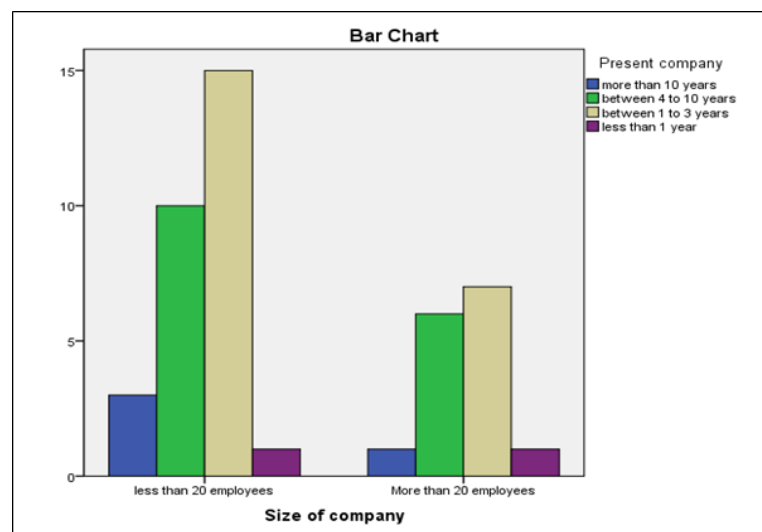


Figure2: Experience in present company

#### 4. RESULTS

In this section, we discuss the obstacles identified by the result of the questionnaire survey and interviews. The obstacles have been classified in order to provide an overview. Figure 3 depicts the categories and the number of statements inside each category.

Activities Management: Figure 3 illustrates that, according to the results of a questionnaire survey, the most commonly reported difficulty was "activities management", which was mentioned by 16 respondents. Several respondents indicated that

they are occasionally unable to determine the most suitable usability evaluation activities. In addition, the short time of iteration could not enabled usability practitioners conduct enough evaluations despite; the usability aspect needs to be addressed extensively during the development stages to achieve usable software.

Another respondent stated, "The usability test results are not always accepted by other people in the organization". This is likely a result of ineffective communication between team members, which negatively affects understanding other

people's opinions. A respondent from a small company also said, "The developers were primarily concerned with the functionality and effectiveness of the programming. For instance, a responder claimed that certain developers believe "...priority-wise it is more important to fix the bugs than it is to participate in the usability test.." or "the developers are more interested in writing beautiful code". These citations demonstrate that developers' minds are preoccupied with coding activities and that there is a fundamental barrier to changing this attitude to one that is more user-centric.

In an agile environment, the development team often neglect usability activities and artifacts because of time constraints and the need to spend much effort [3]. The obtained results in this study proved this issue as our participants unsurprisingly only proposed everyday activities.

Although the benefits of usability have been widely proven, participants still hesitated to imply usability activities. Therefore, the authors think there is a need for developing a management system that supports the timely and relevant selection and application of usability evaluation activities during the development iterations.

In relation to activity management, the interview study uncovered an additional barrier. Some respondents commented that they are unsatisfied with the decisions made for incorporating usability work, since there are no specific criteria or basis for selecting activities of usability evaluation (individual bias). When team members have a shared understanding, disagreements can be handled by achieving consensus, they noted.

**Artefact Management:** The category "artifacts management" was mentioned in the questionnaire survey by 11 respondents. The respondents stated that they do not have the ability to access, understand, or monitor artifacts of both disciplines within the development environment of a given project. In addition, the respondents highlighted that achieving a solid coordination between usability and agile practitioners requires tightly coupled activities and the artifacts of both disciplines. This allows on-going feedback of evaluation while continuing the development.

One respondent from the interview study also stated that the absence of a console repository leads to misunderstanding of the software processes performed by the development team. Another respondent commented on the lack of effective monitoring tools during the development process,

saying "We lack a really good tool for updating progress information during process...".

Furthermore, some respondents mentioned lack of a consoled dashboard used as a guide for development team may lead to fragmentation, loss of focus on overarching objectives of the usability work, and loss of clarity on the links between the evaluation and overall development process.

**Communication Management:** The last category in the questionnaire survey was "communication management", as mentioned by 14 respondents. Some respondents pointed out that they can perform collaboration as agile developers and usability engineers work together in the development stages. Furthermore, although respondents appreciated the involvement of usability specialists, who must be extensively involved during development, there is a need for greater communication between the development team and usability specialists to achieve overall understanding.

However, other respondents reported finding it difficult to deal with multidisciplinary sub-teams of diverse specialists which are the inherent need for diversity in the agile environment. Moreover, some respondents stated that should more improve the communication between team members and increase the flexibility in terms of which work is done by whom.

The interview study also revealed a lack of communication agile team members (usability and developer) as an obstacle. The respondents from the interview study stated that close daily communication with usability engineers helps stakeholders understand how the software will look like before it is implemented and with the little effort they can be used in the usability testing as well. Through give space to conduct activities of usability evaluation with the stakeholders will save work as developer's time is not wasted in wrong or inaccurate development. Another respondent also pointed out the fact that sometimes, limited time or slipped schedules of development process negatively affect the frequent communication between usability and agile practitioners. As a result, producing software which is difficult to learn and use ultimately leads to dissatisfaction among end users.

**General obstacles:** A number of obstacles were identified through the interview study to fall outside the scope of the study. For example, some respondents explained how cost and time

circumstances hindered usability work in general, e.g. “High budget and long-time periods both are necessary for the usability testing’s efficiency...”. Another respondent said, “It is a process which takes an extended amount of time, getting it introduced in the organization...”.

No Problems: Surprisingly, some respondents from both small and medium-sized businesses said that they had not encountered any issues with usability evaluations. This could be due to a lack of motivation to fill out the survey, or because they did not have any difficulties in relating.

The interviews revealed opposition to suggestions, evaluation reports and guidelines produced by usability practitioners, as they are uncommon in agile environments and inessential to the agile process despite the clear advantages of a risk of ignoring the usability approach. However, usability integration cannot be exceeded while applying a method in agile software projects. Though many development organizations describe great success in using agile development processes, none explicitly describe the inclusion of usability activities [5][7].

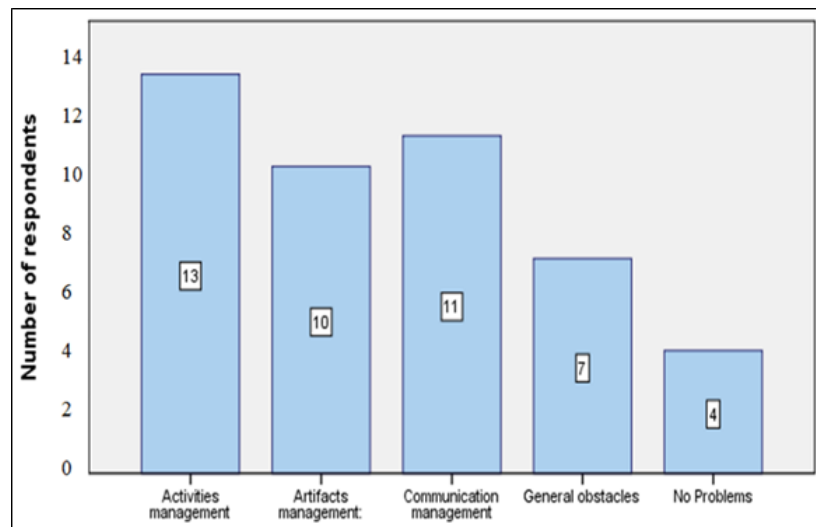


Figure3: Management obstacles found in the questionnaire survey

## 5. MODEL FOR INCORPORATION OF USABILITY EVALUATION

The chosen model includes a set of activities and artifacts derived from meticulously chosen and studied ISO standards. The approach tries to operationalize the standards from usability evaluation and software development processes in order to facilitate their implementation (see Figure 4). This integration consists of three components. First, prospective inclusion points are chosen on the basis of their findings and the overlap between usability evaluation and software development

processes. Second, the linkages between activities and artifacts are evaluated to depict the information flow inside each process. Thirdly, the convergence artifacts of both processes are identified. The model considers a beneficial base for supporting software development team members in nominating appropriate development activities; for documenting and sharing their outcomes; and for establishing domains of competency for roles in software development projects in order to ensure the quality of use.



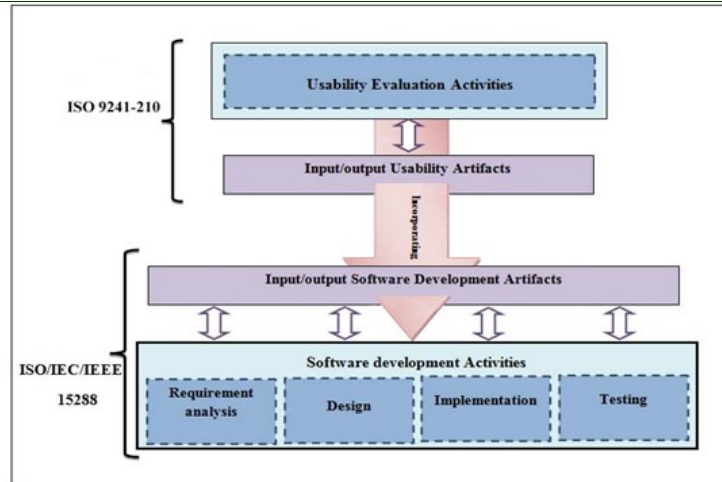


Figure 4: The general incorporation model [13]

## 6. THE APPROACH TO MANAGING ACTIVITIES AND ARTIFACTS

Based on the IUE model, the authors have proposed a unified approach for artifacts of both usability evaluation and agile development process to help development teams in understanding the relationship between them. The essential element of the proposed approach is taking advantage of the convergence points of artifacts as the unifying principle for organizing the diverse information forms. These artifacts are common, relevant outputs or inputs through which all team members' activities take place. The authors' approach demands finding the artifacts that mediate incorporation points and making them into "inhabited spaces", displaying the activities and actions of the communities (usability /developer) who work with them.

### 6.1 The Approach Considerations

The goal here is not simply to help the agile development team members become aware the converging points of artifacts, but also to help them be aware of the artifacts that occur around activities. This focus leads to two main considerations.

**Visualization:** generally, visualization approaches provide visual depictions of information with the aim of allowing users to perceive correlations or patterns. The authors find that each set of development is different, whereby the actual interpretation of activity information rely basically on local factors. Accordingly, this overall approach is a visualization based on the IUE model. Instead

of encoding specific workflows, the authors provide a visual system that allows software development members to explore views of expected artifacts. The motivation behind taking visualization artifacts as a significant approach in an agile environment is that the agile development process is a particularly not simple task, and the needs of individual projects are uniquely dependent in their specific domain and development history. In the light of this unevenness, the authors posit that it will be more effective if provide users with flexibility rather than making assumptions about their needs.

**Concurrent and predictive:** another particularly critical issue is whether this approach is intended primarily for predictive artifacts, or for representing artifacts that are complete or currently in progress. In fact, this approach involves both, in that it should help agile team members to observe and control the development process through iterative steps in which the design of the software project improves gradually.

## 7. THE MANAGEMENT SYSTEM

This section describes the creation of a management system for a unified approach. The proposed system is not an evaluation tool or software development in itself. Rather, it is a supplement to existing evaluation and development tools. One of the key concerns, the system should be flexible enough to manage, monitor, and steer activities of usability evaluation, which has long been considered a crucial aspect of the software development process. However, the emergence of this system is not intended to provide a total solution for smoothly cooperating between software engineering and usability practitioners. Instead, it is

meant to support well-managed activities and artifacts of both fields in addition to existing integration approaches and techniques (see Figure 5).

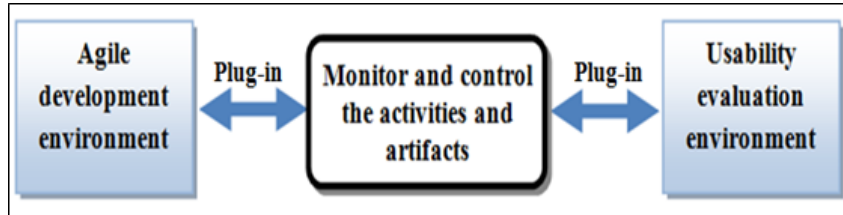


Figure5: Conceptual view of the proposed system

Through the plug-in technique, a system should connect to additional tools such as the Integrated Development Environment (IDE) and usability tools. Three characteristics have driven the proposed system's design. First, it favours dynamic integration into the development process through the online access. Second, it is designed for use by development team members concurrently with development process instead retrospectively for

management analysis. Third, the system's design emphasizes interoperability, so that it must be incorporated into existing development efforts without significant overhead. In short, A system should serve as a guideline to bring the usability and development environments closer together (see Figure 6 ).

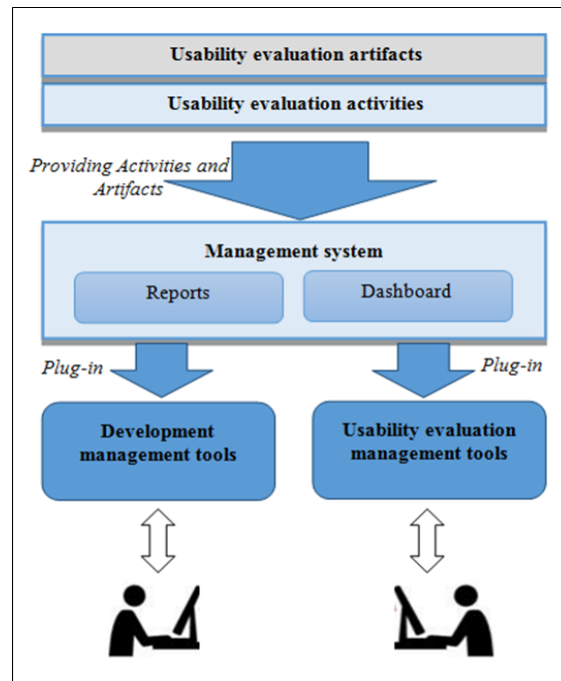


Figure 6: A perspective on software management framework

The proposed system will support two main uses: (1) Monitoring activities and artifacts from both domains, providing agile development teams with an enhanced understanding of the ongoing activities of their colleagues; and (2) Exploring the distribution of activities in time and space, allowing team members to explore the history and expect of particular artifacts.

## 8. CONCLUSION

This paper presented a study to investigate the obstacles to the integration of usability evaluation into an agile environment. The study involved a combined questionnaire survey and interviews with participants from software development companies. Based on the results, a unified approach is proposed



to achieve effective coordination between software developers and usability engineers within an agile environment. The central element of this approach is to exploit the convergence points of artifacts as the unifying principle for organizing many different forms of information. This study is limited to questionnaire and interview techniques; further studies can focus on observation techniques. The observation techniques enable researchers to get information not obtained by verbal information, such as the team's behaviour, feeling and activities. For future work, the authors plan to correlate the findings from different software companies and countries to understand whether the obstacles can be generalized. The authors also plan to design and implement a software management tool based on practitioner case studies to evaluate this approach.

## ACKNOWLEDGEMENTS

The authors would like to acknowledge the Malaysia Minister of Education and Research Management Center at Universiti Malaysia Terengganu (UMT) for funding the grant with the reference code FRGS 59561 to pursue research in this field.

## REFERENCES

- [1] F. A. Salman, A. Bin Deraman, and M. B. A. Jalil, "A Guideline Tool for Ongoing Product Evaluation in Small and Medium-Sized Enterprises," *Int. J. Electr. Comput. Eng.*, vol. 7, no. 6, p. 3632, 2017.
- [2] E.-M. Schön, J. Thomaschewski, and M. J. Escalona, "Agile Requirements Engineering: A systematic literature review," *Comput. Stand. Interfaces*, vol. 49, pp. 79–91, 2017.
- [3] F. A. SALMAN, A. B. I. N. DERAMAN, and M. B. A. JALIL, "THE VALUE OF T-GIUESE-SUPPORTING USABILITY EVALUATION PRACTICES DURING DEVELOPMENT PRODUCT: A CONTROLLED EXPERIMENT.," *J. Theor. Appl. Inf. Technol.*, vol. 95, no. 8, 2017.
- [4] J. C. Lee and D. S. McCrickard, "Towards extreme (ly) usable software: Exploring tensions between usability and agile software development," in *Agile 2007 (AGILE 2007)*, 2007, pp. 59–71.
- [5] B. Jerome and R. Kazman, "Surveying the solitudes: An investigation into the relationships between human computer interaction and software engineering in practice," in *Human-Centered Software Engineering—Integrating Usability in the Software Development Lifecycle*, Springer, 2005, pp. 59–70.
- [6] A. Seffah and A. Andreevskaia, "Empowering software engineers in human-centered design," in *25th International Conference on Software Engineering*, 2003. *Proceedings.*, 2003, pp. 653–658.
- [7] X. Faulkner and F. Culwin, "Enter the usability engineer: integrating HCI and software engineering," *ACM SIGCSE Bull.*, vol. 32, no. 3, pp. 61–64, 2000.
- [8] A. Holzinger, M. Errath, G. Searle, B. Thurnher, and W. Slany, "From extreme programming and usability engineering to extreme usability in software engineering education (XP+ UE/spl rarr/XU)," in *29th Annual International Computer Software and Applications Conference (COMPSAC'05)*, 2005, vol. 2, pp. 169–172.
- [9] M. Singh, "U-SCRUM: An agile methodology for promoting usability," in *Agile 2008 Conference*, 2008, pp. 555–560.
- [10] M. D. C. Tongco, "Purposive sampling as a tool for informant selection," *Ethnobot. Res. Appl.*, vol. 5, pp. 147–158, 2007.
- [11] S. Kvale, *An introduction to the qualitative research interview*, 2nd ed. 1998, Hans Reitzel.
- [12] J. Corbin and A. Strauss, *Basics of qualitative research: Techniques and procedures for developing grounded theory*. Sage publications, 2014.
- [13] F. A. Salman, A. B. Deraman, and M. B. Abdul Jalil, "Towards producing usable software: a model for incorporating usability evaluation activities during product development based on ISO standards," *International Journal of Management and Applied Science (IJMAS)*, vol. 2, pp. 211–215, 2016.