

# A Unified Approach to Support the Coordination of Usability Work in Agile Software Development

Fouad Abdulameer Salman, Aziz Bin Deraman, Masita Binti Abdul Jalil

**Abstract**—Usability evaluation is essential for developing usable software systems, yet its integration within agile software development remains a challenging interdisciplinary endeavour. In this paper, the authors present a study to investigate obstacles of such integration from the management perspective. The study incorporates two methods, namely an online questionnaire survey and a series of interviews with participants that answered the questionnaire. Based on the obtained results, a unified approach is proposed for enabling coordinate the efforts of agile developers and usability engineers to produce usable software systems.

**Keywords**—usability; usability evaluation; software development process; usability management;

## I. INTRODUCTION

THE success of developed usable software systems largely depends on the effectiveness of coordination between software developers and usability engineers [1]. Agile methodologies are trendy interactive software development strategies that lean towards engaging all developers, who not only have a shared understanding of the design but are equally qualified to work on any part of the system [2]. These methodologies may lead to better communication between team members and increased flexibility in terms of who does what work. However, it can be difficult for usability engineers and agile developers to work together, given their differing 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 software developers and usability engineers [3].

Any software development has to deal with two sorts of complexity, namely the complexity of the artifacts being produced or involved e.g. requirement, prototype, code; and the complexity of the activities concerning those artifacts (the distributed software development process). Although the coordination effort will inherently involve both of these sorts of complexity, no studies have yet obtained an overview of the management obstacles that hinder incorporating usability evaluation into the agile software development. This paper

presents a study investigating obstacles to usability evaluation integration from the management perspective. Based on the obtained results, a unified approach is proposed for reconciling usability work with the agile development environment. The contribution of the authors' approach is a set of combination information about the structure of both artifacts and activities.

## II. AGILE SOFTWARE AND USABILITY

There is much research on software engineering and usability, both important in system development, to simplify cooperation between practitioners in these fields. 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 quantity and variety of techniques and methodologies existing in both fields. The differences of aims between SE and HCI practitioners, for example, and their different priorities in software development [4] negatively affect the aforementioned integration. 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 between software developers and HCI practitioners. 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] proposed a U-SCRUM methodology to specifically involve usability evaluation concepts within agile methods. Unlike typical SCRUM,

Fouad Abdulameer Salman, School of Informatics and Applied Mathematics, Universiti Malaysia Terengganu (UMT), 21030 K. Terengganu, Terengganu Malaysia, (Email: fouarese@gmail.com).

Aziz Bin Deraman, School of Informatics and Applied Mathematics, Universiti Malaysia Terengganu (UMT), 21030 K. Terengganu, Terengganu Malaysia, (Email: a.d@umt.edu.my).

Masita Binti Abdul Jalil, School of Informatics and Applied Mathematics, Universiti Malaysia Terengganu (UMT), 21030 K. Terengganu, Terengganu Malaysia, (Email: masita@umt.edu.my).

USCRUM has two product owners. One is focused on usability and the other on more conventional functions. Results indicate that U-SCRUM gives improved usability over SCRUM. Various integration approaches are continually being proposed and developed using different methods and techniques, but how to manage this integration is absent or poorly documented in the literature. Determining how to fill this gap would improve the exchange of common information, making collaboration more successful.

### III. INVESTIGATION THE OBSTACLES TO INCORPORATE USABILITY EVALUATIONS

To study the integration of usability evaluation in context of management and the obstacles against integration, authors conducted a survey involved two stages. The first stage was an online questionnaire survey. The second stage was a series of interviews to obtain more insight into the results from the online questionnaire survey.

#### A. Participants

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

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

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.

#### B. Data collection

By telephone, the authors contacted potential respondents and determined their willingness to participate in the online questionnaire survey for a total of 44 respondents. The authors used face-to-face data collection approach to ensure a clear understanding of all questions and good quality answers from respondents.

To elaborate, clarify, and gain a deeper insight into the results of the online questionnaire survey, the authors decided to interview the respondents who had answered the questionnaire. All respondents were contacted and 12 agreed to participate in an interview.

The 12 interviews were held on-site with the respondents. The interviews were semi-structured and interview guides were designed according to [11].

Each interview involved a respondent and one of the authors participating as an interviewer. The author asked elaborating questions when required. To document the

interviews, consent was obtained to audio record the interview.

#### C. Data analysis

The questionnaire had both open and closed questions. Since the interviews were semi-structured, the authors had to use different analytical approaches. Because of the relatively large amount of data resulting from the questionnaire, and the aim of getting an overview of respondents' opinion of usability evaluation, the authors performed quantitative analysis. However, the results from the open questions had to be quantified. This process was done with open coding, as described in the grounded theory approach by Strauss and Corbin [12]. By putting codes on sentences, the authors analyzed the data from each of the open questions. Then, the code for each sentence was discussed, and a single code was agreed upon. Also, the authors assigned the codes to categories which were then used to get a condensed overview of the results from the questionnaire.

The recordings from the interview sessions were processed the same or following day using the meaning condensation method [11]. This method minimizes interpretation, and allows empirical data to substitute the audio recordings.

#### D. Demographic Data

This section discusses the respondents' experiences and practices relevant software development. Most of the respondents were developers 42%, testers 18%, or usability engineers 21%, while only 4% were project managers and 15% worked in other areas.

For the duration of employment at their present company, most of the respondents were employed between 1 and 3 years 51%, whereas 36% reported 4 to 10 years. 8% of the respondents reported having worked for more than 10 years. Only 5% of respondents were employed for less than a year at their present company.

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 3years 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 Fig. 1 and Fig. 2).

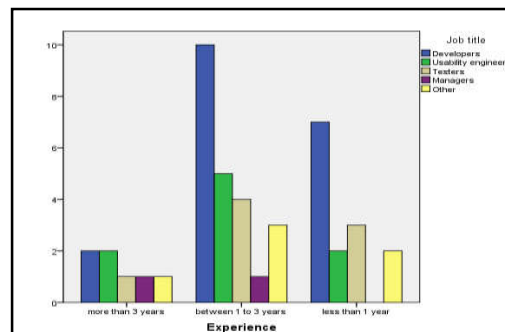


Fig. 1 Respondents profile

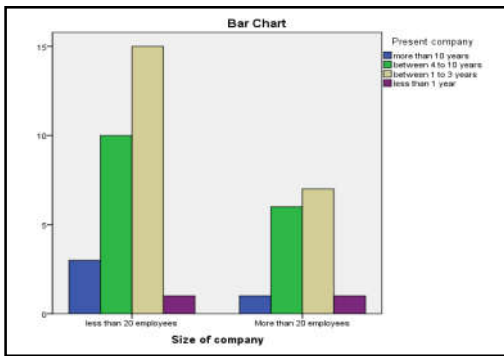


Fig. 2 Experience in present company

#### IV. OBSTACLES TO INCORPORATE USABILITY EVALUATIONS

In this section, we present the obstacles found through the questionnaire survey as well as the interview study. The obstacles have been categorized to provide overview. The categories can be seen in Fig. 3, along with the number of statements for each category.

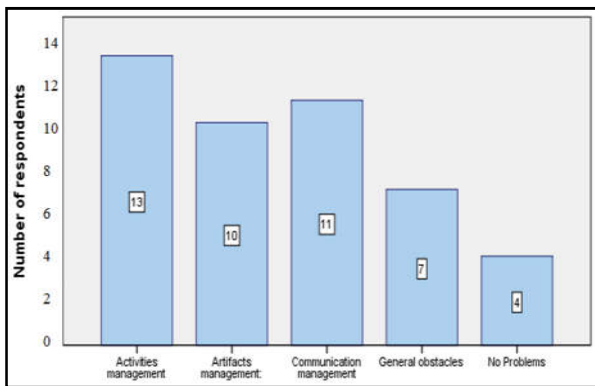


Fig. 3. Management obstacles found in the questionnaire survey

**Activities management:** Fig. 3 shows that the most frequently mentioned obstacles in the questionnaire survey were in the category “activities management”, as mentioned by 16 respondents. Some respondents pointed out that they are sometimes unable to know the appropriate usability evaluation activities. In addition, they could not conduct enough usability activities despite; the usability aspect needs to be addressed extensively during the development stages to ensure that the end product is usable.

Another respondent stated, “The usability test results are not always accepted by other people in the organization”. This is presumably related to the inefficient communication between team members, which negatively affect to understand the other people’s opinions. A respondent from a small company also said, “From a risk assessment we prioritize functionality first” the developers’ main focus and interest was the functionality and efficiency of the code. For example, a respondent stated that some developers have the attitude that “...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 beautiful code”. These citations demonstrate that the developers’ minds are focused on programming activities, and there is a fundamental obstacle to

change this mindset toward a more user-centred one.

In relation to the activities management, the interview study revealed another obstacle. Some respondents commented that they are unsatisfied with the decisions made for incorporating usability activities and methods, since there are not any specific criteria or basis for selecting activities of usability evaluation (individual bias). They also added that conflicts can be resolved by reaching consensus when team members share understanding.

**Artifacts 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 to achieve a solid incorporation between usability and agility requires a continuous process, 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 usability specialists, who must be extensively involved during development, there is a need for greater communication between 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 increased 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 the agile team members. 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, there were some respondents belonging both small and medium companies stated that they had not experienced any problems with usability evaluations. This may be due to a lack of motivation for filling in the questionnaire, or because they did not experience any problems to relate.

## V. MODEL FOR INCORPORATION OF USABILITY EVALUATION

This section proposes the use of the incorporation of usability evaluation (IUE) model [13] as a solid basis for mitigating the management obstacles that hinder interdisciplinary collaboration. The selected model contains a list of activities and artifacts constructed from carefully selected and analyzed ISO standards. The standards are from both usability evaluation and software development processes, which the model aims to operationalize and thus simplify their implementation (see Fig. 4). This incorporation includes three aspects. Firstly, potential incorporation points are mapped based on their end effects and overlap between activities in usability evaluation and software development. Secondly, the dependencies of the activities and artifacts are examined to demonstrate information flow within each process. Thirdly, the convergence artifacts of both processes are identified.

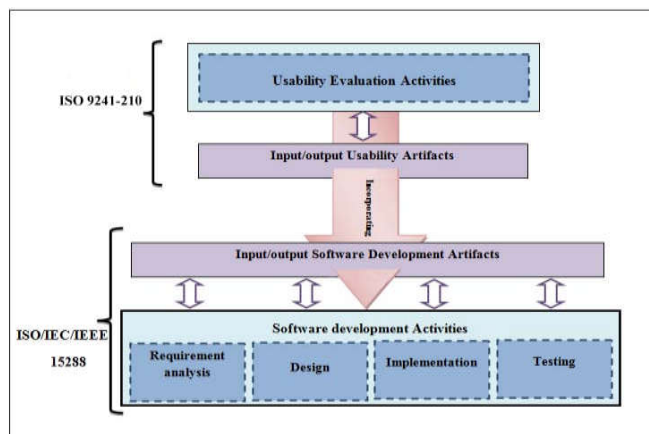


Fig. 4 The general incorporation model [13]

The model considers a beneficial base for supporting software team member in nominating appropriate development activities; for documenting and communicating

their results; and for defining fields of competencies for the roles in software development projects to achieve quality of use.

## VI. THE APPROACH OF 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 understand 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.

### A. 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 setting of development is different, whereby the actual interpretation of activity information rely basically on local factors. Accordingly, their overall approach is visualization-based 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 visualisational artifacts as a significant approach in 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 this will be more effective if providing users with flexibility rather than to make assumptions about their needs.

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

## VII. 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 was that it be flexible enough to



manage, monitor, and steer activities of usability evaluation, which has long been treated as a significant issue in the software development process. However, the emergence of this system is not intended to provide a total solution for incorporating usability issues within the development process. Instead, it is meant to support well-managed activities and artifacts of both fields in addition to existing integration approaches and techniques (see Fig. 5).

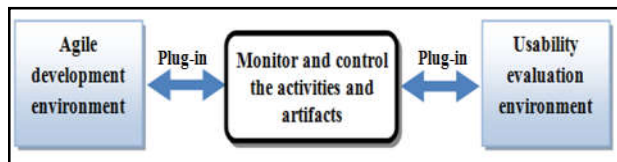


Fig. 5 Conceptual view of the proposed system

A system should connect to other tools such as the Integrated Development Environment (IDE) and usability tools by the plug-in technology. 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 scaffold to bring more closely together the usability and development environments (see Fig. 6).

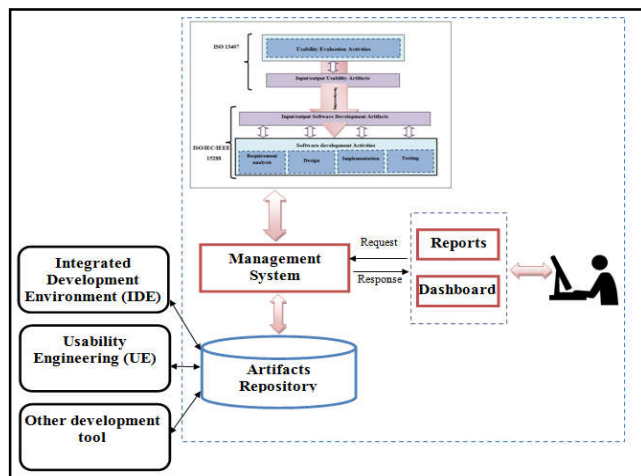


Fig. 6 A view of software management system architecture

The proposed system will support two main uses: (1) Monitoring activities and artifacts of both fields, 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.

## VIII. CONCLUSION

This paper presented a study to investigate the obstacles against integration of usability evaluation into 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 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. Future studies can focus on designing and implementing a software management tool which may be used to evaluate this approach based on practitioner case studies.

## REFERENCES

- [1] F. A. Salman, A. B. Deraman, and M. B. Abdul Jalil, "A guideline tool for ongoing product evaluation in small and medium-sized enterprises," *International Journal of Electrical and Computer Engineering (JECE)*, vol. 6, Dec. 2017, pp. 3632-3642.
- [2] E. M. Schön, J. Thomaschewski, and M. J. Escalona, "Agile requirements engineering: a systematic literature review," *Computer Standards & Interfaces*, vol. 49, Jan. 2017, pp. 79-91.
- [3] F. A. Salman, A. B. Deraman, and M. B. Abdul Jalil, "The value of T-GIUESE-supporting usability evaluation practices during development product: a controlled experiment," *Journal of Theoretical & Applied Information Technology*, vol. 95, Apr. 2017, pp. 1808-1817.
- [4] J. Lee, C. Chong, and D. S. McCrickard, "Towards extreme (ly) usable software: exploring tensions between usability and agile software development," in Proc. IEEE conf. Agile Conf., 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 Netherlands, 2005, pp. 59-70.
- [6] A. Seffah, and A. Andreevskaia, "Empowering software engineers in human-centered design," in Proc. IEEE Computer Society, Conf. of the 25th International on Software eng., 2003, pp. 653-658.
- [7] X. Faulkner, and F. Culwin, "Enter the usability engineer: integrating HCI and software engineering," in Proc. ACM SIGCSE Bull., 2000, vol. 32, No. 3, pp. 61-64.
- [8] A. Holzinger, M. Searle, and B. Thurnher, "From extreme programming and usability engineering to extreme software engineering education," in Proc. In Computer Software and Applications IEEE, Conf. COMPSAC 29th Annual International, 2005, vol. 2, pp. 169-172.
- [9] S. Mona, "U-SCRUM: An agile methodology for promoting usability," in Proc. Agile, AGILE'08, 2008, Conf. IEEE, pp. 555-560.
- [10] M. D. Tongco, "Purposeful sampling as a tool for informant selection," *Ethnobotany Research and Applications*, 2007, vol. 5, pp. 147-158.
- [11] S. Kvale, *An introduction to the qualitative research interview*, 2nd ed. 1998, Hans Reitzel.
- [12] A. Strauss, and J. Corbin, *Basics of qualitative research. Techniques and Procedures for Developing Grounded Theory*, 1998, 2.edition. SAGE Publications.
- [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)*, Dec. 2016, vol. 2, pp. 211-215.
- [14] Corbin, J., & Strauss, A. (2008). *Basics of qualitative research: Techniques and procedures for developing grounded theory*. Thousand Oaks.