

# Managing usability evaluation practices in agile development environments

Aziz Bin Deraman, Fouad Abdulameer Salman

School of Informatics and Applied Mathematics, Universiti Malaysia Terengganu (UMT), Malaysia

---

## Article Info

### Article history:

Received Feb 7, 2018

Revised Oct 18, 2018

Accepted Nov 2, 2018

---

### Keywords:

Agile software development

Usability

Usability evaluation

Usability management

---

## ABSTRACT

Usability evaluation is a core usability activity that minimizes risks and improves product quality. The returns from usability evaluation are undeniable. Neglecting such evaluation at the development stage negatively affects software usability. In this paper, the authors develop a software management tool used to incorporate usability evaluation activities into the agile environment. Using this tool, agile development teams can manage a continuous evaluation process, tightly coupled with the development process, allowing them to develop high quality software products with adequate level of usability. The tool was evaluated through verification, followed by the validation on satisfaction. The evaluation results show that the tool increased software development practitioner satisfaction and is practical for supporting usability work in software projects.

Copyright © 2019 Institute of Advanced Engineering and Science.

All rights reserved.

---

### Corresponding Author:

Fouad Abdulameer Salman,

School of Informatics and Applied Mathematics,

Universiti Malaysia Terengganu (UMT),

Universiti Malaysia, 21030 Kuala Terengganu, Terengganu, Malaysia.

Email: [fouadaug@gmail.com](mailto:fouadaug@gmail.com)

---

## 1. INTRODUCTION

The Poor usability and inefficient end-product design are common issues [1], [2] of failed software products. These causes can be avoided by incorporating usability evaluation activities into the design and development stage of software systems [3]. Using usability evaluations in software development allows for highly significant feedback, making it a critical factor in creating usable and useful software for target audiences [4].

Usability evaluation is applied during the development process to increase product usability, minimize the risks of failure, decrease costs, and increase overall quality [5]. Various priorities and approaches make the incorporation further complicated [6]. Although both fields have similarities, such as a shared focus on delivering value, being iterative in nature and including continuous testing, the differences in how they approach these issues leads to practical difficulties [7]. A key difficulty encountered is the incorporation of activities, artifacts, and methods of the evaluation of existing organizational structures, which typically possess established process models for product development and implementation [8]. A lack of comprehensive strategic approaches to usability evaluation in conventional agile methods and inadequate transfer of knowledge between practice and academia [9] makes it challenging to perform ongoing evaluations. This sometimes leads to its remittance from software development projects [10]. However, despite growing research in both fields, agile development teams often lack a properly-integrated approach that utilizes incorporation during the development practices from end-to-end. In this paper the authors developed a software management tool to fill the gap between the usability evaluation side and the development side. Managing usability evaluation activities while working according to the agile software development approach was suggested [11], [12]. The authors' contribution is connecting both usability evaluation environment and agile environment by a management tool derived from a model for incorporating

usability evaluation into agile environment [13]. The authors think that a more comprehensive basis of incorporation would remove process or organization-specific details that hinder interdisciplinary collaboration and add more value to the incorporation process.

## 2. ENABLING AGILE USABILITY

In existing approaches, many researchers have addressed the issue of integrating usability evaluation into software development, and specifically into the agile development approach. For example, some earlier studies focus on applying several usability evaluation techniques to agile development, while others focus on the benefits a particular technique can bring to agile development. Many approaches have been developed to bridge this gap. Kane stated that none of the main agile development methods explicitly incorporate interaction design practices, and suggests that by incorporating discount usability evaluation techniques as part of agile development, the usability of the software can improve [14]. He stressed that there are still many issues that must be addressed to make the combination effective. It seems that these lightweight techniques can be a good match with agile methods as these techniques do not require any formal usability training and can thereby easily be adapted to work in an agile context. Holzinger presents a development method called Extreme Usability (XU) where all the best usability practices are kept in the planning game of XP, including a think-aloud method on paper mock-ups [15]. Meszaros and Aston argue that some design up front provides better guidance to the development team and provides earlier opportunities for feedback [16]. Paelke and Nebe present using an approach to integrate agile methods with UE and design techniques that offered effective exploration. These techniques are based on mixed reality or the close coupling of real-world objects with computer-generated information and functionality. This has strong potential for the creation of usable user interfaces [17]. Wolkerstorfer presents an approach to combine interaction design techniques (user studies, extreme personas, usability expert evaluations, usability test, and automated usability evaluations) with XP to take advantages of both approaches [18]. They have combined the advantages of XP methodology (on-time delivering, optimized resource investment, short release cycles, working high-quality software, tight customer integration) with the advantages of an interactive design process (usable, accessible, accepted products, end-user integration).

Obendorf proposed an approach using the scenario techniques in XP that preserve the agility of agile development methods [19]. He addressed scenarios as tools for combining XP and usability evaluation and reflects on the experiences they have made adapting it to real projects. However, the integrated approach needs more reflection. Losada integrated UE, evaluation techniques in InterMod and proving its advantages, benefits, and verifiable improvements. This promotes development tailored to user expectations since end users are involved in the beginning of the project and in the continuous usability evaluation [20]. McNeill described techniques of storyboarding as an iterative approach and a low-reliability prototyping in agile that can perk up the software design instead of spending time on code modules. Hence, enhanced versions of these products can develop a complete focus on end-users by adding usability to agile approaches [21]. While various integration approaches are continually being proposed and developed using different methods and techniques, management of such integration is absent or poorly documented in the literature. Determining how to reduce this gap would improve information exchange, making collaboration more successful. In the next section, the authors identify cases in which usability evaluation can be well-managed within the development environment.

## 3. INCORPORATION OF USABILITY EVALUATION CHARACTERISTICS

The main characteristics of the incorporated approach of agile and usability evaluation that authors use are as follows:

- a. Iterative design activities - In many cases, when user-centric techniques are used, the design of the system is refined according to the users' evaluations and this is performed mainly during the design phase. When introducing the agile approach, a design is updated regularly as the product evolves. Combining usability evaluation activities with the agile approach, user evaluation is fostered by performing usability evaluation tasks in each iteration of two to four weeks. The design is then updated according to the evaluation of on-going outcomes that are considered as refactoring tasks.
- b. Roles - Different roles are defined to support software development environments. The agile approach adds roles for better management and development of the project. Combining agile and usability evaluation adds usability evaluation roles, such as the user interface (UI) expert role.
- c. Measures - Taking measurements is a basic activity in software development processes. The agile approach emphasizes this and encourages the tracker role. When combining agile and usability evaluation

- approach, a set of evaluation tools may be built and refined during the process and used iteratively to complement the process and product measures.
- d. Iterative design activities - In many cases, when user-centric techniques are used, the design of the system is refined according to the users' evaluations and this is performed mainly during the design phase. When introducing the agile approach, a design is updated regularly as the product evolves. Combining usability evaluation activities with the agile approach, user evaluation is fostered by performing usability evaluation tasks in each iteration of two to four weeks. The design is then updated according to the evaluation of on-going outcomes that are considered as refactoring tasks.
  - e. Stakeholders (Users, Customers) - Both usability and agile approaches are 'human-centered', in that they value close collaboration with stakeholders. In usability, users are people that actually use the software and that participate in requirements gathered through interviews and field studies. Agile development also values collaboration with users and considers stakeholder participation in the development process to be the most effective means of communicating needs to the developers. Combining agile and usability evaluation allows stakeholders to evaluate software under real conditions.

#### 4. INCORPORATION USABILITY EVALUATION MODEL

The model of incorporation usability evaluation into agile environment [13] has been chosen as a basis for the proposed management tool which supports software team members to nominate appropriate development activities; to document and communicate their results; to define fields of competencies for the roles in software development projects to achieve quality of use. However, the selected model contains a list of activities and artifacts constructed from carefully selected and analyzed ISO standards. The standards form the basis of usability evaluation and software development processes, which the model aims to operationalize and thus simplify their implementation as shown in Figure 1. The incorporation includes three aspects as follows. 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 show information flow within each process. Thirdly, the convergence artifacts of both processes are identified. In the next section, the proposed tool is established as a complete system within its defined scope.

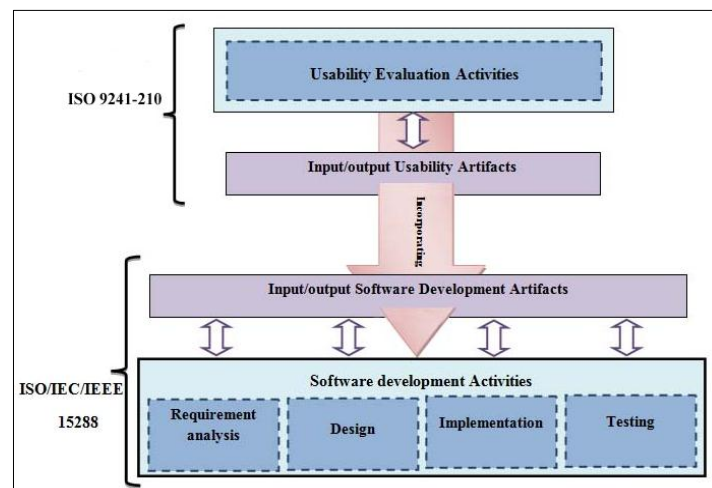


Figure 1. The general incorporation models

#### 5. THE SOFTWARE MANAGEMENT TOOL

Based on the model, a complete management system should be designed to fulfill a critical need of usability evaluation activities, alongside the process of software development, especially in terms of the agile environment. This system attempts to manage, monitor, and steer the usability activities which have long been treated as one of the core issues in the software development process. However, the deployment of this system is not meant to provide a total solution to incorporate usability issues within the development process. It is simply intended to support management activities in addition to the existing incorporation approaches

and techniques. Before proceeding to system implementation, we must first understand the system architecture and environment.

The management system is developed as a software prototype system to implement in a real environment. It is connected to other tools such as the Integrated Development Environment (IDE) and usability tools using plug-in technology. Essentially, it serves as a scaffold to bring together the usability and development environments.

This system was developed in a JavaScript server(JSP). The application of internet WEB and using MySQL to database management also develop the database through the use of the open source program. It was designed to cover four main functionalities: artifact repository, monitor development activities of both fields (usability and software development), the evaluation of the usability evaluation methods and the production of plans. It is assumed that all the software development knowledge managed by the developed system has been extracted from the artifact repository which is created by other tools. Figure 2 contains an overview of the management system architecture.

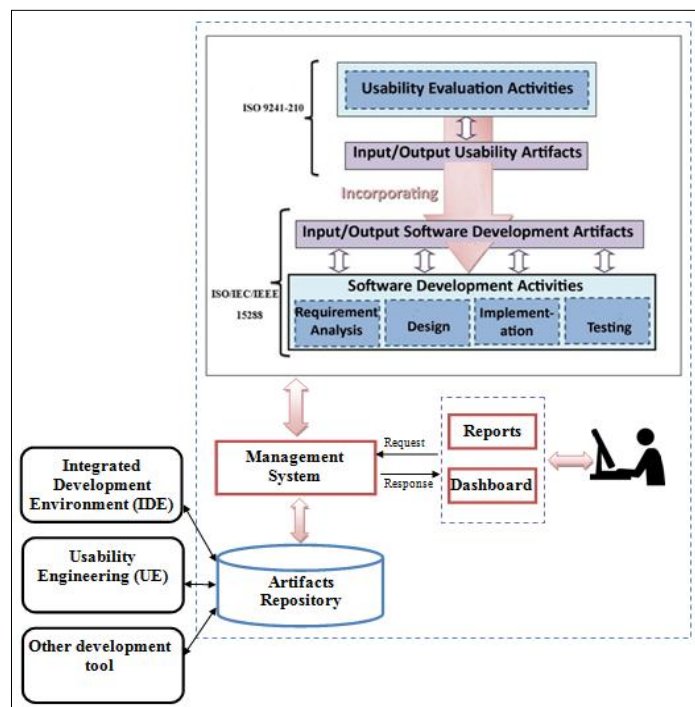


Figure 2. A view of software management system architecture

a. Monitor and control (activities and artifacts)

The objective of monitoring and controlling from this research perspective is to ensure a continuous evaluation process tightly coupled with the development process, thus receiving on-going user feedback while continuing the development. Providing dashboard for displaying stream of the activities and artifacts of both fields strike the balance of these tracks alongside the development process. In this way, the software team can identify the progress of tasks, activities, projects, and other associated information (e.g. Task Name, Start Date, End Date, Percent Complete) via this dashboard.

b. Manage the activities and artifacts in the plan

This encompasses interfaces that cover all phases of development process activities (requirements, design, implementation, and testing) to enable the software development team to set, prioritize, and evaluate usability aspects smoothly during their projects. The lists of these activities show fundamental UE and SE activities and artifacts relevant to the usability evaluation and software development process, with the potential incorporation points being highlighted.

c. Manage the methods of usability evaluation in the plans

The next one is the previous functionality that provides the interface display of all activities of both fields including (requirements, design, implementation, and testing) to enable the software development team to select, suggest, and set proper usability evaluation methods based on certain criteria e.g. Effectiveness,

Dynamicity, Time-cost, Effort-cost, Learnability, Understandability, Results Accuracy, Coverage Area, Attitude, and Satisfaction.

d. Manage artifacts repository

An artifacts repository represents a collection of artifacts and links captured from available resources including documentation, code and other related information. This information needs to be stored into a database of artifact relationship tables to make them readily available for management system implementation.

e. Produce plans

This can be divided into two main tasks, namely suggesting activities for both usability evaluation and development process, and suggesting methods of usability evaluation. This may then be used to produce plan reports consisting of activities, methods and expected artifacts. The authors suggest that the combination of the usability evaluation and agile approaches should be supported by an extension to a contemporary development environment in order to be used in a natural manner. This is elaborated in the next section.

## 6. THE PROJECT

With support from an industry developer, one of the authors developed the management tool described in this paper. The project took three months and was composed of 4 iterations, three of 3 weeks each and one of 4 weeks. Table 1 shows the duration and the main themes of each iteration.

Table 1. The Iterations-duration and Themes

Iter.	Themes	Duration (weeks)
1	Experiments and Roles a. End-to-end plans: defining the activities, methods, expected artifacts, and results view b. Developer role-perspective c. Usability engineer role-perspective d. Other team role-perspective e. Work items can be created, assigned f. The system has one data repository	3
2	Users' interface a. Users' management and permissions b. A user interface to produce the plans c. Client / Server architecture for managing the activities and methods d. Supporting selection method for user evaluation that are derived from certain criteria e. On line help	4
3	Stability a. Testing and Refactoring b. Development refinement	3
4	Observe and User Profiling a. Supporting monitor and control (activities and artifacts) b. Supporting user profiling c. Scale usability evaluation methods	3

### 6.1. Using the management system

The main feature of the management system is its ability to incorporate and manage usability evaluation within the context of an agile environment. Focusing on a specific software project that can be defined by different kinds of required activities, then define suitable methods to apply these activities. That is, the development team chooses the set of activities according to the nature of software project and then selects appropriate methods based on selection criteria (the selection will be from agile perspective) to perform these activities and tasks as shown in Figure 3 and Figure 4.

### 6.2. Related environment

Due to the proposed management tool intended to build a bridge between usability evaluation and development environment; it has to deal with several types of development and evaluation tools. For example, one type of those tools is IDE. It is a software development platform that provides comprehensive facilities to computer programmers for software development. The boundary between an integrated development environment and other parts of the broader software development environment is not well-defined. Sometimes a version control system, or various tools to simplify the construction of a graphical user interface (GUI), are integrated. Our management tool is one of those applications integrated to this environment. A number of UE tools have been developed to facilitate usability practitioners' tasks such as: a) Usability data Capture: collecting usability data, such as task completion time, errors, guideline violations,

and subjective ratings; b) Usability problems analysis: interpreting usability data to identify problems in the interface; and c) Critique (report): suggesting solutions or improvements to mitigate problems.

Similarly, the management tool is connected with those tools. Bringing together these two environments using the management tool could well improve the communication between usability engineers and software developers. This communication link is essential to the development of a system as it is deemed efficient, also that it is pleasing and useful to the end user. In addition, effective communication saves time by reducing the inconsistencies in understanding thereby reducing the number of areas for debate. By using the plug-in feature, the management tool will consolidate both fields and promote the precision and completeness as a result of the development process.

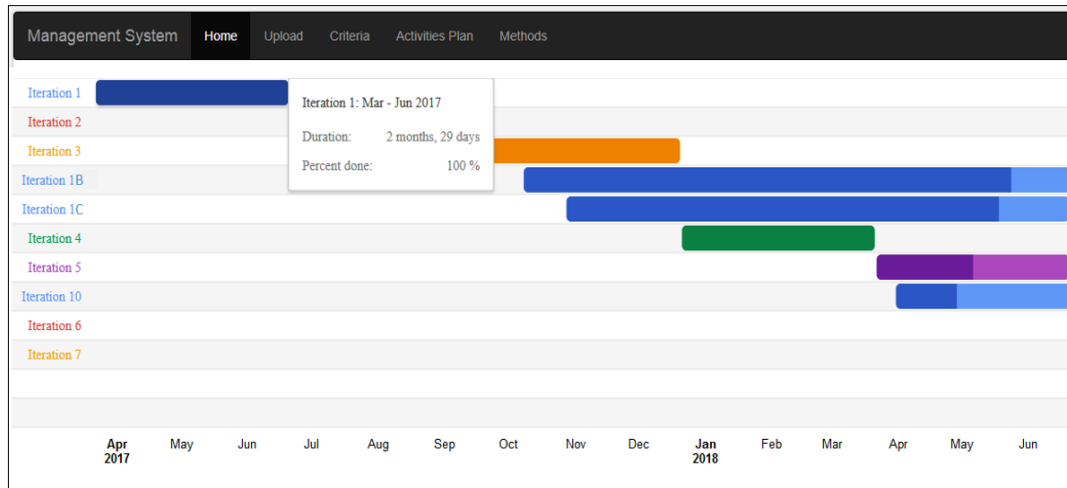


Figure 3. Excerpt of the progress of the development process page

Figure 4. Excerpt of the planning page for suggesting methods of usability evaluation

## 7. EVALUATING THE MANAGEMENT TOOL

The process of evaluating the management tool was carried out by performing the verification and validation stages. The verification stage was performed by experts from the industry field, who are the domain experts. Validation was carried out by the software development practitioners through a case study. These stages are discussed further in the next sub-sections.

### 7.1. Verification through expert reviews

Expert reviews are accepted as a way to detect and remove defects [22], [23]. Therefore, this study adapted this technique for the verification process. Verification is intended to ensure that the developed tool conforms to its specification [24] and that all required components are present in the right quantity [25].

The two main issues that need to be verified in this system are lists of activities, artifacts, dependencies and relationships; and the selection usability methods criteria. Therefore, the potential experts who had knowledge and experience in software quality focus on usability, and those in SE, pays attention to the agile processes identified. They were chosen by following the characteristics of experts as suggested by [26], [27].

### 7.2. Define the experts

Verification was provided by twelve experts after they were contacted through e-mails. Five of them were willing to verify the usability aspect and three of them were willing to verify the incorporation approach from the SE perspective, which is considered sufficient [28], [29]. Consequently, appointments were made and face to face meetings were held. Basically, the following activities were conducted during the review sessions:

- a. Authors presented the overview of the study and the steps for incorporating the usability evaluation activities in the agile environment to the experts.
- b. The experts would then focus on reviewing the usability and software development activities and artifacts plus selection usability methods criteria of the management tool.
- c. The experts reviewed the steps of implementing the management tool. They had the opportunity to ask questions of the authors for further clarification.
- d. The experts gave feedback by completing a verification form.
- e. Researcher then updated the management tool as suggested by the experts.

### 7.3. Results of verification

Lists of the activities, artifacts, and dependencies and relationships, including the tool itself, were verified based on their comprehensiveness, understandability, sufficient, accurateness, and organization. These criteria are adapted from previous studies [30], [31]. The descriptions of these criteria are as listed in Table 2. The experts provided their feedback by completing a checklist form.

Table 2. Descriptions of Verification Criteria

Criteria	Descriptions
Comprehensiveness	This criterion shows that the required activities, artifacts, the dependencies and relationships, and the selection usability methods criteria are included in the tool.
Understandability	The criterion suggests that the activities, artifacts, the dependencies and relationships, and the selection usability methods criteria are decomposed clearly and unambiguously.
Sufficient	The criterion indicates that the activities, artifacts, the dependencies and relationships, and the selection usability methods criteria are decomposed to achieve sufficient incorporation.
Accurateness	The criterion indicates that the activities, artifacts, the dependencies and relationships, and the selection usability methods criteria are adequately decomposed to achieve accurate incorporation.
Organization	The criterion denotes that the activities, artifacts, the dependencies and relationships, and the selection usability methods criteria are organized well.

Most experts agreed that the activities, artifacts, the dependencies and relationships, and the selection usability methods criteria of the tool are comprehensive, understandable, sufficient, accurate, and organized well. However, they had some comments on the understandability and organization of the activities, artifacts and their relationship. For example, Experts A, B, and D suggested that the lists of activities and artifacts should be supported with source reference. Meanwhile, Experts C, E, and H found that there were some activity and artifact descriptions that were not very clear. Expert F concluded that certain texts for the activities and artifacts were misrepresented, such as usability evaluation activity number 3 “feedback”. Experts D and E suggested that the selection criteria should be organized based on three scales rather than five. However, all suggestions of the expert were taken into consideration. Hence, the activities, artifacts, the dependencies, and the selection usability methods criteria were updated accordingly as suggested. Based on the comments from the experts, the system was reorganized and updated.

## 8. VALIDATION BY A CASE STUDY

We present a case study in which development teams in academia worked with agile approach in developing their software projects while applying our management tool. 36 participants were asked to implement the tool by applying it within their final projects that they have been working on. Based on the supporting management tool, they selected appropriate usability evaluation activities and methods within their project plans. An assessment form [32], [33] was provided for them to assign the score for each item after using the tool. In the final project, team members were asked to grade their satisfaction for each: perceived usefulness, perceived ease of use, balance the competing activities, effective of monitor and control, and ability to produce usable results, between 1 to 5, as well as open-ended questions seeking their opinions. The participants' feedback are shown below:

**Perceived usefulness:** The participants pointed out that the management tool is useful for their working environment. This tool allows them to know the appropriate usability evaluation activities. In addition, it helps them to conduct enough usability activities thus leading to produce usable software.

**Perceived ease of use:** According to the responses of the participants, the management tool was perceived as easy to be used because it has a series of well-defined activities, artifacts, selection methods attributes, and dependencies and relationships. The participants found out that the tool is internally consistent, mainly because the components complement each other. In particular, it starts with adding the activities, then adding the methods in the proposed plan. The process is then continued properly. In addition, the proposed plan is prepared by a group (including the development and usability aspects) for obtaining the consensus among the team members on the number and type activities and methods of usability.

**Balance the competing activities:** The participants agreed that the tool is appropriate for enabling balanced development decisions that incorporate the oftentimes conflicting needs of the developers and usability engineers. The participants added that conflicts can be resolved by reaching the consensus when deciding on the criteria and bases in a team. During the planning, the development team will have to make decisions about changes to the design proposed by the usability engineers based on usability evaluation results. A usability engineer will summarize results from usability evaluations and present changes to the design, with each change being assigned a relative priority level.

**Effective monitoring and control:** The participants were satisfied with the ability to monitor and control the usability evaluation activities within the development environment of the project. Furthermore, participants highlighted that the tool helps to steer a continuous process, tightly coupled with the development process, thus receiving on-going user feedback while continuing the development. Thus, it supports an agile environment because it allows the interaction and active team, which is the core value of agile. Other than that, they added that it also enables the derivation of relevant development tasks for the forthcoming iterations for further improvements in the developing product.

**Ability to produce usable results:** Participants pointed out that by having the management tool, the team could understand and perform the best usability practices in the agile software process. Furthermore, the tool is able to produce usable results. From the list of activities and their relationships as well as attributes for selection methods, the participants were able to identify which activities or methods would be better suited in the right place and right time.

Furthermore, descriptive statistics were taken to measure the level of each item. A 5-point Likert scale was used, with 5 for "Very agree" and 1 for "Very disagree". Any answer with an average of above 3.0 was considered to be good as this indicated the level of the respondents' agreement to those statements representing the tested items. Results from the descriptive analysis as shown in Table 3.

Table 3. Descriptive Statistics: Mean Score of Satisfaction Criteria

Satisfaction Criteria	Mean
Perceived usefulness	3.72
Perceived ease of use	3.42
Balance the competing activities	3.63
Effective of monitor and control	3.89
Ability to produce usable results	3.44

From the descriptive statistics for the item in the Table 3, the item with the highest mean (3.89) is "Ability to produce usable results", and the item with the lowest mean (3.42) is "Perceived usefulness". However, all of these items have a mean of above 3.0 and thus belong on the same level. Essentially, based on the feedback from the participants, the management tool gained satisfaction from them and deemed practical to be implemented in the software projects.



## 9. CONCLUSION

This paper describes a software management tool specifically developed to support the incorporation usability evaluation activities within the software development process. This tool serves as scaffolding that helps the usability and agile practitioners to closely collaborate and understand each other. It contributes to activities management, allowing ongoing user and usability expert feedback during development. Verification and validation of the tool are conducted in this paper a) the verification of the tool through expert reviews followed by the validation on satisfaction. The results of evaluations show that the tool satisfied development teams was practical for use in the software projects. Future work should note that although the management tool targets toward agile based development approaches, they are also applicable for other development approaches that come under the umbrella of iterative and incremental development methodology. Therefore, applying this approach to other development environments will be an interesting research opportunity.

## REFERENCES

- [1] Salman FA., Deraman A., Jalil MA., Fahmy S., "On Implementing Usability Evaluation Activities within Agile Environment: A Plan Based-Process," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 9(3-3), pp. 95-9, Oct 2017.
- [2] Bornoe N., Stage J., "Usability Engineering in the Wild: How Do Practitioners Integrate Usability Engineering in Software Development?," In *International Conference on Human-Centred Software Engineering*, pp. 199-216. Springer, Berlin, Heidelberg, Sep 2014.
- [3] da Silva TS, Silveira MS, Maurer F., "Usability Evaluation Practices Within Agile Development," *InSystem Sciences (HICSS), 2015 48th Hawaii International Conference*, IEEE, pp. 5133-5142, Jan 2015.
- [4] Larusdottir M., Bjarnadottir E., Gulliksen J., "The Focus On Usability In Testing Practices In Industry," *Human-Computer Interaction*, pp. 98-109, 2010.
- [5] Bevan N., "Extending Quality In Use To Provide A Framework For Usability Measurement," *Human Centered Design*, pp. 13-22, 2009.
- [6] McInerney P., Maurer F., "UCD in Agile Projects: Dream Team Or Odd Couple?," *Interactions*, vol. 12(6), pp. 19-23, Nov 2005.
- [7] Bak JO., Nguyen K., Risgaard P., Stage J., "Obstacles to Usability Evaluation in Practice: a Survey of Software Development Organizations," In *Proceedings of the 5th Nordic Conference on Human-Computer Interaction: Building Bridges*, ACM, pp. 23-32, Oct 2008.
- [8] Bornoe N., Stage J., "Usability Engineering in the Wild: How Do Practitioners Integrate Usability Engineering in Software Development?," In *International Conference on Human-Centred Software Engineering*, pp. 199-216, Springer, Berlin, Heidelberg, Sep 2014.
- [9] Ardito C., Buono P., Caivano D., Costabile MF., Lanzilotti R., "Investigating And Promoting UX Practice In Industry: An Experimental Study," *International Journal of Human-Computer Studies*, vol. 72(6), pp. 542-51, Jun 2014.
- [10] Salman FA., Deraman AB., Jalil MB., "The Value Of T-GUIESE-Supporting Usability Evaluation Practices During Development Product: A Controlled Experiment," *Journal of Theoretical & Applied Information Technology*, vol. 95(8), Apr 2017.
- [11] Lee JC., Judge TK., McCrickard DS., "Evaluating Extreme Scenario-Based Design In A Distributed Agile Team," In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*, ACM, pp. 863-877, May 2011.
- [12] Humayoun SR., "Incorporating Usability Evaluation In Software Development Environments," *KI-Künstliche Intelligenz*, vol. 26(2), pp. 197-200, May 2012.
- [13] Salman FA., Deraman AB., Jalil MB., "A Guideline Tool for Ongoing Product Evaluation in Small and Medium-Sized Enterprises," *International Journal of Electrical and Computer Engineering (IJECE)*, vol.7(6), pp.3632-42, Dec 2017.
- [14] Kane D., "Finding A Place For Discount Usability Engineering In Agile Development: Throwing Down The Gauntlet," In *Agile Development Conference, ADC 2003. Proceedings of the 2003 Jun 25. IEEE*, (pp. 40-46), 2003.
- [15] Holzinger A., "Usability Engineering Methods For Software Developers," *Communications of the ACM*, vol.48(1), pp. 71-4, Jan 2005.
- [16] Meszaros G., Aston J., "Adding Usability Testing To An Agile Project," In *Agile Conference, 2006, IEEE*, pp. 6-pp, Jul 2006.
- [17] Paelke V., Nebe K., "Integrating Agile Methods For Mixed Reality Design Space Exploration," In *Proceedings of the 7th ACM Conference on Designing interactive systems*, ACM, pp. 240-249, Feb 2008.
- [18] Wolkerstorfer P., Tscheligi M., Sefelin R., Milchrahm H., Hussain Z., Lechner M., Shahzad S., "Probing An Agile Usability Process," In *CHI'08 Extended Abstracts on Human Factors in Computing Systems*, ACM, pp. 2151-2158, Apr 2008.
- [19] Obendorf H., Finck M., "Scenario-Based Usability Engineering Techniques In Agile Development Processes," In *CHI'08 Extended Abstracts on Human Factors in Computing Systems*, ACM, pp. 2159-2166, Apr 2008.

- [20] Losada B., Urretavizcaya M., López-Gil JM., Fernández-Castro I., "Combining Intermod Agile Methodology With Usability Engineering In A Mobile Application Development," In *Proceedings of the 13th International Conference on Interacción Persona-Ordenador*, ACM, (p. 39), Oct 2012.
- [21] McNeill M., *User centered design in agile application development*, ThoughtWorks Ltd., 2000.
- [22] Hodges NJ., Huys R., Starkes JL., "Methodological Review And Evaluation Of Research In Expert Performance In Sport," *Handbook of Sport Psychology, Third Edition*, 159-83, 2007.
- [23] Baker SE., Edwards R., Doidge M., "How Many Qualitative Interviews Is Enough?," *Expert voices and early career reflections on sampling and cases in qualitative research*.
- [24] Martin D., Rooksby J., Rouncefield M., Sommerville I., "'Good' Organisational Reasons For 'bad' software Testing: An Ethnographic Study Of Testing In A Small Software Company," In *Proceedings of The 29th International Conference on Software Engineering, IEEE Computer Society*, pp. 602-611, May 2007.
- [25] Chemuturi M., "Mastering Software Quality Assurance: Best Practices, Tools And Techniques For Software Developers," *J. Ross Publishing*, Sep 2010.
- [26] Hallowell MR., Gambatese JA., "Population And Initial Validation Of A Formal Model For Construction Safety Risk Management," *Journal of Construction Engineering and Management*, vol.136(9), pp. 981-90, Feb 2010.
- [27] Murugesan S., Gangadharan GR., "Harnessing green IT: Principles and practices," *Wiley Publishing*, Oct 2012.
- [28] Shneiderman B., Plaisant C., Cohen MS., Jacobs S., Elmquist N., Diakopoulos N., "Designing The User Interface: Strategies For Effective Human-Computer Interaction," *Pearson*, 2016.
- [29] Nielsen J., Molich R., "Heuristic Evaluation of User Interfaces," In *Proceedings of the SIGCHI conference on Human factors in computing systems, ACM*, pp. 249-256, Mar 1990.
- [30] Al-Tarawneh FH., "A Framework For Cots Software Evaluation And Selection For COTS Mismatches Handling And Non-Functional Requirements," Doctoral dissertation, Universiti Utara Malaysia.
- [31] Kunda D., "STACE: Social Technical Approach To COTS Software Evaluation," *Component-Based Software Quality*, pp. 64-84, 2003.
- [32] Surendro K., Olivia O., "Academic Cloud ERP Quality Assessment Model," *International Journal of Electrical and Computer Engineering*, vol. 6(3), pp. 1038, Jun 2016.
- [33] Roy S., Pattnaik PK., Mall R., "A Cognitive Approach For Evaluating The Usability Of Storage As A Service In Cloud Computing Environment," *International Journal of Electrical and Computer Engineering*, vol. 6(2), pp. 759, Apr 2016.