On Implementing Usability Evaluation Activities within Agile Environment: A Plan Based-Process

Fouad Abdulameer Salman, Aziz Deraman, Masita Abdul Jalil, and Syahrul Fahmy
School of Informatics and Applied Mathematics,
Universiti Malaysia Terengganu, 21030 Kuala Nerus, Terengganu Malaysia.
fouarese@gmail.com

Abstract—**In fast-paced software development cycles, poor governance may cause a loss of focus on the overarching objectives and implementation of usability work. We propose a conceptual framework that would assist software development teams in understanding and implementing usability evaluations in the described agile environments. The framework provides a common language that conveys the "big picture" of software development, details the roles needed for usability, and facilitates the analysis, discussion and communication of responsibilities and decision structures. The framework is supported by a visual mapping guideline that eases discussion and simplifies the process of incorporating usability work in companies.**

Index Terms—**Agile Development Process; Agile Planning; Usability Evaluation; Usability Management.**

I. INTRODUCTION

Linking usability evaluation to software development is essential knowledge [1] for software companies that want to adopt an effective end user-oriented approach in its development decision-making. While initiatives for integrating the perspectives of usability and development have spawned a rich body of literature [2,3,4,5,6], there is ample room for extension. Agile methods, and most other iterative and incremental development methods focus on building releasable software in short, fixed time periods. Additionally, agile methods adopt the viewpoint of single development teams dedicated to individual projects. Thus, usability issues involving multi-project teams have been unaddressed [7].

In practice, poor governance in fast-paced, multiple team development situations may lead to fragmentation, loss of focus on overarching objectives of the usability work, and loss of clarity on the link between the evaluation and overall company strategy [8]. However, usability is not a mandatory part of agile processes despite being perceived as essential when applying a method in software projects [9,10]. This is evidenced by the lack of explicitly described inclusion of usability evaluation activities in agile development processes despite the successes reported by many development organizations [11].

We believe for usability to be prioritized in agile environments, it must first be an essential component in agile software development processes and considered at every stage of iteration and realization of developmental tasks. To achieve this, the planning process has to address usability in a manner that enables successful incorporation into software engineering. Addressing usability in the planning stage prevents it being sidelined throughout the development stages

and thus avert the production of faulty software. In this respect, we propose a conceptual framework supported by a mapping guideline that could help development teams incorporate usability evaluation activates in all stages of agile software development process.

II. AGILE SOFTWARE DEVELOPMENT AND USABILITY

There is much research on software engineering and usability, both important in system development, to simplify cooperation between developers in these fields. According to Lee and McCrickard [12], the main tension between usability and software development approaches stems from differing aims and motivations of Software Engineering (SE) and Human Computer Interaction (HCI) practitioners, which is further complicated by the quantity and variety of techniques and methodologies existing in both fields.

As a partial solution, Seffah and Andreevskaia [13] proposed educating software engineers on certain usability concepts with an economic educational framework. Conversely, Faulkner and Culwin [14] propose the adoption of human-computer interaction (HCI) principles to guide the development of computer systems. Evidently, these approaches do not emphasize cooperation as draw heavily on concepts from either software engineering or usability engineering, as the strategies tilt more towards the strategies in either field. This is disadvantageous to the problem of incorporating usability evaluation approaches within agile software development, which is increasingly acknowledged and addressed [15]. In fact, the processes in both fields have many similar foundational concepts, such as iterative development and being user-focused.

Holzinger et al [16] present a powerful developmental method - Extreme Usability (XU), which combines usability evaluations with the agile method such that all the best practices of usability evaluations are kept in the XP process during game-planning. Singh [5] proposed a U-SCRUM methodology to specifically involve usability evaluation concepts within agile methods. Unlike typical SCRUM, U-SCRUM has two product owners where one is focused on usability and the other on more conventional functions. Results indicate that U-SCRUM gives improved usability over SCRUM. However, a combined approach is difficult because the nature of agile methods (inherently incremental and iterative) does not support the comprehensive overview of the framework which is often crucial towards coordinating the development process with usability principles and values.

Many leading software companies attempt to integrate usability evaluation in agile development, yet the field lacks defined tactics for integrating sound usability evaluation

practices in organizations transitioning into the agile process. Adaptation of usability evaluation techniques and methods is based on the experience of usability professionals, many of which require time and resources that cannot be afforded by agile processes [17]. Additionally, agile processes have no guidelines for the development, or maintenance of usability adaptations [18], and application of usability principles and practices in agile processes seems lacking. It seems that customers or users are often overwhelmed with system functionality concerns instead of usability issues [19]. Factoring in usability evaluations early in the planning process alongside other requirement features could help to avoid rework and wastage due to readjustments, and improve end-user experience.

III. ENABLING AGILE USABILITY

Agile teams recognize the priority and importance of usability evaluation activities. However, dictation of usability evaluation methods and techniques may be insufficient information for agile teams to implement basic usability features correctly and comprehensively. Effective combination of these two practices requires striking a balance between the continuity of the evaluation and the fast paced, flexible agile development methods.

Our approach recognizes that usability issues need to be fixed before (or with) software development, i.e. usability and software engineering as concurrent and coordinated processes in a single development framework. In this sense, we propose a conceptual framework supported by mapping guideline derived from ISO standard [20,21], which aims to provide a common language for communicating the overarching principles ("big picture") of software development, and enable the analysis and communication of usability needed roles and responsibilities. Through providing a basis for development teams to discuss, coordinate and implement their processes, the incorporation aspects for both processes can be identified without bottlenecking the agile process.

IV. FRAMEWORK AND TOOL SUPPORT

This section describes the conceptual framework for linking the usability issues within agile software development planning. A considerable number of studies focused on agile planning [22,23,24,25] have been taken into account. The levels of the conceptual framework (see Figure 1) discussed in this paper are: Business unit, Product and service, Development portfolio, Project, and Iteration.

A. Backlogs and their Items

Central to the framework are the backlogs from the Product, Project and Iteration level and their relationships to backlog items from Regular and Usability. We have chosen the terms Regular and Usability backlog items instead of Requirements to indicate that not everything requiring attention from developers are software requirements. According to Jarno [26], Products, Projects and Iterations have their own backlogs, which are essentially lists of prioritized items that need to be done. Backlog items are anything requiring attention from developers: new features, change requests, bug fixes, service requests and so on. In our framework, we display regular backlog items side to side with usability backlog items within Product and Project backlogs
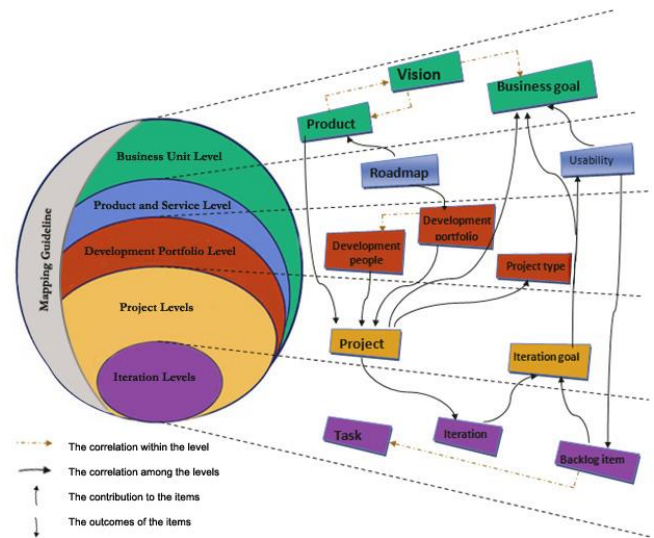
to avoid unnecessary hierarchy.

Figure 1: Linking usability issues within agile development planning

B. Business Unit Level

The Business unit level addresses the competition strategy used by the management of individual business units in a particular industry or product/market segment [27]. "Product" is the generic term used for a company's offering, which could be a software or service that the company is developing, either for commercial or internal usage. Products should contribute to a "Vision", where "Vision" describes the "grand plan" for one or more Products that is conveyed as one or more Business goals. Our approach proposes involves usability as concept within the overall strategy of a business unit, described in terms of Visions and Business goals.

C. Product and Service Level

The Product and Service level is responsible for product and release planning. Product planning is about creating and updating backlog items for a specific product/service proposition to meet Business goals. The plan should be formulated with less emphasis on resource constraints, preferring the following perspective i.e. "If this product had all the resources we wanted, what would be achieved?" In our framework, release planning means describing the future development steps of a product/service through road mapping [28], in which the currently defined Business goals, regular and usability backlog items are mapped to future development Projects.

D. Development Portfolio Level

Portfolio management is a key governance process that links business to development. Hard realities are considered at the Development portfolio level, resulting in the evaluation, selection and prioritization of projects; the acceleration, de-prioritization, or killing of existing projects; and the re-distribution of resources within active projects based on business priorities and constraints such as resource or technical dependencies [29]. In our framework, the supportive mapping guideline is suitable for use in the immediate future as it is synthesized to match available resources and the most important Business goals as closely as possible.

This leads to the release plan, which consists of (1) the set of ongoing usability activities that require attention from the

Development staff, i.e., the product development and/or technical personnel available to the business unit [29], and (2) the assignment of the usability practitioners to these activities.

E. Project and Iteration Levels

Project management deals with the planning and monitoring of individual Projects. The tasks covered are, detailed planning to match project contribution to Business goals, planning initial goals and high level content of project Iterations [25].

Iteration management aims for product development in stages, whereby working intermediate versions with partial functionality of the final release are produced to get feedback in tandem with the development process [25]. Furthermore, it aims to include usability evaluations aspects as a series of reasonably stable activities. Prior to the Iteration, usability backlog items pending implementation are described in greater detail as one or more Iteration backlog items. The goals for Iteration are tangible, Business goal related objectives that summarize, or ideally, encompass the listed individual Backlog items and beyond. The implementation of individual Backlog items in daily work can be listed as Tasks.

V. MAPPING GUIDELINE

This section describes the mapping guideline as a tool derived from incorporation model to guide potential incorporation points (activities, and artifacts) between usability evaluation and software development process. For simplicity and space constraints, only the excerpts relevant to the tool and model are shown and discussed here.

A. Model for Incorporation of Usability Evaluation

Here we describe the model [30] for incorporation of usability evaluation in the software development process, based on 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 Figure 2).
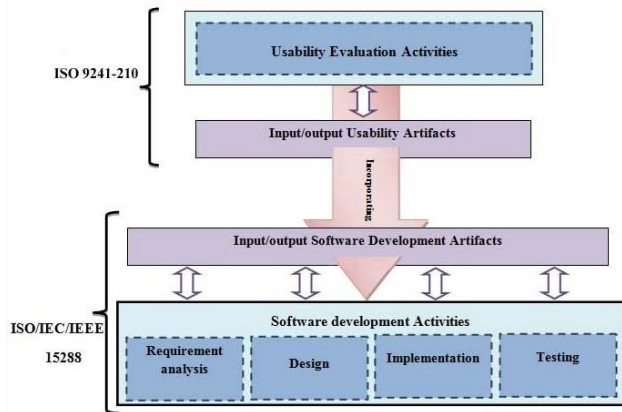

Figure 2: Incorporating usability evaluation into software development process activities based on ISO standards

The incorporation was implemented through three aspects. Firstly, potential incorporation points were mapped based on their end effects and overlap between activities in usability evaluation and software development. Secondly, the dependencies of the activities and artifacts were examined to

show information flow within each process. Thirdly, the convergence artifacts of both processes were identified and shown with a detailed discussion of each aspect as follows.

1) Mapping Potential Incorporation Points

Results from the analysis of usability activities were adapted to software engineering concepts and terminology to correlate with processes in usability evaluation software development. Wherever possible, the basis used to define activities within the software development process is SWEBOK [31]. Relevant activities refer to those interlinked in the abovementioned fields, which were mapped to potential incorporation points based on similar end effects and overlap.

2) Dependencies of Activities and Artifacts

Sufficient interlinking between usability and development activities requires consideration of information flow and artifacts. Therefore, a concrete basis of artifacts was extracted from the chosen standards for further investigation. The dependencies of those activities and artifacts were examined to show the information flow between the activities of each process. Each process contains defined input and output artifacts for clear presentation of production and consumption.

Figures 3 and Figure 4 show an excerpt of the represented and distributed artifacts within activities of each process, numbered according to their position in the origin list and the type of artifacts (Input, I or Output, O).

Figure 3: The excerpt of information exchange of usability evaluation artifacts within activities

Figure 4: The excerpt of information exchange of software development artifacts within activities

3) Convergence Artifacts in Both Disciplines

The identification and display of convergence artifacts in both processes is crucial as it hints at prospective

incorporation points. Artifacts which converge in the same software development process activities were listed based on interlinked usability and development activities. Artifact linkage to the software development activities was presented as input (I) and output (O) and not restricted to a single direction to accurately portray ongoing changes during product development. For instance, "results of the evaluation" are represented frequently as output and input in different phases of development.

B. Visualization of Mapping Guideline

The conventional KJ method [32] was used to create the mapping guideline tool for incorporation, which visually constructs the appropriate structure and relationships of the activities and artifacts. The itinerated activities and artifacts pending delivery were grouped and organized by similarity (see Figure 5). Similarity was evaluated by considering the phases in software development activities, such as requirements, design, implementation, and testing, which were presented separately. After several stages of grouping the activities and artifacts, the final groupings were obtained and spatially spread and arranged on a large sheet of paper. Finally, the correlations, convergence points, and dependencies between the processes were added to arrive at the proposed tool. The tool is divided into three phases, i.e: potential incorporation points, dependencies of activities and artifacts, and convergence points of artifacts for both disciplines.

Arrows were used to highlight the incorporation between activities and artifacts. The following are the typical relationship symbols used:

1) Activities
Interdependence: One item overlaps or affects another (depends on arrow direction).

2) Artifacts
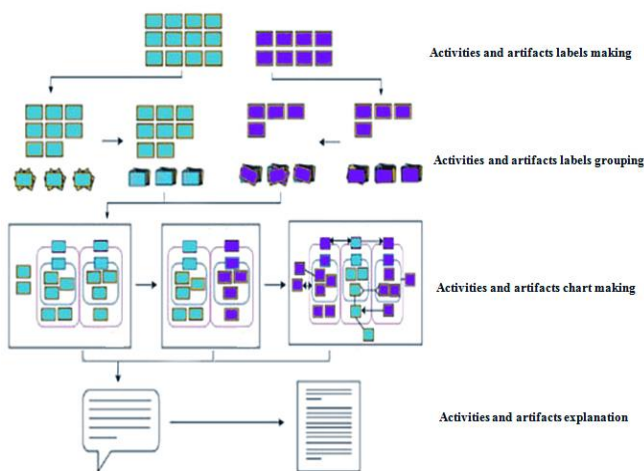Output or Input: The information exchange of artifacts (depends on arrow color)

Figure 5: Steps of constructing structure and relationships of the activities and artifacts (the mapping guideline tool)

Consequently, the tool can support easy comparison and discussion of processes within organizations, thus helping to identify incorporation aspects between usability evaluation and software development process. An excerpt of the tool is shown in Figures 6 and Figure 7.
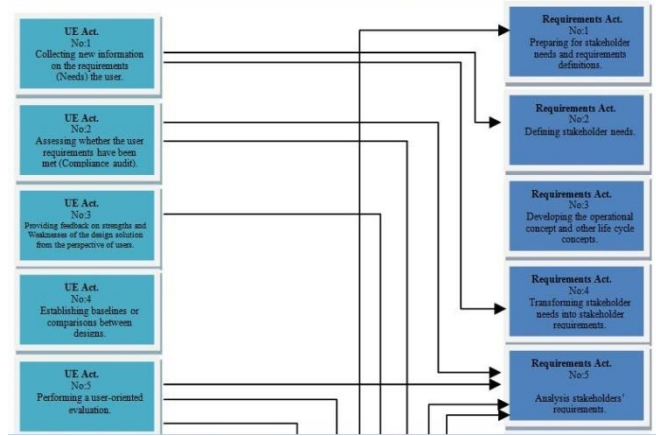

Figure 6: Excerpt of incorporating usability evaluation into software development activities (requirements phase)
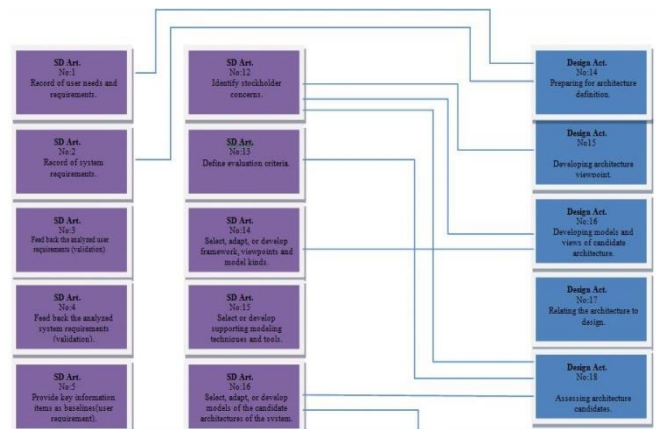

Figure 7: Excerpt of information exchange between software development process artifacts and activities

VI. SUMMARY AND OUTLOOK

This paper presented a conceptual framework that shows the overall linking of usability issues within agile software development planning and an ISO standard derived mapping guideline that supports the framework. Our initial experiences of using the conceptual framework in companies were encouraging. We were able to identify missing responsibilities, decision making structures, and poorly defined roles through discussions with the key personnel in two companies on the framework and concepts. Furthermore, we have also been able to propose tangible improvement suggestions which the companies have acted on. Finally, according to the personnel's input, the framework showed improvement and is partially validated. For future work, we will seek further validations based on practitioner case studies.

REFERENCES

[1] A. Fernandez, E. Insfran, and S. Abrahão, "Usability evaluation methods for the web: A systematic mapping study," *Information and Software Technology*, vol. 53, no. 8, pp. 789-817, 2011.

[2] J. Ferreira, J. Noble, and R. Biddle, "Up-front interaction design in agile development," in *Proc. Extreme Programming and Agile*

Processes in Software Engineering, Springer Berlin Heidelberg, 2007, pp. 9-16.

[3] J. Lee, D. McCrickard, and K. T. Stevens, "Examining the foundations of agile usability with extreme scenario-based design," in *Proc. IEEE Agile Conference*, 2009, pp. 3-10.

[4] M. McNeill, *User Centered Design in Agile Application Development*. Thought Works Ltd, 2000.

[5] S. Mona, "U-SCRUM: An agile methodology for promoting usability," in *Proc. Agile, AGILE' 08. Conf. IEEE*, 2008, pp. 555-560.

[6] S. Chamberlain, H. Sharp, and N. Maiden, "Towards a framework for integrating agile development and user centred design", in *Proc. Extreme Programming and Agile Processes in Software Engineering*, Springer Berlin Heidelberg, 2006, pp. 143-153.

[7] M. Düchting, D. Zimmermann, and K. Nebe, "Incorporating user centered requirement engineering into agile software development," in *Human-computer interaction. Interaction design and usability*, J. A. Jacko, Ed. Berlin, Heidelberg: Springer, 2007, pp.58-67.

[8] G. Meszaros, and J. Aston, "Adding usability testing to an agile project," in *Proc. IEEE. Agile Conference*, 2006.

[9] N. Bornoe, and J. Stage, "Usability engineering in the wild: How do practitioners integrate usability engineering in software development ?," in *Proc. Human-Centred Software Engineering*, 2014, pp. 199-216.

[10] J. Bak, K. Nguyen, P. Risgaard, and J. Stage, "Obstacles to usability evaluation in practice: A survey of software development organizations," in *Proc. 5th Nordic conference on Human-computer interaction: Building bridges*, 2008, pp. 23-32.

[11] O. Sohaib, and K. Khan, "Integrating usability engineering and agile software development: A literature review," in *Proc. Int. Conf. Computer design and applications (ICCDA)*, 2010, pp. 22-32.

[12] J. Lee, and D. S.McCrickard, "Towards extreme (ly) usable software: Exploring tensions between usability and agile software development," in *Proc. IEEE conf. Agile Conference (AGILE)*, 2007, pp. 59-71.

[13] A. Seffah, and A. Andreevskaia, "Empowering software engineers in human-centered design," in *Proc. IEEE Computer Society, Conf. of the 25th International on Software engineering*, 2003, pp. 653-658.

[14] X. Faulkner, and F. Culwin, "Enter the usability engineer: Integrating HCI and software engineering," *ACM SIGCSE Bulletin*, vol. 32, no. 3, pp. 61-64.

[15] S. Blomkvist, *User-Centred Design and Agile Development of IT Systems*. Doctoral dissertation, Uppsala universitet, .

[16] A. Holzinger, G. Searle, B. Thurnher, and W. Slany, "From extreme programming and usability engineering to extreme usability in software engineering education," in *9th Annual International Computer Software and Applications Conference (COMPSAC'05)*, 2005, vol. 2, pp. 169-172.

[17] J. Gil, M. Lopez, M. U. Loinaz, B. Losada, and I. F. Castro, "Field vs. laboratory usability evaluations: A study on a context dependent mobile application developed with an agile methodology," *IEEE Latin America Transactions*, vol. 14, Nno. 1, pp. 339-348, 2016.

[18] D. Silva, T. Silva, M. S. Silveira, and F. Maurer, "Usability evaluation practices within agile development," in *Proc. 48th Hawaii International Conf. on System Sciences (HICSS)*, 2015, pp. 5133-5142.

[19] W. F. Ahmad, S. M. Butt, and L. Rahim, "Usability evaluation of the agile software process," in *Proc. Springer International Publishing, Conf. International Visual Informatics*, 2013, pp. 640-651.

[20] ISO 9241-210: *Human-centered Design Processes for Interactive Systems*. ISO, Genf, 2010.

[21] ISO/IEC/IEEE 15288: *Systems and Software Engineering - System Life Cycle Processes*, Genf, 2015.

[22] K. Rautiainen, and C. Lassenius, "An Experience in integrating strategic product planning and agile software development practices," in *Proc. ACM-IEEE International Symposium on Empirical Software Engineering*, Rome, Italy, 2003, pp. 28-37.

[23] J. Vähäniitty, "A tentative framework for connecting long-term business and product planning with iterative & incremental software product development," in *Proc. of the 7th International Workshop on Economic-Driven Software Engineering Research (EDSER-7)*, St. Louis, USA, 2005.

[24] J. Vanhanen, and I. Sulonen, "Improving the interface between business and product development using agile practices and the cycles of control framework," in *Proc. Agile Development Conference, Salt Lake City*, Utah, USA, 2003, pp. 71-80.

[25] K. Rautiainen, "*Cycles of Control: A Temporal Pacing Framework for Software Product Development Management*," Licentiate Thesis. Helsinki University of Technology, 2005.

[26] J. Vähäniitty, and K. Rautiainen, "Towards a conceptual framework and tool support for linking long-term product and business planning with agile software development," in *Proc. ACM, the 1st International Workshop on Software Development Governance*, 2008, pp. 25-28.

[27] M. Hitt, and I. Hoskisson, *Strategic Management: Competitiveness and Globalization: Concepts*. West Publishing Company, 2011.

[28] L. Lehtola, and M. Vähäniitty, "Strengthening the link from business decisions to requirements engineering: Long-term product planning in software product companies," in *Proc. of the IEEE, Conf. 15th IEEE International Requirements Engineering*, New Delhi, India, 2007, pp. 153-162.

[29] J. Vähäniitty, "Do small software companies need portfolio management, too?," in *Proc. EIASM, Conf. the 13th International Product Development Management*, Milan, Italy, 2006, pp. 1471-1486.

[30] F. A. Salman, A. Deraman, and M. A. Jalil, "Towards promoting usable software: Incorporating usability evaluation into software development environments based on ISO standards," in *Proc. world research library, Melbourne, Australia*, 2016, pp.1-5.

[31] *IEEE Software Engineering Coordinating Committee. Guide to the Software Engineering Body of Knowledge -Trial Version 1.00*. IEEE Computer Society, Los Alamitos, California, May 2001.

[32] R. Scupin, "The KJ method: A technique for analyzing data derived from Japanese ethnology," *Human Organization*, vol. 56, no.2, pp.233-237.