

# A FRAMEWORK FOR INTEGRATING USABILITY PRACTICES INTO SMALL-SIZED SOFTWARE DEVELOPMENT ORGANIZATIONS

Lulwa Alyahyan<sup>1</sup>, Khalid Alnafjan<sup>2</sup>, and Hamza Aldabbas<sup>3</sup>

<sup>1</sup>King Abdulaziz City for Science and Technology, Riyadh, KSA

<sup>2</sup>Software Engineering Department, King Saud University, Riyadh, KSA

<sup>3</sup>Prince Abdullah bin Ghazi Faculty of Information and Technology, Al-Balqa' Applied University, Salt- Jordan

## ABSTRACT

*Usability now appears to be a highly important attribute for software quality; it is a critical factor that needs to be considered by every software-development organization when developing software to improve customer satisfaction and increase competition in the market. There exists a lack of a reference model or framework for small-sized software-development organizations to indicate which usability practices should be implemented, and where in the system-development life cycle they need to be considered. We offer developers who have the objective of integrating usability practices into their development life cycle a framework that characterizes 10 selected user-centered design (UCD) methods in relation to five relevant criteria based on some ISO factors that have an effect on the selection of methods (ISO/TR16982). The selection of the methods for inclusion in the framework responds to these organizations' needs; and we selected basic methods that are recommended, cost-effective, simple to plan and apply, and easy to learn by developers; and which can be applied when time, resources, skills, and expertise are limited. We favor methods that are generally applicable across a wide range of development environments. The selected methods are organized in the framework according to the stages in the development process where they might be applied. The only requirement for the existing development life cycle is that it to be based on an iterative approach.*

## KEYWORDS

*Integration, framework, usability, UCD, small-sized software development organization, small- and medium-sized enterprises (SMEs)*

## 1. INTRODUCTION

Usability and user-centered design methods are recognized as important quality aspects in the software-development life cycle when looking at today's industry; However, the integration of usability engineering (UE) and software engineering (SE) still remains a challenge in practice [1]. Different human computer interaction (HCI) engineering methods have been proposed for developing highly usable software, but the integration of these methods into software-development life cycles is seldom realized in industrial software settings [2]. There is a gap in usability knowledge between research and industry practice [3]. Many techniques have been proposed in the literature that provide usability practitioners and software developers with knowledge on how to integrate usability practices into existing organizational work practices; however, they still lack empirical research that focuses on introducing usability knowledge in small-sized organizations [3]. Aikio [3] indicated that usability integration is limited to large

organizations; therefore, many large organizations recognize usability as an important quality aspect in the software-development process, and employ usability engineering (UE) methods and standards. On the other hand, there is a lack of knowledge and experience regarding UE methods and standards in small- and medium-sized enterprises (SMEs) [1]. Current studies show that the adoption of UE methods in SMEs is up to 28% in a sample of 2,000 participating companies, while 53% stated that they do not use or rarely use a method. Furthermore, only 15% of the companies indicated that they use one of the usability standards [4]. Aikio [3] recommends the development of organizational strategies or approaches for exporting and adopting UCD knowledge in small organizations.

A number of benefits are related to the successful implementation of usability in organizations. According to a survey conducted by the Xerox corporation [5], usability provides important benefits in terms of quality, product, cost, and customer satisfaction. It improves productivity through fewer code revisions and more efficient design, reduces development times and costs, reduces training costs, improves user productivity, and increases customer satisfaction. Recently, software developers in SMEs have become aware of the importance of usability issues, but they face difficulties with respect to integrating usability practices into the development process. Furthermore, usability methods are still unknown to common developers in SMEs [1]. In this paper, we offer developers a framework for usability integration in small software-development organizations. The framework characterizes 10 selected UCD methods in relation to five relevant criteria (General Applicability, User Involvement, Limited Resources, Limited Time, Limited Skills, and Expertise). The selection of the methods for inclusion in the framework responds to these organizations' needs; and we selected basic, recommended, cost-effective methods that are simple to plan and apply; easy to learn by developers; and can be applied when time, resources, skills, and expertise are limited. The framework offers a flexible tool for supporting developers in small organizations to integrate UCD practices into the development process. It is based on general software-development processes that can be applied easily by small software-development organizations regardless of their development processes.

This paper is organized into six sections. The first places our research into context by briefly outlining issues related to usability and usability integration in software-development organizations. Section 2 describes usability in small-sized software-development organizations. Section 3 reviews many approaches for usability integration. Section 4 explains the selected UCD approaches and methods. Section 5 delineates the creation of our proposed framework and details the usage of the framework, and Section 6 presents conclusions and future directions.

## **2. USABILITY AND SMALL SOFTWARE DEVELOPMENT ORGANIZATIONS**

The European Union commission [6] defines small organizations as enterprises that have fewer than 50 employees, and usually the total sheet for their annual balance cannot be more than 10 million EUR. Small software organizations have the following strengths: Employees perform tasks in a cross-functional pattern, and small organizations use resources efficiently because each employee is expected to be actively involved in the organization's activities at all times [7]. On the other hand, small organizations suffer from limited resources relating to time, money, and people [7]; and lack of skills, knowledge, and experience regarding UE methods and standards [1]. Aikio [3] indicated that usability methods are unknown, underused, and inaccessible to common developers and small- and medium-sized software-development teams. However, in order to consider usability in software-development teams, there are four ways to do so, which are listed below [8]:

- Revert to third-party specialized companies in usability engineering.
- Involve an external consultant expert in UCD.
- Create a separate usability group or department.
- Provide training to some members of the development team, who can then function as usability specialists.

Depending on the size of the company, project characteristics, and time/cost constraints, one of these four approaches may be appropriate over another. For small organizations, we prefer that developers on the development team integrate usability into their development life cycle and ignore other choices that might be expensive. In addition, usually small-sized software organizations develop small-sized projects. For such projects, software-development teams can usually avoid the direct involvement of usability experts or practitioners due to the availability of methods that can be applied by non-usability experts such as developers on development teams [8]. Furthermore, small organizations utilize resources (employee) efficiently. Therefore, it may be appropriate for developers in small organizations to act as developers and usability practitioners.

### **3. RELATED WORK**

There is much research on usability integration in organizations; however, there is limited research on usability integration in small organizations. A case study by [9] described how usability activities were introduced into a small organization. A department was established for conducting usability activities and incorporated several usability activities into the development process. However, that case created solutions only for a particular software-development organization and was not general enough to be applied to other organizations. Ferre [10] indicated that one of the reasons why usability techniques are not regularly used in software development is the lack of integration with software engineering (SE) concepts, terminologies, and processes. Ferre tried to approach the integration of usability activities and techniques in a general software-development process by adapting usability concepts to SE terminology; a grouping of 51 selected usability techniques are packaged as increments at different places in a software-development process according to moments of application in the development time. This approach is proposed for developers who use these increments in their software-development process. In 2005, frameworks [11] and patterns [12] were developed for usability integration or UCD integration. They complemented each other, but their approach in representing usability knowledge was different; frameworks provide criteria for evaluating the effectiveness and suitability of a given UCD method, whereas patterns provide suggestions and practical UCD knowledge for UCD practitioners while integrating UCD into organizations [3]. Although patterns and frameworks are used for UCD integration, they do not clearly specify how their approach suits organizations of different sizes or different software-development approaches. Aikio [3] applied Battle's patterns [12] in a case project where an external UCD consultant provided a small-sized software company with knowledge on how to improve the usability of one of their web-based products. A combination of patterns B and C was used, where pattern B recommended external consultants conducting usability testing or heuristic evaluations for existing products, and pattern C recommended UCD during the early phases of development. Therefore, four usability tests based on the think-aloud method were conducted for the product, and a usability test report was delivered to the developers. The personas method was introduced to company stakeholders in a collaborative workshop session. The approach supported the organization in current and future usability work. It was, however, conducted by UCD practitioners and was not based on collaboration between UCD practitioners and software engineers or developers.

There are other approaches to integration based on the level of standards. However, according to limited human resources and smaller iterations in a process, the adoption of standards in SME hardly exists [1]. The famous international standard (ISO 13407) proposed a framework for the integration of usability at all stages of a development process. This standard is replaced by (ISO 9241-210), which provides requirements and recommendations for human-centered design principles and activities throughout the life cycle of computer-based interactive systems. First, integration approaches on the level of standards already exist in the technical report ISO/TR 16982, which addresses the integration of UE and SE and supports the selection of UE methods. However, the report only defines 12 method categories; it does not, however, focus on selecting a suitable method [13]. Fischer et al. [14] analysed corresponding UE and SE standards to introduce a model that consists of activities and artifacts highlighting dependencies, similarities, and possible points for integration. A year later, Fischer [1] proposed an approach that focused on the collection of standards to integrate UE and SE. However, in practice, the UE standards often seem very different to common developers. This is because they are formulated using different terminology, notations, and language [15].

## **4. UCD APPROACHES AND METHODS SELECTION**

There are a number of UCD approaches and a huge number of HCI methods in the literature (95) for a developer to consider when integrating usability practices into the development process. We have considered the recommended methods on TRUMP [16], UsabilityNet [17, 18] and Discount Usability Engineering [19] approaches. These three approaches seem to be more appropriate for the integration purpose, as they offer cost-effective methods that can be applied when time, resources, and expertise are limited. Below is a description of the three approaches.

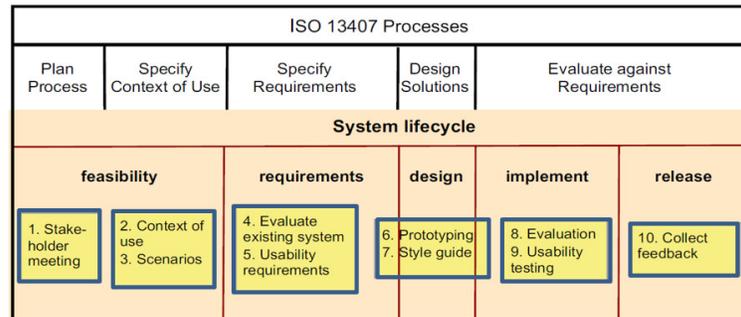
### **4.1. Cost-Effective User-Centered Design**

Serco Usability Services worked on the TRUMP project in conjunction with two organizations in the UK and Israel. The project aims to improve the usability capability of the development processes in two organizations [17]. The TRUMP project trialled the use of user-centered design methods based on ISO 13407 in these organizations. The project had successfully raised the usability maturity of two organizations. Furthermore, to help other organizations achieve similar results, the project recommended 10 simple user-centered methods based on the common experience of these trials. These methods are:

- Generally applicable across a wide range of development environments.
- Simple to plan and apply, and easy for developers to learn.
- Able to improve the usability of products and systems.
- Cost-effective; the methods' costs were extremely low, and the results were obvious in the very short term.

Figure 1 shows how each of the 10 recommended methods relates to the system life-cycle stages and ISO 13407 processes.

Figure 1: TRUMP methods



### 4.2. UsabilityNet Methods

UsabilityNet [17, 18] is a project funded by the European Union to provide resources for usability practitioners. UsabilityNet partners reviewed an extensive range of methods based on their experience in EC and commercial projects. They selected 35 methods that had a track record of cost-effective application in a commercial environment. These were mapped into general system development life cycles using the same stages of the development process as in TRUMP, except that testing and measuring was identified as a separate stage at the end of implementation. They represent the methods in a table, with a column for each development-process stage, as shown in Figure 2.

Figure 2: UsabilityNet basic methods highlighted [20]

	<input checked="" type="checkbox"/> limited time/resources	<input type="checkbox"/> No direct access to users	<input checked="" type="checkbox"/> Limited skills/expertise		
Planning & Feasibility	Requirements	Design	Implementation	Test & Measure	Post Release
Getting started	User Surveys	Design guidelines	Style guides	Diagnostic evaluation	Post release testing
Stakeholder meeting	Interviews	Paper prototyping	Rapid prototyping	Performance testing	Subjective assessment
Analyse context	Contextual inquiry	Heuristic evaluation		Subjective evaluation	User surveys
ISO 13407	User Observation	Parallel design		Heuristic evaluation	Remote evaluation
Planning	Context	Storyboarding		Critical Incidence Technique	
Competitor Analysis	Focus Groups	Evaluate prototype		Pleasure	
	Brainstorming	Wizard of Oz			
	Evaluating existing systems	Interface design patterns			
	Card Sorting				
	Affinity diagramming				
	Scenarios of use				
	Task Analysis				
	Requirements meeting				

### 4.3. Discount Usability Engineering

Nielsen [19] proposed Discount Usability Engineering, which is a usability engineering approach that can be applied when there is minimal time and a limited budget. It is based on the use of the following four techniques:

- *User and task observation:* Developers can visit and observe users in their locations to study their characteristics and analyze the tasks they perform.
- *Prototyping through scenarios:* Scenarios describe a single-interaction session where the user interacts with the system. Scenarios can be implemented as paper mock-ups or in simple

prototyping environments. Scenarios can be used during the design of a user interface to support in understanding the way users interact with the system.

- *Simplified thinking aloud*: Thinking aloud is a usability testing method evaluating the usability of the system by involving users in using the system for a given set of tasks while being asked to “think out loud” to verbalize their thoughts, while observers watch and take notes. The major difference between simplified and traditional thinking aloud is that tests are conducted quickly and cheaply, as opposed to more expensive and traditional thinking-aloud approaches.
- *Heuristic evaluation*: Nielsen suggests the use of a small number of heuristics (just 10 rules) that developers or interface designers follow. This method was developed by Nielsen and Molich [21], and it can be applied by nonexperts to detect usability problems. The method is effective, inexpensive, easy, and fast to perform, and can result in major improvements to user interfaces, It has the further benefit that it is less costly than methods that involve the user in usability testing [22].

## 5. INTEGRATING USABILITY PRACTICES INTO THE SOFTWARE PROCESS: A FRAMEWORK

We have created the framework based on three stages. The first stage involves method selections for the proposed framework, which is described in Section 5.1. The second stage is the selection of the basic methods for our proposed framework, which is described in Section 5.2. The last stage is methods characterization, which is described in Section 5.3.

### 5.1. Methods Selection

The primary purpose of this stage is to select appropriate usability methods for inclusion in the framework. We have selected the methods based on the following three steps. **First**, two filters are applied in UsabilityNet approach; time and/or resources are limited, and skills and/or expertise are limited [17]. These filters are most important in the case of small development organizations. According to Bevan [17], with the filters application, the results involve 13 basic recommended methods that can be applied when time or resources are limited, and skills or expertise are limited. Figure 2 shows these methods highlighted.

**Second**, based on the result of the first step, the basic methods recommended by UsabilityNet have been compared with those recommended by the TRUMP approach to determine the recommended methods by both approaches, as shown in Table 1. Development-process phases that are used in the UsabilityNet approach are considered. Since time, resources, skills, and expertise are limited with the filters application, the UsabilityNet approach eliminates machine prototypes and usability evaluations of machine prototypes, as this method is recommended by the TRUMP approach but is not recommended by the UsabilityNet approach. According to Bevan [18], machine prototyping can be replaced with paper prototyping. In addition, the getting-started method and affinity diagramming are eliminated, as these methods are recommended by UsabilityNet but are not recommended by the TRUMP approach. Analysis of the context of use is ignored in the planning and feasibility phase and considered only in the requirements phase; the method (collect feedback from users) that is recommended by the Trump approach is matched by the subjective assessment that is recommended by the UsabilityNet approach. Furthermore, the diagnostic evaluation that is recommended by the UsabilityNet approach is matched by the usability testing method that is recommended by the Trump approach.

From the above comparison, methods that are recommended by both approaches are selected to be applied by small organizations to integrate usability into the development process. These methods are suitable for small organizations in which there is a need for simple methods that are

easy to learn and apply by developers; and can be used when time, resources, skills, and expertise are limited.

**Third**, the methods recommended by the Discount Usability Engineering approach are compared with the recommended methods by both approaches (UsabilityNet and TRUMP), and we have found that there is a need to consider heuristic evaluation method that is recommended by Discount Usability Engineering in our proposed framework. According to the comparison, we conclude the following:

- User and task observation that is recommended by Discount Usability Engineering is not recommended by both approaches (UsabilityNet and TRUMP). However, the context-of-use analysis method that is recommended by both UsabilityNet and TRUMP is considered to be an alternative method for user and task observation, especially when users participate in the context-of-use analysis [18]. Therefore, this method is not considered in our framework.
- Prototyping through scenarios that are recommended by Discount Usability Engineering is already recommended by both approaches (UsabilityNet and TRUMP).
- Simplified thinking aloud, which is recommended by Discount Usability Engineering, matches the diagnostic evaluation that is recommended by UsabilityNet, and matches the usability testing that is recommended by TRUMP.
- Heuristic evaluation, which is recommended by Discount Usability Engineering, is not recommended by our approach. However, it is considered by UsabilityNet, TRUMP, and Discount Usability Engineering as a complement to user-based testing. Heuristic evaluation, which is recommended by Discount Usability Engineering, needs to be added to our proposed framework.

From the three steps, we have selected 10 methods, as shown in Table 2. These methods are generally applicable across a wide range of development environments. We recommend that they be applied in small-sized organizations to integrate usability practices into their development processes. These methods are suitable for small organizations where there is a need for simple methods that can be applied when time, resources, skills, and expertise are limited. Table 2 shows the 10 selected methods for the proposed framework, which are mapped into the general system-development life cycle. In this table, we have mapped “evaluate existing system” method in the requirement phase into test-and-measure methods.

Table 1: Recommended methods by the UsabilityNet approach and the TRUMP approach.

Method	UsabilityNet Approach	TRUMP Approach
<b>Planning and feasibility</b>		
Getting Started	✓	
Stakeholder Meeting	✓	✓
Analyze Context of Use	✓	
<b>Requirements</b>		
Analyze Context of Use	✓	✓
Evaluate Existing System	✓	✓
Affinity Diagramming	✓	
Scenarios of Use	✓	✓
Usability Requirements	✓	✓
<b>Design</b>		
Paper prototyping and	✓	✓

Evaluation		
Style Guides and Usability guidelines	✓	✓
<b>Implementation</b>		
Style Guides and Usability guidelines	✓	✓
Machine Prototypes and Evaluation		✓
<b>Test and Measure</b>		
Usability Testing (Diagnostic Evaluation)	✓	✓
Subjective Assessment	✓	
<b>Post-Release</b>		
Subjective Assessment	✓	✓

Table 2: The selected methods for the proposed framework after mapping “evaluate existing system” method in the requirement phase into test-and-measure methods.

Development Phase	Method	
<b>Planning and feasibility</b>	Stakeholder Meeting	
<b>Requirements</b>	Analyze Context of Use	
	Evaluate Existing System	Usability Testing (Diagnostic Evaluation)
		Heuristic Evaluation
	Scenarios of Use	
Usability Requirements		
<b>Design</b>	Paper Prototyping and Evaluation	
	Style Guides and Usability Guidelines	
<b>Implementation</b>	Style Guides and Usability Guidelines	
<b>Test and Measure</b>	Usability Testing (Diagnostic Evaluation)	
	Heuristic Evaluation	
<b>Post-Release</b>	Subjective Assessment	

## 5.2. Basic Methods for the Proposed Framework

The primary purpose of this stage is to determine the basic methods for the proposed framework. These methods are suggested to the framework user as the minimum method that needs to be applied to get the main benefits of user-centered design. According to TRUMP [23], we can attain the main benefits of user-centered design by using three simple techniques:

- Stakeholder meetings
- Paper prototyping and evaluation
- Usability testing with real users

Therefore, these three techniques will be considered the basic methods for our framework. In order to provide some additional guidance to the user of the framework, these methods are highlighted and presented as the first ones in each development stage, as shown in Table 3. They are suggested as the minimum methods that need to be applied by the user in order to reap the primary benefits of user-centered design.

### 5.3. UCD Methods Characterization

There are many factors that affect the selection of methods (ISO/TR16982). To simplify the framework, we have characterized the UCD methods according to the following five criteria:

- *User Involvement (UI)*: One of the basic principles of a user-centered approach is the active involvement of future system users. Some usability methods are specifically designed to encourage this involvement; others are not.
- *General Applicability (GA)*: This criterion reflects the scope of the method—that is, how applicable it is across a wide range of development environments. Some UCD methods—for example, card sorting—are applicable to web development. Others are applicable across a wide range of development environments, such as heuristic evaluation.
- *Limited Time (LT)*: This criterion determines whether the method can be applied when there is limited time. Some UCD methods, for example, are not relatively time-consuming, such as heuristic evaluation; other methods require more time, such as thinking aloud.
- *Limited Resources (LR) (cost/price control)*: This criterion determines whether the method can be applied when there are limited resources. Some methods are less costly than others. For example, heuristic evaluation is less expensive than methods that involve the user in usability testing. Therefore, it can be applied when there are limited resources (i.e., money).
- *Limited Skills and Expertise (LS)*: This criterion determines whether the method can be applied when there are limited skills and expertise in the development team. Whenever human-factors expertise is available in-house or outside, all methods can be used, including the ones that require high skill levels. Some UCD methods—for example, style guides and usability guidelines, can be applied when there is limited expertise and skills.

Table 3 shows our proposed framework for usability integration into small organizations after assigning values for each criterion, according to (ISO/TR16982) values, and according to the literature for each method. The framework allows developers to take into account decisions about the specific usability method in their development process in order to improve the usability of the developed product. It condenses important information developers need to apply to the solution we are proposing. In the framework, we offer references for each usability method, where developers may refer to apply the method. The framework aims to be flexible so that a particular process model is not required for its application.

In order to use the proposed framework, developers must look for the development-process phase they are interested in. At that point, they must translate the generic phases in the framework to the specific activity (phase) names considered in their organization, and then, select one or more of the proposed usability methods, according to the characteristics the organization values most (UI: User Involvement, GA: General Applicability, LT: Limited Time, LR: Limited Resources, LS: Limited Skills and Expertise), and refer to the reference for each usability method to apply it (WR: Web Reference, B: Book Reference, PS: Published Study). Basic methods are represented on a solid-gray background, while other methods are represented on a white background. The values for method characterization are represented as follows: Y: Yes; N: No; H: High; M: Medium; L: Low; ++: Recommended; +: Appropriate; When the Cell is Empty: Neutral; -: Not Recommended.

The framework offers a flexible solution for easing the usability-integration endeavor. In order to provide some additional guidance to the user of the framework, basic methods are highlighted and presented as the first ones in each development phase. However, developers may take into account the specific constraints of the organization or the project at hand, and considering the characterization criteria for the methods, they may select a set of methods that fit better their needs. The framework is considered to be a solution for supporting developers in integrating usability practices into the development process. For this purpose, it condenses some knowledge about the UCD field in a way that is understandable by developers with a SE background.

Table 3: Our proposed Framework for usability integration into small organization

Development Phase	Method	UI	GA	LT	LR	LS	References	
<b>Planning and feasibility</b>	Stakeholder Meeting	Y	M	+	+	+	WR: [24]	
<b>Requirements</b>	Analyze Context of Use	Y	M	+	+	+	WR: [25] B: [26]	
	Evaluate Existing System	Usability Testing (Diagnostic Evaluation)	Y	M	+	+	+	WR: [27] B: [28]
			Heuristic Evaluation	N	H	++	++	+
	Scenarios of Use	Y	M	++	++	+	WR: [30] B: [31]	
	Usability Requirements	N	M	+	+	+	WR: [32] B: [33]	
<b>Design</b>	Paper Prototypes and Evaluation	Y	H	++	++	++	WR: [34] B: [35]	
	Style Guides and Usability Guidelines	N	M	+	++	+	WR: [36] B: [35]	
<b>Implementation</b>	Style Guides and Usability Guidelines	N	M	+	++	+	WR: [36] B: [35]	
<b>Test and Measure</b>	Usability Testing (Diagnostic Evaluation)	Y	M	+	+	+	WR: [27] B: [28]	
	Heuristic Evaluation	N	H	++	++	+	WR: [29] B: [19]	
<b>Post-Release</b>	Subjective Assessment	Y	H	-		++	WR: [37] B: [35]	

## 5. CONCLUSION AND FUTURE DIRECTIONS

Although the benefits of usability have been widely proven, most small- and medium-sized software-development organizations do not apply usability while developing applications. The main reasons are related to the time and costs associated with the integration of usability practices into the software-development process [38]. In addition, there is a lack of knowledge and experience regarding usability in SMEs, and the lack of a reference framework or approach for integrating usability practices into the software-development process—particularly for small software-development organizations. Furthermore, usability-practices integration in the

development process is not an easy issue because of the existing gap between SE and HCI, where both fields have a different approach to development and also have different terminology.

Therefore, we propose a framework for integrating usability practices into the development life cycle; the framework characterizes 10 selected user-centered design (UCD) methods in relation to five relevant criteria that may be of interest to developers in the usability integration, and may respond to these organizations' needs. We organize the methods according to the stages in the development process where they may be applied. Our integration approach is cost-effective and based on a general software development process that can be applied to any small organizations regardless of their development processes. For future work, there is a need to apply the framework to see the results; in addition, we will release a website presenting the framework for easing the application of the framework. With the release of the framework on the web, we expect to refine it after receiving feedback from developers and usability specialists, who may use it for their particular needs.

## REFERENCES

- [1] H. Fischer, "Integrating usability engineering in the software development lifecycle based on international standards," in Proceedings of the 4th ACM SIGCHI symposium on Engineering interactive computing systems, 2012, pp. 321-324.
- [2] L. Carvajal, A. M. Moreno, M.-I. Sanchez-Segura, and A. Seffah, "Usability through software design," *Software Engineering*, IEEE Transactions on, vol. 39, pp. 1582-1596, 2013.
- [3] K.-P. Aikio, "Exporting Usability Knowledge into a Small-Sized Software Development Organization—A Pattern Approach," in *Human Interface and the Management of Information. Methods, Techniques and Tools in Information Design*, ed: Springer, 2007, pp. 3-11.
- [4] M. Woywode, A. Mädche, D. Wallach, and M. Plach, "Gebrauchstauglichkeit von Anwendungssoftware als Wettbewerbsfaktor für kleine und mittlere Unternehmen (KMU): Abschlussbericht," 2012.
- [5] M. J. Aamir and A. Mansoor, "Testing Web Application from usability perspective," in *Computer, Control & Communication (IC4)*, 2013 3rd International Conference on, 2013, pp. 1-7.
- [6] E. Commission. (05/10/2015, 11/10/2015). What is an SME? Available: [http://ec.europa.eu/growth/smes/business-friendly-environment/sme-definition/index\\_en.htm](http://ec.europa.eu/growth/smes/business-friendly-environment/sme-definition/index_en.htm)
- [7] S. Voruganti, *A Quality Framework for Small Software Organizations*, 1998.
- [8] A. Seffah and E. Metzker, "The obstacles and myths of usability and software engineering," *Communications of the ACM*, vol. 47, pp. 71-76, 2004.
- [9] C. B. Fellenz, "Introducing usability into smaller organizations," *interactions*, vol. 4, pp. 29-33, 1997.
- [10] X. Ferre, "Integration of Usability Techniques into the Software Development Process," in *ICSE Workshop on SE-HCI*, 2003, pp. 28-35.
- [11] X. Ferre, N. Juristo, and A. M. Moreno, "Framework for integrating usability practices into the software process," in *Product focused software process improvement*, ed: Springer, 2005, pp. 202-215.
- [12] L. Battle, "Patterns of integration: Bringing user centered design into the software development lifecycle," in *Human-Centered Software Engineering—Integrating Usability in the Software Development Lifecycle*, ed: Springer, 2005, pp. 287-308.
- [13] ISO, "TR 16982: Usability methods supporting human-centred design," *International Organisation for Standardisation*, 2002.
- [14] H. Fischer, K. Nebe, and F. Klompaker, "A holistic model for integrating usability engineering and software engineering enriched with marketing activities," in *Human Centered Design*, ed: Springer, 2011, pp. 28-37.
- [15] A. Seffah, J. Gulliksen, and M. C. Desmarais, "An introduction to human-centered software engineering," in *Human-Centered Software Engineering—Integrating Usability in the Software Development Lifecycle*, ed: Springer, 2005, pp. 3-14.
- [16] N. Bevan, "Cost benefits evidence and case studies," *Cost-justifying usability: An update for the internet age*. San Francisco: Morgan Kaufmann, 2005.

- [17] N. Bevan, "UsabilityNet methods for user centred design," *Human-Computer Interaction: theory and Practice*, vol. 1, 2003.
- [18] N. Bevan, Claridge, N., Fröhlich, P., Granlund, Å., Kirakowski, J., Tscheligi, M. (2006, 16/10/2015). UsabilityNet. Available: <http://www.usabilitynet.org/home.htm>
- [19] J. Nielsen, *Usability engineering*: Elsevier, 1994.
- [20] N. Bevan, Claridge, N., Fröhlich, P., Granlund, Å., Kirakowski, J., Tscheligi, M. (2006, 16/10/2015). methods Table. Available: <http://www.usabilitynet.org/tools/methods.htm>
- [21] J. Nielsen and R. Molich, "Heuristic evaluation of user interfaces," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1990, pp. 249-256.
- [22] N. Belkhit, M.-M. Boulet, S. Baffoun, and C. Dupuis, "Usability Inspection of the ECONOF System's User Interface Visualization Component," *Usability evaluation of online learning programs*, p. 177, 2003.
- [23] N. Bevan. (2001, 17/10/2015). Basic methods for User Centred Design. Available: <http://www.usabilitynet.org/trump/methods/basic/index.htm>
- [24] Usabilitybok.org. (2012, 17/10/2015). Stakeholder Meeting. Available: <http://www.usabilitybok.org/stakeholder-meeting>
- [25] Usabilitybok.org. (2009). Context of Use Analysis. Available: <http://www.usabilitybok.org/context-of-use-analysis>
- [26] H. Beyer and K. Holtzblatt, *Contextual design: defining customer-centered systems*: Elsevier, 1997.
- [27] N. Bevan, Claridge, N., Fröhlich, P., Granlund, Å., Kirakowski, J., Tscheligi, M. (2006, 16/10/2015). UsabilityNet: Diagnostic evaluation. Available: <http://www.usabilitynet.org/tools/diagnostic.htm>
- [28] J. Rubin, "Handbook of Usability Testing New York," ed: John Wiley & Sons, Inc, 1994.
- [29] Usabilitybok.org. (2007, 17/10/2015). Heuristic Evaluation Available: <http://www.usabilityevaluation.org/heuristic-evaluation>
- [30] Usabilitybok.org. (2011, 17/10/2015). Scenario of Use. Available: <http://www.usabilitybok.org/scenario-of-use>
- [31] M. B. Rosson and J. M. Carroll, *Usability engineering: scenario-based development of human-computer interaction*: Morgan Kaufmann, 2002.
- [32] N. Bevan, Claridge, N., Fröhlich, P., Granlund, Å., Kirakowski, J., Tscheligi, M. (2006, 16/10/2015). UsabilityNet: Requirements meeting. Available: <http://www.usabilitynet.org/tools/requirements.htm>
- [33] D. Hix and H. R. Hartson, *Developing user interfaces: ensuring usability through product & process*: John Wiley & Sons, Inc., 1993.
- [34] N. Bevan, Claridge, N., Fröhlich, P., Granlund, Å., Kirakowski, J., Tscheligi, M. (2006, 16/10/2015). UsabilityNet: Paper prototyping. Available: <http://www.usabilitynet.org/tools/prototyping.htm>
- [35] D. J. Mayhew, "The usability engineering lifecycle," in *CHI'99 Extended Abstracts on Human Factors in Computing Systems*, 1999, pp. 147-148.
- [36] N. Bevan, Claridge, N., Fröhlich, P., Granlund, Å., Kirakowski, J., Tscheligi, M. (2006, 16/10/2015). UsabilityNet: style guide. Available: <http://www.usabilitynet.org/tools/style.htm>
- [37] N. Bevan, Claridge, N., Fröhlich, P., Granlund, Å., Kirakowski, J., Tscheligi, M. (2006, 16/10/2015). UsabilityNet: Subjective Assessment. Available: <http://www.usabilitynet.org/tools/subjective.htm>
- [38] C. Lallemand, "Toward a closer integration of usability in software development: a study of usability inputs in a model-driven engineering process," in *Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems*, 2011, pp. 299-302.

## Authors

**Lulwah Alyahyan** received her Bsc in Computer Applications from King Saud University (Riyadh, KSA) and Msc in Software Engineering from the same university. Since 2009, she has been a research faculty member of the Computer Research Institute at King Abdulaziz City for Science and Technology (KACST), Riyadh, KSA, she has done research in several areas including natural language processing and web usability.

**Hamza Aldabbas** I am currently an Assistant Professor at Al-Balqa'a Applied University /Prince Abdullah Bin Ghazi Faculty of Science and Information Technology-Jordan , Previously a lecturer at De Montfort University/UK, with responsibility for teaching and project supervision at B.Sc & M.Sc levels (2010-to 2012). My Main research interests are in Ad hoc Networks, Grid computing, Context-aware Systems and E-commerce.