An Agile Information-Architecture-Driven Approach for the Development of User-Centered Interactive Software

Luis A. Rojas and José A. Macías
Escuela Politécnica Superior, Universidad Autónoma de Madrid,
Tomás y Valiente 11, 28049 Madrid, España
luisalberto.rojas@estudiante.uam.es, j.macias@uam.es

ABSTRACT
For the most part, Information Architecture processes include sets of activities and techniques to be carried out by the development team to create interactive applications effectively, involving usability concerns at every development step. In fact, plenty of process models have already been proposed to bridge the gap between User-Centered Development and Information Architecture, empowering the development team to build usable applications successfully. However, the combination of User-Centered Development and Information Architecture paradigms sometimes results in cumbersome process models containing lots of phases and activities to be considered, which increases the cycle time to have partial and validated software increments readily. As less effort has been devoted to speed up the usable Information Architecture development, the aim of this paper is to address such problem. To do so, we present Scrum-UIA, an agile and usable development process driven by the Information Architecture. This process is intended to develop web applications by splitting up responsibilities and tasks, and decreasing the time to perform technical activities, in order to readily obtain usable software increments.

Categories and Subject Descriptors
H.1.2 [Information Systems]: User/Machine Systems - Human factors; H.5.2 [Information Interfaces and Presentation]: User Interfaces – User-Centered Design and Prototyping

General Terms
Management, Design, Human Factors.

Keywords
Information Architecture, Agile Development, User-Centered Design, Usability

1. INTRODUCTION
Today, even though we live immersed in the information society, information in many organizations is still not properly managed, which affects negatively the costs of the organization in terms of errors and inefficiencies, and also the client's perception in terms of service quality [1]. Furthermore, environments move quickly and are constantly evolving in today's world, which is an additional factor that increases the complexity of information management in organizations.

The current scenario is at once an opportunity and an obligation for both researchers and information professionals, allowing to address the different challenges, opportunities and critical aspects of the Information Architecture (IA). Martin et al. [1] reported that one of the critical aspects of IA is the need for methodological proposals to develop it. There are currently several methodologies for the IA development [2,3,4,5,6]. However, these proposals are based on traditional development process models, making it difficult to carry them out in changing environments, where an agile and quick response is often required.

On the other hand, agile methodologies emerge as a response to the need of adapting quickly to changing environments. Nevertheless, this also reports difficulties when incorporating User-Centered Design (UCD) in agile environments [7], which is an overriding factor when developing interactive software. Several studies have addressed the need of involving end-users into the agile development, supplying specific recommendations but without providing a comprehensive or complete vision, which has been reported by [8] as an incentive to develop new methodologies for the integration of the UCD and the agile paradigm.

The aim of this paper is to address such drawbacks by providing an agile methodology, called Scrum-UIA (Scrum driven by Usable Information Architecture), for the user-centered development of interactive applications, also involving IA as a building block to guide and drive the development to make it agilely adaptable to changing environments. Our proposal features the integration of UCD in Scrum to carry through an agile approach. Also, we provide a set of agile AI techniques to support development tasks. In addition, an end-user vision is incorporated and considered throughout the whole process.

This paper is organized as follows: Section 2 presents a comparative analysis of methodologies for IA and a discussion of agile ones. Next, section 3 introduces the practices used in agile and UCD integration, as well as specific recommendations to integrate UCD in the Scrum methodology. Then, section 4 describes our proposed methodology for agile UCD-IA development based on Scrum. Finally, section 5 presents conclusions and future work.

2. ANALYSIS AND DISCUSSION ON RELATED METHODOLOGIES
In this section, a comparative analysis is performed to check the current features of the different methodologies for the IA development (section 2.1). On the other hand, the feasibility of

integrating user-centered activities into agile methodologies is also addressed (section 2.2).

2.1 Comparative Analyses of Information Architecture Methodologies

Five IA methodologies [2,3,4,5,6] have been selected according to their popularity and broad usage. In order to have a common framework to analyze and compare the main features of these methodologies for the IA development, the following criteria has been used: *User-Centered criterion, IA Elements Covered, Level of Description, Scope of the Proposal* and *Flexibility and Adaptability*. These will be briefly described below.

Table 1. Comparative analysis of IA methodologies.

Method	User-Centered	IA Elements Covered	Level of Descr.	Scope of the Prop.	Flex. and Adpt.
[2]	Partially	-Navigation -Organization -Labeling -Search and recovery -Information presentation	High	A, D, I (-), M (-)	No
[3]	Partially	-Navigation -Organization -Labeling	Low	A, D	No
[4]	Partially	-Navigation -Organization -Labeling	Low	A,D, E (-)	No
[5]	Yes	-Organization -Labeling	Average	E	No
[6]	Yes	-Navigation -Organization -Labeling -Search and recovery -Information presentation	High	A, D, E, I, M	No

The *User-Centered* criterion is used to identify whether the proposals design their products focused on the needs of end-users. This criterion can take Yes, Partially or No values to indicate that the proposals are fully, partially or not user-centered, respectively. The *IA Elements Covered* criterion is used to identify the aspects of the IA that proposals attempt to cover. According to Erlin et al. [9] the aspects of the IA that are involved in the development of a product correspond to: navigation, organization, labeling, search and recovery, and information presentation. The *Level of Description* criterion is used to identify the level of detail in which methodologies for the IA development are described. This criterion can take High, Average or Low values to indicate that the proposals are described with sufficient, moderate or general detail, respectively. The *Scope of the Proposal* criterion is used to identify the lifecycle phases of IA development that proposals attempt to cover. The considered phases are: analysis (A), design (D), evaluation (E), implementation (I), and management (M). The phases that are incompletely covered (without providing or prescribing the necessary information to carry them out) are pointed out with an (-). Finally, the *Flexibility and Adaptability* criterion is used to identify whether methodologies are able to adapt to changing environments and respond quickly. In order to assess this criterion, IA methodologies are analyzed to know whether they require extensive planning or have a development process model (i.e., waterfall) that needs numerous controls and policies/standards to be implemented.

The results of the comparative analysis are presented in Table 1.

As we can see in Table 1, most of the proposals do not cover all of the lifecycle phases of IA development, focusing primarily on the analysis and design phases. It is important to highlight that proposals [6] and [5] are the only ones that present recommendations of activities and techniques for the IA evaluation (*Scope of the Proposal* criterion: E). The proposal [4] also indicates IA evaluation, but it is only described and shallowly defined (-). It is worth noting that methodologies [5] and [6] are the only ones that provide a user-centered approach to IA development. By contrast, in most of the proposals this aspect is carried out partially, as the end-users are only included in the initial phases.

Finally, methodologies [6] and [2] are the only ones that have an adequate level of proposal description (*Level of Description* criterion: High) and cover all IA aspects (*IA Elements Covered* criterion) in the different lifecycle phases of IA development (*Scope of the Proposal* criterion). However, none of the analyzed methodologies present flexibility and adaptability characteristics to respond in an agile and flexible way to changing environments, or they require considerable effort to be adapted (*Flexibility and Adaptability* criterion).

Such drawback can be addressed through the flexibility and adaptability offered by agile methodologies, which allow to respond quickly to changing environments. However, agile methodologies include specific issues that can prevent against usage in user-centered product design, which is an essential factor to consider for the IA-driven development.

Therefore, it becomes necessary to analyze the feasibility of agile methodologies to integrate user-centered activities, in order to be able to clearly discern which agile methodologies are more suitable for the development of user-centered interactive software.

2.2 Feasibility of Agile Methodologies to Integrate User-Centered Activities

This section presents a study and analysis of several agile methodologies in order to know the extent to which these can integrate user-centered activities. This will enable to find out both the aspects to envision a hypothetical integration support and the strengths that make it impossible.

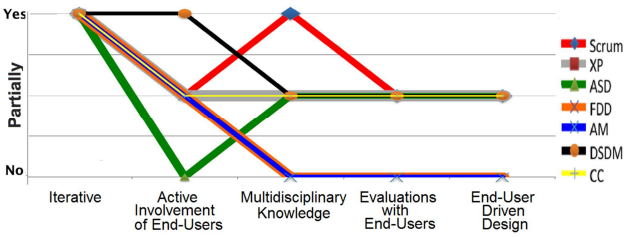

Figure 1. Overview of the feasibility of agile methodologies to integrate user-centered activities.

This way, the seven most-popular agile methodologies were analyzed against a set of reference attributes based on the principles of ISO 9241-210 [10]. The agile methodologies analyzed were Scrum, Extreme Programming (XP), Adaptive Software Development (ASD), Feature-Driven Development (FDD), Agile Modeling (AM), Dynamic Systems Development Method (DSDM) and Crystal Clear (CC).

Figure 1 depicts the analysis in relation to the degree of feasibility of agile methodologies to integrate user-centered activities. Axis X represents the ISO 9241-210 attributes used to evaluate the agile methodologies, whereas axis Y indicates the degree to which the agile methodologies comply with such attributes (Yes, Partially or No).

As shown in the Figure 1, all methodologies comply with the *Iterative* attribute, so that the project is planned and executed iteratively.

Regarding the *Active Involvement of End-Users* attribute, in 5 out of the 7 analyzed methodologies (Scrum, XP, FDD, AM and CC) end-users are partially involved in the process. The DSDM agile methodology is the only one that allows the active participation of end-users by proposing roles that are directly assumed by them. Although 4 out of the 7 analyzed methodologies (Scrum, XP, AM and CC) also describe roles related to end-users, those are more related to aspects concerning functional requirements of the system. It is also important to highlight that Scrum includes a review process (*Sprint Review*), which would facilitate the direct involvement of end-users.

With respect to *Multidisciplinary Knowledge* attribute, 5 out of the 7 analyzed methodologies (Scrum, XP, ASD, DSDM and CC) consider work teams with different skills and knowledge. However, the team building is primarily based on the search of efficiency in the development and maintenance of functionalities from the technical perspective of a software programmer. In the case of Scrum methodology, it explicitly states that the work team formation will be based on all competencies needed to accomplish the work without depending on others being not part of the team.

In relation to *Evaluations with End-Users* attribute, 5 out of the 7 analyzed methodologies (Scrum, XP, ASD, DSDM and CC) accomplish assessments, which are partially focused on end-users. In fact, these methodologies consider specific roles on behalf of end-users, even considering roles as client or expert user with knowledge about the system procedures or functional requirements, which would allow partial involvement of end-users in assessments. Nevertheless, agile assessments are mainly oriented towards unit testing and system integration, as most of the agile approaches are principally focused on the client rather than on the end-user. It is worth noting that Scrum defines a specific event (*Sprint Review*) in order to verify each product increments with the participation of stakeholders. This feature allows the participation of end-users in evaluations.

As for the last evaluated *End-Users-Driven Design* attribute, in 5 out of the 7 analyzed methodologies (Scrum, XP, ASD, DSDM and CC) the design is in part based on understanding end-users, tasks and environments. In 3 out of the 7 analyzed methodologies (Scrum, DSDM and CC) there exists the use of non-functional prototypes and participation of various roles related (directly or partially) to end-users. This facilitates the inclusion of environmental and end-user tasks aspects. However, the priority is set on getting a usable version of the software. In this respect, it is important to stress that in Scrum the requirements are dynamically managed by the *Product Owner* role, which has a profile oriented to the end-user's needs. Hence, this would lead the design based on understanding end-users, tasks and environments.

All in all, DSDM and Scrum are principally the two agile methodologies enabling a better adoption of user-centered activities. These methodologies have in common that both of them involve end-users by considering specific roles and using practices to actively involve end-users. For instance, DSDM presents the user's role descriptions and the use of prototypes. Similarly, Scrum facilitates end-users to prioritize the list of requirements and provides a review process that facilitates end-user involvement. On the other hand, the DSDM methodology has some weaknesses that prevent its application for the stated purpose. To cite a few, DSDM methodology requires a specific institutional framework for software development process, which is neither cheap nor easy to implement, and it also demands significant change of consciousness in any organization. By contrast, Scrum methodology presents a framework that is easy to implement, providing flexibility and adaptation to end-user and business requirements.

On the other hand, AM and FDD methodologies provide less facilities for the UCD integration. These methodologies have in common their orientation towards optimization of coding techniques and modeling for systems development, as well as the use of such methods for technical purposes.

Therefore, the Scrum agile method has been selected as ideally suited to integrate UCD in a development environment that requires agility and end-user focusing.

3. INTEGRATING UCD AND AGILITY

This section describes and analyzes the different practices that are commonly used to integrate UCD in agile methodologies (section 3.1). Furthermore, specific recommendations that enable the integration of UCD in the Scrum methodology are also reviewed (section 3.2).

3.1 Practices and Common Artifacts

Silva da Silva et al. [11] identified, by means of a systematic review of bibliography, what needs, artifacts and common practices are used to support collaboration between designers and developers in the integration of UCD and agility. Silva da Silva et al. analyzed 58 papers concluding that the most important practices and artifacts correspond to: *little design up front* (LDUF), *low fidelity prototypes, users testing, user stories, inspection methods, one sprint ahead and big picture*. Similarly, Jurca et al. [12] conducted a systematic mapping study to identify relevant research and understand Agile-UCD combination. Some of the recommended practices and artifacts identified were the following: *concept maps, low fidelity prototypes, interviews, scenarios and meetings with users*. Recently, Brhel et al. [13] also conducted a systematic review of the literature on aspects of integration of UCD in agile methodologies. The identified papers were analyzed using a coding system of four levels: process, practices, people and technological dimensions. Thus, the most common practices identified were: *prototypes, scenarios, usability evaluation* (expert), *usability testing* (user) *and user stories*, among others. Finally, Jia et al. [14] conducted a study with the aim of exploring how usability techniques were integrated during the software development in Scrum projects. In this case, the most used usability techniques found in Scrum projects were: *workshops, low fidelity prototypes, interviews and meetings with users*.

Summarizing all, Table 2 identifies correspondences between techniques, reported by [11,12,13], used to integrate UCD in agile methodologies. In addition to this, the correspondence between the techniques mentioned above and the usability techniques used in the Scrum methodology [14] has been identified as well. These correspondences allow to identify matches between usability techniques used in the Scrum and techniques used to integrate UCD in agile methodologies.

Table 2. Summary, according to reviewed bibliography, describing correspondences between techniques used to integrate UCD in agile methodologies and usability techniques used in the Scrum methodology.

Techniques Used to Integrate UCD in Agile			**Usability Techniques Used in Scrum** [14]
[11]	[12]	[13]	
Low fidelity prototyping	Low fidelity prototyping	Prototyping	Low fidelity prototyping
Scenarios	Scenarios	Scenarios	Scenarios
Inspection methods	Cognitive walkthrough variants	Usability evaluation (expert)	Heuristic evaluation
Users testing		Usability testing (user)	Usability evaluation with users
Personas		Personas	Personas
	Workshops	Focus groups	Workshops
	Interviews	Interviews	Interviews
	Meetings with users		Meetings with users
		Contextual inquiry	Field studies
User stories		User stories	
Guidelines		Guidelines	

As shown in Table 2, *low fidelity prototyping, scenarios, heuristic evaluation, usability testing, people, workshops and interviews* are the most frequent techniques found in the systematic review of the literature regarding UCD and agile integration. This consolidates such techniques as mandatory when integrating aspects of usability in agile projects. Moreover, it is important to note that *user stories* and *guidelines* techniques are not mentioned in [14] as common techniques in Scrum projects because these are mainly associated to XP methodology.

Lastly, it can be stated that all these techniques have in common the ability to adapt to changing environments that require a rapid response to the needs of end-users and business value.

3.2 Specific Recommendations to Integrate UCD in Scrum

In this section, the different practices and recommendations used to integrate UCD in the Scrum methodology have been collected and analyzed.

Definition of "Done" for UCD-Related Tasks: Kniberg [15] highlighted the importance of the fact that Product Owner and Development Team should agree on a clear definition of "Done". This would facilitate obtaining a common understanding regarding the scope and demands of the requirements presented by the Product Owner, as well as the tasks that will be carried out by the Development Team. Felker et al. [16] proposed to use a different definition of "Done" for UCD-related tasks. Authors suggested that this strategy would facilitate the monitoring of work in order to know the right moment to move on to the next task.

Product Backlog Management: Singh [17] noticed that one of the key challenges to usability in the Scrum projects is the study of end-user needs and context. Therefore, selected requirements should fit usability concerns and be prioritized accordingly. To address these drawbacks, Budwig et al. [18] proposed the creation of a Product Backlog for issues related to UCD, which helps the UCD team allocate corresponding resources to projects. Similarly, Singh [17] proposed to structurally maintain the Product Backlog but incorporating elements including greater awareness of usability, that is, a higher priority for the requirements that impact on usability, especially regarding potential acceptance criteria for requirements. Furthermore, Kuusinen [19] proposed that UCD specialists were given more influential roles in regard to product level decisions in order to improve managing the *big picture* and understanding and fulfilling end-user needs.

Usability Evaluation Management: Lárusdóttir et al. [20] indicated that it is difficult to find a good moment for UCD evaluation in Scrum. On the one hand, a very early assessment in the project is complicated, because the available features are still insignificant to have a UCD assessment. On the other hand, when characteristics are significant to be evaluated, it is difficult to make important changes because some parts of the product have already been delivered and there is not enough time to evaluate before the next delivery. There have been different proposals attempting to minimize this drawback. Among others, Felker et al. [16] proposed to schedule assessments before knowing what is going to be evaluated, analyze end-user feedback just after UCD assessments, and carry out this at the end of Sprint. Lárusdóttir et al. [21] suggested using informal ways to involve end-users in the evaluation and apply different methods to successfully perform each of the user-centered assessments.

Completing the Contextual Inquiry Beforehand: The contextual inquiry is a method to inspect and understand end-users and their workplace, tasks and preferences. Rannikko [22] recommended the contextual inquiry to be completed before starting the software development. Felker et al. [16] reported on a successful usage of such guideline, so the authors noted that possessing the results of a contextual inquiry was incredibly helpful and it allowed them to focus on the design and implementation, helping establish initial priorities.

Close Collaboration: Lárusdóttir et al. [20] suggested that the UCD specialists should work closely with developers in Scrum Teams. Moreover, Kuusinen [19] stated that it was necessary to identify the right moment for the UCD specialists to work. These issues have been addressed in different ways, among these, it has been suggested that the UCD should occur in parallel tracks to implementation [23], set the UCD teams to work in one or two Sprints ahead of the development teams [18] and design a Sprint ahead of implementation [16,21].

Big Picture of the Project: The term big picture refers to a holistic view of the whole project in Scrum. Lárusdóttir [20] reported that the big picture of user experience is usually missing in Scrum projects. On the one hand, one of the reasons why this happens is that programmers have the responsibility to deliver a small piece of software, but often they do not feel responsible for the user experience or the entire system. On the other hand, it has been reported that the big picture of user experience is not present because the responsibility for particular activities of user experience in Scrum projects is often not clearly defined [20]. To address these difficulties, Budwig et al. [18] proposed to quarterly incorporate, throughout a common Sprint cycle, activities oriented to update the big picture, in order to have a clear vision of the design to be carried out in the proyect and keep up the overall coherence. Another proposal is to use overall quality goals to help deliver the overall design direction [8]. Finally, Lárusdóttir et al. [20] indicated the need for strategic vision and user experience objectives to be defined before starting the current project, that is,

before the Sprint. However, the strategic vision should also be considered when defining what will take place in different Sprints [8]. Therefore, different authors recommend that a view of user experience must be considered before starting the implementation, but also it needs to be applied during the iterations of the Scrum project.

Assign Responsibility for End-User Concerns: The results of the work of Cajander et al. [8] showed that the responsibility for the end-user's perspective is not clear in Scrum projects, and end-user perspective is often neither discussed nor described in the projects. However, the end-user's perspective is often present through informal feedback used to understand the context of use and report design. Cajander et al. [8] aimed to strengthen the emphasis on the end-user's perspective by clarifying and explicitly communicating the responsibility of working through usability. This includes both who will work with usability and who is responsible for the quality of the final product. However, this proposal does not solve what could be done in the context of Scrum, where there are no formal responsibilities for any quality aspects, such as security, privacy and performance. Cajander et al. [8] provided some examples of the organizational support needed: sufficient mandate, support from the management, organizational competence as well as an adequate position in the team to be able to contribute to better usability.

Systematize the Process of End-User Inclusion: Cajander et al. [8] indicated that general agile processes do not support end-user participation. Rather, end-users are informally involved. Often this is done on an ad hoc basis, and mostly based on personal initiative and knowledge of the team members about the end-user's perspective rather than being systematically planned in the Scrum process. Cajander et al. [8] suggested that it could be useful to systematize the process through showing end-user involvement and design feedback as general activities in the development process.

4. PROPOSAL

In order to address commented drawbacks taking also into consideration analyzed proposals to integrate the UCD in the Scrum methodology, an IA-driven approach is proposed with the aim to integrate agility into the user-centered development process of interactive software. The proposal, called Scrum-UIA (Scrum driven by Usable Information Architecture), is based on the Scrum methodology, and it includes roles, events, artifacts and associated rules [24], as well as a combination of practical and specific recommendations, as analyzed before in the literature, to integrate UCD in the Scrum process (see Figure 2).

The team is composed following the same traditional structure of Scrum (a Product Owner, the Development Team, and a Scrum Master) but the Information Architect is incorporated as a primary role in order to lead the contextual inquiry, support the Product Backlog management, promote the IA-driven incremental development, ensure UCD and encourage end-user involvement.

As shown in Figure 2, our proposal is based on establishing a contextual inquiry as the starting point of the project with the aim of studying and analyzing the needs of end-users and prioritizing the contents, in order for the gained knowledge to provide the basis to set up a big picture of the project. Requirements included in the Product Backlog are improved by incorporating output from the contextual inquiry (end-user and content priorities). This is carried out in coordination between the Product Owner and the Information Architect.

The Sprint planning is carried out after the requirements management through the Sprint Planning Meeting event. In the Sprint planning the work to be done in each Sprint is defined. Thus, the highest prioritized requirements of the Product Backlog are selected for the Sprint Backlog. (This is explained in detail in Section 4.1)

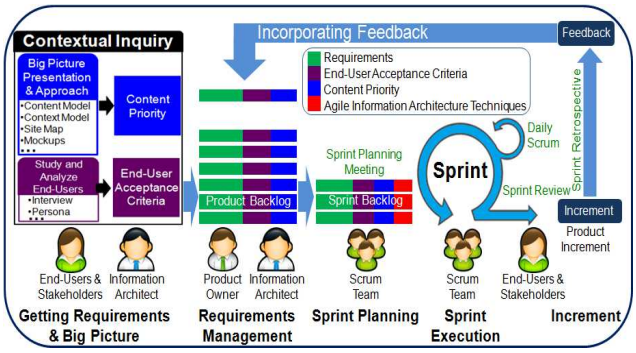

Figure 2. Detail of Scrum-UIA.

The Product Backlog's items selected for this Sprint together with the plan to carry them out made up the Sprint Backlog. The Development Team splits requirements selected for the Sprint Backlog into specific development tasks, and those are developed during the Sprint execution. Moreover, specific development tasks can be associated to techniques for Agile IA development that are auto-assigned by the Development Team.

Daily Scrum meetings are carried out during the Sprint execution every day in order to inspect and adapt the daily work accomplished by the Development Team, and also to manage individual development tasks so that the visibility and consistency with the big picture can be maintained.

The Sprint Review meeting takes place at the end of Sprint and is carried out involving both end-users and other secondary stakeholders, in order to review the potential product Increment generated by the Development Team.

Finally, the Scrum Team analyzes the working practices during the iteration and seeks improvement opportunities through the Sprint Retrospective meeting. Overall, our proposal is based on three essential components:

- *Contextual inquiry-driven Product Backlog management.*
- *Information architecture-driven Sprint planning.*
- *End-user-driven inspection and Continuous Improvement processes.*

These essential components are described below in detail.

4.1 Contextual Inquiry-Driven Product Backlog Management

The objective of this component is twofold: first, to ensure that the end-user's perspective can be discussed, described and considered (problem reported by [8]) throughout the Scrum process. And the second is to provide a big picture of the project to help obtain a global and inclusive vision of the product regarding usability priorities, content, business value and end-users.

In order to fulfill the objectives indicated above, we propose to initiate the project with a contextual inquiry (see Figure 3), as recommended in [22] and successfully implemented in [16], with the aim of obtaining knowledge about the priorities of end-users

and content. This simultaneously provides the basis to set up the big picture of the project to support the Product Backlog management.

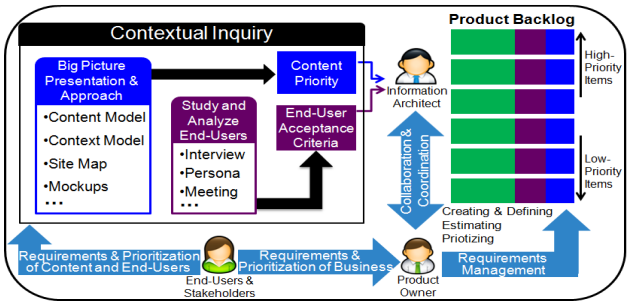

Figure 3. Contextual inquiry-driven Product Backlog management.

Lárusdóttir et al. [20] reported that Scrum projects do not have a big picture and, according to [25], this is perceived as a problem among UCD professionals. We propose to address this problem by specifying a low fidelity vision of the following artifacts: *content model*, *context model*, *site map* and *mockups*. These artifacts make it possible to report the design of contextual navigation of a site, determine the critical content and visualize relationships between pages and other content components. Thus, these artifacts can be processed to obtain an overview of a site and facilitate discussion of the organization and content management, as well as access priorities desired by end-users [2,26].

As a result, suggested artifacts can support the management of the project's big picture by providing a clear view of the most critical content elements of a site, and also facilitate discussion among the team members on the aspects of end-users and contents related to business needs.

Therefore, it is proposed to develop the suggested artifacts before starting the current project [20] during the contextual inquiry. Moreover, it is suggested to quarterly incorporate work to update the artifacts throughout a common Sprint cycle [18] and use overall quality goals to support the global design management [8]. Furthermore, it is suggested to obtain and specify end-users acceptance criteria through interviews, Personas and meetings techniques that are usually used in Scrum projects [14].

This way, we propose, similarly to Kuusinen's suggestion [19], to provide the Information Architect with the facility for defining, estimating and prioritizing the Product Backlog collaborating in coordination with the Product Owner. In this case, the Information Architect provides the vision of IA and usability obtained from contextual inquiry, and the Product Owner provides business requirements and prioritization.

Finally, we propose, in a similar manner to [17], to maintain the same structure of Product Backlog but incorporating elements to include greater consideration on content priority, end-users acceptance criteria , business value priorities, as well as identification and association of requirements with the big picture's elements. In particular, the Information Architect identifies, among Product Backlog items, high-level conceptual representations of content that evolve towards the solution domain and facilitate the IA-driven incremental development – i.e., *content models* created in the problem domain evolving towards models closer to the solution domain [27]. The results obtained are then used in the Sprint Planning Meeting.

4.2 Information Architecture-Driven Sprint Planning

This component has three main objectives: the first is to ensure (during the Sprint Planning Meeting) that requirements development is driven by IA priorities in an agile and user-centered way. The second is to promote that the development is performed incrementally through the different fidelity levels in evolution with IA deliverables (*blueprints*, *wireframes*, *content mapping*, *inventory* and *content models*). And the third is to provide a common understanding through a clear definition of "Done", as recommended [16], regarding the scope and demands of the requirements set by the Product Owner (in collaboration with the Information Architect) and the tasks that are required to be performed by the Development Team according to specific and selected techniques.

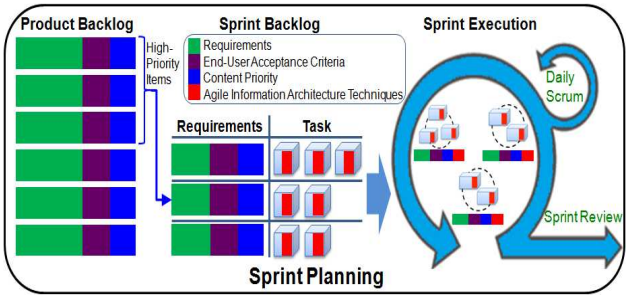

Figure 4. Information Architecture-driven Sprint planning.

As shown in Figure 4, the requirements of the Product Backlog having the higher priority are selected to be included in the Sprint Backlog. Each of the selected requirements (which simultaneously contain elements regarding the priority of end-users, business value and content) is split into specific development tasks by the Development Team. Each requirement priority has an IA-driven weighing provided by the Information Architect during the requirements management, in order to specify IA priorities for the incremental development.

A set of agile techniques for IA development (see Table 3) is provided to support the development of specific tasks during the Sprint execution. The agile techniques for IA development are auto-assigned by the Development Team, as well as the tasks to carry out for each of the specific development tasks.

We have obtained the agile techniques for IA development from a second analysis of the five IA development methodologies [2,3,4,5,6]. The second analysis was performed in order to obtain a set of activities, techniques and products for IA development considering an agile and user-centered approach. In addition the coincidences between the resulting techniques and the usability techniques used in Scrum, reported by [14], are identified.

In Table 3, the aforementioned set of agile techniques for IA development is presented, as well as the recommended phases where they can be applied: analysis (A), design (D), evaluation (E), implementation (I) and management (M). Furthermore, the techniques presented by [14] are pointed out with an asterisk (*) in order to demonstrate the high correspondence with the Agile IA techniques. Also, each associated IA methodology is provided denoting its corresponding bibliographical reference.

Finally, the Development Team performs the Sprint execution according to the stated planning and the techniques to ensure usability during the development, having in mind the big picture of the project and a common understanding of the requirements.

Table 3: Recommended techniques for agile IA development.

Techniques	IA Meth.	A	D	E	I	M
Affinity diagram	[4,6]		X			
Background investigation	[2]	X				
Benchmarking	[2,6]	X	X			X
Card Sorting	[2,4,5,6]	X	X	X		X
Consistency inspection	[6]		X			
Consolidated evaluation	[6]	X	X			
Diagramming	[4]		X			
Entity–relationship model	[6]			X		
Feedback analysis	[2,5,6]			X		X
Field studies*	[6]	X				X
Focus group discussion	[2,6]	X				X
Goodness rating	[5]			X		
Heuristic evaluation*	[2,5,6]	X		X		
Interface design patterns	[6]	X	X		X	
Interviews*	[2,3,6]	X		X		
Low fidelity prototyping*	[6]	X	X			
Meetings*	[2,6]	X			X	
Mock-up prototype	[6]	X	X			
Participatory design	[2,6]		X		X	
Personas*	[2,4,6]	X	X			
Predictability and efficiency evaluation	[5]			X		
Questionnaires*	[4,5]	X		X		
Scenarios*	[2,6]	X	X			
Speeded sentence verification	[5]			X		
Sponsor-driven structure evaluation	[5]			X		
Storyboards	[6]			X		
Structure evaluation	[6]		X			
Usability evaluation*	[6]			X		
Survey	[5]	X		X		
Workshops*	[2,6]	X			X	

4.3 End-User-Driven Inspection and Continuous Improvement Processes

The last issue of our proposal is aimed at enhancing traditional inspection and continuous improvement processes in Scrum (Daily Scrum, Sprint Review and Sprint Retrospective) as a way of encouraging UCD and end-users involvement. On the one hand, individual tasks are inspected to keep up with the big picture and track compliance of end-users acceptance criteria from the early stages during the Sprint (Daily Scrum). On the other hand, the potential product Increment is evaluated through the straight involvement of end-users and other secondary stakeholders (Sprint Review). And last but not least, all team members reflect on the completed Sprint to find out what improvements could be made in the next one – i.e., process improvement (Sprint Retrospective).

During the Sprint execution a "Done", useable, easy-to-use and potentially releasable product Increment is created. As shown in Figure 5, the Sprint execution is supported by a set of activities, techniques and products for IA development in an agile and user-centered way, as commented before.

Every day, the Daily Scrum takes place during the Sprint execution. The Daily Scrum is done by inspecting the work since the last Daily Scrum and forecasting the work that needs to be done before the next one. Moreover, we encourage the Information Architect participation in the Daily Scrum to review individual tasks in order to preserve the big picture and identify, from early stages, the fulfillment of the end-user's acceptance criteria.

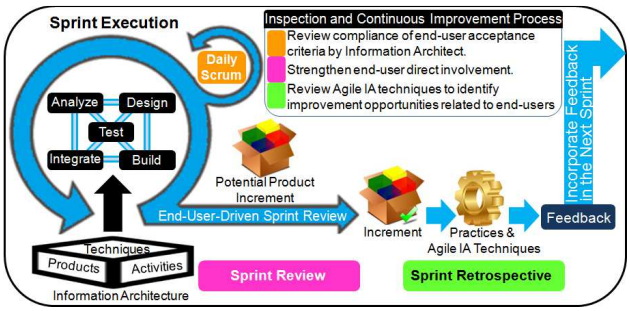

Figure 5. End-user-driven inspection and continuous improvement processes.

A Sprint Review is carried out at the end of the Sprint execution in order to inspect the Increment and sort out the Product Backlog if needed. It is proposed to strengthen end-users involvement through an end-user-driven Sprint Review that, according to [8], could be also beneficial. In this way, the end-user's direct involvement is materialized through formally assigning the prerequisite that end-users must interact with the potential product Increment during Sprint Review. That is to say, the Development Team demonstrates the work that is "Done" through the end-user's direct involvement interacting with the Increment. Moreover, it is suggested to reflect on the results of these ceremonies just after the evaluations [16] and apply different methods to successfully perform each of the user-centered assessments [21].

Finally, the Sprint Retrospective is held with the aim of reviewing aspects concerning the practices and Agile IA techniques used by the Scrum Team during Sprint execution, and also creating a plan for improvements to be enacted during the next Sprint. The Information Architect uses this event as an opportunity to review the performance of the practices and techniques used in the development of the IA, in order to identify improvement opportunities regarding responsiveness to the demands of agility and management of issues related to end-users. The idea is to improve the process, incorporating ideas for next Sprints.

5. CONCLUSIONS

This paper presents an IA-driven approach for the agile development of usable software, which considers the dynamic demands of the environments where organizations operate, as well as the needs of end-users. Concretely our approach, named Scrum-UIA, aims at integrating agility and UCD by contributing the following issues:

- The end-user's perspective is discussed, described and considered during the Scrum process through a *Contextual Inquiry-Driven Product Backlog Management*.
- Development of requirements is driven by IA priorities in an agile and user-centered way through an *Information Architecture-Driven Sprint Planning*. Furthermore, a set of agile techniques for IA development is provided to support the development of specific tasks during the Sprint execution.
- Compliance with the end-user's acceptance criteria is checked from the early stages of the Sprint, and the potential product Increments are evaluated through an *End-User-Driven Inspection and Continuous Improvement Processes*.

In addition, this paper presents different contributions involving analysis and discussion on the following issues:

- A comparative analysis checking the current features of the methodologies for the IA development shows that none of them present flexibility and adaptability characteristics to respond in an agile and flexible way to changing environments, and require considerable effort to be adapted.
- A analysis to know the feasibility of integrating user-centered activities into agile methodologies, showing that the Scrum methodology presents a framework that is easy to implement, providing flexibility and adaptation to end-user and the business requirements.
- A analysis of the practices that are commonly used to integrate UCD in agile methodologies, showing that *low fidelity prototyping, scenarios, heuristic evaluation, usability testing, people, workshops and interviews* are the most frequent techniques used in the UCD and agile integration.

As future work, we expect to build an easy-to-use CASE tool, which can be used to implement Scrum-UIA, in order to automatically provide scheduling, recommendation, activities and techniques to carry out an agile UCD-IA development, also dealing with end-user demands, priorities, usability, business value and content during the agile development process.

6. ACKNOWLEDGMENTS

This work has been supported by the funding projects «eMadrid», granted by the Madrid Research Council (project code S2013/ICE-2715) and «Flexor», granted by the Spanish Government (project code TIN2014-52129-R).

7. REFERENCES

[1] Martin, A., Dmitriev, D., and Akeroyd, J. a resurgence of interest in information architecture. *International journal of information management*, 30, 1 (2010), 6-12.

[2] Morville, Peter and Rosenfeld, Louis. *Information architecture for the world wide web*. O'Reilly, 2006.

[3] Sharlin, Micole, Tu, Evelyn, and Bartus, Thomas. Guide to Creating Website Information Architecture and Content. *Princeton University* (2009).

[4] Lamar, L. Introduction to a user interface design/information architecture process for Web sites. *In Professional Communication Conference* (2001), 185-197.

[5] Toub, Steve. Evaluating information architecture: a practical guide to assessing web site organization. *Argus Center for Information Architecture* (2000).

[6] Reichenauer, A. *LUCIA: Development of a comprehensive information architecture process model for websites*. (Doctoral dissertation, University of Regensburg, Germany), 2005.

[7] Salah, D., Paige, R. F., and Cairns, P. A systematic literature review for agile development processes and user centred design integration. *In Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering* (May 2014), p. 5.

[8] Cajander, Å., Larusdottir, M., and Gulliksen, J. Existing but not explicit-The user perspective in scrum projects in practice. *In Human-Computer Interaction–INTERACT 2013* (2013), 762-779.

[9] Erlin, E., Yunus, Y., and Abdul Rahman, A. The evolution of information architecture. *Institute of Electrical and Electronics Engineers* (2008).

[10] DIS, I. 9241-210: 2010. Ergonomics of human system interaction-Part 210: Human-centred design for interactive systems. *International Organization for Standardization (ISO), Switzerland.* (2009).

[11] Da Silva, T. S., Martin, A., Maurer, F., and Silveira, M. S. User-Centered Design and Agile Methods: A Systematic Review. *In AGILE* (2011), 77-86.

[12] Jurca, G., Hellmann, T. D., and Maurer, F. Integrating Agile and User-Centered Design: A Systematic Mapping and Review of Evaluation and Validation Studies of Agile-UX. *In Agile Conference (AGILE)* (July 2014), 24-32.

[13] Brhel, M., Meth, H., Maedche, A., and Werder, K. Exploring Principles of User-Centered Agile Software Development: A Literature Review. *Information and Software Technology* (2015).

[14] Jia, Y., Larusdottir, M. K., and Cajander, Å. The usage of usability techniques in Scrum projects. *In Human-Centered Software Engineering* (2012), 331-341.

[15] Kniberg, H. Scrum and XP from the Trenches. *Lulu* (2007).

[16] Felker, C., Slamova, R., and Davis, J. Integrating UX with scrum in an undergraduate software development project. *In Proceedings of the 43rd ACM technical symposium on Computer Science Education* (2012), 301-306.

[17] Singh, M. U-SCRUM: An agile methodology for promoting usability. *In Agile, 2008. AGILE'08* (August 2008), 555-560.

[18] Budwig, M., Jeong, S., and Kelkar, K. When user experience met agile: a case study. *In CHI'09 Extended Abstracts on Human Factors in Computing Systems* (April 2009), 3075-3084.

[19] Kuusinen, K. Improving UX Work in Scrum Development: A Three-Year Follow-Up Study in a Company. *In Human-Centered Software Engineering* (2014), 259-266.

[20] Lárusdóttir, M. K., Cajander, Å., and Gulliksen, J. The big picture of UX is missing in Scrum projects. *In Proc. I-UxSED* (2012).

[21] Lárusdóttir, M., Cajander, Å., and Gulliksen, J. Informal feedback rather than performance measurements–user-centred evaluation in Scrum projects. *Behaviour & Information Technology*, 33, 11 (2014), 1118-1135.

[22] Rannikko, P. User-centered design in agile software development (2011).

[23] Sy, D. Adapting usability investigations for agile user-centered design. *Journal of usability Studies*, 3 (Feb. 2007), 112-132.

[24] Schwaber, K. and Sutherland, J. The Scrum guide. *Scrum Alliance* (2011).

[25] da Silva, T. S. *A framework for integrating interaction design and agile methods (Doctoral dissertation, Pontifical Catholic University of Rio Grande do Sul)*. 2012.

[26] Brown, D. M. *Communicating design: developing web site documentation for design and planning*. New Riders, 2010.

[27] Rojas, L. A. and Macías, J. A. Bridging the gap between information architecture analysis and software engineering in interactive web application development. *Science of Computer Programming*, 78, 11 (2013), 2282-2291.