Modelling Agile Requirements using Context-based Persona Stories

Jorge Sedeño[1], Eva-Maria Schön[2,3], Carlos Torrecilla-Salinas[4], Jörg Thomaschewski[5], Maria José Escalona[4] and Manuel Mejias[4]

[1]*Agencia Andaluza de Instituciones Culturales and University of Seville, Seville, Spain*
[2]*CGI, Hamburg, Germany*
[3]*University of Seville, Spain*
[4]*Department of Computer Languages and Systems, University of Seville, Seville, Spain*
[5]*University of Applied Sciences of Emden/Leer, Emden, Germany*

Keywords: User Involvement, User Experience Design, Human Computer Interaction, Assistive Technologies, Agility in IS Development, Model-driven Web Engineering, User Centered Design, Personas, User Stories, Agile Software Development.

Abstract: In recent years hybrid approaches focusing on user needs by integrating Agile methodologies (e.g. Scrum, Kanban or Extreme Programming) with Human-Centered Design (HCD) have proven to be particularly suitable for the development of Web systems. On the one hand, HCD techniques are used for requirements elicitation and, on the other hand, they can be utilized to elicit navigation relationships in Web projects. Navigation is one of the basic pillars of Web systems and also a fundamental element for the methodologies within the Model-Driven Web Engineering (MDWE) field. This paper presents an approach to model Agile requirements by means of integrating HCD techniques into Agile software development. We contribute to the software development body of knowledge by creating the concept of a Context-based Persona Story (CBPS) and formalizing it through a metamodel. Our approach covers the modelling of users and stakeholders by personas as well as the visualization of the context of use by storyboards. The attributes of the context of use enable us to elicit acceptance criteria for describing the scope of an Agile requirement.

1 INTRODUCTION

Agile approaches can be seen as an evolution of the previously existing iterative and incremental approaches, with the objective of providing organizations with tools to quickly adapt to changing requirements and also to ensure the early value-delivery of results to business. Agile is as a label grouping different frameworks and methodologies sharing a common set of principles and values that can be found in what is known as “Agile Manifesto” (Beck et al., 2001). Some examples of Agile approaches are Lean Software Development (Poppendieck and Poppendieck, 2003), Scrum (Sutherland and Schwaber, 2011) eXtreme Programming (XP) (Beck, 2000) or Kanban (Anderson, 2010). Besides, Web systems are those developed to be published and consumed on the Internet, being the subject of study of Web Engineering (Deshpande et al., 2002). This field can be defined as the systematic, quantifiable and structured application of methodological approaches to development, evaluation and maintenance of Web systems (Deshpande et al., 2002). The navigational model, which defines how users can navigate through the information, is a basic element in Web Engineering approaches (Valderas and Pelachano, 2011), especially in requirements elicitation and analysis phases.

Web-based systems are characterized, among other aspects, by flexible approach to requirements and quick user-feedback, in order to easily adapt and adjust to changing needs (Hu et al., 2008). Besides, Web-based development has also special characteristics that differentiate it from other type of development projects, such as: complex navigational structure (Escalona and Aragon, 2008), (Escalona et al., 2004) or quick adaptation to changing

Sedeño, J., Schön, E-M., Torrecilla-Salinas, C., Thomaschewski, J., Escalona, M. and Mejías, M.
Modelling Agile Requirements using Context-based Persona Stories.
DOI: 10.5220/0006220301960203
In *Proceedings of the 13th International Conference on Web Information Systems and Technologies (WEBIST 2017)*, pages 196-203
ISBN: 978-989-758-246-2

requirements (Mendes and Mosley, 2005) (Pressman, 2000) (Reifer, 2000). The latter is becoming a key success factor in Web-based systems. As it is known, one of the principles of Agile methodologies is embracing changes (Beck et al., 2001), thus Agile approaches might offer a suitable framework for the exposed Web development characteristics (Hu et al., 2008).

The classical approach to requirements is based on an up-front detailed requirements engineering phase. This kind of approach demands a stable environment, which is not often the case in Web projects, as requirements might have a fluidic scope (Mendes and Mosley, 2005) (Pressman, 2000). The incremental and iterative way of processing requirements that Agile approaches provide (Dyba et al., 2008) may better fit this particular case. There are examples, as the one described by Torrecilla-Salinas et al. (Torrecilla-Salinas et al., 2015) that shows successful application of Agile approaches to Web environments.

One of the main changes that Agile approaches bring is the move from a plan-driven deliverable centered approach to a value-driven one (Schön et al., 2017), in which human interactions become a relevant element, as the Agile manifesto (Beck et al., 2001) states. In this context, Agile proposes several approaches to requirements, among which user stories (Cohn, 2004) and personas (Cooper, 1999) (Maguire, 2013) are some of the most popular ones. On the one hand, user stories technique represents a way to express user needs in an "in-formal" way, which allows discovering the details of the requirement through interactions with customers or their representatives at the same time functionality is developed. On the other hand, persona technique depicts an imaginary person that will represent a certain target group of users. It is a common approach on Agile projects to refine user roles and user profiles. Several authors propose the combinations of these two techniques into what is called "Persona-driven user stories" (Winter et al., 2012), which tries to combine user needs and user profiling simultaneously.

Our paper goes a step forward to this approach, by including in addition to the "need" (user story) and "who" has this need (persona), the "context" where this need appears (Context of Use, ISO 9241-210, 2010). This can be defined as Context-based Persona Story (CBPS).

Based on the foregoing, our paper has the following goals:

- Define CBPS concept.
- Illustrate, by means of different proposed techniques, how CBPS can be elicited.
- Propose a metamodel to formalize the definition of CBPS.
- Draft meaningful conclusions and suggest further lines of research.

For this purpose, it is organized into the following sections: after this introduction, Section 2 will offer a view of the related work. Afterwards, Section 3 will describe, through an example, how to obtain CBPS and it will also present the proposed metamodel. Section 4 will define the limitations of the model, and finally Section 5 will draft the main conclusions of the paper and will propose further lines of research.

2 RELATED WORK

User stories have their origin in eXtreme Programing (XP) (Beck, 2000). They consist of three essential elements (Cohn, 2004) (Jeffries, 2001): the written part, which is used as a reminder to the requirement; the conversation around the story, which supports building shared understanding; and the acceptance criteria, which serve as boundaries for the scope of a story. During the last years, this established schema of user story has evolved. In the field of Human-Centered Agile Development (HCAD), different types of user stories exist that aim to describe requirements from users perspective considering their needs and motivations.

Cohn (Cohn, 2004) suggests replacing the role of story with persona. The benefits of the integration of persona are also discussed in Winter et al. (Winter et al., 2012) and Jeffries (Jeffries, 2001). Personas support the project team to gain common understanding concerning user and stakeholder, as well as their needs and behaviors by utilizing them as actors in user stories. The concept of persona stories is also discussed by Hudson (Hudson, 2013). He stresses that personas assist understanding how and when tasks are performed. In addition, he states that one of the main benefits is that their resulting scenarios and visual designs are prescriptive by means of doing user research rather than just guessing how users will interact with the system.

Näkki (Näkki et al., 2011) recommends using "needs-based user stories". They collect users everyday needs and challenges regarding a specific domain to create this type of stories. Furthermore, Näkki et al. involve users during requirements

elaboration, by commenting and rating features in order to allow prioritization.

Harbers (Harbers et al., 2015) introduces the concept of "value-based user stories", which are created during a "Value Story workshop", where different stakeholders participate. This contribution enables stakeholder values to be embedded into requirements.

Analyzing the existing work, we can conclude that there are multiple ways of including user and stakeholder needs into system development by using persona stories, needs-based user stories or value-based user stories. One common goal of these approaches is making Agile Software Development (ASD) more human-centric. However, one major gap they share is not considering that value or needs might vary, even from the same user type, depending on the context in which the story is executed. We can therefore conclude that the related work lacks in defining this context of use (ISO 9241-210, 2010), which usually plays an important role when it comes to estimating and prioritizing requirements. Taking into account the attributes of the context of use (users, tasks, equipment and physical and social environments in which a system is used) enables us to elicit the acceptance criteria of a persona story and also to give it a scope. To this end, and as main contribution, we introduce the concept of "Context-based Persona Stories" (CBPS). On the one hand, needs and values are covered by using personas; on the other hand, the context of use is comprised by means of a storyboard.

3 MODELING CONTEXT-BASED PERSONA STORY

Our proposed model will include all the specific goals of Web systems requirements engineering: identification of content requirements, identification of functional requirements in terms of navigation needs and businesses processes, and definition of interaction scenarios for different groups of Web users (Escalona and Koch, 2007).

As it has been mentioned, we propose the CBPS approach that will allow us to formalize the aforementioned objectives by taking users needs and stakeholders value as our starting point, and by utilizing, in a coordinated way, a set of Agile techniques that will ensure the involvement of users. In this section, we will present our proposal by introducing the application of persona technique to model user needs and values and explaining how to identify the context of use by means of storyboards. Later, we will describe the formalization of CBPS and finally we will introduce a metamodel representing CBPS Agile requirements.

The next sections will illustrate an example of how to identify the modelled CBPS by means of different Agile techniques.

3.1 Modelling User Needs and Stakeholder Values

Personas are used as representatives of real users during system development in HCD (Cooper, 1999), (Pruitt and Adlin, 2010). They became an established artefact in the Agile community during the past years. Persona specifically describes a potential user of the system to be developed and it represents a larger part of the target group. In addition to modelling users through personas, we can also use this technique for modelling stakeholders. One of their main contributions to system development is enabling project members to generate an empathetic focus on user needs and values (Pruitt and Adlin, 2010). It allows preventing self-referential design, where project members lead their own needs and motivations to the user.

Figure 1 presents an example of persona. It shows how a representative of the Andalusian civil servant community is modelled by deeply identifying his character, behaviour and motivations. Additionally, elements like the way the application is consumed (e.g. What devices does it use? When and how often it is used?) are included in the definition of persona.

3.2 Modelling Context of Use

Storyboards can be used to visualize the workflow and user-system interaction (Truong et al., 2006). They consist of a sequence of pictures, which show significant steps of the workflow. In particular, they support visualizing the conditions of Web projects (e.g. light conditions and connection to the Internet by means of connection type and bandwidth) and they can be linked to one or more personas. Thus, storyboards can be used to model the context of use as defined by ISO (ISO 9241-210, 2010). This context of use is composed of users, tasks, equipment (hardware, software and materials), and physical and social environments in which the product is used.

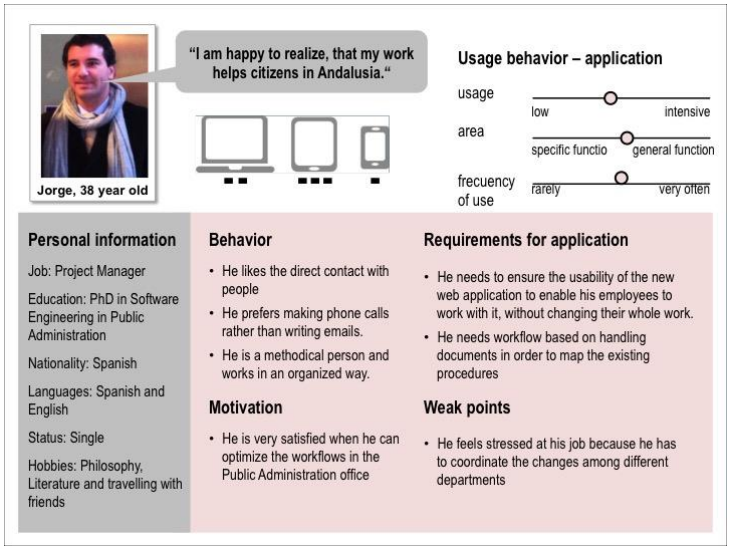

Figure 1: Example of a persona providing photo, personal information (e.g. name, age or nationality, among others) and data regarding user behavior, motivation and pain points.

Figure 2 shows an example of a storyboard named "requesting a sight visit", which shows the workflow associated with how a citizen of Andalusia gets permission to a sight. On the one hand, it displays the users who are involved in the process and the interactions among them; on the other hand, it shows how to communicate with the software they use through a storyboard. As seen, the already defined personas are the actors of our story in the storyboard with the aim of representing graphically the context where the application is used.

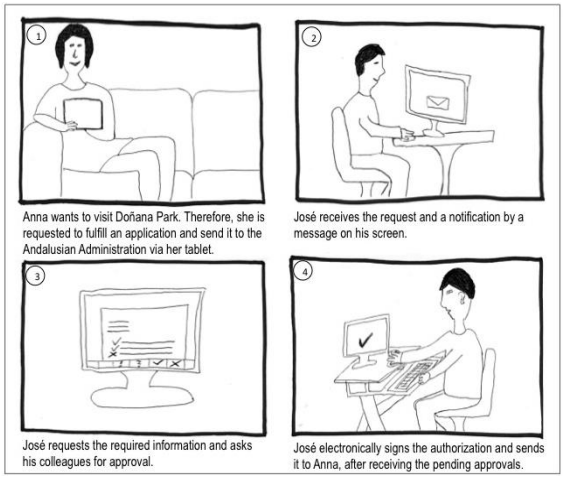

Figure 2: Example of a storyboard in the area of E-government describing the workflow to get permission for a tourist visit.

3.3 Creating Context-based Persona Story

In Agile Software Development user stories are utilized in order to describe the functional requirements of a system from users viewpoint, according to a specific format as shown below:

As a <role>, I want < feature> so that I can achieve <goal> (1)

As discussed above, this formal representation of a user story lacks in defining a specific user, his/her needs and the environment in which the specified feature of the story is applied. To this end, we recommend using CBPS instead of this well-known format. Firstly, we can model users, stakeholders and their needs by including the information given by our already defined personas. Secondly, we can take into account the context of use represented by our storyboards.

Our metamodel (see Figure 4) offers support by means of formalizing CBPS as follows:

<Persona> in the <Context of Use>, wants to achieve <Goal>, so that <benefit> (2)

Based on the information provided by personas and storyboards, we can create these CBPS (as Figure 3 shows). This approach allows connecting

the functionality described by different CBPS to each other and we can declare how interaction among different personas takes place in order to achieve one common business objective ("reduce processing time of a case").

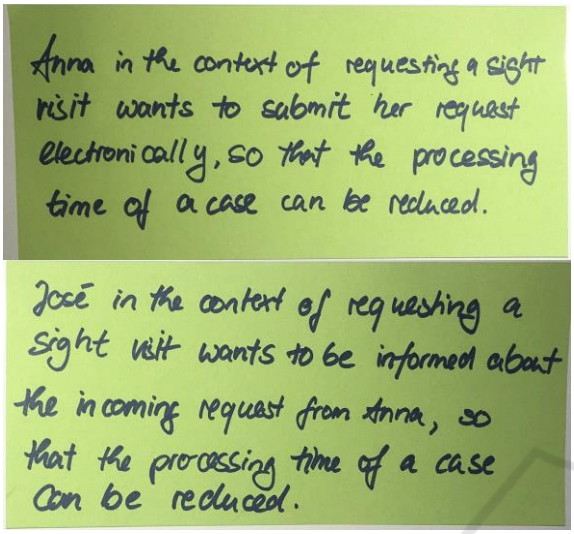

Figure 3: Example of CBPS.

3.4 Metamodel for a Context-based Persona Story

The aforementioned objectives in section 3 can be formalized by an UML metamodel representing an Agile requirement, which is shown in Figure 4. It describes the relationships between the main Agile artifacts and includes the key elements and relationships of a CBPS.

As Figure 4 displays, three key entities can be found in the metamodel: *"User Story"*, *"Persona"* and *"Navigation Relationships"*.

Together with the key entities, the metamodel will include the minimum and mandatory set of attributes that will allow a real understanding of the modeled Agile requirement. The main goal is to obtain a set of requirements as much homogeneous and complete as possible at a certain moment of the development process. Our metamodel enables us to define the meaning of homogeneity and completeness by means of a set of attributes. Among them, we will now highlight the following ones:

- From the ones related to "*User Story*" entity, we would like to point to those like *business value*, *size* and *ROI* (*Return Of Investment*) that will help order and prioritize the list of Agile requirements. "*User Story*" entity describes a value-based perspective of a particular user need by means of the ROI attribute. This attribute is calculated as the result of dividing Story *business value* (the value it brings to the business, given by the business representatives) by Story *size* (how much building the story will cost, indicated by the development team). We can also find "*Acceptance Criteria*" and "*Definition of Done*" entities, linked to "*User Story*". They will allow better defining the boundaries of each user need in order to narrow its scope.
- From the ones related to "*Persona*" entity, we would like to refer to those making possible the differentiation among the diverse types of users and stakeholders and the relationships between them and the identified user stories (like *motivations* or *personal information*). The metamodel also includes the mandatory attributes that describe in-depth "*Context of Use*", defining the environmental aspects (e.g. physical or social environment in which the functional need is present). Linked to "*Persona*" entity, we can also find "*UI Graphic Elements*" entity, which will help better describe the functionality contained in user stories. This supports a formal description of the navigation within the system, expressed as an interaction of users (represented by "*Persona*" entity) with their needs (represented by "*User Story*" entity).
- From the ones related to *"Navigation Relationship*" entity, we would like to remark those associated with the navigational model. These attributes represent the interactions between personas and user interface graphical elements (as wireframes or mock-ups) to materialize a particular User Story.

As it can be seen, the metamodel has helped us cover the above-mentioned three goals of Web Systems requirements elicitation phase: identification of content requirements (by means of *"User Story"*, *"Acceptance Criteria"* and *"Definition of Done"* entities), identification of functional and non-functional requirements in terms of navigation needs and businesses processes (by means of *"Navigational Relationships"* and *"UI Graphic Element"*, and their relationships with *"User Story"* entity), and definition of interaction scenarios for different groups of Web users (by means of *"Persona"* and *"Context of Use"* entities).

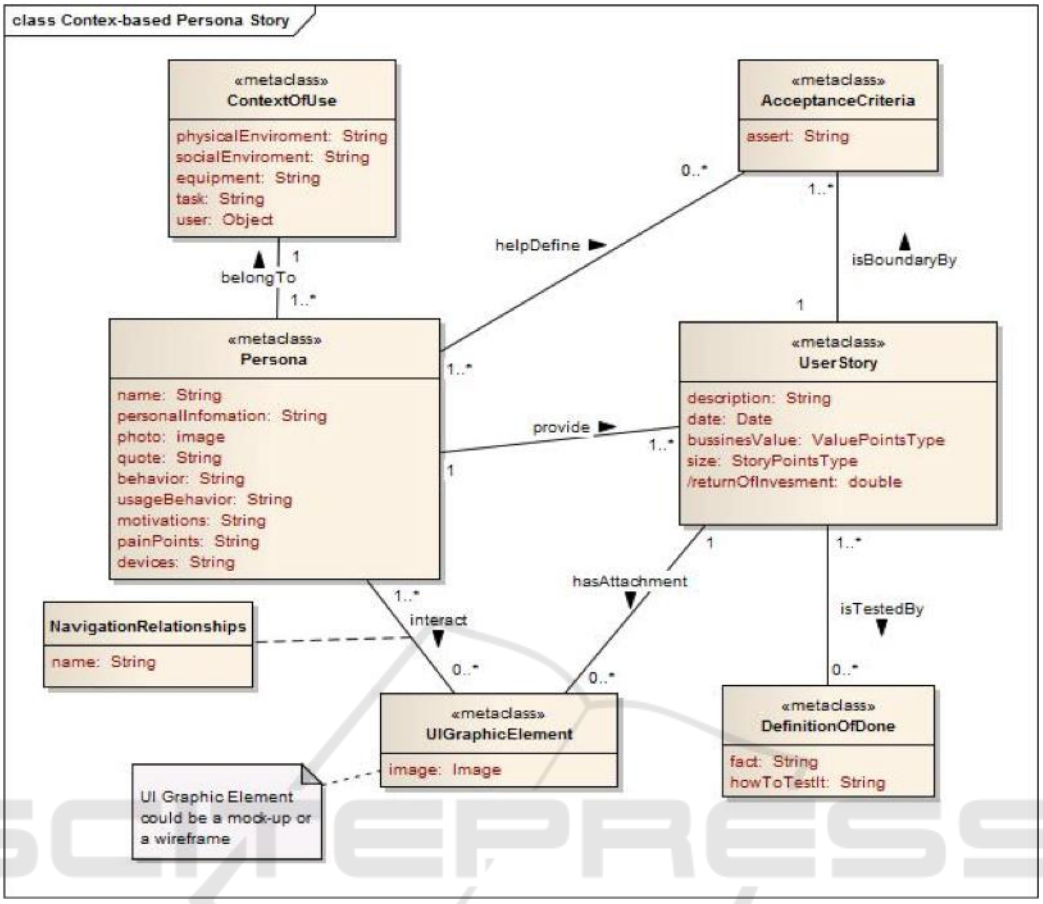

Figure 4: Context-based Persona Story Metamodel.

3.5 An Example in the Industry

It was in between 2008-2012 that the Regional Ministry of Culture of Andalusia, Spain, (Junta de Andalucía, España) transformed itself into an organization able to operate under SOA paradigm, but it was in between 2012-2015 that it started using an own Agile methodology (Torrecilla et al., 2015). However, there was something left: the integration of an activity in the early stages of ASD where executing the process of discovering services that cover the requirements represented by User Stories. This problem is related to software reusability in early stages of development of the lifecycle. In light of this, a process that consisted of five steps was proposed, as shown below:

- Indexing the Services Portfolio.
- Formalizing the Agile requirement (Figure 4).
- Transforming this formalization into a query.
- Launching this query against this index.
- Obtaining the Services that could cover this Agile requirement.

This is an example of how the use of CBPS approach could connect Agile requirements elicitation to other fields of software engineering.

4 LIMITATIONS AND FURTHER DISCUSSION

This section will present some limitations and discussions the proposed model poses in order to clarify when and how it can be used.

Regarding drawbacks, we have identified a list of criteria that should be present to define good CBPS. This is what we have named as “must-have criteria”, that is to say, their presence or absence conditions that make the definition of CBPS succeed or fail.

As it has been stated in this paper, our approach mainly depends on Agile and HCD techniques (by means of techniques like persona, storyboards and user stories). Thus, the use of CBPS will require a HCAD environment, which in turn will demand full

users involvement, as it will clarify notions like context of use (stakeholders, not only users of the systems, but also people interested in outputs and outcomes, can help clarify this context). Finally, our approach is also conceived in terms of Web Systems, as it includes elements like navigation relationships that may not make too much sense in other type of software development projects.

In addition to "must-have criteria", we have identified some other criteria that are very useful to have, although they are not mandatory to define successfully CBPS. We have named them as "nice-to-have criteria". This latter category includes the participation of stakeholders. Table 1 summarizes both "must-have" and "nice-to-have" criteria:

Table 1: Must-have and nice-to-have criteria.

Criteria	Must-have	Nice-to-have
Agile Environment	**X**	
User involvement	**X**	
Web System	**X**	
User interaction with UI techniques	**X**	
HCD techniques	**X**	
Involvement of stakeholders		**X**

Even though the model has some limitations, expressed by the described criteria, we can also highlight that CBPS might be useful to initially populate the Product Backlog of a Web Project in what is known as "Sprint-0" or "Iteration-0" (Torrecilla-Salinas et al., 2015). In this case, the Product Backlog might be an ordered list of CBPS. This way of defining Agile requirements might allow a systematic definition of roles, user needs and scenarios of usage (by means of a gathering personas, user stories and storyboards) without a significant increase in the overhead of the development process (in the cost/benefit ratio).

Once the Product Backlog is initially defined, the technique should be used along "Product Backlog grooming" sessions (Cohn, 2009), as well. The idea is that those sessions will review and enrich the process of gathering personas, storyboards and user stories, by adding, modifying or deleting elements so as to keep the Product Backlog always updated.

5 CONCLUSIONS AND FUTURE WORK

The main contribution of this paper is the definition of the concept CBPS. Together with this new concept, its formalization through a metamodel allows linking it to different fields of Software Engineering. A CBPS can be utilized as a way to formalize Agile requirements in Web projects. Along this paper, we have shown how user and stakeholder needs can be modeled by personas as well as how the context of use can be visualized by storyboards. These artifacts enable us to create this new type of user stories in which all the relevant elements and its interactions have been presented. Furthermore, we have provided a real life example of our artifacts related to a Web project in the area of E-government. Finally, we have discussed the limitations of our approach, establishing the different criteria that should be required in order to successfully apply the technique. Together with the restrictions identified, we have also pointed out that the proposed model will need more empirical validation, which could be fulfilled in future work.

As it can be explained, formalizing an Agile requirement by means of a CBPS provides an important link between Agile approaches and other Software Engineering fields that might benefit from the HCAD approach proposed in our work.

An example of this link, and also a first future line of research, is Model-Driven Web Engineering, which could take advantage of the use of the presented techniques in order to define formally requirements in an Agile way, that could lead to the automatic generation of a "draft Product Backlog" to be implemented during the development project.

Additionally, as a second future line of research, this approach can also be useful for those researchers and practitioners that work on connecting Agile to maturity models like those of CMMI family (Torrecilla-Salinas et al., 2016). The idea of formalizing requirements in an Agile way without a significant increasing overhead might help in the process of institutionalizing Agile without loosing agility.

As a third and last future line of research, the proposed model might be an initial step in order to associate Agile requirements engineering with Human-Centered Design, with the goal of making Agile Software Development more human-centric.

Finally, it must be stated that the elicitation of "Agile UI techniques" could be an interesting research topic derived from our approach.

ACKNOWLEDGEMENTS

This research has been supported by the MeGUS project (TIN2013-46928-C3-3-R), Pololas project (TIN2016-76956-C3-2-R) and by the SoftPLM Network (TIN2015-71938-REDT) of the Spanish Ministry of Economy and Competitiveness.

REFERENCES

Anderson, D.J., 2010. *Kanban - Successful Evolutionary Change for your Technology Business*, Blue Hole Press.

Beck, K. et al., 2001. Manifesto for Agile Software Development. *2001. http://www.agilemanifesto.org. Accessed April 1, 2016.*

Beck, K., 2000. *Extreme Programming Explained: Embrace Change*, Boston: Addison-Wesley.

Cohn, M., 2009. *Succeeding with Agile Using Scrum*, Boston: Addison-Wesley.

Cohn, M., 2004. *User Stories Applied: For Agile Software Development*, Boston: Addison-Wesley.

Cooper, A., 1999. *The inmates are running the asylum*, Sams. Indianapolis, USA.

Deshpande, Y., Marugesan, S., Ginige, A., Hanse, S., Schawabe, D., Gaedke, M., White, B., 2002. Web Engineering. *Journal of Web Engineering Vol. 1 Nº 1, pp. 3-17.*

Dybå, T., et al., 2008 Empirical Studies of Agile Software Development: A Systematic Review. *Inf. Software Technology 50 (9-10), 833-859.*

Escalona, M.J., Aragón, G., 2008. NDT: A Model-Driven Approach for Web requirements. *IEEE Transactions on Software Engineering, 34 (3), 370-390*.

Escalona, M.J., Koch, N., 2007. Metamodeling the Requirements of Web Systems. *Web Information System and Technologies, pp. 267–280, Springer, Heidelberg*.

Escalona, M.J., Mejías, M., Torres, J., 2004. Developing systems with NDT & NDT-Tool. *In: Proceeding of 13th International conference on information systems development: methods and tools, theory and practice, pp. 149-159. Vilna, Lithuania*.

Harbers, M., Detweiler, C., Neerincx, M.A., 2015. Embedding Stakeholder Values in the Requirements Engineering Process. *In: Proceedings of 21st International Working Conference, pp. 318-332. Germany.*

Hu, R., Wang, Z., Hu, J., Xu, J., Xie, J., 2008. Agile Web Development with Web Framework. *In: Proceedings of 4th International Conference on Wireless Communications, Networking and Mobile Computing. IEEE, Dalian, China.*

Hudson, W., 2013 *User stories don't help users*, Interactions, pp. 50-53.

ISO: ISO 9241-210:2010 - *Ergonomics of human-system interaction* - Part 210: Human-centred design for interactive systems.

Jeffries, R., 2001. *Essential XP: Card, Conversation, Confirmation*. http://ronjeffries.com/xprog/articles/expcardconversationconfirmation/. Accessed April 1, 2016.

Maguire, M., 2013. Using human factors standards to support user experience and agile design. *In: Proceedings of the International Conference, UAHCI 2013, Held as Part of HCI International 2013. Las Vegas, USA.*

Mendes, E., Mosley, N., 2005. *Web Cost Estimation: An Introduction. Web Engineering: Principles and Techniques, pp 182-202*, IGI Global.

Näkki, P., Koskela, K., Pikkarainen, M., 2011. Practical model for user-driven innovation in agile software development. *In Proceedings of the 17th International Conference on Concurrent Enterprising, pp. 1-8. Germany*.

Poppendieck, M., Poppendieck, T., 2003. *Lean Software Development. An Agile Toolkit*, Boston: Addison-Wesley.

Pressman, R.S., 2000. What a Tangled Web We Weave. *IEEE Software (1-2), pp 18-21*.

Pruitt, J., Adlin, T., 2010. *The persona lifecycle: keeping people in mind throughout product design*, Morgan Kaufmann.

Reifer, D.J., 2000. Web Development: Estimating quick-to-market software. *IEEE Software (1-2), pp. 57-64.*

Schön, E.-M., Thomaschewski J., Escalona, M. J., 2017. Agile Requirements Engineering: A systematic literature review. *Computer Standards & Interfaces (49). pp. 79–91*.

Sutherland, J., Schwaber, K., 2011. *The Scrum Guide: The Definitive Guide to Scrum*, http://www.scrum.org/Scrum-Guides. Accessed April 1, 2016.

Torrecilla Salinas, C.J., Sedeño, J., Escalona, M.J., Mejías, M., 2015. Estimating, planning and managing Agile Web development projects under a value-based perspective. *Information and Software Technology 61, 124–144* .

Torrecilla Salinas, C.J., Sedeño, J., Escalona, M.J., Mejías, M., 2016. Agile, Web Engineering and Capability Maturity Model Integration: A systematic literature review. *Information and Soft. Technology 71, 92-107.*

Truong, K.N., Hayes, G.R., Abowd, G.D.: Storyboarding: An Empirical Determination of Best Practices and Effective Guidelines. *In: Proceedings of the 6th Conference on Designing Interactive Systems, pp. 12-21, New York, USA (2006).*

Valderas, P., Pelechano, V., 2011. A Survey of Requirements Specification in Model-Driven Development of Web Applications. *ACM Trans. Web 5 (2), 10*.

Winter, D., Holt, E.M., Thomaschewski, J., 2012. Persona driven agile development. Build up a vision with personas, sketches and persona driven user stories. *In: Proceedings of the 7th Conference on Information Systems and Technologies. Madrid, Spain*.