Using i* Models to Enrich User Stories

Aline Jaqueira[1], Márcia Lucena[1], Eduardo Aranha[1],
Fernanda Alencar[2] and Jaelson Castro[3]

[1]Departamento de Informática e Matemática Aplicada – UFRN
alinejaqueira@ppgsc.ufrn.br, {marciaj, eduardo}@dimap.ufrn.br
[2]Departamento de Eletrônica e Sistemas - UFPE
fernanda.ralencar@ufpe.br
[3]Centro de Informática - UFPE
jbc@cin.ufpe.br

Abstract. In agile methods the user stories are widely used to describe requirements. However, the user stories are an artifact too narrow to represent and detail the requirements. Issues like software context and dependencies between stories are also limited with the use of only this artifact. The lack of documentation in agile development environment is identified as one of the main challenges of the methodology. This work proposes the use of i* model that aims to reduce this lack of existing documentation in agile methods. We propose a set of heuristics to perform the mapping of the requirements presented as user stories in i* models. The i* models are used as a form of documentation in agile environment, thus the user stories can be viewed more broadly and with their proper relationships according to the business environment that they will meet.

Keywords: i* Models, User Stories, Agile Requirements

1 Introduction

Agile methods have been gaining much interest among practitioners and researchers [1], because they use more simplified processes with less bureaucratic activities associated with the development [2]. However, requirements engineering and agile development are often seen as incompatible activities [3], as requirements engineering is a traditional process of software engineering and has the documentation to manage the project and share knowledge, while the agile methods focus on face-to-face collaboration among stakeholders to address the same goals.

In addition, the elicitation is held with clients that are part of the development team. In order to do this, customers write stories according to the system needs to do (user stories) and prioritize them according to the value of the concerned business. A user story describes the functionality that is valuable to the customer and is used for project planning, acting as a reminder to the team since subsequent conversations about the story are essential to convey the details [4]. User stories are used to establish a common understanding of the software requirements using a flexible approach, low overhead and focusing on the user [5]. They are widely used by agile development [6] and are therefore considered in this work as an agile artifact.

The requirements documentation is seen as a bureaucratic activity in the Agile methods [2], but their absence is singled out as one of the main challenges of the methodology [8]. In addition, the user stories are very limited artifacts to process issues such as progress tracking software and that there is a lack of detailed information about software in development [9]. The control of the system performed only by way of user stories is a challenge both for customers and for the team. To make decisions based only on the stories, without any documentation, becomes risky especially for complex systems [14].

This work proposes the use of i* model that aims to reduce this lack of existing documentation in agile methods. We propose a set of heuristics to perform the mapping of the requirements presented as user stories in i* models. The i* models are used as a form of documentation in agile environment, thus the user stories can be viewed more broadly and with their proper relationships according to the business environment that they will meet.

In the following section, briefly define the research objectives. In Section 3 we discuss the scientific contributions. Section 4 provides the conclusions and Section 5 presents future work.

2 Objectives of the research

This work proposes the use of i* models as an additional resource that aims to reduce the lack of documentation of agile software development projects and improve view dependencies between user stories and the system and also provide information and understanding of the system as a whole. We seek to support the stakeholders, through models i* providing a graphical and comprehensive vision of the user stories and their relationships.

The common concern with stakeholders justifies the choice of the technique i* to represent the user stories. The focus on stakeholders and their relationships to the description of requirements is a feature of the technique i*, where actors depend on each other to achieve their goals. The agile methods also focus on human factors and bet in delivering value to the customer [10], recognizing people as fundamental part of the project success [11]. Therefore, the concepts of user stories are employed together with the concepts of the i* model [12].

Although user stories are written in natural language by the client, Mike Cohn suggests a format for writing the stories that has been widely used: "as <role>I want <action>to <goal> " [7]. Therefore, for this proposal it is assumed that the same format proposed by Cohn [7] will be used.

The user stories are mapped to i* models generating a graphical and comprehensive vision of the software requirements and their relationships. The concepts and notations of the i* model are used according to i* Wiki [13], that represents a simplified version of the technique. It is also important to report that only a few elements of i* are used in accordance with the need of the proposal. In this way, to build the SD model, the elements used are the actor, the goals and the IS_A Association. For SR model the tasks, resources, and also the connection of decomposition are used.

By using i * models from user stories, agility feature of the projects should be maintained because the view of stories complemented by i * models can make better understanding, easier and simpler. The following presents the correspondence of elements when mapping the user stories to i* model: (i) Role in User Story is mapping to Actor in i* model; (ii) Action in User Story is mapping to Task in i* model; and (iii) Goal in User Story is mapping to Goal in i* model.

To simplify the understanding of the mapping, heuristics were established as a resource to arrive at the solution of the mapping proposed in this paper. It is noteworthy that the heuristics set must be executed in the order they were made for the mapping occur in a more objective. The heuristics to map user stories to SD model are: SD-H1: Create the Actor System; SD-H2: Create an Actor in i * model for each different role of user stories; SD-H3: Create a meta in i* model for each goal of user stories. If repeated the same goals will be defined only once in the model; SD-H4: If repeated goals for different actors, create a Generic Actor; SD-H4.1: Create a relationship is_a the Actor generic to other specific actors who share the same goal; SD-H5: Connect the dependencies of each actor with their goals.The heuristics used to map user stories for the SR model are: SR-H1: Create a Task within the Actor System for each share of user stories; SR-H2: If there are different actions for the same goal, to create a generic task; SR-H2.1: Decomposing the generic task into sub tasks that represent the actions associated with the same goal; SR-H3: List the dependencies of each goal with the corresponding tasks according to user stories; SR-H4: If there are tasks that depend on own Actor that are related, generate a resource with the name of the task; SR-H5: Relate the resource depending on the Actor.

According to the proposed heuristics, the SD model is mapped by first creating the System actor. Subsequently, it creates the actors for the roles of user stories, which, in this case, the actors are User and Administrator. The goals of the user stories are created as goals for the SD model and linked as dependencies leaving from the actors that are associated and coming to System Actor. We omitted the result of heuristics this example for SD model mapping (from user stories to i* SD model) due to lack of space.

To demonstrate this proposal a login system was used as an example, considering the prospect of a user and an administrator. Table 1 presents the stories of user login system and the figure 1 shows the result of the mapping.

Table 1. User stories Login System (Source: IBM, 2012)

	Role	Action	Goal
1	User	Having username	Access secure content
2	User	Having password	Access secure content
3	User	Choose your username	Customize account
4	User	Change the default password	Customize password
5	Administrator	Assign the user password	Automated registration
6	Administrator	Send email registry	Confirm the account activation email
7	Administrator	Request to login user	Ensuring security of con-

			tent
8	User	Register password reminder	Remember the password
9	Administrator	Request password reminder	Confirm user

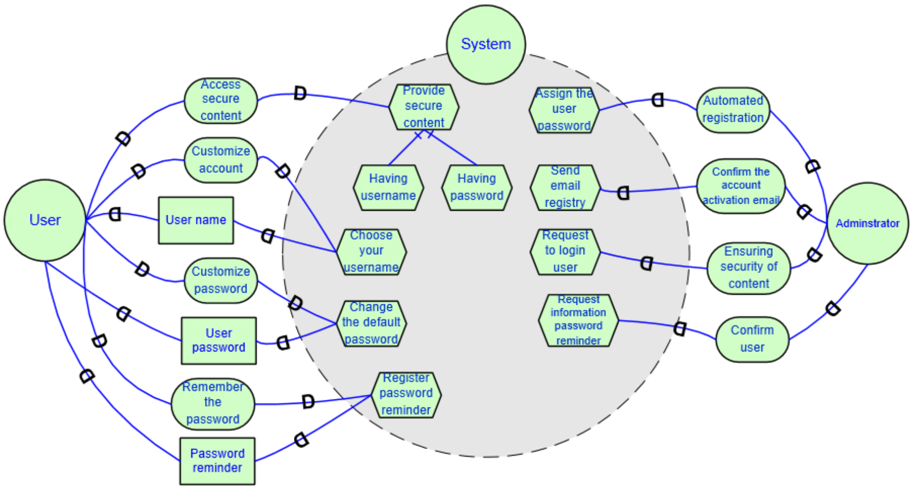

Fig. 1. Model SR in the sample application

To generate the SR model, every action of the user stories have been generated as a task within the system actor, once it is the system actor that will operate them, performing the task in a particular manner in order to meet the goals of the actors. As there are, in this example, different actions for the same goal, a generic task was created in SR model which was decomposed in the actions in the form of sub-tasks. The tasks that depend on the actor himself generate a resource that depends on the actor and that has the same name as the task.

3 Scientific contributions

Even in an agile environment it is necessary to develop some models before any implementation to ensure a shared understanding by the development team, so that it is synchronized with the goals of the business value and context of the project [14]. Visual models assist in understanding of how users will need to use the system. In addition, those models are effective for the stakeholders to understand the proposed solution and also to keep them interested and involved.

The most important contribution of this work is the development of a proposal that uses visual models provided by i* to alleviate the lack of documentation in agile development environments that was cited in the systematic review conducted by Jaqueira et al. [8]. A set of heuristics is proposed to perform the mapping of user stories for i* models.

Since i * models are graphical representations of the requirements, they would have a comprehensive and visual way to see the user stories, thus mitigating the lack of documentation in agile software development projects, improving the visualization of the system context, facilitating access to requirements, contributing to decision-making in the development environment.

In addition, other contributions may be cited as the improvement in understanding the context of the system to be developed, the use and easier access to information of user stories from the preview of the visual model, improving the decision-making process in accordance with the analysis of the stories, as they are described in i* models, and tooling support enabling the automation of some of its stages (construction of SD and SR models).

4 Conclusions

To evaluate the approach this work, we performed a case study analyzing qualitative issues. From the participants' impressions of use, it was found that the use of i * models contribute to complement user stories [16]. An approach based on visual models can provide more direct and traceable links to system development, promoting an analysis with the greatest impact in the design and implementation of software. It also works to facilitate communication, understanding, and detecting problems or explore what-if scenarios and potential solutions.

We find that when viewing models, errors and/or neglect could be more easily recognized [15]. All this facilitates the process of analysis and discussion of the system to be developed.

From the mapping of the user stories to SD and SR i* models, we can organize and represent all the stories in a model that provides an overview of the stories and their relationships. In addition, all the stories of the same actor were presented in the same model, allowing to find them more easily. In this way, it is possible to understand the context of the system, its main actors and their goals.

Viewing through the models makes it easier to identify dependencies between user stories and the identification of system tasks to meet each specific actor involved with the software. Thus, it is possible to notice that the use of i* models enriches the user stories to provide a better view (broader and general); to enable showing the dependencies between the stories, contributing to a better understanding of the context of the system to be developed; to provide visualization of system tasks associated with the goals of each actor; to allow the recognition of possible errors or negligence in the stories. Therefore, this work proposes a form of documentation of the requirements on agile development, thus a visual artifact will be provided supplementing the user stories allowing analysis, communication, discussion and better understanding of the system.

5 Ongoing and future work

In order to continue the research for this work a few suggestions of further work can be cited. The development of a tool to perform the transformation of user stories the format Mike Cohn [7] used in this work, in order to use this proposal to all user stories. The development of a tool for the purpose of this work in order to fully automate the mapping of user stories for i* models. The treatment of scalability for the system actor. Identify and treat the relationships and connections between tasks in System Actor. Furthermore, the development of guidelines to perform the mapping back from i* models to user stories.

References

1. Cao, L. and Ramesh, B., Requirements Engineering Practices: An Empirical Study, IEEE Software, Volume: 25, Issue: 1. page(s): 60-67 (2008).
2. Fowler, M. The new methodology. Disponível em: <http://www.martinfowler.com/articles/newMethodology.html>. Acesso em 03/12/2011.
3. Paetsch, F., Eberlein, A. and Maurer, F. Requirements Engineering and Agile Software Development, Proc.12th IEEE Int'l Workshops Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE 03), IEEE CS Press, 2003, pp. 308–313.
4. Cohn, M.: User Stories Applied: For Agile Software Development (The Addison-Wesley Signature Series), March. Addison-Wesley Professional, Reading (2004).
5. O'hEocha, C., & Conboy, K. (2010). The role of the user story agile practice in innovation. In P. Abrahamsson & N. Oza (Eds.), Lean enterprise software and systems (pp. 20-30). New York: Springer.
6. Cockburn, A. (2007). Agile Software Development: The Cooperative Game. Boston, Pearson.
7. Cohn, M. Agile Estimating and Planning. Prentice Hall, 2006.
8. Jaqueira A., Andreotti, E., Lucena, M., Aranha, E. Desafios de Requisitos em Métodos Ágeis: Uma Revisão Sistemática 3rd WBMA, São Paulo, 2012.
9. Sharp, H., Robinson, H. Segal, J. and Furniss, D. (2006) 'The Role of Story Cards and the Wall in XP teams: a distributed cognition perspective', Proceedings of Agile 2006, IEEE Computer Society Press, pp65-75.
10. Bassi, D. Experiências com desenvolvimento ágil, Dissertação de Mestrado, Universidade de São Paulo, 2008.
11. Highsmith, J. and Cockburn, A., "Agile Software Development: The Business of Innovation," Computer, vol. 34, pp. 120-122, 2001.
12. Yu, E. "Modelling Strategic Relationships for Process Reengineering". PhD thesis.University of Toronto, Department of Computer Science. 1995.
13. i* Wiki Home, Disponível em <http://istar.rwth-aachen.de/tiki-index.php> Acesso em 10/08/2012.
14. Beatty, J. e Chen, A. Visual Models for Software Requirements. Washington, Microsoft Press: 2012.
15. Horkoff, J., Yu, E.: A Qualitative, Interactive Evaluation Procedure for Goal- and Agent-Oriented Models. In: CAiSE Forum. CEUR Workshop Proceedings (2009).
16. Jaqueira, A.: Uso de Modelos i* para Enriquecer Requisitos em Métodos Ágeis. Dissertação de Mestrado. Departamento de Informática e Matemática Aplicada – UFRN. Natal (2013)