Integrating User Stories and i* Models
A Systematic Approach

Aline Jaqueira[1], Márcia Lucena[1], Fernanda Alencar[2,3], Celso Agra[3] and Eduardo Aranha[1]
[1]*Departamento de Informática e Matemática Aplicada, UFRN, Natal, Brazil*
[2]*Departamento de Eletrônica e Sistemas, UFPE, Recife, Brazil*
[3]*Programa de Pós-Graduação em Engenharia da Computação, UPE, Recife, Brazil*

Keywords: Agile Requirements, User Stories, i* Models.

Abstract: User stories are a common way to describe requirements in Agile methods. However, the use of user stories is restricted, since they offer only a limited view of the whole system. In contrast, one of the features of the i* framework is provides a visual representation of the actors involved in a system and the goals that are to be met. This allows for a better understanding of the problem as well as for a better overview and evaluation of alternative solutions. In addition, i* models consider the early phases of requirements engineering, while user stories cover the later phases. In this context, this paper presents an approach to map user stories to i* models and vice versa, aiming at providing a bigger picture of the system as a whole. A case study to evaluate this work is also presented, suggesting the viability of the approach.

1 INTRODUCTION

In Agile methodology, the development of a software project starts by creating requirements statements to express what the system should do. But in some projects only with the user stories there is little understanding of why the system is needed (Sharp, 2006). Thus, these requirements statements are limited in providing a general understanding of the system, and the goals of a software project are not clearly defined.

Within the context of requirements engineering, the goal-oriented approach GORE (Lamsweerde, 2001) uses the organizational goals of stakeholders to elicit the requirements of the software. The i* framework (Yu, 1995) is a GORE approach that provides a graphical view of the actors involved in the system with their dependency relations. Also, this framework can easily be related to other requirements engineering artifacts, such as Goal Decomposition Analysis (Liaskos et al., 2012), deriving i* models into UML diagrams (Santander and Castro, 2002) (Martinez et al., 2002), tranformating of i* models into matrices (Laue, 2011), and also using of Agent-Oriented Conceptual Modeling with Tropos methodology (Bhuiyan et al., 2007). The choice of a well-suited framework to represent organizational requirements and to capture initial requirements lead us to focus on the i* model. The i* framework supports the understanding of stakeholders' intentions and captures why the software is developed (Bresciani et al., 2004). On the other hand, user stories represent what and how a functional requirement is developed (Cohn, 2004), but there is a lack of identification of non-functional requirements and the rationale in the i* model.

According to Yu (1995), the process of conceptualizing an initial idea, called early phase, assists in understanding and developing clear, concise, and well-defined requirements during the elicitation phase, called late phase. This paper presents an approach to map user stories and i* models to each other, as a way to connect the two phases of requirements engineering. This provides a visual representation of the user stories and the system's context as a whole in an Agile development environment. The i* models cover the early phase while the user stories cover the late phase. The goal is to integrate i* models with user stories to reap the benefits that the requirements technique brings to the development environment. For instance, i* models are suitable for analysis of conflicting requirements. Mapping from i* models to user stories is important because some details captured by the i* framework can be added as non-functional requirements (softgoals). Additionally,

Lucena M., Agra C., Alencar F., Aranha E. and Jaqueira A..
Integrating User Stories and i* Models - A Systematic Approach.
DOI: 10.5220/0005400904280438
In *Proceedings of the 17th International Conference on Enterprise Information Systems* (ICEIS-2015), pages 428-438
ISBN: 978-989-758-097-0

user stories are best understood in a language that users and stakeholders are familiar with. They work to facilitate communication and understanding as well as the exploration of other possible scenarios and potential solutions. Moreover, GORE´s semantic, allows the modelling of intentions and their dependencies, however, these approaches are not commonly accepted by industry. On the other hand, user stories are widely accepted in industry due the their simplicity. Thus, the main objective of this paper is to integrate these two approaches to introduce GORE approaches to industry and enrich user stories with GORE elements, such as non functional requirements. At the end, we performed a simple case study of our approach, which is presented in this paper along with its assessment.

The remainder of this paper is organized as follows. Section 2 presents the requirement activity in Agile methods. In Section 3, the i* framework is presented. Section 4 presents the approach of this paper and an example of using the approach. Section 5 presents the case study to analyze the approach as well as the results of the assessment. Section 6 presents the related work and finally Section 7 shows conclusions and future work.

2 REQUIREMENTS IN AGILE METHODS

To elicit software requirements in Agile methods, teams use abstractions in the form of user stories. The user stories are used to meet the requirements using a user-focused approach. They are written in an incremental process, according to the priorities of the client. Although user stories are not universally accepted as the best way to capture software requirements (Cockburn, 2007), they are widely used by agile development and are therefore considered this paper as an agile technique. According to Cohn (Cohn, 2004), a user story describes the functionality that has value for the client. There are three parts to consider when writing a good user story. The first one is to identify who will perform an action. The second one is related to a functionality that will be executed. The last one is related to the reason (why the user needs this functionality). Cohn (Cohn, 2006) suggests a format for the writing of user stories that has been used in practice: “As <role>, I want <action> for <goal>”. Therefore, in this work, we use user stories in this format and we consider as artefact produced in the environment agile development.

By thoroughly analyzing the activities of an Agile team, Sharp et al., (2006) concluded that user stories are very limited artifacts in showing detailed information about the software under development. For instance, dependencies among the stories are omitted or not fully understood, making it difficult to obtain a global understanding of the system.

3 i* FRAMEWORK

In Goal-Oriented Requirements Engineering (GORE) (Lamsweerde, 2001), the goals of the stakeholders are used to elicit, elaborate, structure, specify, analyze, negotiate, document, and modify requirements

The i* model (Yu, 1995) is one framework used in the GORE approach. Its focus is on the intentional properties of the actors and their relationships. This framework provides a visual representation of the actors involved with the system and their dependencies.

The i* framework requires knowledge of what the actor wants; how they achieve it; and who they depend on to achieve it. In i*, there are two types of models: the Strategic Dependency model (SD) and the Strategic Rationale model (SR). SD model describes the dependencies among actors and express the network of intentional and strategic relations between the actors. It is represented by a set of nodes and links, where each node is an actor and each connection between two actors indicates that an actor depends on the other to achieve a goal. While, SR model demonstrates how actors achieve their goals, express and represent the reasons behind the dependency, having several types of nodes and links. In this model, the relationships are analyzed in the context of a single actor. Each actor has its own intentional boundary having internal elements.

Extracting information and understanding the system as a whole are challenges when analyzing user stories (Sharp et al., 2006); (Beatty et al., 2012). The i* framework fills this gap by providing a graphical view of the actors involved in the system and providing a better understanding of the system and its dependencies (Alencar, 1999). This is our main reason for proposing the integration of user stories and i* models. According to Beatty et al. (Beatty et al., 2012), also in an Agile environment it is necessary to develop some models before starting implementation in order to guarantee a shared understanding by the development team. This way the team can be synchronized with the business goals and with the value and context of the project. For Beatty et al. (Beatty et al., 2012), visual models

facilitate the understanding of what users need in the system. They are also very useful for the stakeholders to understand the proposed solution and to maintain commitment. In other words, an agile method should not oversimplify the development process to the point of risking to lose relevance with business goals.

Models help organize and present a vast quantity of information and give context to details. They provide visual groups that allow for analysis of great quantities of information quickly in a short time period (Beatty et al., 2012).

Therefore, in this work, user stories are mapped to i* models (and vice versa), generating a diagram with i* concepts and facilitating the visualization and understanding of the software requirements and their relationships.

The i* framework concepts and notations used in this work are in accordance with the i* proposed at i* Wiki (2012), which represents a simplified version. In addition, it is important to note that only some elements of the i* framework are used, according to the need of the proposed approach.

4 MAPPING BETWEEN USER STORIES AND i* MODELS

Software requirements must give priority to communication between all stakeholders (Cohn, 2004). Therefore, the requirement elicitation and specification process can use a set of techniques that might be applied together: interviews, questionnaires, ethnographies and workshops, are example of those techniques. However, this process can present some problems due conflicts communication and comprehension within all stakeholders (Liaskos et al., 2012).

I* Framework can be useful to improve the understanding of system since their models are related to why those requirements are needed (Bresciani et al., 2004). In this way, i* model presents a high-abstraction requirements modeling which explores a business macro vision. On other hand, User Stories can represent a lower abstraction, such as functional requirements.

Although both approaches deal with requirement problems using a diversity of artifacts, they only cover a particular scope from whole system. Thus, it is possible relate them by a mapping and transformation process, using a set of heuristics.

This paper proposes mapping from user stories to i* models and vice versa. This approach provides a understanding of the system as a whole to stakeholders, providing a wide vision of the requirements described in the user stories. Projects based only on the user stories can bear risks, especially in complex systems, because a global understanding of the system-to-be is needed. Therefore, this work presents an approach which provides a graphical view of the system as a whole, showing the involved actors and their dependencies with the system. Figure 1 and Figure 2 present the activities to accomplish the mapping processes, respectively, from user stories to i* models and from i* models to user stories. Both take two kinds of heuristics into account: (i) the i* SD model; and (ii) the i* SR model.

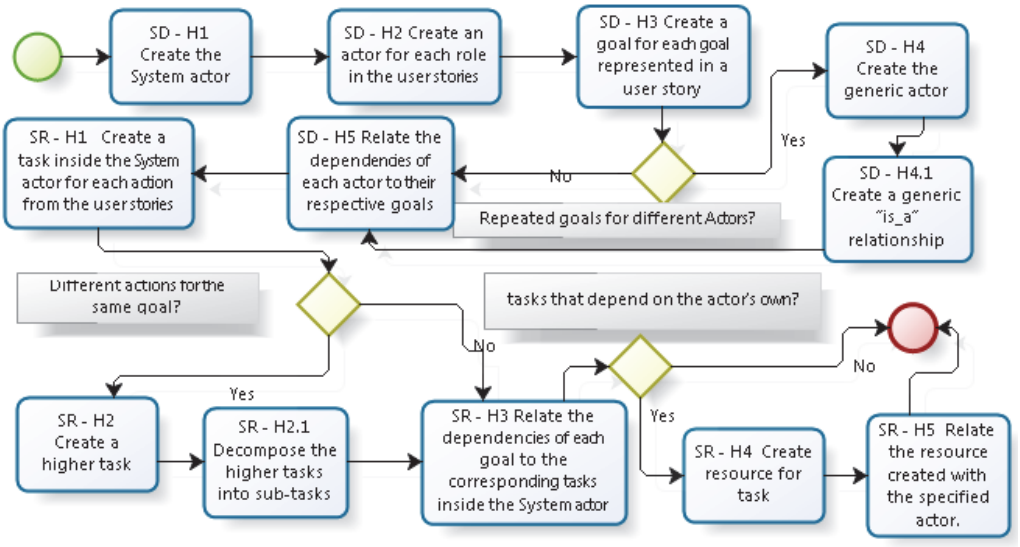

Figure 1: Mapping from user stories to i* models.

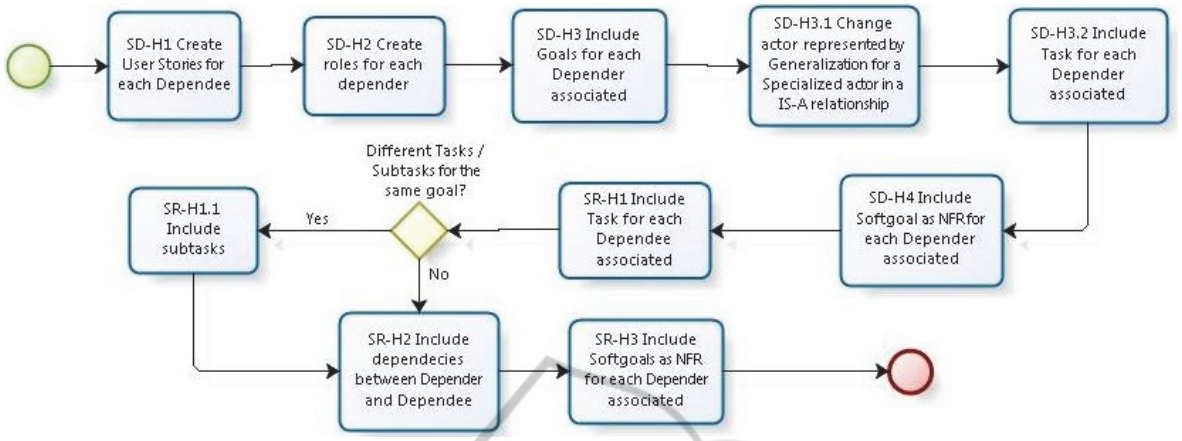

Figure 2: Mapping i* models into user stories.

4.1 From User Story to i* Models

To demonstrate an application of the approach, a login system was used, taking into account the perspective of a user and an administrator. Table 1 presents the user stories of the login system (Data sets, 2012).

Table 1: User Stories of the login System.

	Role	Action	Goal
01	User	Have username	Access secure conten
02	User	Have password	Access secure conten
03	User	Choose username	Customize account
04	User	Change default password	Customize account
05	Administrator	Assign user password	Confirm the account activation email
06	Administrator	Send registration email	Confirm the account
07	Administrator	Request user to login	Ensure security of content
08	User	Register password remind	Remember the password
09	Administrator	Request password remind	Confirm user

To simplify the understanding and to make the mapping more systematic, the following heuristics were established for the SD and SR models. The heuristics used to create the SD models are:

• SD-H1: Create the System actor. If the system is the main actor, it will meet the goals by each actor involved in the context. The System actor is created at the beginning since the users stories require the existence of a system to be developed.

• SD-H2: Create an actor in the i* model for each different role represented in the user stories. This occurs because an actor in the i* model has the same meaning as a role in a user story.

• SD-H3: Create a goal in the i* model for each goal represented in a user story. In case of repeated goals, it will be considered only once in the model. The goal has the same meaning as in the user stories domain.

• SD-H4: If there are repeated goals for different actors, a generic actor needs be created since this generic actor will relate to shared goals to facilitate the understanding of model.

• SD-H4.1: Create a generic "*is_a*" relationship of the actor to other specific actors that share the same goal. This heuristic is a consequence of SD-H4.

• SD-H5: Relate the dependencies of each actor to their respective goals, i.e., the actors will be connected with their goals. The System actor will be the dependee since the other actors depend on it.

The following heuristics are used to accomplish the mapping from user stories to the i* SR model:

• SR-H1: Create a task inside the System actor for each action from the user stories since the System actor will fulfill the goals that other actors depend on.

SR-H2: If there are different actions for the same goal, a higher-level task is created to concentrate the different actions. This is done to better visualize the model.

• SR-H2.1: Decompose a higher-level task into sub-tasks that represent the actions related to the same goal. This heuristic is a consequence of SR-H2.

• SR-H3: Relate the dependencies of each goal to the corresponding tasks inside the System actor according to the user stories. This is necessary to complete the corresponding i* model.

• SR-H4: If there are tasks that depend on the same actor, a resource dependency between the System actor and this actor is created, showing that the System actor depends on this actor to have this resource.

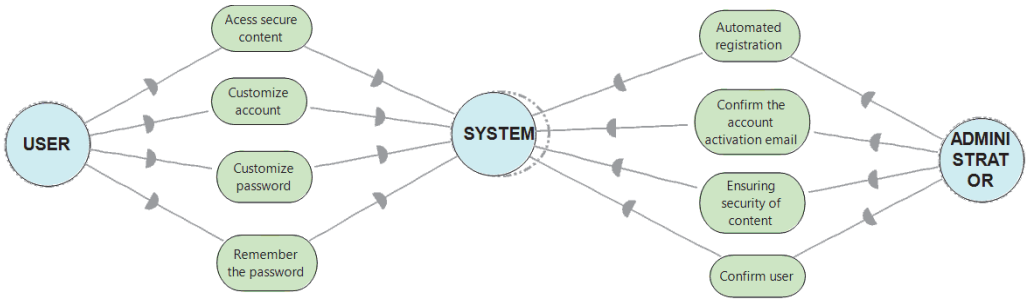

Figure 3: The resulting SD model from mapping user stories to the i* SD model.

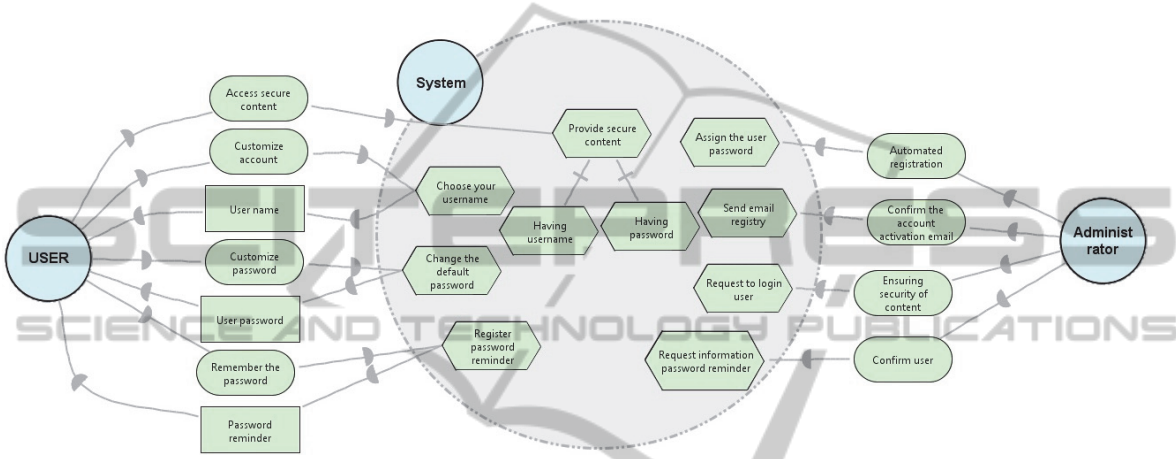

Figure 4: The resulting SR model from mapping user stories to the i* SR model.

• SR-H5: Relate the resource created with the specified actor.

According to the proposed heuristics, Figure 3 presents the SD model and Figure 4 presents the SR model. Both are the result of the mapping of user story to i* models.

4.2 From i* Models to User Stories

As an alternative, we use i* models to create and enrich a set of user stories. This will be useful in linking the early phase and the late phase. To perform the mapping of i* models into user stories, another set of heuristics is proposed. The following heuristics are used to Map the SD models to user stories:

• SD-H1: For all "dependee" it must be created a User Story. Each one is represented by a set of role, action and goal (Cohn, 2006). Moreover the softgoal was added as a NFR.

• SD-H2: For all "depender", it must create a role into User Story which belongs to dependee.

• SD-H3: All Goal dependencies relationship must include its dependum as a goal into User Story which is associated with dependee, in the same role associated to its depender; The related roles is set using the above Heuristic (SD-H2).

• SD-H3.1: If an actor is a "depender" in SD model, and also represents a generalization of IS_A relationship. So, the role associated to this actor will be changed for role which is related to actor that represent a specialization in IS_A relationship. If this actor is a "dependee", so nothing happen.

• SD-H3.2: For all "depender" which is related to a "dependee", from a Task dependency relationship, and also is associated to a User Story. The correspondent role associated to depender, is gonna link a task as an action which is perform by this role.

• SD-H4: All Softgoal dependency relationship, will be added a NFR (Non-Functional Requirement) into correspondent actor (depender) which is a role in a User Story, represented by dependee on this dependency.

The heuristics to accomplish the mapping of the SR i* model to user stories are:

• SR-H1: For each task that represents a "dependee" in i* model, will be created an action into User Stories related to this dependee. In case of this internal task is associated to a goal dependency relationship,so, the goal will be linked to this internal task, as an objective.

• SR-H1.1: If there's internal task from dependee in a goal dependency, and that is decomposed into different subtasks. So, these subtasks will be transformed as actions, and will be related to a goal in the User Story represented by "dependee".

• SR-H2: Is needed to establish a relation dependencies for each goal related to a "depender", and that has correspondent actions into "dependee" according to User Story.

• SR-H3: For each internal task from dependee, related in a goal dependency, that is decomposed into a softgoal. It will be added into User Story as a non-functional requirement (NFR), for its related role, or actor.

Figure 5 presents the SD model and Figure 6 presents the SR model that leads to Table 2, the result of mapping i* models to user stories according to the proposed heuristics.

Table 2: User stories of the login system with softgoals.

	Role	Action	Goal	Softgoal
01	User	Have username	Access secure content	-
02	User	Have password	Access secure content	Security
03	User	Choose username	Customize account	Flexibility
04	User	Change default password	Customize account	
05	Administrator	Assign user password	Confirm the account activation email	Reliability
06	Administrator	Send registration email	Confirm the account	Performance, Usability
07	Administrator	Request user to login	Ensuring security of content	-
08	User	Register password reminder	Remember password	-
09	Administrator	Request password reminder	Confirm user	-

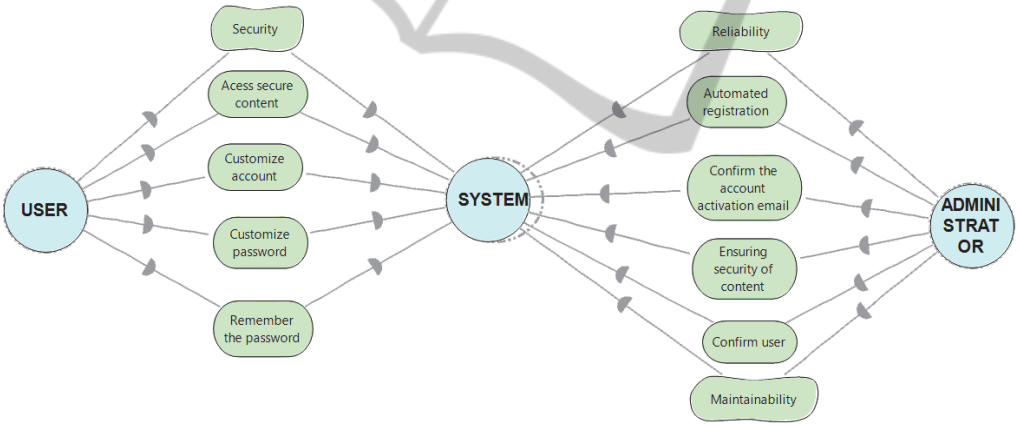

Figure 5: The resulting SD model of mapping from i* SD model to user stories.

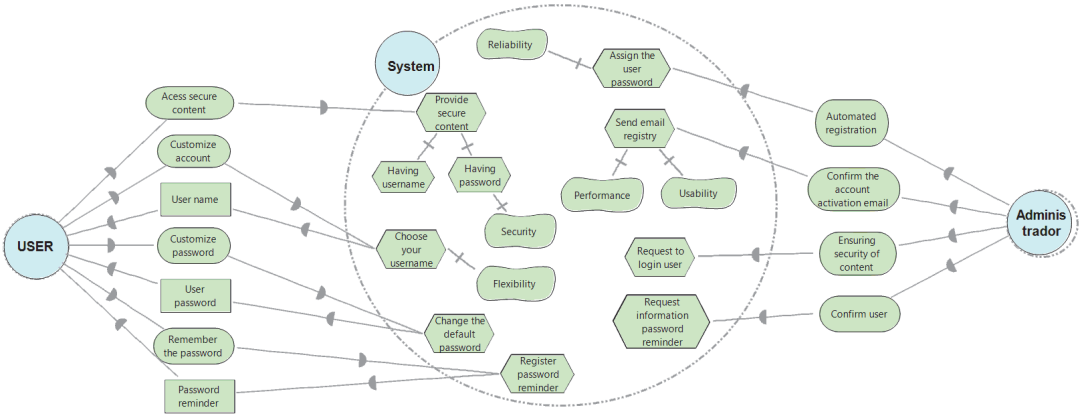

Figure 6: The resulting SR model of mapping from i* SR model to user stories.

5 CASE STUDY

In order to evaluate the mapping of user stories to i* models, a case study was conducted. The goal of the case study was to verify, through the impressions of user participants, if the mapping of user stories to i* models brings benefits for the Agile development. After using the approach, the participants reported their impressions, which are analyzed and discussed next.

5.1 Research Questions

In order to assess the results reported by the participants while using the approach, 5 research questions were developed and answered through the execution of a case study:

• Question 1: How difficult is the learning and understanding of the approach?

• Question 2: How was the performance of the requirements engineers for mapping user stories to i* models?

• Question 3: What were the perceived benefits of using the approach?

• Question 4: What was the perceived usefulness of the approach?

• Question 5: Would the participants use the approach in their work environments?

5.2 Input Artifacts and Participants

This case study considered a project management system developed by a small software company. The artifacts used from this system were short explanatory texts of the software scope and system user stories. 13 volunteers with experience in Agile development participated in this case study.

5.3 Case Study Execution

The participants were organized into two groups, according to their current occupations and abilities: requirements engineers (6) and software developers (7). Both groups evaluated the usefulness of the i* models, but only the requirements engineers mapped user stories to i* models.

During the study, the participants used their own computers (after the installation of the OME tool), the heuristics for mapping user stories into i* models, the text with the system scope, and the user stories. They were requested to create the SD and SR models based on the provided user stories.

After these activities, data was collected through a questionnaire to collect the impressions of each participant about using the approach. The questionnaire and the responses are available at http://goo.gl/8hZcs7.

5.4 Data Analysis

Next, we present and analyze the answers collected for each research question in this case study.

• Question 1: How difficult is the learning and understanding of the approach?

The participants assessed themselves and rated their level of learning and understanding of the approach after using it. According to the participants' answers, it was possible to conclude that the approach was well understood by them. This result is highlighted in Table 3 and suggests a low learning curve for the approach.

Table 3: Learning and understanding of the approach.

	Requirements Engineers	Developers
Bad	-	-
Reasonable	-	1
Good	2	4
Great	4	2

• Question 2: How was the performance of the requirements engineers for mapping user stories to i* models?

Assessing the performance is an opportunity to point out the improvement needed for the approach. Five of the requirements engineers agreed that the use of the approach was good and only one considered it bad. As this was the first contact of the participants with the approach, this result suggests that the heuristics for mapping user stories to i* models are clear and objective.

• Question 3: What were the perceived benefits of using the approach?

To assess the enhancements achieved by the approach, the participants were asked about possible enhancements in the visualization of the system under development, the facilitation of the requirements access and the support for decision-making. The results are summarized in Table 4.

Table 4: Benefits obtained from approach.

	Visualization of the context		Access of the requirement		Decision making	
	Yes	**No**	**Yes**	**No**	**Yes**	**No**
Requirements Engineers	6	-	5	1	5	1
Developers	7	-	7	-	6	1

According to the assessments, most of the participants believe that the approach enhances the

understanding of the system to be developed, what is considered important for better knowledge the requirements. By contributing to the understanding of the general system context, the i* models are enriching the development environment. According to Sharp, Robinson and Peter (2009), the user stories by themselves are not enough to fulfill this role. Taking into account the assessment of the participants, the mapping of the user stories for i* models makes it easier to use and to access requirement information. Therefore, there is evidence that the visual model generated by the approach provides more familiarize and expertise in understanding of the software requirements..

Most of the participants also agreed that the use of i* models contributes to the requirements in decision-making. Therefore, the approach enhances the Agile requirements by providing ease. It also provides a platform for decision-making in the development environment through more ample visualization of the system requirements.

• Question 4: What was the perceived usefulness of the approach?

The participants were asked if they considered the application of the i* models in visualizing the user stories useful. Most of the participants judged the usefulness of the approach as positive. Therefore, there is evidence that the approach is interesting because it was considered useful for the Agile development environment.

• Question 5: Would the participants use the approach in their work environment?

After using the approach, the participants were asked if they would use it in their work environment. Their answers are presented in Table 5.

Table 5: Use of the approach.

	Requirements Engineers	Developers
Yes	4	6
No	2	1

Most of the software developers showed acceptance for the approach, which corroborates the complaints about the restriction of the user stories in the Agile environment (Sharp et al., 2006), (Beatty et al., 2012). Considering both groups, most of them asserted they would use it (10 out of 13). Therefore, there is interest in this approach in enhancing user stories in the Agile development environment since it has generated interest among the participants.

5.5 Threats to Validity

Taking into account the concepts defined by Travassos et al., (2002), this section points out the main threats to the validity of the results achieved by the presented case study. The threats to internal and external validity are presented.

5.5.1 Internal Validity

The internal validity defines if the relationship observed between the treatment of the study and its result is causal and not influenced by another factor that is not controlled or even not measured.

The first threat to the internal validity in this study is related to the level of knowledge of Agile requirements, user stories and the i* technique by the participants, because the level of knowledge of each one could be different. To diminish this threat, training was performed to present the main concepts applied in the study to all participants before the beginning of the case study. This reduced the differences among the levels of knowledge of the participants. The groups were also defined according to the professional profile of each participant in order to diminish this threat.

The second threat to the internal validity of the case study is related to the availability and enthusiasm of the participants. To diminish this threat, the case study was scheduled on a convenient day and at a convenient hour, according to the availability of the participants, and the importance of their contribution to the research was highlighted.

5.5.2 External Validity

The external validity is related to the conditions that limit the generalization of the results achieved by the case study. In this way, two threats to the external validity of this study were identified.

The first threat is related to the number of considered scenarios. Only one scenario was used to assess the approach. It is important for new studies involving new scenarios to be conducted in the future.

The second threat to the external validity is related to the scenario chosen. A relatively small scenario was used. Thus, it is important to conduct new studies involving large and medium-sized scenarios in order to verify the results achieved by using the approach in those contexts.

5.6 Discussion

The approach presented in this work may seem incompatible with the culture of Agile software development, since one of the main characteristics of Agile methods is simplicity. Agile methods focus

on face-to-face collaboration among stakeholders to address the requirements of the software being less document-centric and more code-oriented. But there are some challenges with using these methods (Jaqueira, 2012): having information and understanding of the system as a whole from a user story, especially for complex systems, can be very difficult; the client may not be available in the development environment; or the client may be limited in explaining a requirement or replying to questions about the system. Therefore, it is important to evaluate the trade-off with the use of the proposed approach. According to the case study, the performance of the participants using the approach in the Agile development environment was good. For most of them, its use enhances the visualization of the system context to be developed, making the access to the requirements easier and contributing to the decision making. Regarding its use in the Agile environment, most of the participants confirmed its usefulness and showed interest in using it in their work environment. In this way, the solutions brought about by the visual models of the i* technique represent the dependency relationships and make the social comprehension an integral part of the development process, and they allow for a better understanding and analysis of the problem in an Agile development environment.

By mapping the user stories to the SD and SR models of the i* technique, it was possible to organize and represent all stories in a model that provides a general visualization of them and their relationships with the system. All stories of the same actor were presented in the same model, allowing for them to be identified more easily. In this way, it is possible to understand the context of the system with its main actors and their goals.

The visualization through the models made it easier to identify dependencies between the actors and the system as well as to identify the system tasks that serve each specific actor involved with the software.

According to Horkoff (Horkoff, 2009), it is thought that by visualizing the models, possible mistakes and negligence can be recognized more easily. All of this can facilitate the analysis and discussion about the system to be developed.

The use of i* models provides a better visualization of user stories. By allowing the visualization of the dependencies between the actors and the system, they contribute to a better understanding of the context of the software to be developed; by providing a visualization of the tasks of the system related to the goals of each actor; and by allowing possible mistakes or negligence in the requirements to be recognized through the visualization provided. Hence, the approach presented in this work can significantly contribute to Agile development environments because the models generated allow, through their visualization, for better analysis, communication, discussion and a better understanding of the system.

6 RELATED WORK

Patton (Patton, 2008) proposes a way to organize and prioritize user stories in order to provide a more comprehensive visualization by creating a map of stories without using the GORE approach. The user story cards are organized horizontally one next to the other on the wall according to the order that their functionalities are executed in by the users. The goal is to facilitate the exploration of new solutions and the understanding of the more ample context of interaction with stories, allowing the visualization of the software as a whole instead of focusing on one single story. This proposal uses card threads and a wall, limiting its use when the stories are not written on cards.

There are some works that associate the i * framework with agile methodologies. Scheidegger (Scheidegger, 2010) suggests the Scrum i* modeling language, that integrates Scrum with the i* framework. This language focuses on supporting the context understanding of the software to be developed, considering its relationships and aiming to solve the lack of understanding about the dependency between actors observed by the authors in Scrum. The i* modeling technique was simplified to be integrated into Scrum in order to keep the characteristics of agility (Scheidegger, 2010). Only part of the SD model was used.

The differentiating characteristic of the approach is that it is directly related to the requirements of the software, and that the actors of the models are the same as the owners of the user stories, in other words, the requirements. In Scheidegger (Scheidegger, 2010), organizational actors are used in the models, people interact with the organization and with the system, but they are not necessarily related to the requirements. The approach of our work can be used by any Agile development team that uses user stories in the format proposed by Mike Cohn (Cohn, 2006). In our approach, other elements of the i* technique, such as the SR models, the decomposition, and *is_a* association links, are also used.

According to Esfahani et al., (2010), Scrum is used to illustrate the use of the GORE approach in the descriptions of social aspects of the Agile methodology aimed at identifying the factors that contribute to the success or failure of adopting a Agile methodology, providing orientation during the introduction of the methodology in a company. The i* framework is used to model a social perspective, addressing the actors involved in the Scrum process. The actors are the Scrum Master, the owner of the product and the members of the development team. Through the representation of the social relationships provided by the i* technique, the Scrum adoption can be assessed according to the probabilities of success shown by the social aspects of the process, allowing for adjustments for the members of the current team and for identifying the vulnerability that are specific to the company.

The work of Santander and Castro (Santander and Castro, 2002) proposes an approach to assist requirements engineers in transforming i* models into use cases through guidelines. A set of guidelines was established, and applied to an i* model to generate a use case. This research is related to the traditional approach of software engineering. Moreover, it is possible to represent functional and non-functional requirements, covering all phases of requirements engineering, such as the early and late phases. The difference of our approach compared to this work is that our focus is on the social relationships based on the software requirements. The actors are the "owners of the requirements". Therefore, the contribution of the models is relative to the understanding and decision making regarding to the requirements.

7 CONCLUSION AND FUTURE WORK

The contribution of this work is the integration of user stories with i* models in order to deliver the benefits provided by the i* model with respect to visualization and analysis of the system through their models. This is an important contribution, since Agile methods do not include such an artifact to support the analysis and understanding of the system as a whole. The following contributions have also been made: (i) the enhancement of the context understanding as a way to consider both phases of requirements engineering: the early phase and the late phase, (ii) an easier access to the requirement information through the visual model, (iii) an enhancement of the decision making process according to the analysis of the requirements described in i* models, and (iv) a study case conducted to assess the mapping of user stories to i* models, showing how the use of i* models contributes to the visualization of user stories and to the understanding of those approaches.

Regarding the heuristics created to map the i* models and user stories, an experiment is under development to gather more evidence about those proposals. This solution aims to facilitate and optimize their use and the understanding of the requirements environment. As part of the development of the tool, two domain-specific languages (DSL) (Fowler, 2010) must be generated from the set of heuristics presented in Section 4, in order to: (i) map a user story to an i* model through a specific generator of this DSL; (ii) perform the opposite mapping to the first DSL through mapping and conversion of i* models to user stories with another specific language generator. The tool is divided into a set of steps to accomplish the mapping of approaches (user stories to i* or vice versa): (i) provide as input the code based on a DSL specific to a chosen approach, (ii) validate the domain language according to the choice made in the first stage, and (iii) present the visual model for the case of mapping a user story to an i* model or present the artifact of the user story for the case of mapping the i* model to a user story, showing all the features, including the nonfunctional requirements (NFR). A graphical interface will also be created for end user who are not used to working with textual DSLs. Thus, the tool will facilitate the definition and specification of requirements both in a GORE solution as well as in an Agile solution within the scope of requirements engineering.

As future work, we suggest: (i) developing a tool to convert the user stories into the format suggested by Mike Cohn (2006) which was used in this work, and an extension that considers the inclusion of NFRs (softgoals) in order to use this resource for all user stories, (ii) improving the mapping of the i* models back to user stories, considering more elements from i* model, such as resources, (iii) treating the scalability of the actor system, (iv) represent tasks in other actors of the model, (v) representing the relationship among the other actors in the model, and (vi) conducting other case studies that compare this work to other related work in order to verify the differences and similarities with more accuracy as well as to identify possible improvements needed.

REFERENCES

Alencar, F.M.R. de., 1999. *Mapeando a Modelagem Organizacional em Especificações Precisas*. 1999. 302 fls. Tese de Doutorado. UFPE.

Beatty, J. e Chen, A., 2012. A. *Visual Models for Software Requirements*. Washington, Microsoft Press.

Bhuiyan, M.; Islam, M. M.; Krishna, A.; Ghose, A.K., 2007. *Integration of Agent-Oriented Conceptual Models and UML Activity Diagrams Using Effect Annotations.* Computer Software and Applications Conference, v. 1, p. 171-178.

Bresciani, P.; Perini, A.; Giorgini, P.; Giunchiglia, F.; Mylopoulos, J., 2004. *Tropos: An Agent-Oriented Software Development Methodology.* Autonomous Agents and Multi-Agent Systems, v. 8, n. 3, p. 203-236.

Cockburn, A., 2007. *Agile Software Development: The Cooperative Game*. Boston, Pearson.

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

Cohn, M., 2006. Agile Estimating and Planning. Prentice Hall.

Data sets: Example of User Stories Available in <http://www58.ibm.com/software/analytics/manyeyes/datasets/example-of-user-stories/versions/1> Access in 05/12/2012.

Esfahani, H. C., Cabot, J. and Yu, E., 2010. *Adopting agile methods: can goal-oriented social modelling help?*. Conf. on Research Challenges in Information Science (RCIS'10).

Fowler, M. and Parsons, Rebecca., 2010. *Domain Specifc Language*. Addison-Wesley.

Horkoff, J. and Yu, E., 2009. *A qualitative, interactive evaluation procedure for goal- and agent-oriented models*. In: CAiSE Forum. CEUR Workshop Proceedings.

i* Wiki Home., 2012. Available in<http://istar.rwth-aachen.de/tiki-index.php> in 08/10/2012.

Jaqueira A., Andreotti, E., Lucena, M., Aranha, E., 2012. *Desafios de Requisitos em Métodos Ágeis: Uma Revisão Sistemática* 3rd Brazilian Workshop on Agile Methods, São Paulo.

Lamsweerde, A.van., 2001. *Goal-oriented requirements engineering: a guided tour*, in RE'01: Proceedings of the 5th IEEE International Symposium on Requirements Engineering. Washington, DC, USA: IEEE Computer Society.

Laue, R., 2011. *Expressing and analysing goal models in design structure matrices.* In Proceedings of the 13th International DSM Conference, p. 229-244.

Liaskos, S.; Jungblut, M. Daoud; Mylopoulos, J., 2012. *From intentions to code: exploring the life of user goals in two web-based systems*. IET Software, v. 6. n. 2, p. 103-113.

Martínez, A.; Estrada, H.; Sánchez, J.; Pastor, O., 2002. *From Early Requirements to User Interface Prototyping: A methodological approach.* In Proceedings of ASE, p. 257-260.

Patton, J., 2008. *The new user story backlog is a map*. Available in agileproductdesign.com.

Santander, V. F. And Castro, J.F., 2002. *Deriving Use Cases from Organizational Modeling.* IEEE Joint International Requirements Enginnering Conference. RE´2002, University of Essen, Germany, September 9-13, 2002, pp. 32-39

Scheidegger, M.E.S., 2010. *Integrando Scrum e a Modelagem de Requisitos Orientada a Objetivos por meio do SCRUM i**. Dissertação de Mestrado. Centro de Informática, Universidade Federal de Pernambuco.

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. IEEE Computer Society Press, pp. 65-75

Sharp, H., Robinson, H., Petre, M., 2009. *The role of physical artefacts in agile software development: two complementary perspectives*, Interacting with Computers 21. pp. 108–116.

Travassos, G. H.; Gurov, D. e Amaral, E. A. G., 2002. *Introdução à Engenharia de Software Experimental*. Relatório Técnico. ES-590/02-Abril. Programa de Engenharia de Sistemas e Computação, COPPE/UFRJ.

Yu, E., 1995. *Modelling Strategic Relationships for Process Reengineering.* PhD thesis. University of Toronto, Department of Computer Science.

Yu, E., 1997. *Towards Modeling and Reasoning Support for Early-Phase requirements engineering.* 3rd IEEE International Symposium on Requirements Engineering. IEEE Computer Society. p. 226.