A Proposal of Using Conceptual Models for User Story Development and Maintenance

Abhimanyu Gupta  
Ghent University  
abhimanyu.gupta@ugent.be

Geert Poels  
Ghent University  
geert.poels@ugent.be

Palash Bera  
St. Louis University  
palash.bera@slu.edu

ABSTRACT

While agile methodologies are commonly used in software development, researchers have identified many issues related to the requirements elicitation in agile projects. Some of these requirements relate to documentation and more specifically the development, maintenance, and management of user stories. This research addresses some of the user stories challenges by proposing the use of conceptual models while developing user stories. The proposal considers development of such conceptual models automatically (with the help of a tool) while user stories are developed. A detailed research plan has been developed to conduct this research.

Keywords
User stories, Agile methodologies, conceptual models

INTRODUCTION

In Agile software development processes, the software requirements documentation is limited to creation of user stories (Inayat, Salim, Marczak, Daneva, & Shamshirband, 2015). The user stories are simple descriptions of a feature of the working software written from the user’s perspective (Cohn, 2004; Leffingwell, 2011). A user story is a brief statement that describes the systems needs to do for the user (Leffingwell, 2011). Because of the substantial number of user stories that are developed in an Agile software development project, the Agile team finds difficulty in maintaining, tracing, and managing the user stories (Ramesh, Lan, & Baskerville, 2010). Also at a later stage the Agile team loses the high-level view of the various functions and their integrations (Trkman et al., 2016). The problem gets magnified when the team changes and/or there are new members added to the team who fail to comprehend the high-level architecture of the software under development (Vithana, 2015). Even for moderately complex software, the number of user stories grows exponentially. More the user stories are created; more details are required in the acceptance criteria to accommodate the complexity so that the scope becomes clear and explicit to the developers and testers. Acceptance criteria are the performance or other metrics that define the acceptable functionality of a story (Cohn, 2004). Therefore, if the user stories are not synchronized, integrated, and aligned with each other, they create more ambiguity, and as a result make the software unstable.

To alleviate this problem, we suggest using conceptual models in the process of developing user stories. Conceptual models are visual representations that are commonly used for understanding the domain of business functions and communicating with the stakeholders (Wand & Weber, 2002). The conceptual models can link individual user stories and help the Agile team in creating an understanding of the domain. It is suggested that user stories should be linked and mapped with the conceptual models of the domain (e.g. Figure 1). The research question that we investigate is “how conceptual models can help in creating, maintaining and managing user stories?”

This research agenda is executed through a PhD thesis and the various steps of the proposed methodology are the various stages of the PhD theses. In the next section, the proposed methodology is discussed followed by initial results. The conclusion section suggests possible contributions.

BACKGROUND

In Agile development, working software is valued over comprehensive documentation (Agile manifesto, 2017). However, many Agile practitioners often misinterpret this with “no documentation” (i.e. do not create any documentation) other than the creation of user stories. This creates a problem when at a later stage the Agile team loses the high-level view of the various functions and their integrations (Trkman et al., 2016). The problem gets magnified when the team changes and/or there are new members added to the team who fail to comprehend the high-level architecture of the software under development (Vithana, 2015). Even for moderately complex software, the number of user stories grows exponentially. More the user stories are created; more details are required in the acceptance criteria to accommodate the complexity so that the scope becomes clear and explicit to the developers and testers. Acceptance criteria are the performance or other metrics that define the acceptable functionality of a story (Cohn, 2004). Therefore, if the user stories are not synchronized, integrated, and aligned with each other, they create more ambiguity, and as a result make the software unstable.

Figure 1 shows examples of such mapping where each user story is mapped with certain part of the conceptual model. As more and more user stories are created, they are linked with the model.
Two assumptions are made in the conceptual model creation process. First, more than one type of model should be developed. For example, apart from the process model, a conceptual data model (e.g. ER model) can be created to show how domain concepts are interrelated. Second, these conceptual models are to be created by a tool instantaneously as soon as the user stories are written or modified. The second assumption is aligned with the minimum documentation philosophy of agile methodology.

Using the above approach, this mapping can have the following contribution:

1. The conceptual models will help align and synchronize all the user stories.
2. The Agile team members will get the overall high picture of the domain. The mapping will subsequently help to identify the incompleteness of the requirements (e.g. user stories that are not mapped with any part of the model).
3. The conceptual models will help the new members of the Agile team to learn about the requirements quickly.
4. The conceptual models will help facilitate the communication between the Agile teams that are often dispersed geographically.

Trkman et al. (2016) have proposed a business process modeling framework where the user stories are integrated in the business process model. They conducted an empirical study and found that when users are provided with user stories that are integrated as a business process then they understand the domain better than those who are provided only with a list of user stories. Although the business process model framework is similar to our proposal, but we propose linking user stories with more than one type of conceptual model. A data-centric conceptual model will help users to provide a high level of domain understanding that a process model cannot. Moreover, our methodology includes development of a tool that will automatically create multiple conceptual models.
among the categories. The cause-effect analysis will result into a framework of agile requirements challenges.

To create the above-mentioned framework, a Delphi study is planned. A Delphi study involves obtaining a collective view from individuals about issues where there is little definite evidence and where opinion is important (Thangaratinam & Redman, 2005). In a Delphi study, a three phase process is involved - brainstorming, narrowing down, and ranking (Keil, Koo Lee, & Deng, 2013). In this Delphi study, a group of experts in agile methodologies will interact anonymously in a web-based platform to create the framework. The tasks performed by the experts are:

- Classify the list of challenges to causes and effects
- Summarize the cause and effects lists to a small set of constructs
- Rank these constructs in terms of importance
- Establish the causal relationship among the constructs
- Develop a research framework on the relationships among the constructs

Stage 4: Prototype tool development

In agile development, where there is less focus on documentation, it will be unrealistic to expect that users will develop and maintain the conceptual models in the process of creating the user stories. Therefore, a prototype tool will be developed that will automatically create and update conceptual models when user stories are fed to it. The objective of the prototype will be to demonstrate the feasibility of creation of such tool. The exact details of the prototype will be identified later.

Stage 5: Validation of use of conceptual models

In step 3, a research framework will be developed as an outcome. This framework will show the relationships among multiple constructs (group of challenges). It is expected that one or more of the links in this framework will be related to the use of conceptual models in developing user stories. This link of the research framework will be empirically tested and validated in this thesis.

Prior to continue with the rest of the steps of the thesis, it is important to validate that it is possible to develop conceptual models from user stories (as suggested in Figure 1) using the developed tool. For this purpose, a case study approach will be taken. In this approach, an organization will be contacted who would share a set of user stories for a software development project. By analyzing the user stories and interviewing the Agile team who was involved in developing the user stories, conceptual models will be developed using the tool. The user stories will then be mapped to the conceptual models and analyzed how the models can be useful. Feedback will be taken on the usefulness of these models in creating, maintaining, and managing the user stories.

Stage 6: Testing the recommendation of using conceptual models

To test the recommendation that conceptual models can be helpful for developing user stories, a laboratory study will be done with real users as subjects. Subjects will be asked to extend and develop a set of user stories from a specific set of existing user stories. A between group design study will be used where a group of subjects will have access to the conceptual models (related to the domain) and another group of subjects will not have any access to these models. An eye tracking device will be used to identify: (1) whether the conceptual models that are provided to one set of users are indeed being used to develop the user stories. (2) if these models are used then which specific parts of the models are used, (3) between the two groups, whether there is a difference in the pattern of using the existing user stories.

Eye tracking offers a window into how individuals read and scan information that is displayed to them (Rayner, 1998). Eye movements provide a valid measure of distribution of attention (Glaholt & Reingold, 2011). By relating eye movements with tasks, one can obtain a picture of the decision-making process. A common eye movement metric is eye fixation that is often measured with respect to time or count (Sharif & Maletic, 2010). Eye tracking metrics such as fixation duration can be used to identify which parts of the conceptual models’ users are referring to while developing the user stories. Eye tracking devices also allow counting how many times the models have been accessed to and in what sequence. Fixation duration on overall conceptual models and specific parts of the model will provide insights on how the conceptual models have been used. Comparative fixation duration analysis between the two groups on existing user stories will throw light on the pattern of user stories usage.

At the end of the eye tracking study, the quality of the user stories that are developed by the two groups will be independently assessed by experts. It is expected that the quality of the user stories will be higher for those who referred to conceptual models when compared to the other group.

INITIAL RESULTS

Out of the above mentioned six steps, the first two steps have been completed so far. A detailed literature review followed by interviews with the experts have been conducted.

From 80 IS journal papers a list of 50 challenges were identified. The top three challenges that were identified are: customer unavailability, frequent changes in requirements, and insufficient information in user stories.

It was identified that the words-domain knowledge, shared understanding, communication, and learning occurred repeatedly in these challenges. Incidentally, two common uses of conceptual models are to: (1) facilitate communications between users and analysts and (2)
support the analysts’ understanding of the domain (Kung & Solvberg, 1986). Based on the definition and use of conceptual models, it seems that the cause of some agile requirements issues can be attributed due to lack of development of conceptual models.

Five Agile practitioners were interviewed in terms of what are the challenges they face working in Agile projects. The recruitment was based on convenient sampling and these practitioners had more than average 5 years of Agile experience. From the interviews, four new challenges were identified. These are: translation of features into user stories, unavailability of success criteria at feature level, insufficient system documentation, and agile not implemented at organization level.

It was identified that the words- domain knowledge, shared understanding, communication, and learning occurred repeatedly in these challenges. Incidentally, development of conceptual models involves building a representation of a domain (Hoffer, George, & Valacich, 2011). Some common uses of conceptual models are to: (1) facilitate communications between users and analysts, (2) support the analysts’ understanding of the domain, and (3) serve as the basis for system design and implementation (Kung & Solvberg, 1986). Based on the definition and use of conceptual models, it seems that the cause of some agile requirements issues can be attributed due to lack of development of conceptual models. Of course, this proposition needs to be identified and tested in the later part of the thesis.

CONCLUSION

This research can have several contributions. First, the consolidated literature review and interviews can provide insights to the requirements challenges in agile projects. Second, the Delphi study will help to develop a research framework on the agile requirements challenges. Third, an empirical study will be conducted to test this proposition. The results of the study can provide insights on how users can use conceptual models in creating new user stories. These insights can help the Agile teams in managing user stories and better communicate with each other in the process of developing software. Finally, a key component of this research is to create a prototype to demonstrate that conceptual models can be created automatically when user stories are developed. This prototype can help practitioners to develop and use tools in practice.

REFERENCES