8-6-2011

Generating User Stories in Groups with Prompts

Aaron Read
University of Nebraska at Omaha, aread@mail.unomaha.edu

Andy Callens
University of Nebraska at Omaha, acallens@mail.unomaha.edu

Cuong Nguyen
University of Nebraska Omaha, cdnguyen@mail.unomaha.edu

Gert Jan de Vreede
University of Nebraska Omaha, gdevreede@usf.edu

Generating User Stories in Groups with Prompts

Aaron Read
Center for Collaboration Science
aread@mail.unomaha.edu

Andy Callens
Center for Collaboration Science
acallens@mail.unomaha.edu

Cuong D. Nguyen
Center for Collaboration Science
cdnguyen@mail.unomaha.edu

Gert-Jan De Vreede
Center for Collaboration Science
gdevreede@mail.unomaha.edu

ABSTRACT

Communicating about system requirements with user stories is a distinctive feature of Agile Software Development methods. While user stories make system requirements intelligible to both customers and technical developers, they also create new challenges for the requirements elicitation process such as personal bias and requirements coverage. In this study we propose that when elicited from groups instead of individuals, and with prompts, the number of stories generated and comprehensiveness of the stories is likely to increase. A lab experiment was conducted to examine these hypotheses is delineated in this paper. We found that prompting significantly increased the number of user stories generated as well as the comprehensiveness of the stories generated. We did not find a difference in user stories generated or comprehensiveness of stories generated by groups and individuals.

KEYWORDS

Requirements elicitation, user stories, group story-telling, Agile Software Development.

INTRODUCTION

Software development remains a challenging process with only one third of projects successfully completed; other projects are canceled, considered late, over budget, and completed with fewer features than planned (Rubenstein, 2007). Poorly defined requirements are considered to be a leading factor in project failure (Hofmann and Lehner, 2001). Agile software development methodologies address difficulties of developing requirements resulting from rapidly changing customer needs by allowing a development team to respond quickly to changing requirements (Highsmith and Cockburn, 2001). They encourage incremental releases, cooperation between customer and developers, simplicity (ease of learning), and adaptability (Abrahamsson, Warsta, Siponen, and Ronkainen, 2003). User stories are an integral part of several Agile methodologies including XP and Scrum (Beck and Fowler, 2000; Cohn, 2004). A user story is a a short, one or two sentence account in the user's own words of a way that (s)he would like to use the software. It includes a user, an action, and a goal (Cohn, 2004). A story enables the communication of software requirements between developers and customers without needing familiarity with a specific method of delivery or jargon (Cohn, 2004). Although using stories as a means of gathering requirements has been shown to be beneficial in a number of studies (Alvarez and Urla, 2002; Ima and Benyon, 1999; Sutcliffe, 2003), collecting system requirements in the form of user stories can also be problematic for several reasons: A customer or potential user's tacit knowledge may be partially hidden, stories may be subject to multiple interpretations and personal bias, and the completeness of the set of stories may be difficult to determine (Sutcliffe, 2003). These problems may be addressed by collecting stories in groups. Group story telling can create an environment that supports evaluation of experience and promotes problem-solving (Banks-Wallace, 1998). This can help surface conflicts in goals among users and enable them to create shared understanding. Group story telling can also help elicit the tacit knowledge of participants with the richness of several different perspectives (Valle, Prinz, and Borges, 2002). The purpose of our research is to understand to what extent groups will outperform individuals in generating user stories. A key criterion to assess the quality of a set of requirements is completeness in fulfilling goals with a minimal amount of conflicts and overlaps in requirements (Grünbacher, Halling, Biffl, Kitap, and Boehm, 2004). We therefore assess the extent to which groups are able to generate a comprehensive set of requirements without duplicates in a lab experiment. We also assess the effectiveness of prompting techniques in stimulating groups and individuals to generate requirements in the forms of stories. Prompting has been found to be effective means of input stimulation in research studying general brainstorming (Santanen, Briggs, and Vreede, 2004) and in requirements elicitation (Pitts and Browne, 2007).

In the following section we define stories and explain their use in requirements engineering. We then explain the design and results of our study. The paper concludes with a discussion of our contributions, the limitations of this work, and directions for future research.

BACKGROUND

Requirements elicitation

System requirements gathering or elicitation is the first and one of the critical steps in requirements engineering (Hickey and Davis, 2004; Nuseibeh and Easterbrook, 2000). The purpose of the requirements elicitation activity is to arrive at a description of the goals of the new system, with an understanding of the needs of the stakeholders and the constraints of the system (Nuseibeh and Easterbrook, 2000). The process of eliciting requirements consists of several steps including elicitation, analysis, specification, and verification. In elicitation, the needs of customers are discovered. During analysis, information from stakeholders is analyzed through the creation of models or prototypes for incompleteness and inconsistency. Specification involves documenting the required behaviors of the system. Finally in verification, the requirements are validated with stakeholders (Hickey and Davis, 2004).

A major concern of the requirements elicitation process is to understand stakeholder needs and discover a set of requirements that completely represents the needs of stakeholders. Towards that end, various requirement elicitation techniques have been devised and implemented, such as observation, interviews or protocol analysis (Maiden and Rugg, 1996).The techniques may differ from one another in the difficulty levels of implementation, in the kinds of data format they might acquire and the time and effort to implement the techniques (Maiden and Rugg, 1996). Due to their strengths and weaknesses, each technique finds its fit in different contexts depending on the purposes of requirements and the type of knowledge that requirement engineer wants to elicit (Maiden and Rugg, 1996). For example, observation is a simple technique to execute, but might result in a lot of irrelevant data. It might be a good method for discovering tacit knowledge but not a recommended technique for eliciting non-tacit knowledge and requirements for a future system. In recent years, along with the increasing popularity of agile software development method, eliciting system requirements in form of user stories have gained momentum as a new elicitation technique with certain beneficial features. A discussion of the advantages and disadvantages of the technique is provided in the following section.

Stories in Requirements Engineering

In requirements engineering, stories are used and structured in at least two different ways: as long accounts of user's interactions with a system, and as short, one or two sentence descriptions of interactions with a system that are used in Agile Softwar edevelopment technqiues. First, stories may be used to capture the experiences of users with a current system as well as aid in capturing the desired attributes of a system (Alvarez and Urla, 2002). Secondly, stories used this way are often long narratives that provide rich information about the users' habitual work practices as well as their specific difficulties and needs (Alvarez and Urla, 2002).

In the second structure, user stories play an important role in the requirements elicitation process in agile development (Cohn, 2004). Eliciting requirements in the form of user stories allows stakeholders to convey their needs in a way that is natural to them, allowing them to relate more tacit knowledge (Alvarez and Urla, 2002). Documenting system requirements in the form of user stories allows customers to communicate desired features of a system without having to know a specific modeling language (Davies, 2002). Details of stories are worked out through oral communication between the users and requirements engineers, thus avoiding errors of interpretation which may occur with written requirements (Jeffries, Anderson, and Hendrickson, 2000). However, stories consist of an individual's view of a system and may therefore make it difficult for requirements engineers to grasp a complete view of the system (Sutcliffe, 2003). Valuable information might not be volunteered by users as they might assume it is already known to the requirements engineers, or forget abnormal cases (Sutcliffe, 2003).

Group Storytelling

Allowing users to generate stories in groups can alleviate some of the aforementioned shortcomings of collecting user requirements with stories individually. Story telling in groups allows a problem to be seen from multiple perspectives (Valle et al., 2002). When users tell stories in groups, the knowledge of one user can be verified and expanded by another user, since the knowledge of one user helps to activate the knowledge of another group member (Leal, 1993). When users meet in groups, they are able to evaluate one another's information and ask for clarification, or provide the clarification (Banks-Wallace, 1998). A similar phenomenon is likely to be experienced by groups brainstorming requirements. In a brainstorming setting, the exposure of member's groups to the ideas of other members may prompt them to fill in gaps left in the requirements left by their peers.

With this understanding of the benefits of group storytelling and brainstorming techniques, we propose to test the following hypotheses:

H1a: Groups generate a larger quantity of stories than those working individually.
H1b: Groups generate a more comprehensive collection of stories than those working individually.

Prompting

Different collaboration processes can create different group performance. In this study, we also examined the two brainstorming techniques i.e. brainstorming with prompts and brainstorming without prompts, or free brainstorming influence the group performance in story telling task. In a brainstorming session with prompts, the requirements engineers will add some prompts, or questions or suggestions that can direct the users' thoughts to areas that they might ignore while in a free brainstorming session, people tell their stories freely without the intervention of the requirement engineers. While free brainstorming is traditional and has been applied in requirements gathering for a long time (Maiden and Rugg, 1996), brainstorming with prompts is a relatively new technique that was proposed by Santanen and colleagues (Santanen, Briggs, and de Vreede, 2000) under the name "directed brainstorming". The technique was claimed to make a group produce more unique and straight to the point solutions for problem solving tasks than free brainstorming (Santanen et al., 2000). Brainstorming with prompts is also considered to reduce cognitive challenges like human memory constraints during a requirement elicitation process (Pitts and Browne, 2007). Based on the findings of these previous studies, we propose the following hypotheses:

H2a: Groups using brainstorming with prompts generate a larger quantity of stories than those using brainstorming without prompts.
H2b: Groups using brainstorming with prompts generate a more comprehensive set of stories than those using brainstorming without prompts.

METHODS

The method applied in this study to test the hypotheses was lab experiment. The description of the experiments was provided in this section.

Participants

Seventy one students enrolled in course in the psychology, business, and management information systems department classes participated in this study. Of the students who provided demographic information, 77 percent of participants were female. Participants were predominantly white (84%), with 8% Latino and 4% each of Asians and Blacks.

Procedure

The experiment utilized the book exchange system scenario to elicit stories of requirements for the development of a theoretical online book exchange system (see Appendix A). This task was deemed sufficiently complex such that users would be able to generate requirements for a full 40 minutes and also because textbook exchange is a subject matter of high familiarity to college students. User stories were generated and captured electronically using GroupSystem's Thinktank version 2.4 group decision support software.

After obtaining informed consent and demographic information, each session began by presenting each participant with a written description of the book exchange scenario that the experimenter read aloud. This description included (a) a description of the task, (b) an explanation of the key components of a valid user story, and (c) multiple examples of good user stories.

Verbal instructions indicated that subjects were to generate as many user stories as they could over a forty-minute time period. At this time, participants were divided into groups of three or four and were told to begin the brainstorming activity. Groups were stopped before forty minutes if no contributions were submitted for at least 120 seconds. Following user story generation, individuals completed additional measures of satisfaction with the process and outcomes and were debriefed.

Variables

This study employed a between-participants factorial design composed of two independent variables (shared vs. unshared ideas & facilitator prompting vs. non-prompting). Sharing of ideas was manipulated within the GDSS software by allowing/denying participants the ability to view and comment upon the user stories created by the others in their ad-hoc group. Prompting was manipulated by the experimenter reading seven scripted prompts at approximately five-minute intervals (see appendix B for the list of prompts).

The dependent variables of interest to the study were quantity and comprehensiveness of user stories. Quantity was defined as the total number of user stories generated by subjects. Comprehensiveness was calculated as the number of 101 predetermined categories that were covered by the user-generated requirements generated in earlier pilot testing of this experiment. Coders worked in pairs to independently rate data from each of the 22 groups for comprehensiveness. Inter-rater reliability was 88.8%. Disagreements in comprehensiveness scores were resolved during a consensus meeting.

RESULTS

Across all conditions, groups generated an average of 72.38 (SD = 48.32) user stories that covered 38 (SD = 11.66) of the pre-determined categories. Further descriptive statistics are shown in Table 1 and 2. No support existed for our hypothesis that groups that could view the contributions of others would generate more stories (M = 66.5) than those groups whose team member had to work alone (*M* = 76.00), *F* (1,19) = 0.18, *p* =.67; nor were groups that were able to share contributions able to generate a more comprehensive list of features (*M* = 34.56), than those groups that worked alone (*M* = 38.62) *F* (1, 19) = .09, *p* = .76.

As expected, prompted groups generated more user stories (*M* = 122.14) than unprompted groups (M = 47.5), *F* (1, 19) = 23.85, *p* < .01. Further, prompted groups generated a more comprehensive list of software features (*M* = 50.86) than unprompted groups (*M* = 31.57), *F* (1,19) = 33.56, *p* < .01.

Raw User Stories Generated		
	Shared	Unshared
Prompted	131.5	118.40
Unprompted	44.83	49.50

Table 1. Mean of number of generated user stories by group type

Key Categories Hit		
	Shared	Unshared
Prompted	42.00	49.00
Unprompted	30.83	32.13

Table 2. Mean of number of key categories hit by group type.

DISCUSSION

This paper has explored the question of whether generating user stories in groups, with prompted and unprompted facilitation will result in significantly more user stories and features of a simple, real-life online textbook marketplace. We found support for our hypothesis that prompted user story generation would outperform unprompted generation in terms of number of raw contributions and comprehensiveness or number of features generated. We did not find support for our hypothesis that groups would generate a higher number user stories and features than pooled individuals. Our findings provide insight for practitioners and researchers wishing to understand the benefits of group storytelling in the requirements elicitation setting. Practitioners employing Agile Methods to develop software are likely to see additional benefits in terms of the quantity of stories generated as well as a more comprehensive set of features if they choose to prompt users. Our findings support the role of prompting in the requirements elicitation process, further validating the findings of Pitts and Browne's study (Pitts and Browne, 2007).

Our study extends understanding of the role of prompting requirements elicitation by exploring the elicitation with multiple individuals brainstorming ideas separately or in groups. In each of our treatments, we have explored how multiple viewpoints can be pooled to understand the requirements of the system. At this point, it appears that it makes no difference in the number of stories or comprehensiveness of features generated whether or not the elicited individuals are exposed to the ideas of others, even if they are prompted to read and build upon preexisting ideas.

The lack of support for a group requirements brainstorming effect is surprising. Groups have long been thought to generate a much broader set of ideas than individuals (Osborn, 1957). Some research may indicate that this lack of group advantage may be due to social effects such as free riding or evaluation apprehension (Warr and O'Neill, 2005). However, in each prompted group setting, we observed that participants were making contributions the entire time. In fact, we feel that social pressure may have accounted partially for the lack of difference between group and individual treatments, as participants in the same room were able to observe others entering ideas and may have felt some pressure to avoid being among a minority who were participating less. Groups may not have differed greatly from individuals in generating requirements because of the participants' familiarity with existing online textbook marketplaces. Such familiarity may have made it unlikely for participants to vary in their understanding of what requirements such an online marketplace should have. They may not have been inspired from access to the ideas about requirements generated by others because they were so similar. However, by being exposed to the features generated by others, a participant in a group treatment may have decided not to generate the same requirements. In a future analysis, we plan to assess whether or not groups had fewer duplicate ideas than individuals in both prompted and unprompted settings. Finding that groups generated fewer duplicates could provide evidence that groups are more efficient at generating requirements.

CONCLUSION

The current study represents a novel approach to requirements elicitation research and group brainstorming research at the same time. While we did not find support for a group effect, on one hand, we demonstrate the benefits of prompting in brainstorming to reach a fixed set of ideas. A majority of the brainstorming research has assessed the role of prompts in generating a set of unique ideas (Santanen et al., 2004). Instead of generating a loosely connected set of ideas, the group had the task of generating a complete solution. Future research can build on our study to make groups more effective at this goal. We also contribute to the requirements literature by introducing new research that combines GSS supported brainstorming and prompting with traditional requirements techniques. Such an approach has been proposed earlier, but without empirical validation (Boehm, Grunbacher, and Briggs, 2001). While this method was highly successful in the field, it has not been validated in an experimental setting. Future research in the field of requirements engineering could also focus on the benefits of brainstorming and prompting techniques with subjects of mixed expertise or with a more complex/less defined system**.**

REFERENCES

1. Abrahamsson, P., Warsta, J., Siponen, M. T., and Ronkainen, J. (2003). New directions on agile methods: a comparative analysis. *Proceedings of the 25th International Conference on Software Engineering*, Portland, Oregon, 244-254
2. Alvarez, R., and Urla, J. (2002). Tell me a good story: using narrative analysis to examine information requirements interviews during an ERP implementation. *SIGMIS Database,* 33, 1, 38-52
3. Banks-Wallace, J. (1998). Emancipatory Potential of Storytelling in a Group. *Journal of Nursing Scholarship,* 30, 1, 17-22
4. Beck, K., and Fowler, M. (2000). Planning Extreme Programming Addison-Wesley Longman Publishing Co., Inc.
5. Boehm, B., Grunbacher, P., and Briggs, R. O. (2001). Developing groupware for requirements negotiation: Lessons learned. *IEEE Software,* 18, 3, 46-55.
6. Cohn, M. (2004). User Stories Applied: For Agile Software Development Addison Wesley Longman Publishing Co., Inc
7. Davies, R. (2002). The Power of Stories
8. Grünbacher, P., Halling, M., Biffl, S., Kitap, H., and Boehm, B. W. (2004). Integrating Collaborative Processes and Quality Assurance Techniques: Experiences from Requirements Negotiation. *Journal of Management Information Systems,* 20, 4, 9-29
9. Hickey, A. M., and Davis, A. M. (2004). A Unified Model of Requirements Elicitation. *Journal of Management Information Systems,* 20, 4, 65-84
10. Highsmith, J., and Cockburn, A. (2001). Agile software development: the business of innovation. *Computer,* 34, 9, 120-127.
11. Hofmann, H. F., and Lehner, F. (2001). Requirements Engineering as a Success Factor in Software Projects. *IEEE Software,* 18, 4, 58-66.
12. Ima, M., and Benyon, D. (1999). How Stories Capture Interactions: IOS Press.
13. Jeffries, R. E., Anderson, A., and Hendrickson, C. (2000). Extreme Programming Installed Addison-Wesley Longman Publishing Co., Inc.
14. Leal, D. J. (1993). The Power of Literary Peer-Group Discussions: How Children Collaboratively Negotiate Meaning. *Reading Teacher,* 47, 2, 114-120.

15. Maiden, N. A. M., and Rugg, G. (1996). ACRE: selecting methods for requirements acquisition. *Software Engineering Journal*, 11, 3, 183-192.
16. Nuseibeh, B., and Easterbrook, S. (2000). Requirements engineering: A roadmap. *Proceedings of the Conference on The Future of Software Engineering*, Limerick, Ireland, 35-46
17. Osborn, A. P. (1957). Applied Ituagination: Principles ami Procedures of Creative Thinking (2 ed.). New York Scribner.
18. Pitts, M. G., and Browne, G. J. (2007). Improving requirements elicitation: an empirical investigation of procedural prompts. *Information Systems Journal,* 17, 1, 89-110
19. Rubenstein, D. (2007). Standish Group Report: There's Less Development Chaos Today. *Software Development Times*, from http:// www.sdtimes.com/content/article.aspx?ArticleID=30247
20. Santanen, E. L., Briggs, R. O., and de Vreede, G.-J. (2000). The cognitive network model of creativity: a new causal model of creativity and a new brainstorming technique. *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*.
21. Santanen, E. L., Briggs, R. O., and Vreede, G.-J. D. (2004). Causal Relationships in Creative Problem Solving: Comparing Facilitation Interventions for Ideation. *J. Manage. Inf. Syst.,* 20, 4, 167-198.
22. Sutcliffe, A. (2003). Scenario-based requirements engineering, *Proceedings of the 11th IEEE International on Requirements Engineering* (pp. 320-329).
23. Valle, C., Prinz, W., and Borges, M. (2002). Generation of group storytelling in post-decision implementation process. *Proceedings of the 7th International Conference on Computer Supported Cooperative Work in Design*., 361-367.
24. Warr, A., and O'Neill, E. (2005). *Understanding design as a social creative process*. Paper presented at the Proceedings of the 5th conference on creativity & cognition.

APPENDIX A - EXPERIMENT SCRIPTS

Every subject in the groups will be given the same overview description of the system to be designed as follows:

The Book Exchange is a website which will be designed to allow students at this university to buy and sell text books at a reasonable price. The website will not provide payment services; it will simply allow sellers to post items for sale, allowing potential buyers to search for their textbook offerings. The website will also have features that facilitate a buyer's search for textbooks. For example, the website will have access to which textbooks are required for a given course.

Next, the subjects receive the following instructions:

- Provide as many user stories as possible. A user stories is a story that provide a feature that the system to be designed should have in your opinion. A recommended form for a user story is:
  "As a <type of user>, I want <some goal> so that <some reason>." (Type of user = buyer, seller, professor, administrator).

  e.g. "As a buyer, I want to be able to see the prices of all the books so that I can decide whether to buy the book or not.

- Continue to brainstorm user stories

Your stories should not be more than two sentences in length. You are NOT being asked to come up with a technical description of the website (i.e., it will use mySQL database for data storage). Instead we are asking you to describe what the website can do from the perspective of the website's users.

The experiment will be conducted in one hour long sessions and the experiment processes applied to each of four treatment groups are as follows:

- *Individual story telling – Unprompted* group: For this group, the session starts with letting the subjects complete a questionnaire. After that, the investigators give a short presentation about the purpose and the procedure of the session. Then, the subjects are trained on using GroupSystems. Next each subject is required to generate his/her user stories individually, i.e. no contact with other subjects is allowed, by typing them in a computer.
- *Individual story telling –Prompted* group: For this group, the session starts with letting the subjects complete a questionnaire. After that, the investigators give a short presentation about the purpose and the working process of the session. Then, the subjects are trained on using GroupSystems. Next each subject is required to generate his/her user stories individually, i.e. no contact with other subjects is allowed, by typing them in a computer. After the group has been brainstorming for 10 minutes, prompting will begin. Try not to explain the prompt for more than 1 minute, as this time cuts into the productive time with the group. A prompt should be given to the group once every seven minutes, so that in the 35 minutes that they have left to brainstorm, they are given 5 prompts. Before the first prompt, assign the group members to their own bucket (1-4). After each prompt is given, switch the group members to the next bucket.

- *Group story telling – Unprompted* group: For this group, the session starts with letting the subjects complete a questionnaire. The subjects are suggested to provide no more than two sentence length stories. Then the subjects are divided into groups of three people. After that, the investigators give a short presentation about the purpose and the working process of the session. Then, the subjects are trained on using GroupSystems. Next, each subject group is required to generate their user stories together by contributing user stories to the same electronic page or list at the same time.
- *Group story telling – Prompted* group: For this group, the session starts with letting the subjects complete a questionnaire. The subjects are suggested to provide no more than two sentence length stories. Then the subjects are divided into groups of three people. After that, the investigators give a short presentation about the purpose and the working process of the session. Then, the subjects are trained on using GroupSystems. Next, each subject group is required to generate their user stories together by contributing user stories to the same electronic page or list at the same time. After the group has been brainstorming for 10 minutes, prompting will begin. Try not to explain the prompt for more than 1 minute, as this time cuts into the productive time with the group. A prompt should be given to the group once every seven minutes, so that in the 35 minutes that they have left to brainstorm, they are given 5 prompts. Before the first prompt, assign the group members to their own bucket (1-4). After each prompt is given, switch the group members to the next bucket.

APPENDIX B – PROMPT QUESTIONS

1. Look at the requirements written. What other requirements or features do they make you think of?
2. As you look at the features described on the page, think what they enable the user to do. What will the user do before or after? Are their features to support those activities? Try to think from the beginning to the end of your experience with the website.
3. Look at the features in the list. Are their features missing that would need to be included to support those features?
4. Think about the goals that the features of the stories support. For example, a user may want to manage his profile. What functionality is needed to support this goal? What other features would be needed to support those goals?
5. Are there any details missing from the stories on your list? Elaborate on information that is missing from the features.