Students' perceptions of user stories

V. Mahnic

University of Ljubljana
Ljubljana, Slovenia

ABSTRACT: In order to acquaint students with agile methods for software development, a new capstone course was designed at the University of Ljubljana, Ljubljana, Slovenia, that introduces Scrum as a framework for the planning and management of students' projects. Students must develop a quasi-real project on the basis of requirements, which are formulated as user stories; a lightweight technique advocated by agile methods. A survey was conducted among students in order to find out: 1) how students rate the use of user stories within the course; 2) how their opinions change when they obtain more experience; and 3) what the most important factors are that affect the use of user stories for requirements specification. Results indicate that students' opinions mostly are positive and improve when they gain more experience, thus, confirming that agile methods are best taught through practical work. Team-project work enables students to successfully grasp the main concepts behind user stories and recognise the importance of good communication between the *Scrum Team* and the *Product Owner* for their successful use in practice.

INTRODUCTION

Agile software development methods take a novel, lightweight approach to requirements specification [1]. It is assumed that full, detailed requirements for a software package cannot (and need not) be developed as these requirements inevitably will change over time. Instead, each requirement is recorded as a user story, consisting of three parts: a written description (used for planning and as a reminder); conversations about the story (to flesh out the details); and acceptance tests (to determine when a story is *done*) [2].

A story description is formulated in the language of the customer and is intentionally short enough to be hand-written on a card. It serves merely as a reminder for conversations with the customer in order to clarify the story details and document the project users' expectations of acceptance tests. The written description typically follows a simple template:

As a <type of user>, I want <some goal> so that <some benefit>.

As such, user stories strongly shift the focus from writing about features to discussing them.

In order to prepare students for an increasing use of agile methods in industry, teaching these methods is becoming an important part of the computer science and software engineering curricula. Experience has shown that the best results are achieved if teaching is done through practical work [3]; therefore, the capstone course seems to be an appropriate place for the introduction of agile software development. In light of the above, a new software engineering capstone course [4] was designed at the University of Ljubljana, Slovenia, that introduces Scrum as a framework for the planning and management of students' projects, and uses user stories for specifying required functionality [5][6].

Scrum was chosen since it is the most widespread agile method and its use is increasing in popularity. According to the latest State of Agile Survey, it is used by 66% of agile practitioners [7]. The course lasts 15 weeks and consists of Sprint 0 (a preparatory Sprint lasting 3 weeks) and three regular Scrum Sprints, each of them lasting 4 weeks. Students work in groups of four, each group playing the role of a Scrum Team, which is collectively responsible for developing the required functionality. During Sprint 0, students are presented with the initial Product Backlog consisting of user stories, that must be implemented by the end of the course.

The stories are written and prioritised by a domain expert (the teacher or a representative of a co-operating company) playing the role of the Product Owner. A member of the teaching staff also plays the role of the ScrumMaster, taking care that everybody obeys Scrum rules and practices. A sample user story is shown in Figure 1.

As a clerk in the student records office I want to enrol a student so that data for all students are complete and available for further processing.

Test that a student can be enrolled only if all required data are present and valid.

Check the student matriculation number for freshmen (no duplicates).
Check all codes (postcode, municipality code, country code, study programme code, department code) and dates.
Check name and surname (letters only).

Test for different years of study and different study programmes/modules.
Test invalid combinations of the academic year, year of study, and study programme/module.
Test that the mandatory courses are allocated automatically.

Figure 1: A sample user story.

At the end of Sprint 0, each team estimates the stories using planning poker and prepares the release plan [8]. Following the Scrum method, each Sprint starts with a Sprint planning meeting at which student teams negotiate the contents of the next iteration with the Product Owner, and develop the initial version of the Sprint Backlog. During the Sprint the teams have to meet regularly at the Daily Scrum meetings and maintain their Sprint Backlogs, adding new tasks if required and updating data on work spent and work remaining. At the end of each Sprint, the Sprint review and Sprint retrospective meetings take place. At the review meeting the students present their results to the instructors, while at the retrospective meeting students and instructors meet to review the development process in the previous Sprint, giving suggestions for improvements in the next. After three Sprints, the first release should be complete and delivered to the customer.

The aim of this paper is to analyse students' perceptions of user stories and present factors affecting their use within the scope of a course. When conducting the course for the first time in the Academic Year 2008/09 [9], the author noticed that students' opinions were mixed: while some of them grasped the essence of the lightweight approach to requirements specifications, many of them complained that user stories did not provide enough information regarding desired functionality. In the next two years special attention was devoted to this issue and an on-going communication with the Product Owner was strongly encouraged to promote the awareness that the purpose of user stories is not to document details, but to serve as a reminder for conversation. In order to analyse this problem in more detail, the author surveyed students who attended the course in the Academic Years 2009/10 and 2010/11. The aim of the survey was to find out: 1) how students rate the use of user stories in the course; 2) how their opinions change when they obtain more experience; and 3) what are the most important factors that affect the use of user stories for requirements specification.

In the Academic Year 2009/10, the survey concentrated only on the first two of the aforementioned goals and was conducted twice, at the end of Sprint 1 and at the end of Sprint 3, thus, providing the possibility to study the before and after effect by comparing grades after each of the two Sprints. It was expected that the answers would help to improve the teaching of the course in subsequent years and show how students' opinions change when they get more practice. In the Academic Year 2010/11, the survey was extended so as to also gather students' opinions regarding the use of user stories for requirements specification in general.

The survey consisted of two questionnaires: in addition to the first questionnaire, which remained the same as in the previous year, a new questionnaire was added, asking students to rate their agreement with a set of assertions describing the benefits of user stories and justifying the lightweight approach to requirements specification. The new questionnaire was used at the end of the course and it was expected that it would help to identify those factors that (according to students' opinions) affect successful use of user stories in general, as well as to analyse how well the students grasped the main ideas behind the lightweight approach to requirements specification.

The remainder of the paper is organised as follows: the next section describes answers to the first questionnaire and analyses the difference between students' opinions after the first and the third Sprint. Then, the answers to the second questionnaire are presented and discussed. Finally, the most important conclusions are summarised.

STUDENTS' SATISFACTION WITH USER STORIES

To analyse students' satisfaction with the use of user stories in the capstone course, the students were asked to answer seven questions about appropriateness of user stories and other activities and artefacts associated with their use. The questions had to be answered using a 5-point Likert scale, grade 1 representing the most negative (strong *No*) and grade 5 the most positive opinion (strong *Yes*). The survey was anonymous. In the Academic Year 2009/10, the number of respondents was 52 (the survey was answered by all students enrolled in the course), while in the Academic Year 2010/11 the survey was answered by 42 students out of 51 enrolled in the course. Results are presented in Table 1. Columns 2009/10 and 2010/11 represent average (mean) grades after Sprint 1 and Sprint 3 for each academic year.

The first question asked whether user stories together with acceptance tests represent an appropriate way to specify requirements for students' capstone projects. Given that the Product Backlog was simply a set of user stories, the aim of the second question was to find out if the Product Backlog was clear enough to understand what the Product Owner really wanted. Scrum requires that the effort to implement each user story be estimated using planning poker [8]. Therefore, the third question was intended to gather student opinions regarding the appropriateness of this technique for estimating the size of user stories. The fourth question required students to evaluate the accuracy of their effort estimates. Accurate effort estimates are a prerequisite for adequate release and Sprint planning. It was expected that the accuracy of students' estimates would improve from Sprint to Sprint.

The fifth question dealt with co-operation with the ScrumMaster. The ScrumMaster is responsible for the appropriate flow of the Scrum process, which also includes the appropriate use of user stories. Therefore, good co-operation between the SrumMaster and students is very important for the success of a project. However, the role of the Product Owner is even more important since it is the Product Owner who writes user stories, maintains the Product Backlog and provides details regarding the implementation of each user story. It is important that he/she promptly answers questions posed by student teams and makes quick evaluations of the work being done. Co-operation between students and the Product Owner was evaluated by Question 6. Finally, the seventh question required students to provide their opinions about the Scrum method as a whole.

Table 1: Results of the survey at the end of Sprint 1 and Sprint 3 (mean on 5-point Likert scale).

Question		2009/10		2010/11	
		After Sprint 1	After Sprint 3	After Sprint 1	After Sprint 3
1	Appropriateness of user stories	3.75	4.04	3.82	3.93
2	Clarity of the Product Backlog	4.06	4.42	3.80	3.79
3	Appropriateness of planning poker	3.14	3.44	3.31	3.60
4	Accuracy of effort estimates	3.00	4.02	2.63	3.42
5	Co-operation with the ScrumMaster	4.47	4.67	3.61	4.14
6	Co-operation with the Product Owner	4.39	4.67	3.29	4.07
7	General evaluation of Scrum	3.78	4.04	3.90	3.90

Students' opinions were most positive with regard to Questions 5 and 6 concerning co-operation with the ScrumMaster and Product Owner. In the Academic Year 2009/10, their grades were very high after Sprint 1 and became even higher after Sprint 3. In the Academic Year 2010/11, these grades were not as high after Sprint 1, but increased significantly after Sprint 3. It seems that in the Academic Year 2009/10, the students recognised very early that agile projects require good communication between all parties involved, while the students enrolled in the Academic Year 2010/11 needed more time to become aware that the lightweight approach to requirements specification is only feasible if the details of user stories are clarified in conversations with the Product Owner.

The students admitted that there was not enough initiative for such conversations on their part at the beginning of the project, but the situation improved after Sprint 1, when the instructors additionally motivated them to increase co-operation with the ScrumMaster and, especially, the Product Owner. On the other hand, the high grades of Questions 5 and 6 indicate that the course was well prepared and conducted, which is very important when user stories are introduced to novices.

The results also show that the majority of grades increased after Sprint 3 clearly indicating that students' opinions about user stories improve when they get more experience. In the Academic Year 2009/10, the grades of all questions were higher at the end of the project than after Sprint 1. In the Academic Year 2010/11, the trend was similar: the grades of five questions increased, while the grades of two questions remained approximately the same. This fact additionally supports the hypothesis that the best way for imparting agile methods to students is through practical work. The students can fully grasp the essence of the agile approach only after trying it in practice.

Additionally, the results indicate that students' opinions regarding user stories and related issues were mostly positive. In both academic years, the average grades of all questions except Question 4 were on the positive side of the Likert scale after Sprint 1. The results were even better at the end of the project when the grades were between 3.42 and 4.67. Students considered user stories and acceptance tests appropriate for requirements specification (Question 1) and were mostly satisfied with Scrum (Question 7).

Their positive attitude towards user stories is also evident from their answers to Question 2, which indicate that the Product Backlog (which was simply a set of user stories) described user requirements in an adequate way. Nevertheless, it can be noticed that the grades were higher in the Academic Year 2009/10 than in 2010/11. This phenomenon is difficult to explain since the instructors, the course content and the project that students had to develop were the same. It seems the only difference was in the students themselves, who (according to instructors' observations) showed more enthusiasm and put more effort into their projects in 2009/10 than in 2010/11.

Questions 3 and 4 were rated lowest; however, the grades improved after Sprint 3. With regard to Question 3, it seems students were unsure how to rate the planning poker technique since most of them had no previous experience with any of the effort estimation methods. Consequently, their estimates were quite inaccurate at the beginning, but improved substantially after Sprint 3 (Question 4). Considering that Question 4 required students to rate the accuracy of their estimates, the low grades for this question do not represent a deficiency of planning poker and user stories as such, but indicate students' inability to provide accurate estimates when these techniques are used for the first time. A detailed study of students' abilities of estimating and planning revealed that after three Sprints almost all teams were able to define accurate Sprint plans [10]. However, the author found that planning poker, when used by students, increases over-optimism instead of providing more realistic estimates [11].

BENEFITS OF USER STORIES AND FACTORS THAT AFFECT THEIR USE

The second questionnaire consisted of 12 assertions describing the benefits of user stories and prerequisites for their successful application in agile projects. For each assertion students were asked how much they agree (or disagree) using a 5-point Likert scale (1 - *Strongly disagree* to 5 - *Strongly agree*). On the one hand, the questionnaire helped to determine the extent to which the students grasped the main concepts of user stories; on the other hand, it made it possible to identify those factors that (according to students' opinions) most affected their use. The survey was conducted at the end of the course in the Academic Year 2010/11 and was answered by 50 students out of 51. Results are presented in Table 2.

Table 2: Results of the survey at the end of the course (mean on 5-point Likert scale).

Assertion		Mean
1	User stories significantly shorten the time to write requirements specifications (compared to other more detailed techniques).	3.98
2	User stories describe the requirements in a language that is comprehensible to the end user.	4.40
3	User stories require ongoing and intensive communication with the Product Owner.	4.38
4	Even a very detailed specification does not prevent misunderstandings between the customer and developers.	4.02
5	Comunication with the Product Owner prevents misunderstandings that can occur even with the most detailed specification.	4.18
6	Appropriate decomposition of user stories into individual tasks (just-in-time design) can significantly contribute to better understanding of user requirements and the timely resolution of ambiguities.	3.38
7	Communication with the Product Owner (in order to clarify details) requires significantly less time compared to the development of detailed requirements specification.	3.54
8	The decision to use user stories for requirements specification largely depends on the availability of a responsive and competent Product Owner.	4.52
9	User stories encourage deferring details. Details are added when the implementation of a story starts.	4.14
10	User stories are useful for project planning because they describe components of appropriate size that are convenient for estimation, programming and testing.	3.78
11	User stories require more skilled and experienced developers than other, more detailed, techniques.	3.12
12	In most cases you can count on the Product Owner or customer representative to write user stories and maintain the Product Backlog.	3.74

Students agreed most with Assertions 8, 2 and 3. Their answers indicate they successfully grasped that the use of user stories requires a competent and responsive Product Owner (Assertion 8), since an ongoing and intensive communication with him/her is needed in order to clarify the details of each user story (Assertion 3). A nonresponsive Product Owner can cause unproductive work periods, which make iteration planning more difficult or even impossible. Students also recognised that communication between the development team and the Product Owner is facilitated by the fact that user stories describe requirements in the language of the customer (Assertion 2). However, the Product Owner should be sufficiently knowledgeable about Scrum to be able to write, maintain and prioritise user stories. The author was rather surprised that students were quite optimistic with regard to this problem. Results of the survey revealed a prevailing opinion that in most cases the developers can count on the customer representative (i.e. the Product Owner) to do this job (Assertion 12).

Students also agreed with Assertions 5, 9, 4 and 1. Being short enough to be written on a paper note card, user stories serve merely as reminders for conversation, thus, deferring details until the implementation actually starts (Assertion 9). This reduces the amount of time needed for writing requirements specification compared to other more detailed techniques used by disciplined approaches (Assertion 1). Students agreed that even a very detailed requirements

specification cannot prevent misunderstandings between the customer and developers (Assertion 4). Instead, communication with the Product Owner provides a better way for preventing misinterpretation of customer needs (Assertion 5).

There was only moderate agreement with Assertion 10 that user stories are useful for project planning, although the suitability for planning is often cited as one of the main advantages (e.g. [2] p. 148). The author thinks this is a consequence of the fact that students did not have enough experience in planning real projects to recognise that user stories describe the components of appropriate size that are convenient for estimating, programming and testing. Additionally, computer science students prefer technical activities and often perceive planning as less important and uninteresting.

In spite of significant agreement with Assertions 1, 4 and 5, the students expressed rather weak support to Assertion 7 that conversations with the Product Owner, which are necessary to clarify the details of user stories, require less time than the development of detailed requirements specification. The author expected stronger agreement; however, this result also can be attributed to the fact that students did not have enough experience with working on real projects and were, consequently, more cautious in their opinions.

The students agreed least with Assertions 6 and 11, although their grades still were on the positive side of the Likert scale. Assertion 6 referred to just-in-time design as advocated by agile software development methods. It was assumed that most questions regarding the details of user stories arise (and can be answered immediately by the Product Owner) during the Sprint planning meeting when developers decompose stories into constituent tasks in order to develop the Sprint Backlog. However, students' answers provided just weak support to this assumption.

Assertion 11 was based on Boehm and Richardson's claim that the agile approach to software development requires more skilled and experienced developers than the traditional disciplined approach [12]. Although the author fully agrees with Boehm and Richardson, it seems the students were rather perplexed when answering this question, which indicates that the difference between both approaches should be emphasised more during the course.

CONCLUSIONS

Students' answers to both questionnaires indicate that:

1. Their opinions regarding the use of user stories in the capstone course were mostly positive.
2. Their opinions improved when they gained more experience, thus, indicating that the lightweight approach to requirements specification is best taught through practical project work.
3. Students successfully grasped the main concepts behind user stories and recognised the importance of communication between the Scrum Team and the Product Owner for their successful use.

With regard to the first questionnaire, the author was gratified that the students rated the highest, co-operation with the ScrumMaster and Product Owner. This indicates the course was well designed and the instructors who played these roles contributed significantly to the successful completion of the students' projects. The fact that the students' satisfaction increased after gaining more experience confirmed once more that the decision to teach agile methods through team-project work was right. Only practical experience with user stories helps to overcome initial doubts regarding their usefulness and, especially, claims that user stories are too vague and do not provide enough details regarding user requirements.

Answers to the second questionnaire clearly indicate that the students recognised the role of the Product Owner to be crucial for the successful application of user stories in a Scrum project. User stories can only be applied if a responsive Product Owner, knowledgeable about Scrum, is available, and good communication between the Product Owner and the development team is ensured. Intensive collaboration between the customer representative and developers is inevitable since misinterpretations of user requirements occur even in the case of extensive written documentation. The students also agreed that user stories significantly shorten the time to write requirements specifications (compared to other more detailed techniques) since the details are deferreed until the implementation of a story actually starts.

On the other hand, the survey revealed some areas that deserve more attention in future. The accuracy of effort estimates provided by student teams need to improve and the awareness of importance of just-in-time design by decomposing the stories into tasks needs to increase. Students should exploit the Sprint planning meeting not only to agree about user stories that should be implemented in the next Sprint, but also to fully clarify every detail of each story.

REFERENCES

1. Williams, L., Agile software development methodologies and practices. *Advances in Computers*, 80, 1-44 (2010).
2. Cohn, M., *User Stories Applied for Agile Software Development*. Boston: Addison-Wesley (2004).
3. Devedzic, V. and Milenkovic, S.R., Teaching agile software development: a case study. *IEEE Transactions on Educ.*, 54, **2**, 273–278 (2011).

4. Mahnic, V., A capstone course on agile software development using Scrum. *IEEE Transactions on Educ.*, 55, **1**, 99-106 (2012).
5. Schwaber, K. and Beedle, M., *Agile Software Development with Scrum*. Upper Saddle River: Prentice Hall (2002).
6. Schwaber, K., *Agile Project Management with Scrum*. Redmond: Microsoft Press (2004).
7. VersionOne, State of Agile Survey 2011 (2011), 9 October 2012, http://www.versionone.com/pdf/2011_State_of_Agile_Development_Survey_Results.pdf.
8. Grenning, J., Planning poker or how to avoid analysis paralysis while release planning (2004), 23 October 2012, http://www.renaissancesoftware.net/files/articles/PlanningPoker-v1.1.pdf
9. Mahnic, V., Teaching Scrum through team-project work: students' perceptions and teacher's observations. *Inter. J. of Engng. Educ.*, 26, **1**, 96-110 (2010).
10. Mahnic, V., A case study on agile estimating and planning using Scrum. *Electronics and Electrical Engng*, 111, **5**, 123-128 (2011).
11. Mahnic, V. and Hovelja, T., On using planning poker for estimating user stories. *J. of Systems and Software*, 85, **9**, 2086-2095 (2012).
12. Boehm, B. and Turner, R., *Balancing Agility and Discipline, A Guide for the Perplexed*. Boston: Addison-Wesley (2004).