On Developing an E-Assessment Tool for Agile Practices

Kerstin Jacob[1], Daniel Hallmann[1], Gerald Lüttgen[1], Vanessa Völpel[2]

Abstract: As agile software development processes are widely applied in industry, students need to develop a good understanding of agile principles and practices as part of their education. During our ten years of experience with teaching the Scrum framework in team project modules, we noticed that students frequently struggle with writing good user stories and concise sprint goals. To support students in mastering basic practices and to free limited teaching resources for more in-depth discussions on advanced topics, e-assessment tools should be developed to provide timely feedback on at least common mistakes. This paper argues that, and sketches how, research on quality criteria for artifacts of agile development can serve as the basis for such a tool. However, multiple challenges remain that we invite lecturers and researchers to discuss.

Keywords: agile software development; e-assessment tool; Scrum

1 Introduction

Agile development processes are ubiquitous in the software industry today. Therefore, it is highly relevant for computer science students to develop a thorough understanding of agile principles and practices during their software engineering education. To meet this need, our Chair replaced the waterfall model with a Scrum-based model [SS20] in student team projects about ten years ago. In particular, the Bachelor project modules cover multiple skills beyond programming practices, such as discussing requirements with stakeholders, casting requirements into user stories, and estimating user story sizes and priorities. Giving students the practice they need in order to learn these skills and to build a habit of agile thinking requires time and intensive mentoring by lecturers.

We have observed that a vast amount of the time spent in supervision meetings with students is devoted to the same basic practices: phrasing of user stories and sprint goals, and splitting software features sensibly into user stories. Not only do students struggle with these topics, but professionals do, too [Co04]. Thus, it is important to find ways to support students with the writing of these artifacts, and provide formative feedback [Sh08]. Because contact hours between lecturers and students are limited due to the difficulty of finding sufficiently many tutors, we believe that e-assessment tools are needed that identify at least the more shallow mistakes and highlight them to students.

However, generating feedback on sprint goal phrasing and user story writing in an automated manner is challenging due to the diverse and often unstructured textual nature of these

[1] University of Bamberg, Software Technologies Research Group, Germany, firstname.lastname@uni-bamberg.de
[2] University of Bamberg, Germany, vanessa@voelpel.de

doi:10.18420/seuh2023_01

artifacts of interest. While it is difficult to define clear and measurable quality criteria, there are multiple guidelines (see, e.g., [Co04; Je01; Wa03]) derived from practical experience that have been taken up and evaluated by the scientific community (see, e.g., [Ha20; Lu16]) and that may provide a basis for the desired e-assessment tool.

This paper states our vision for the e-assessment tool on agile practices (Sect. 2) and reports on our current prototyping efforts (Sect. 3). Our approach is built upon natural language processing (NLP) techniques [JM09] and informed by recent research on measuring user story quality [Ha20]. While our tool is not yet complete and has not been deployed in our teaching, we encountered multiple challenges and open research questions during prototyping (Sect. 4) to which we believe the SEUH community can contribute.

2 Vision

We introduce the concepts of agile practices in an introductory lecture module to software engineering and practice the writing of user stories and sprint goals with students in accompanying practicals. Additionally, we start off our project modules with tutorials on these practices. Nevertheless, students in our team projects still struggle with the writing of user stories and sprint goals throughout the semester.

User Story 1: Database Implementation	**Sprint goal:**
As a user, I want to save and load things to and from a database. The story is done, when - a current plan can be saved on a MariaDB - all entities are correctly related to each other and the relations in the database reflect the planned design of the developers	Implement the basic functionalities, such as database connection, JSON imports and the basic view so that we have a reliable, stable foundation on which we can build and so that we will be able to deliver a high quality product.

Fig. 1: Exemplary user story and sprint goal suggested by a student team.

Fig. 1 (left) shows an exemplary user story from one of our student teams, which includes multiple common mistakes: the user story misses the rationale ([why]), and thus does not fully follow the pattern "[title]–As [persona], I will [what] so that [why]–[acceptance criteria]–[attachments]"[1]. The story does not describe a vertical application slice, but concerns only the database connection and uses technical terms such as "MariaDB" instead of the language of the customer. Additionally, the story includes unspecific terms such as "things" and is partly phrased in an unreadable manner (e.g., the long second acceptance criterion).

[1] https://www.agilealliance.org/glossary/user-story-template (last accessed: 28 Oct 2022)

User stories are a central artifact based on which a team estimates effort, creates an architecture, builds the system, and sets up the test strategy. Thus, a poorly phrased user story can lead to lengthy discussions and divergent views between team members, which increases the risk of incorrect feature implementation and expensive rework [Co04].

Similar to user stories, a well-phrased sprint goal is essential to communicate to stakeholders why the sprint is valuable, and to provide focus in the sprint meetings. However, we observe that students stumble over formulating sprint goals. Fig. 1 (right) shows a sprint goal as suggested by one of our student teams. It contains mistakes similar to the aforementioned user story: unspecific phrasing (e.g., "Implement the basic functionalities"), low readability (e.g., due to the long nested sentence), and usage of technical keywords (e.g., "JSON imports").

We envision an e-assessment tool that supports the students by detecting such common mistakes and giving formative feedback with actionable suggestions on how to improve the artifacts. While summative feedback is given at the end of a module to assess the students' learning outcomes, formative feedback is provided to the students throughout the module to improve their learning process. Because a sprint in our project module is only two weeks long and included in the normal semester schedule with limited hours each week, it is very important that students receive feedback on artifacts in a timely manner so that they can improve them while the sprint is still in progress. This is difficult with supervision sessions that are only conducted weekly, but can likely be accomplished by an e-assessment tool. The application of the e-assessment tool shall leave more time in the supervision meetings for discussing deeper aspects of agile practices, such as user story splitting and size estimation, and for reflecting on the underlying agile principles.

3 Approach

Our current e-assessment prototype receives user stories and sprint goals as well as meta-data such as authors and creation date from our process management tool (GitLab[2]), and outputs an email with textual feedback to the artifacts' authors. The tool's quality measurements are based on recent research [Ha20] on the quality of artifacts of agile development. In particular, the measurements use NLP techniques [JM09] to analyze textual information, and GitLab meta-data such as revision counts. This section details the employed quality criteria, their measurement, and the feedback generation.

3.1 Quality Criteria and their measurement

There exists a variety of guidelines that can aid authors in writing high-quality user stories, e.g., Cohn's guidelines [Co04], the INVEST criteria by Wake [Wa03], or CCC (Card,

[2] https://about.gitlab.com/solutions/agile-delivery/ (last accessed 28 Oct 2022)

Conversation, Confirmation) [Je01], which have predominantly been derived by consultants from their practical experience. An empirical study [Lu16] found that practitioners consider the principles helpful for creating a shared understanding and increasing productivity within the team, although the principles are highly qualitative and thus difficult to assess automatically.

Recently, Hallmann [Ha20] defined quality criteria for user stories, which are unique in that their measurements can be automated. However, the practical utility and validity of this work has only been assessed preliminarily. To address this shortcoming, and in parallel to our e-assessment prototype, we have initiated an expert study with practitioners from industrial development projects to evaluate Hallmann's criteria and their measurements. Hallmann has selected the following four quality criteria based on a literature review and brainstorming sessions with experts:

Complete: This quality criterion captures whether all form fields of the user story template by Connextra[3] have been filled.

Readable: This quality criterion measures the readability of a given user story using the "Flesch reading ease" [Fl48], which estimates the readability of a document based on the average sentence length and word length.

Valuable: This quality criterion describes the business value that a user story represents. The measurement is focused on the usage of domain language and calculates the percentage of domain words relative to the story's total number of words.

Saturated: A user story should be the basis of discussion between the development team and the stakeholders, and also within the team, and be continuously refined. This quality criterion depicts how often such refinements have taken place, based on data retrieved from the employed process management tool.

Measuring the stated criteria is non-trivial due to the semi-structured (in the case of user stories) and unstructured (sprint goals) text formats. To measure the quality criterion *complete*, we have implemented a pattern matcher using the NLP libraries *spaCy*[4] and *sklearn*[5] to retrieve the individual parts such as the persona or the rationale from a user story written in German or English. Pattern matching typically succeeds as long as the text rather strictly follows the template. However, it quickly fails if the structure of the inputs differs from the expected pattern, e.g., given sentence constructs such as "I, as an experienced user" or "As a doctor or pharmacist". Thus, we have additionally experimented with a supervised learning approach using a tokenizer, a vectorizer, and a support vector machine as a classifier, with promising first results.

Similarly, we employ a pattern matcher to measure the quality criterion *valuable*, which

[3] https://www.agilealliance.org/glossary/user-story-template (last accessed: 28 Oct 2022)
[4] https://spacy.io/usage/spacy-101 (last accessed: 28 Oct 2022)
[5] https://scikit-learn.org/stable/ (last accessed: 28 Oct 2022)

relies on finding domain-specific terms in the story text. To identify such terms, we use a vetted glossary generated during the student project blastoff. In addition to measuring the degree of domain words as is done in [Ha20], we also identify technical keywords using a glossary based on the technology stack of our project modules. To calculate the sentence and word length for the quality criterion *readable*, we employ the sentencizer, tokenizer and syllables annotator of the *spaCy* library. Criterion *saturated* is calculated from retrieved GitLab meta-data, and thus does not require NLP techniques.

While Hallmann [Ha20] considers each user story in isolation, we have added a quality criterion *independent*, which calculates the semantic similarity for each pair of user stories in a given set; user stories with high similarity might be duplicates or not functionally independent. We have done so because student teams do, at least at the start of the semester, seldom act as a team and several students come up independently with user stories, which unsurprisingly, are often neither orthogonal nor complementary.

As suggested by the example sprint goal in Fig. 1 (right), the quality criteria *readable* and *valuable* can also be applied to sprint goals. Nevertheless, it is an open research problem how to measure whether sprint goals are phrased in a way that motivates development teams and fosters focus during Scrum meetings.

3.2 Feedback Generation

Interpreting and presenting the results of our measurements for a particular user story or sprint goal, and thus providing meaningful formative feedback, has proven to be particularly challenging. There is a large body of research surveyed in [Sh08] which is concerned with how to provide feedback to students wrt. content, phrasing, timing, and frequency. Our e-assessment prototype currently reports compliance or deviation from the above criteria in an email sent to the artifacts' authors. Evaluating our prototypical feedback generator in a user study with students, and enhancing its feedback capability, is ongoing work.

Unclear value to the user

Your user story has been checked for the quality criterion *valuable*. We detected the use of the technical terms *database* and *MariaDB*, and calculated a domain terminology degree of 0%.

Recall that the aim of a story is to convey value to the user. Thus, the story should be written in the language of the user and avoid technical terms.

Fig. 2: Exemplary feedback for the quality criterion *valuable*.

Fig. 2 shows an exemplary feedback for the quality criterion *valuable*. It consists of a short description of the identified problem, including the concrete markers in the text on which the feedback is based. Additionally, the report explains why following the quality criterion is relevant.

4 Conclusions

Teaching an agile framework like Scrum in a team project module requires timely and constant feedback to students. Due to the scarcity of teaching resources, we developed a prototypical e-assessment tool for automating this feedback process for common, but not semantically deep mistakes made by students when writing user stories and sprint goals. The tool is based on NLP techniques [JM09] and recent research on quality indicators for user stories [Ha20]. Before it can be deployed in teaching, its feedback generator needs to be optimized. It should also be evaluated user stories and sprint goals written in past student team projects.

Multiple questions and challenges remain, which we invite lecturers, researchers, and practitioners to discuss:

(1) Which agile practices are most difficult for students (and practitioners) to master? Which of the practices can be evaluated by an e-assessment tool?
(2) What is a sound methodology for deriving quality criteria from agile artifacts? How can such criteria be measured, and how should measurements be interpreted?
(3) How can e-assessment tools be integrated best in a module's teaching concept? When should feedback be distributed?

References

[Co04] Cohn, M.: User Stories Applied: For Agile Software Development. Addison-Wesley Professional, 2004.

[Fl48] Flesch, R.: A New Readability Yardstick. J. Appl. Psychol. 32/3, p. 221, 1948.

[Ha20] Hallmann, D.: "I Don't Understand!": Toward a Model to Evaluate the Role of User Story Quality. In: XP. Pp. 103–112, 2020.

[Je01] Jeffries, R.: Essential XP: Card, Conversation, Confirmation, 2001, URL: https://www.ronjeffries.com/xprog/articles/expcardconversationconfirmation.

[JM09] Jurafsky, D.; Martin, J. H.: Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, 2nd ed. Pearson Prentice Hall, 2009.

[Lu16] Lucassen, G.; Dalpiaz, F.; v. d. Werf, J.; Brinkkemper, S.: The Use and Effectiveness of User Stories in Practice. In: REFSQ. Springer, pp. 205–222, 2016.

[Sh08] Shute, V. J.: Focus on Formative Feedback. Review of Educational Research 78/1, pp. 153–189, 2008.

[SS20] Schwaber, K.; Sutherland, J.: The Scrum Guide. The Definitive Guide to Scrum: The Rules of the Game, 2020, URL: https://scrumguides.org.

[Wa03] Wake, B.: INVEST in Good Stories, and SMART Tasks, 2003, URL: https://www.xp123.com/articles/invest-in-good-stories-and-smart-tasks.