Agile Requirements Engineering with User Stories

Fabiano Dalpiaz and Sjaak Brinkkemper
Requirements Engineering Lab
Utrecht University
Utrecht, The Netherlands
{f.dalpiaz, s.brinkkemper}@uu.nl

***Abstract*—90% of agile practitioners employ user stories for capturing requirements. 70% of these practitioners follow a simple template when creating user stories: As a ⟨role⟩, I want to ⟨action⟩, [so that ⟨benefit⟩]. User stories' popularity and their simple yet strict structure make them ideal candidates for automatic reasoning based on natural language processing. In our research, we have found that circa 50% of real-world user stories contain easily preventable errors that may diminsh their potential. To alleviate this problem, we have created methods, theories and tools that support creating better user stories. This tutorial combines our previous work into a tool-supported pipeline for working with user stories. After presenting the basics of creating user stories, we explain (1) how to improve user story quality by identifying and resolving quality defects; (2) how to generate conceptual models from user stories; and (3) how to cope with terminological ambiguity and incompleteness. Our toolset is demonstrated with results obtained from 20+ software companies that employ user stories.**

***Index Terms*—Agile requirements engineering, user stories, natural language processing.**

I. Motivation

The widespread adoption of agile software development methods like Scrum and Kanban has led to the rise of user stories in industry. Although 45% of practitioners employ user stories for requirements elicitation [1], many still struggle with effectively applying this method in practice. Our previous research has shown that a significant number of user stories (in our datasets, almost 50%) are ill formulated [2].

This tutorial presents an innovative method for working with user stories that is based on empirical research with dozens of software companies. Our method makes extensive use of natural language processing techniques that are embedded in our developed automated tools.

II. Research Baseline

User stories follow a standard predefined format [3] to capture three aspects of a requirement:

1) *Who* wants the functionality;
2) *What* functionality the stakeholders want from the system;
3) *Why* the stakeholders need this functionality (optional).

A. Quality User Stories

Through an empirical study of user stories from the software industry, we have built the Quality User Story (QUS) framework, a set of 13 quality criteria that user story writers should strive to conform to [4]. An overview of the QUS framework is shown in Figure 1.

Based on QUS, we have developed the Automatic Quality User Story Artisan (AQUSA) software tool, which relies on natural language processing techniques to detect quality defects and suggest possible remedies [5]. AQUSA shows good performance in practical settings with simple types of quality detects, with a precision of 72% and a recall of 93%.

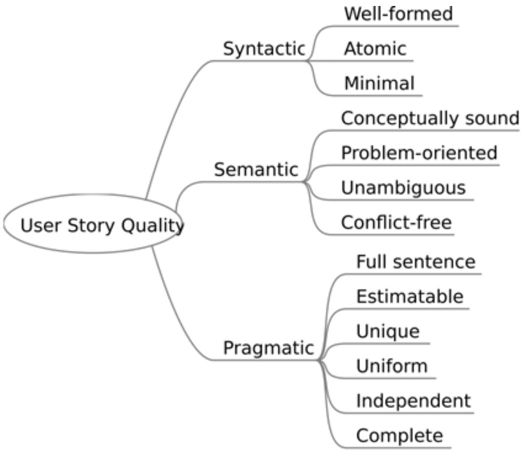

Figure 1: The Quality User Story framework

B. Extracting Conceptual Models

To assist (new) team members in obtaining an overview of the concepts used in a collection of user stories, we have designed and implemented the Visual Narrator tool [6], [7] , which implements 11 heuristics from the NLP literature for extracting conceptual models from user stories. The Visual Narrator (github.com/MarcelRobeer/VisualNarrator) is built in Python and relies on the natural text processor *spaCy* (http://spacy.io), a recent proposal in NLP that implements algorithms needing minimal to no tuning and with excellent performance.

Our application of the Visual Narrator to real-life user story collections shows a very good accuracy of the heuristics [7]: both precision and recall are above 90%.

The outputs of the Visual Narrator can be visualized using the Interactive Narrator tool[1], which adopts information visualization principles (overview, zoom, filter) to help navigate through the many concepts and relationships that are extracted from large collections of user stories. For a screenshot of the Interactive Narrator in action, see Figure 2.

[1] https://interactivenarrator.science.uu.nl/

RE 2018, Banff, Canada
Tutorial

2332-6441/18/$31.00 ©2018 IEEE
DOI 10.1109/RE.2018.00075

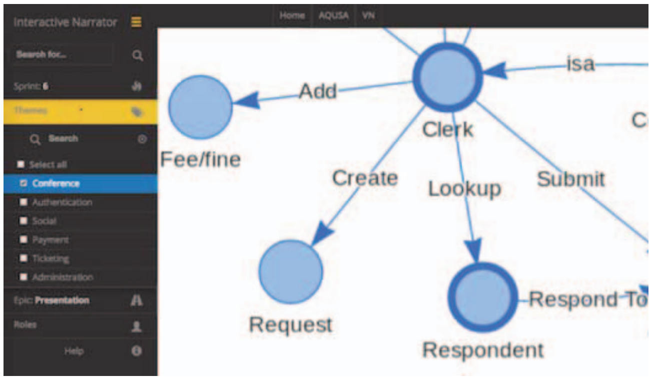

Figure 2: The Interactive Narrator tool

C. *Pinpointing Ambiguity and Incompleteness*

The application of the Visual and Interactive Narrator in practice has shown that additional intelligence is required to assist the analyst in identifying linguistic defects and for improving the collection of user stories. To such extent, we have developed the REVV tool [8] that combines natural language processing and information visualization and assists in pinpointing terminological ambiguity and missing requirements.

The REVV tool (screenshot is in Figure 3, available online: http://www.staff.science.uu.nl/~dalpi001/revv/) leverages two main techniques: (1) an algorithm for identifying near-synonyms that orchestrates state-of-the-art semantic similarity algorithms; and (2) a visualization inspired by Venn diagrams that organizes the terms extracted by the Visual Narrator according to their role (User, Administrator, Manager, . . .), and that emphasizes terminological ambiguity using colors.

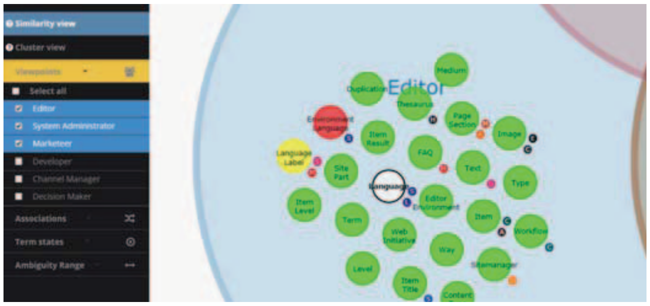

Figure 3: The REVV tool

III. Tutorial Outline

This half-day tutorial is divided into 4 distinct educational learning units in which the audience makes use of the tooling described in Section II.

1) User story basics [45 minutes]: After a brief introduction to the origins and the format, we let the audience create a few user stories, and we discuss the variety of formats and dialects of user stories in industry.
2) User story quality [45 minutes]: Based on our conceptual model of user stories [5], the QUS framework for writing high-quality user stories, and the AQUSA support tool, we explain how to (automatically) detect such defects and discuss strategies for resolving them.
3) Extracting an visualizing user story concepts [45 minutes]: We explain how the Visual Narrator tool [7] can be used to extract conceptual models from a collection of well-formed user stories, and how these models can be used to explore the main concepts used in the requirements using the Interactive Narrator tool.
4) Ambiguity and Incompleteness [45 mintues]: We introduce the REVV tool that enables the identification of potential ambiguity and incompleteness. Besides practicing with the tool, this part of the tutorial will also feature an open discussion about future work.

IV. Conclusion

Through this tutorial, the audience will experience the automated tools—based on natural language processing—that the RE-Lab of Utrecht University has developed in the past few years. Our aim is not only to make the audience aware of the existence and potential of such tools, but also to push the adoption of NLP techniques to assist requirements in their daily tasks [9].

Acknowledgments

We would like to thank all the RE-Lab members who contributed to the research and development of the tools: Garm Lucassen, Marcel Robeer, Jan Martijn van der Werf, Govert-Jan Slob, Ivor van der Schalk, and Fatma Başak Aydemir.

References

[1] M. Kassab, "The Changing Landscape of Requirements Engineering Practices over the Past Decade," in *Proceedings of the International Workshop on Empirical Requirements Engineering*. IEEE, 2015, pp. 1–8.

[2] G. Lucassen, F. Dalpiaz, J. M. E. M. van der Werf, and S. Brinkkemper, "The Use and Effectiveness of User Stories in Practice," in *Proceedings of the International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ)*, ser. LNCS, vol. 9619. Springer, 2016, pp. 205–222.

[3] Y. Wautelet, S. Heng, M. Kolp, and I. Mirbel, "Unifying and Extending User Story Models," in *Proceedings of the International Conference on Advanced Information Systems Engineering (CAiSE)*, ser. LNCS, 2014, vol. 8484, pp. 211–225.

[4] G. Lucassen, F. Dalpiaz, J. M. E. M. van der Werf, and S. Brinkkemper, "Forging High-Quality User Stories: Towards a Discipline for Agile Requirements," in *Proceedings of the International Requirements Engineering Conference (RE)*, 2015, pp. 126–135.

[5] ——, "Improving Agile Requirements: the Quality User Story Framework and Tool," *Requirements Engineering*, vol. 21, no. 3, pp. 383–403, 2016.

[6] M. Robeer, G. Lucassen, J. M. E. M. v. d. Werf, F. Dalpiaz, and S. Brinkkemper, "Automated extraction of conceptual models from user stories via nlp," in *Proceedings of the International Requirements Engineering Conference (RE)*, 2016, pp. 196–205.

[7] G. Lucassen, M. Robeer, F. Dalpiaz, J. M. E. M. van der Werf, and S. Brinkkemper, "Extracting conceptual models from user stories with visual narrator," *Requirements Engineering*, vol. 22, no. 3, pp. 339–358, 2017.

[8] F. Dalpiaz, I. van der Schalk, and G. Lucassen, "Pinpointing Ambiguity and Incompleteness in Requirements Engineering via Information Visualization and NLP," in *Proceedings of the 24th International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ'18)*, 2018.

[9] F. Dalpiaz, A. Ferrari, X. Franch, and C. Palomares, "Natural language processing for requirements engineering: The best is yet to come," *IEEE Software*, vol. 35, no. 5, 2018.