# AQUSA: The Automatic Quality User Story Artisan for Agile Software Development

Garm Lucassen, Fabiano Dalpiaz, Jan Martijn E.M. van der Werf
and Sjaak Brinkkemper
{g.lucassen, f.dalpiaz, j.m.e.m.vanderwerf, s.brinkkemper}@uu.nl

Utrecht University, The Netherlands

**Context and motivation.** User stories are an increasingly popular textual notation to capture requirements in agile software development [6]. User stories only capture the essential elements of a requirement: *who* it is for, *what* is expected from the system, and, optionally, *why*. Popularized by Mike Cohn [2], the most well-known format is: *"As a ⟨type of user⟩ , I want ⟨goal⟩, [so that ⟨some reason⟩]"*. For example: *"As a Marketeer, I want to receive an email when a contact form is submitted, so that I can respond to it"*.

**Question/Problem.** Despite this popularity, the number of methods to assess and improve user story *quality* is small. Existing approaches either employ highly qualitative metrics, such as the six mnemonic heuristics of the INVEST framework [10], or propose generic guidelines for quality in agile RE [4]. This prompted us to introduce the Quality User Story (QUS) framework in earlier work [5], formulating a comprehensive linguistic approach to user story quality. The QUS framework separates the algorithmic aspects that natural language processing (NLP) techniques can automatically process from the thinking-required concerns which necessitate involving human requirements engineers. Our earlier work illustrates each quality criterion with a real-world example to demonstrate that the quality defect occur in practice [5].

**Principal ideas/results.** We take advantage of the potential offered by NLP. However, existing state-of-the-art NLP tools for RE such as Dowser [8] and RAI [3] are unable to transcend from academia into practice because their output is too inaccurate. The ambitious objectives of these tools necessitate a deep understanding of the requirements' contents [1]. This necessity is currently unachievable and will remain impossible to achieve in the foreseeable future [9]. Instead, to be effective, tools that want to harness NLP should focus on the *clerical* part of RE that software can perform with 100% recall and high precision, leaving other work to human requirements engineers [1]. Additionally, they should conform to what practitioners actually do, instead of what the published methods and processes advise them to do [7]. The popularity of user stories among practitioners and their simple yet strict structure make them ideal candidates for applying NLP tools and techniques.

**Contribution.** The Automatic Quality User Story Artisan tool[1] or AQUSA takes a set of user stories as input and outputs errors and warnings that expose possible defects. Specifically, defects are identified by comparing the user stories with a subset of QUS criteria [5]. A first release of this tool was completed in October 2015 and we

[1] https://www.aqusa.nl - visit website for example error report

are currently developing an improved release based on an evaluation with 1,000+ user stories from 18 organizations.

**Demonstration Outline.** The demonstration consists of the following steps where one participant or a small group of participants actively interacts with the researchers and uses the AQUSA tool:

1. *Choose requirements set*: the researcher first invites the participant to create some user stories herself for a special REFSQ dataset, bring her own requirements or choose one of the available data sets that we provide.
2. *Execute AQUSA checks*: next the participant executed the AQUSA automated reasoning techniques, resulting in a report of the quality defects.
3. *Discuss the identified defects*: the researchers asks the participant to explain the intuition behind the defects, and helps the participant understand the rationale when unclear. This way, the theory underlying the protoytpe is discovered by the participant rather than taught in a traditional manner.
4. *Fix defects*: based on the discussion and tool feedback, the participant manually alters the text of a couple user stories to try to fix the identified defects.

## References


1. Berry, D., Gacitua, R., Sawyer, P., Tjong, S.: The Case for Dumb Requirements Engineering Tools. In: Proceedings of International Conference on Requirements Engineering: Foundation for Software Quality (REFSQ), LNCS, vol. 7195, pp. 211–217. Springer (2012)
2. Cohn, M.: User Stories Applied: For Agile Software Development. Addison Wesley (2004)
3. Gacitua, R., Sawyer, P., Gervasi, V.: On the Effectiveness of Abstraction Identification in Requirements Engineering. In: Proceedings of the IEEE International Requirements Engineering Conference (RE). pp. 5–14 (2010)
4. Heck, P., Zaidman, A.: A Quality Framework for Agile Requirements: A Practitioner's Perspective (2014), http://arxiv.org/abs/1406.4692
5. Lucassen, G., Dalpiaz, F., van der Werf, J.M., Brinkkemper, S.: Forging High-Quality User Stories: Towards a Discipline for Agile Requirements. In: Proc. of RE. pp. 126–135. IEEE (2015)
6. Lucassen, G., Dalpiaz, F., van der Werf, J.M., Brinkkemper, S.: The Use and Effectiveness of User Stories in Practice. In: Requirements Engineering: Foundation for Software Quality. Springer International Publishing (2016), accepted for publication
7. Maiden, N.: Exactly How Are Requirements Written? IEEE Software 29(1), 26–27 (2012)
8. Popescu, D., Rugaber, S., Medvidovic, N., Berry, D.M.: Reducing Ambiguities in Requirements Specifications Via Automatically Created Object-Oriented Models. In: Innovations for Requirement Analysis. From Stakeholders' Needs to Formal Designs, LNCS, vol. 5320, pp. 103–124. Springer (2008)
9. Ryan, K.: The Role of Natural Language in Requirements Engineering. In: Proceedings of the IEEE International Symposium on Requirements Engineering (ISRE). pp. 240–242. IEEE (1993)
10. Wake, B.: INVEST in Good Stories, and SMART Tasks. http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/ (2003), accessed: 2015-02-18