Preface: Research on User Stories: Past, Present and Future

Yves Wautelet[1], Palash Bera[2] and Geert Poels[3]

[1] Research Centre for Information Systems Engineering (LIRIS), KU Leuven, Warmoesberg 26, 1000 Brussel, Belgium
[2] Richard A. Chaifetz School of Business, Saint Louis University, 3674 Lindell Blvd, St. Louis, MO 631086, United States
[3] Department of Business Informatics and Operations Management, Faculty of Economics and Business Administration, Ghent University, & CVAMO core lab FlandersMake@UGent, Ghent, Belgium

Abstract

When launching the international workshop on Agile methods for Information Systems Engineering (Agil-ISE) workshop in 2022, the organizers aimed to create an event and build a scientific community dedicated to the study of agile methods and artifacts. Although rooted in the Agile Manifesto, the evolution of industrial methods claiming to be agile has complicated the true nature and meaning of agility. We have consistently argued that only through scientific study can we understand how adopted artifacts and practices genuinely contribute to or hinder agility in project execution and conduct. This preface to the third edition of the Agil-ISE workshop compiles a set of scientific analyses focused on the requirements engineering of agile methods. It particularly highlights research on user stories and behavior-driven development scenarios, aiming to improve the use and quality of these artifacts as well as enhance the conduction of agile projects based on them. This research serves as a relevant illustrative example of how scientific inquiry into agile artifacts and practices can be conducted, and it encourages further research of this nature for the future of the workshop.

1. A Brief History of User Stories Practice and Research

Agile software development methods have been existing for over a quart of a century. These are performing as innovation incubators because they furnish adequate tools to deal with the rapid pace of technological, industrial and environmental change and with the complex needs of users and stakeholders. Since their inception, agile methods have nevertheless had to continually adapt and evolve notably in terms of support for life cycle management but also because of the integration of various techniques often referred to as agile practices [1]. Indeed, they have progressively been used in larger contexts triggering the need for more complex development management and some new approaches destined to favor agility have been discovered and generalized.

As an example of an approach incorporated in agile methods since the early eXtreme Programming (XP) [2], continued by the popular Scrum [3] and nested into the Scaled Agile Framework (SaFe) [4], we can highlight user stories [5]. The latter are short sentences written in structured natural language that encapsulate user needs and expectations into actionable and relatable narratives for development

teams. User stories are relevant and representative within agile development because their creation is a well-known, widespread and documented practice but also their use does also have an heavy impact on the agile project's life cycle management (because they determine and drive Sprint's content).

Since their formal introduction by Mike Cohn in [5], user stories have played a central role in the requirements elicitation process of agile methodologies. This seminal work has not only popularized user stories but also served as first reference for further academic and practical research aimed at optimizing their formulation and use in agile projects. The interaction between academic research and practical application forms a virtuous cycle, wherein scholarly insights lead to enhanced practices, which in turn, pose new questions and challenges for researchers.

Ten years after the book of Mike Cohn, the authors of [6] made a thorough collection and study of the user story templates but also examples both found in academic and professional sources over the internet. The study of the available templates lead to define, study and evaluate all of the mentioned keywords in order to select and only retain a few of them (complementary and non-overlapping) to eventually allow to describe a user story set within each of the necessary dimensions and, for functional aspects, with different granularity levels. Using this work as foundation, [7] approaches a way to build a conceptual representation of user stories called the Rationale Tree. The technique of building a visual model out of them presents itself as an alternative to the user story mapping technique presented in [8]. Also the conceptual model can be used as a way to further transform the user-story-based representation into a UML class diagram [9] or agent-oriented software design [10]. These papers as well as a few others have been part of the Phd thesis of Samedi Heng that was presented on the 27th of February 2017 at UCLouvain. Samedi presented a paper on the forward engineering of user stories and Behavior-Driven Design (BDD) scenarios at Agil-ISE22 in Leuven [11]. His supervisor was Yves Wautelet, who is a co-organizer of Agil-ISE24.

Abhimanyu Gupta conducted related research as part of his PhD project, which he defended on September 5, 2023, at Ghent University in Belgium. He presented his work at the Agil-ISE22 workshop in Leuven [17]. His supervisors, Palash Bera and Geert Poels, are co-organizers of Agil-ISE24. In their literature review, Gupta et al. [18] systematically identify challenges in requirements engineering for system development projects using Agile methods. They surveyed 16 Agile methodology experts to confirm these challenges and uncover new ones not mentioned in the literature. Through thematic analysis, the authors argue that most challenges relate to the primary purposes of conceptual models in software development: enhancing communication and understanding requirements. The paper outlines conditions necessary for the effective use of conceptual models in Agile projects and demonstrates how these conditions can be met by automatically generating conceptual models from requirements specifications, using user stories as an example.

In [19], they develop and evaluate a solution for automatically generating models from user stories. They define four stylized versions of Unified Modeling Language (UML) diagrams—use case diagrams, class diagrams, activity diagrams, and state machine diagrams—as the targets for model generation. Despite their simplified abstract syntax, these stylised UML diagrams provide various perspectives on the software system, which can be useful for requirements and software engineering. The paper introduces an automated model generation approach implemented through a Natural Language Processing (NLP) tool. A key aspect of this approach is the use of the Behavior-Driven Development (BDD) scenario template to document user stories. By using an example set of BDD scenarios as the source for model generation, the paper demonstrates the approach's feasibility. Additionally, an empirical study was conducted with agile software development experts who interactively used the NLP tool under the guidance of the researchers. The study highlights the perceived usefulness of the generated models and identifies their various applications and benefits for requirements analysis, system design, software implementation, and testing in agile projects.

Quality attributes of user stories had, until 2015, been poorly or not studied at all [20]. In [12], Lucassen et al. study how practitioners perceive user stories in the context of requirements engineering. Their research involved a survey of 182 practitioners with follow-up interviews; it highlighted that the use of

user stories, templates, and quality guidelines like the INVEST mnemonic significantly improves productivity and the quality of work deliverables. The study presents 12 key findings on the usage and perception of user stories by practitioners, underscoring the practical benefits and challenges associated with their implementation. In their pioneering work, Lucassen et al. [13] propose the Quality User Story (QUS) Framework, comprising 14 quality criteria for user stories, and introduce the AQUSA tool [14] to enhance user story quality by identifying and correcting defects using natural language processing techniques. Further building on this, their study in [15] refines the QUS framework to include 14 quality criteria and provides a comprehensive evaluation of the AQUSA tool across 18 different organizations. This improves the overall precision and recall in detecting user story defects. The later work offers a more robust and validated approach to ensuring high-quality requirements in agile projects. Further expanding on this research, Lucassen et al. [16] introduces the Grimm Method, which combines the QUS framework with the AQUSA tool. Their multiple case study involving 30 practitioners across three companies demonstrated that while the intrinsic quality of user stories improved, the perceived impact on work productivity and quality was mixed. Notably, the Grimm Method led to more constructive conversations about user stories, reducing unnecessary rework despite the lack of statistically significant changes in project management metrics. These papers as well as a few others have been part of the Phd thesis of Garm Lucassen that was presented on the 30 of February 2017 at Utrecht University. One of Garm's supervisors, Fabiano Dalpiaz co-organized Agil-ISE22 in Leuven. The other supervisor, Sjaak Brinkkemper, acted as a keynote speaker at the same edition of our workshop.

Anis Amna, a participant in Agil-ISE22 in Leuven, focuses her PhD research on specific requirements quality problems arising from the ambiguity of user stories. In [21], she reports on a Systematic Literature Review (SLR) that revealed ambiguity in user stories as an under-researched problem. Consequently, it was unclear how and to what extent this issue, which affects the effectiveness of user stories in supporting system development, had been addressed. In her SLR, Amna reviewed the limited number of studies (n = 36) that investigate or propose solutions for problems related to ambiguity in user stories. From her analysis, she identified three research gaps. First, there is a need for more research on human behaviours and cognitive factors contributing to ambiguity. Second, ambiguity is rarely studied as an issue within a set of related user stories, such as a theme or epic in Scrum. Third, there is a lack of holistic solution approaches that address ambiguity at multiple linguistic levels.

Following the SLR, Anis Amna developed the AmbiTRUS framework for analyzing ambiguity in sets of related user stories. This framework identifies thirteen distinct types of ambiguity problems, classified across four linguistic levels: lexical, syntactic, semantic, and pragmatic. These problems impact the quality of requirements in terms of vagueness, inconsistency, incompleteness, and redundancy. For each type of ambiguity problem, a corresponding user story quality criterion was proposed, which can be verified to detect ambiguity issues. AmbiTRUS was implemented using an NLP/ML-based tool that automatically verifies these quality criteria and provides recommendations for rewriting user stories when criteria violations are detected. The perceived effectiveness and usability of the AmbiTRUS tool were evaluated in a usability study involving thirty-one experienced software developers.

Finally, we can also mention the work of Konstantinos Tsilionis that participated in Agil-ISE22 in Leuven and defended a Phd dissertation on 10 May 2023 at KU Leuven with Yves Wautelet as supervisor. Konstantinos' dissertation describes the conduct of a controlled experiment to analyze two distinct techniques for organizing sets of user stories. Each of these techniques—Rational Tree, a conceptual modeling-driven approach (citation), and User Story Mapping [8] — has its own level of complexity and capability to illustrate dependencies and decompositions [22]. In the experiment one group of master-level students used the Rational Tree method to create artifacts from a set of US, while another group used the User Story Mapping method and studied the result to find out which method of the aforementioned two helps to better understand the software problem. The results show that although the Rational Tree method is initially harder to understand than User Story Mapping, students who received clear, step-by-step instructions were able to produce high-quality representations of the software problem. Konstantinos' dissertation also outlines details the development of a unified

ontological framework designed to simplify the creation of BDD test scenarios in an attempt to reinforce the functionality described in user stories. In particular, the studies described in [23, 24, 25] study how these test scenarios can be directly linked with their respective user stories, thereby improving the overall process of validating requirements when written in an agile manner.

All of these works show very concrete examples of how we can scientifically challenge agile methods and artifacts. This has been the willingness set-up in our workshop from its start [26]. We aim to continue, in this new edition but also for the future ones, to approach agility from a scientific point of view and build-up a research community around this specific topic.

2. Summary of the Agil-ISE'24 Workshop Papers

This section summarises the papers presented at the 3rd International Workshop on Agile Methods for Information Systems Engineering (Agil-ISE'24). Six papers were presented, including this 'editorial' paper of the workshop chairs. Four papers were peer-reviewed by the workshop program committee members and accepted for presentation after revision. For the first time, the Agil-ISE workshop also included an invited 'journal-first' paper. This paper is not included in this volume.

The next paper in this volume, by Jon W. Beard, Veda C. Storey, Binny Samuel, Roman Lukyanenko, Anna Wiedemann, David Schuff, Shawn Ogunseye, Zohra Islamzada and Fereshta Islamzada, is entitled *Agile Development: The Promise, The Reality, The Opportunity*. The promise refers to the origins of Agile Development, as found in the Agile Manifesto, and its conception as a lightweight information systems development approach that embraces uncertainty, changes, learning, and close interaction with customers, as opposed to the highly structured and planned Waterfall approach that is averse to these values. The reality is that nowadays 95% of developers have adopted at least some of the Agile practices in their work. At the same time, the Agile methodology has been adapted to the current complexity of systems development, including DevOps (for continuous development and support) and Scaled Agile approaches like SAFe (for upscaling the self-regulatory small team endeavours promoted by Scrum). The reality also is that Agile approaches come with their challenges and pitfalls including scope creep, technical debt and higher overall costs in the long run. Finally, the opportunity refers to new domains and new technologies for applying Agile development, specifically the emerging domain of Agile Analytics (for Agile development of AI/ML applications), the use of Generative AI to reshape system development practices to boost efficiency and effectiveness, and the reintegration of Conceptual Modeling into Agile Development to facilitate model-driven development, rapid prototyping and low-code development.

The following paper, *Insights on Agile Contracting: Bridging Theory and Practice*, by Bert de Brock, Konstantinos Tsilionis, and Aleksis Mogilnijs, focuses on a specific challenge related to Agile Development, contracting. Due to the inherent flexibility of development projects that follow an Agile approach, negotiating and agreeing upon a contract that fixes scope, budget and schedule at the start of the project is a utopia. The authors conducted a literature review to identify key issues in contracting and the contract types (e.g., T&M – hourly rate with flexible time and scope) that are recommended. They next confront these findings with what is practised. Drawing on both literature and empirical findings, the paper concludes by presenting a 'best practice' contract template.

Next, the paper *Experiences from Combining Merode and Scrum* by Yaimara Granados, Monique Snoeck, Jenny Ruiz and Gheisa Ferreira, presents a concrete implementation of the Agile MERODE method that was presented at the workshop as a 'journal-first' paper (i.e., M. Snoeck and Y. Wautelet, "Agile MERODE: a model-driven software engineering method for user-centric and value-based development," Softw Syst Model, vol. 21, no. 4, pp. 1469–1494, 2022, doi: 10.1007/s10270-022-01015-y). The paper details the use of Agile MERODE in developing a curriculum management system at the Central University of Las Villas (UCLV). Conducted in collaboration with a leading Cuban software company, the project used Scrum as its Agile Development method. Agile MERODE was used to generate functional prototypes at the beginning of each sprint to validate stakeholder

requirements. This approach facilitated stakeholder acceptance without major project disruptions. The study identified challenges in applying model-driven approaches within Agile environments in the Cuban software industry. Future work will address these challenges, aiming to refine software development practices in Cuba.

Finally, *The Impact of Model-Driven Development on Agile Practices within Knowledge-Intensive Systems Engineering* by Ghazaleh Aghakhani, Konstantinos Tsilionis and Sara Shafiee continues the theme of reconciling Agile Development and model-driven system development. The paper focuses on Product Configuration Systems (PCS) and observes that Agile Development varies across software engineering domains, particularly in PCS, where inter-dependent features necessitate detailed pre-planning. PCS development requires explicit knowledge and robust software modelling. The paper investigates the blend of Model-Driven Development (MDD) and Agile methodologies for creating knowledge-intensive systems like PCS. By reviewing three pivotal studies, it assesses the impact of this integration on project management, teaching, and lifecycle processes. The findings reveal both challenges and opportunities, providing insights into best practices and future research areas for effectively combining MDD with Agile practices.

3. Acknowledgements

4. References

[1] S. Kiv, S. Heng, Y. Wautelet, S. Poelmans, M. Kolp, Using an ontology for systematic practice adoption in agile methods: Expert system and practitioners-based validation. Expert Syst. Appl., Elsevier 195 (2022): 116520.
[2] K. Beck. *Extreme programming explained: embrace change*. addison-wesley professional (2000).
[3] K. Schwaber, *Agile project management with Scrum*. Microsoft press (2004).
[4] R. Knaster, & D. Leffingwell, *SAFe 5.0 distilled: achieving business agility with the scaled agile framework*. Addison-Wesley Professional (2020).
[5] M. Cohn, *User stories applied: For agile software development*. Addison-Wesley Professional (2004).
[6] Y. Wautelet, S. Heng, M. Kolp, & I. Mirbel, (2014). Unifying and extending user story models. In *Advanced Information Systems Engineering: 26th International Conference, CAiSE 2014, Thessaloniki, Greece, June 16-20, 2014. Proceedings 26* (pp. 211-225). Springer International Publishing.
[7] Y. Wautelet, S. Heng, M. Kolp, I., Mirbel, & S. Poelmans, Building a rationale diagram for evaluating user story sets. In *2016 IEEE Tenth International Conference on Research Challenges in Information Science (RCIS)* (pp. 1-12). IEEE (2016).
[8] J. Patton, & P. Economy, User story mapping: discover the whole story, build the right product. " O'Reilly Media, Inc." (2014).
[9] Y. Wautelet, S. Heng, D. Hintea, M. Kolp, & S. Poelmans, Bridging user story sets with the use case model. In *Advances in Conceptual Modeling: ER 2016 Workshops, AHA, MoBiD, MORE-BI, MReBA, QMMQ, SCME, and WM2SP, Gifu, Japan, November 14–17, 2016, Proceedings 35* (2016), pp. 127-138, Springer International Publishing.
[10] Y. Wautelet, S. Heng, S. Kiv, & M. Kolp,. User-story driven development of multi-agent systems: A process fragment for agile methods. *Computer Languages, Systems & Structures*, *50*, (2017): 159-176.

[11] S. Heng, M. Snoeck, & K. Tsilionis. Building a Software Architecture out of User Stories and BDD Scenarios: Research Agenda. In *CEUR Workshop Proceedings (CEUR-WS. org)* (2022) (Vol. 3134, pp. 40-46).

[12] G. Lucassen, F. Dalpiaz, J. M. E. Van Der Werf, & S. Brinkkemper, The use and effectiveness of user stories in practice. In *Requirements Engineering: Foundation for Software Quality: 22nd International Working Conference, REFSQ 2016, Gothenburg, Sweden, March 14-17, 2016, Proceedings 22* (2026), pp. 205-222, Springer International Publishing.

[13] G. Lucassen, F. Dalpiaz, J. M. E. Van Der Werf, & S. Brinkkemper, Forging high-quality user stories: towards a discipline for agile requirements. In *2015 IEEE 23rd international requirements engineering conference (RE)* (2015), pp. 126-135, IEEE.

[14] G. Lucassen, F. Dalpiaz, J. M. E. Van Der Werf, & S. Brinkkemper, AQUSA: The Automatic Quality User Story Artisan for Agile Software Development. In *REFSQ Workshops* (2016).

[15] G. Lucassen, F. Dalpiaz, J. M. E. Van Der Werf, & S. Brinkkemper, Improving agile requirements: the quality user story framework and tool. *Requirements engineering*, *21*, (2016) 383-403.

[16] G. Lucassen, F. Dalpiaz, J. M. E. Van Der Werf, & S. Brinkkemper, Improving user story practice with the Grimm Method: A multiple case study in the software industry. In *Requirements Engineering: Foundation for Software Quality: 23rd International Working Conference, REFSQ 2017, Essen, Germany, February 27–March 2, 2017, Proceedings 23* (2017), pp. 235-252, Springer International Publishing.

[17] A. Gupta & G. Poels. The effectiveness of conceptual models auto-generated from a set of user stories written as BDD scenarios: a proposed study. In 1st International Workshop on Agile Methods for Information Systems Engineering (Agil-ISE 2022) 2022 (Vol. 3134, pp. 33-39). CEUR.

[18] A. Gupta, G. Poels & P. Bera. Using conceptual models in agile software development: a possible solution to requirements engineering challenges in agile projects. IEEE Access. 2022 Nov 10;10:119745-66.

[19] A. Gupta, G. Poels & P. Bera. Generating multiple conceptual models from behavior-driven development scenarios. Data & Knowledge Engineering. 2023 May 1;145:102141.

[20] A.R. Amna & G. Poels. Systematic literature mapping of user story research. IEEE Access. 2022 May 9;10:51723-46.

[21] A.R. Amna & G. Poels. Ambiguity in user stories: A systematic literature review. Information and Software Technology. 2022 May 1;145:106824.

[22] K. Tsilionis, J. Maene, S. Heng, Y. Wautelet, & S. Poelmans. Conceptual modeling versus user story mapping: Which is the best approach to agile requirements engineering?. In *International Conference on Research Challenges in Information Science*, 2021 pp. 356-373. Cham: Springer International Publishing.

[23] K. Tsilionis, Y. Wautelet, C. Faut, & S. Heng. Unifying behavior driven development templates. In *2021 IEEE 29th International Requirements Engineering Conference (RE)* 2021, pp. 454-455. IEEE.

[24] K. Tsilionis, Y. Wautelet, & S. Heng. Building a unified ontology for behavior driven development scenarios. In *International Conference on Product-Focused Software Process Improvement*, 2022, pp. 518-524. Cham: Springer International Publishing.

[25] S. Heng, K. Tsilionis, & Y. Wautelet. Building user stories and behavior driven development scenarios with a strict set of concepts: Ontology, benefits and primary validation. In *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing*, 2023, pp. 1422-1429.

[26] P. Bera, F. Dalpiaz, Y. Wautelet. Challenging the Artifacts and Practices Adopted in Agile Software Development. In : *Short Paper Proceedings of the First International Workshop on Agile Methods for Information Systems Engineering (Agil-ISE 2022) co-located with the 34th International Conference on Advanced Information Systems Engineering (CAiSE 2022)*. CEUR Workshop Proceedings, 2022. p. 1-8.