

Challenging the Artifacts and Practices Adopted in Agile Software Development

Palash Bera¹, Fabiano Dalpiaz² and Yves Wautelet³

¹Richard A. Chaifetz School of Business, Saint Louis University, 3674 Lindell Blvd, St. Louis, MO 631086, United States

²Utrecht University, Buys Ballot Laboratory, Princetonplein 5, De Uithof, 3584CC Utrecht, The Netherlands

³Research Centre for Information Systems Engineering (LIRIS), KU Leuven, Warmoesberg 26, 1000 Brussel, Belgium

Abstract

Roughly speaking, agile software development methods include the adoption of an iterative life cycle and of special kinds of artifacts and practices. The life cycle is the result of years of improvement in software development starting from waterfall, going through planned long iterations like in the Rational Unified Process, to finally end-up with short, unplanned sprints. The used artifacts and practices nevertheless deserve more research in order to measure development performance, to analyze their optimal uses, and to determine the opportunity of their integration in a custom agile process. This paper highlights the need for challenging the artifacts and practices from a scientific standpoint. While doing so, we briefly discuss the research conducted in the field user stories—that have often been studied in the literature in combination with conceptual modeling—, before outlining the first edition of the Agil-ISE workshop and discussing future directions.

Keywords

agile development, agile practices, scalability in agile, user stories, Agil-ISE

1. Agility in Information Systems Engineering

The Merriam-Webster defines the word agile as being “marked by ready ability to move with quick easy grace” but also “having a quick resourceful and adaptable character”. These two broad definitions do not relate directly to software development; however, they do emphasize fundamental properties of agile software development. On the one side, they mention the ability to move (or *react*) quickly with an appropriate response to environmental change and, on the other side, the execution of this reaction with adequate resources and mindset.

The concept of agility has taken rise in software development at the start of the 2000s after the creation of the agile manifesto [1] at the *Object-Oriented Programming, Systems, Languages & Applications (OOPSLA)* conference. The core concept related to agility is *value*, more precisely delivering only user-valued features in a quick manner. To this end, agile methods propose a life cycle that is based on short iterations (mostly called sprints) for which (high user-value) features are selected for development, while the development team’s vision is limited to the

Agil-ISE 2022: Intl. Workshop on Agile Methods for Information Systems Engineering, June 06, 2022, Leuven, Belgium

✉ palash.bera@slu.edu (P. Bera); f.dalpiaz@uu.nl (F. Dalpiaz); yves.wautelet@KU Leuven.be (Y. Wautelet)

🌐 <https://www.slu.edu/business/about/faculty/bera-palash.php> (P. Bera); <https://webpace.science.uu.nl/~dalpi001/> (F. Dalpiaz); <https://www.KULeuven.be/wieiswie/en/person/00071732> (Y. Wautelet)

🆔 0000-0002-8364-0972 (P. Bera); 0000-0003-4480-3887 (F. Dalpiaz); 0000-0002-6560-9787 (Y. Wautelet)

© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

next sprint's content only. In parallel, various artifacts and techniques (often called practices) have been developed and been badged as compliant to agile development. Some of these (e.g., pair development, refactoring, user stories, etc.) have been *de facto* integrated into some agile methods' basic description, but no method integrates them all, and only the ones relevant or needed ones should be integrated by the development team in a custom agile project [2, 3].

The success of Scrum and other agile software development methods used in software development has also contributed to the popularization of the concept of agility. In parallel, non-software development teams have started to adapt Scrum to general projects. Also, the trend of agility has gained more abstract levels than the development of a software project only, and entire organizations started to adopt agility as a way to adapt to environmental change; this practice is known as enterprise, organizational or strategic agility [4]. The present paper and workshop focus only on agility as a property of software engineering techniques and projects.

Very handy for innovative software projects of limited size, agile methods have been facing scalability issues. To deal with such issues, the amount of effort related to project management has risen within the core description of agile methods. eXtreme Programming (XP) [5] used a very simple iterative canvas and was mostly focused on the adoption of a set of practices. Later on, Scrum [6] complexified the canvas with an ex-ante sprint, a project backlog, sprint backlogs, etc. Ultimately, the Scaled Agile Framework (SAFe) [7] brings management and coordination practices to a much more complex level by prescribing Scrum teams coordination through a metaphor inspired by train transportation: parallel teams start and stop a Sprint when the train is in the station (so at the same moment) only.

Capitalizing on the agile trend, gigantic frameworks including complex project management did appear, including lots of different agile-related and trendy software engineering practices and intertwining multiple decision levels. These frameworks have gained significant popularity in the industry. Unfortunately, these popular techniques have gained adoption without a reflection on their nature and the development needs, thereby creating environments where the *overall software project agility is threatened*. This leads to questions such as: When is agility hampered by too much project management, team coordination or documentation? How much of these can be considered as just enough to be agile yet dealing with unavoidable constraints? We argue that we need to rethink what agility truly means, and how agile artifacts and practices can be used and integrated in practice for delivering it.

Lots of research in the information systems community has focused on the results and satisfaction of users of software created from agile-based projects [8]. More limited research, however, challenges the intrinsic agile nature of individual artifacts or practices as well as the impact of their integration into the development environment. By conducting engineering-oriented studies, a more significant (i.e., agile) use can be made of the agile artifacts and practices. As we show in Sec. 2, conceptual modeling has been used as an approach to complement agile artifacts and practices, notably in the context of user stories. This approach allows to identify and characterize the fundamental characteristics of artifacts and practices, their interlinking, and the interaction with the software development environment at large. Also, this enables further supporting the use of the agile artifacts once instantiated, in order to gather empirical evidence of the effectiveness in practice.

This paper focuses on an overview of the development of user stories as an example of the research that has been conducted following the path described above. This is meant to be

illustrative rather than exhaustive. The Agil-ISE workshop aims to give voice to and to stimulate discussion between researchers studying, questioning or developing agile artifacts and practices through an approach following the precepts of science.

2. Samples of Research at the Intersection of User Stories and Conceptual Modeling

Considering AI based technologies such as Natural Language Processing (NLP), several research initiatives have been conducted recently. Many of these projects focus on the generation of conceptual models from unstructured text of natural language requirements or structured requirements (e.g., user stories). Honkisz *et al.* [9] proposed a spreadsheet-based technique that converts unstructured textual requirements to semi-structured format to generate process models. Caporale [10] developed a tool that uses a predefined sentence template (“If [CONDITION], then I do [ACTIVITY]”) to derive process models. Herchi and Abdesselam [11] automated the transformation of textual requirements to UML class diagrams using NLP and domain ontology. As requirements are often written in natural language and not in user stories, Vemuri *et al.* [12] developed a probabilistic-based algorithm that identifies actors and use cases by learning from the patterns in requirements.

Lucassen, Robeer and colleagues [13] developed a NLP heuristics based tool, called Visual Narrator, that automatically generates conceptual domain model from a collection of user stories. Based on the work of [14], Mesquita *et al.* [15] introduced an automated tool (US2Star) to convert a set of user stories to i^* goal models. Gunes and Aydemir [16] proposed NLP-based techniques to extract who (role), what (action, object), and benefit information from the user stories and develop goal models. Elallauoi *et al.* [17] created algorithm to automatically generate use case diagrams from user stories. Using NLP, the algorithm identifies the use case concepts (i.e., actor, use case and association) from different parts of the user stories. A similar approach was developed by Azzazi [18] where user stories were segmented by the use of NLP and these segments were assigned to actors, use cases and associations.

Wautelet *et al.* [19, 20] created a reference canvas and algorithms to produce a goal-based representation and functional decomposition (called the rationale tree) out of user stories. The rationale tree can be mapped to agent-oriented concepts to support the forward engineering transformation process. Lucassen *et al.* [21] proposed the Quality User Story framework for assessing the syntactic, semantic, and pragmatic quality of a collection of user stories. Part of this framework is supported by an NLP-based tool called AQUASA. Finally, Snoeck *et al.* [22] build-up an entire software architecture out of a set of user stories and Behavior-Driven Development (BDD) scenarios through a set of well defined rules.

3. Relevant Research Venues

Agil-ISE has been conceived as a response to the lack of a dedicated venue for discussing the roots of agile artifacts and practices within the engineering of software systems. To better characterize its positioning in the research landscape, we provide an overview of some related events that focus on the themes of Agil-ISE.

The *International Conference on Agile Software Development*, also known as the eXtreme Programming (XP) conference, is a large event that mixes practice with research under the umbrella of the Agile Alliance. Agil-ISE aims to kick-start the discussion around agility in the information systems community and encourages the fundamental study and questioning of agile artifacts and practices prior to their application in a real-life environment. This is largely disconnected from the XP conference community which generally speaking focuses more on the practicality of approaches.

At CAISE'21, the *International Workshop on Model-driven Organizational and Business Agility* emerged as a dedicated outlet for discussing aspects regarding agility of an organization. This differs from the scope of Agil-ISE, which is not interested in the agile practices of an organization as a whole, but is rather concerned with agile artifacts and practices for information systems engineering. These two workshops offer complementary perspectives on agility within the CAiSE community. Thanks to its focus at the intersection of information systems and engineering approaches [23], CAiSE is the most suitable venue that can serve as a hub for discussions around the Agil-ISE topics.

There are related workshops that are co-located with requirements engineering conferences, but they do not focus on agility as a goal to attain for software development. Some of these workshops also look into how novel technologies, in particular AI, such as in the *Artificial Intelligence and Requirements Engineering* workshop [24] at the *IEEE International Requirements Engineering conference*, and the *Natural Language Processing for Requirements Engineering* workshop [25] at the *International Working Conference on Requirements Engineering: Foundation for Software Quality*. Neither of these, however, have a focus on methods, artifacts, and techniques used in agile information systems engineering.

4. The First Edition of Agil-ISE

The first edition of Agil-ISE (<https://agilise.github.io/2022/>) was co-located with the 34th International Conference on Advanced Information Systems Engineering, and it took place in the afternoon of June 6th, 2022. We solicited submissions of short papers to reflect ongoing research in the field.

4.1. Workshop's Topics of Interest

The topics of interest that we defined for the first edition of the workshop aimed to provide a comprehensive view on agile information systems engineering, with a focus on artifacts, methods, automation, and empirical work. The non-exhaustive list of topics includes:

- Artifacts in agile ISE
- Techniques for using agile ISE artifacts
- Novel methods for agile ISE
- Conceptual models for agile ISE
- Empirical studies on ISE practices
- Automation in agile ISE, including AI, ML, NLP
- Cognitive aspects of agile ISE (e.g., shared understanding)

- Experience with transition to agile methods in ISE

4.2. Paper Selection Process

Agil-ISE 2022 featured six accepted papers, which were selected from seven submissions. Each of the papers that were presented at the workshop were reviewed by three members of the program committee, and the decisions were taken by the workshop co-organizers based on the reviewers' score. The program committee of Agil-ISE 2022 consists of the following members:

- Pär Ågerfalk, Uppsala University, Sweden
- Fatma Başak Aydemir, Boğaziçi University, Turkey
- Sjaak Brinkkemper, Utrecht University, Netherlands
- Maya Daneva, University of Twente, Netherlands
- Fabian Gilson, University of Canterbury, New Zealand
- Samedi Heng, Université de Liège, Belgium
- Soreangsey Kiv, UCLouvain, Belgium
- Jil Klünder, Leibniz Universität Hannover, Germany
- Marco Kuhrmann, Reutlingen University, Germany
- Isabelle Mirbel, Université Côte d'Azur, France
- Rahul Mohanani, Indraprastha Institute of Information Technology, India
- Sridhar Nerur, University of Texas at Arlington, USA
- Stephan Poelmans, KU Leuven, Belgium
- Marcela Ruiz, Zurich University of Applied Sciences, Switzerland
- Binny Samuel, University of Cincinnati, USA
- Thiago Rocha Silva, University of Southern Denmark, Denmark
- Konstantinos Tsilionis, KU Leuven, Belgium

4.3. Workshop Program

The program included two sessions. The first session focused on *development methods*, and it included three presentations after the workshop introduction:

Agility in Conversational Systems Engineering: Can't we just talk instead of creating all these artefacts? The keynote address by Sjaak Brinkkemper (Utrecht University), engaged the audience to reflect on whether there is still a need to document the IS engineering process, or whether the automated analysis of conversations could replace the artifacts we currently create.

Rapid Software Prototyping from Business Artifacts. This paper by Wahler and colleagues argue for the role of model-driven engineering in the context of agile information systems engineering. They study whether business artifacts such as spreadsheets can be a starting point to automatically generate a prototype.

Watch your (Development) Steps. De Brock's position paper sketches an approach for integrating requirements engineering with agile software development. The proposed

method, which focuses on functional requirements, combines natural language and graphical notations into an end-to-end approach from user wishes to implementation.

The second session focuses on the study of *notations*, with a strong focus on user stories [26], which are the most used notation for expressing requirements in agile software development [27]. This session included four presentations and a workshop wrap-up session:

On Identifying Similar User Stories to Support Agile Estimation based on Historical Data. Duzkiewicz and colleagues propose initial steps toward the construction of a tool that automatically estimates the effort for implementing a user story based on similar user stories that can be found in previous projects.

Controlled Experiments on User Stories' Modeling: Past, Present, and Future. Tsilionis and colleagues review the existing experiments that study the relationship between user stories and conceptual modeling. Based on the past experience of the community, the paper puts forward a planned experiment concerning ambiguity in user stories.

The Effectiveness of Conceptual Models Auto-generated from a Set of User Stories Written as BDD Scenarios: A Proposed Study. Gupta and Poels describe and illustrate their research concerning the automated generation, via NLP tools, of conceptual models from user stories complemented with acceptance criteria.

Building a Software Architecture out of User Stories and BDD Scenarios: Research Agenda. Heng and colleagues propose techniques for transforming user stories and BDD scenarios to different types of agent architectures, and to object-oriented architectures, with the goal of investigating the transition from requirements to architectures.

5. Conclusion and Outlook

This paper motivated the importance of challenging agile artifacts and practices in a scientific way and introduced the Agil-ISE workshop, which aims to serve as a *community building and gathering event* for researchers in the years to come.

The evolution of the workshop topics will depend on the research conducted by the community. We will nevertheless continue to promote the fundamental study and questioning of agile artifacts and practices. For now, we envision that a few directions will play an increasingly prominent role:

- *Conceptual frameworks.* While the *information systems* community (within the umbrella of the Association for Information Systems: <https://aisnet.org/>) has a strong emphasis on theories (see, e.g., [28]), researchers in *information systems engineering* take a more practical approach in which technological innovation takes the priority, as shown in Section 2. We invite the Agil-ISE community to conduct research towards conceptual frameworks and theories on top of which tools and technologies can be built;
- *Informal artifacts.* Two principles of the agile manifesto are “Individuals and interactions over processes and tools” and “Working software over comprehensive documentation”.

These principles are challenged by scaled agile frameworks such as SAFe, which offer heavy methodological guidance. We expect the Agil-ISE community to study how agile principles can be maintained in scaled projects by making effective use of the informal artifacts (user story maps, whiteboard sketches, Kanban boards, ...) that are created;

- *Advanced AI techniques.* The advancements in AI (led by the quick progress in machine and deep learning) are not going to stop, and they will inevitably shape the research in the Agil-ISE field. We expect the use of these techniques not only to support the tasks listed in Section 2, but also to enable information retrieval using communication modalities that differ from pure text, e.g., informal diagrams and conversations.

Acknowledgments

We would like to thank Samedi Heng for setting up and for maintaining the workshop's website, our program committee members for their diligent and thorough work, and Geert Poels for initiating the conversation around organizing this workshop at CAiSE'22.

References

- [1] M. Fowler, J. Highsmith, et al., The agile manifesto, *Software development* 9 (2001) 28–35.
- [2] K. Eilers, B. Simmert, C. Peters, Doing agile vs. being agile-understanding their effects to improve agile work, in: *Proc. of the International Conference on Information Systems*, 2020.
- [3] 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.* 195 (2022) 116520.
- [4] S. Denning, *The next frontier for Agile: Strategic management*, Strategy & Leadership (2017).
- [5] K. Beck, M. Hendrickson, M. Fowler, *Planning extreme programming*, Addison-Wesley Professional, 2001.
- [6] K. Schwaber, J. Sutherland, *The scrum guide*, Scrum Alliance 21 (2011).
- [7] K. Richard, D. Leffingwell, *Safe 5.0 distilled: Achieving business agility with the scaled agile framework*, 2020.
- [8] P. Diegmann, T. Dreesen, B. Binzer, C. Rosenkranz, Journey towards agility: Three decades of research on agile information systems development, in: *Proc. of the International Conference on Information Systems*, 2018.
- [9] K. Honkisz, K. Kluza, P. Wiśniewski, A concept for generating business process models from natural language description, in: *Proc. of the International Conference on Knowledge Science, Engineering and Management*, Springer, 2018, pp. 91–103.
- [10] T. Caporale, A tool for natural language oriented business process modeling, in: C. Hochreiner, S. Schulte (Eds.), *Proc. of the ZEUS Workshop*, volume 1562, CEUR, 2016.
- [11] H. Herchi, W. Abdessalem, From user requirements to uml class diagram, in: *Proc. of the International Conference on Computer Related Knowledge*, 2012.

- [12] S. Vemuri, S. Chala, M. Fathi, Automated use case diagram generation from textual user requirement documents, in: Proc. of the Canadian Conference on Electrical and Computer Engineering, IEEE, 2017.
- [13] G. Lucassen, M. Robeer, F. Dalpiaz, J. M. E. Van Der Werf, S. Brinkkemper, Extracting conceptual models from user stories with visual narrator, Requirements Engineering 22 (2017) 339–358.
- [14] A. Jaqueira, M. Lucena, F. Alencar, C. M., E. Aranha, Using i* models to enrich user stories, in: Proc. of the International i* Workshop, volume CEUR 978, 2013, pp. 55–60.
- [15] R. Mesquita, A. Jaqueira, C. Agra, M. Lucena, F. Alencar, Us2startool: Generation i* models from user stories, in: Proc. of the International i* Workshop, 2015.
- [16] T. Gunes, F. Aydemir, Automated goal model extraction from user stories using nlp, in: Proc. of the IEEE International Requirements Engineering Conference, 2020.
- [17] M. Elallaoui, K. Nafil, R. Touahni, Automatic transformation of user stories into UML use case diagrams using NLP techniques, Procedia Computer Science 130 (2018) 42–49.
- [18] A. Azzazi, A framework using NLP to automatically convert user-stories into use cases in software projects, International Journal of Computer Science and Network Security 17 (2017) 71–76.
- [19] Y. Wautelet, S. Heng, M. Kolp, I. Mirbel, Unifying and extending user story models, in: Proc. of the International Conference on Advanced Information Systems Engineering, volume 8484 of LNCS, 2014, pp. 211–225.
- [20] 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.
- [21] 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.
- [22] M. Snoeck, Y. Wautelet, Agile MERODE: A model-driven software engineering method for user-centric and value-based development, Software and Systems Modeling (2022).
- [23] M. Jarke, M. C. Pham, R. Klamma, Evolution of the cause author community: A social network analysis, in: Seminal contributions to information systems engineering, Springer, 2013, pp. 15–33.
- [24] F. Dalpiaz, N. Niu, Requirements engineering in the days of artificial intelligence, IEEE Software 37 (2020) 7–10.
- [25] F. Dalpiaz, A. Ferrari, X. Franch, C. Palomares, Natural language processing for requirements engineering: The best is yet to come, IEEE software 35 (2018) 115–119.
- [26] M. Cohn, User stories applied: For agile software development, Addison-Wesley Professional, 2004.
- [27] G. Lucassen, F. Dalpiaz, J. M. E. van der Werf, S. Brinkkemper, The use and effectiveness of user stories in practice, in: Proc. of the International Working Conference on Requirements Engineering: Foundation for Software Quality, Springer, 2016, pp. 205–222.
- [28] R. Weber, Evaluating and developing theories in the information systems discipline, Journal of the Association for Information Systems 13 (2012) 2.