

On the Use of ChatGPT to Support Agile Software Development

Palash Bera¹, Yves Wautelet² and Geert Poels³

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

² Research Centre for Information Systems Engineering (LIRIS), KU Leuven, Warmoesberg 26, 1000 Brussel, Belgium

³ Department of Business Informatics and Operations Management, Faculty of Economics and Business Administration, Ghent University, & CVAMO core lab FlandersMake@UGent, Ghent, Belgium

Abstract

So-called agile development has been widely adopted in the software industry for more than 20 years now. Agile is, in the end, a generic term encompassing methods themselves made of specific practices that, if applied correctly, make the software development flexible, user-oriented and, most importantly, focused on value. To set up an agile development life cycle, some support for the involved team is needed especially if the practitioners are novice ‘agilists’. For this purpose, the roles of agile coach or scrum master – played by experts in the application of such methods and practices – appeared. Recently, large language models accessible by AI-based chatbots have become very popular and these could furnish relevant generic knowledge on agile methods, give tailored advice for a development project on the basis of contextual information, or even perform work according to agile practices themselves based on furnished project data. In other words, we could envision that a chatbot like ChatGPT becomes a virtual member of the development team able to inform, coach and execute a share of the development work. Adopting such a tool is nevertheless not trivial. First of all, the true capacity of furnishing valid information, understand contexts and apply techniques correctly should be investigated. Second, it is also unsure how such a digital and virtual team member would be perceived and integrated by its human colleagues. This paper starts with introducing the new context of large language models and analyzing the sentiment expressed in Twitter feeds related to the use of ChatGPT in the context of agile development. Then, it performs a small empirical experiment to assess the performance of ChatGPT when furnished with some tasks that would typically be performed by an agile coach or Scrum master. Then, as an introduction to the second International Workshop on Agile Methods for Information Systems Engineering (Agil-ISE23), it summarizes the papers presented during the event.

Keywords

agile development, agile practices, ChatGPT, Twitter feeds, Sentiment analysis, Scrum master, agile coach, Agil-ISE

Agil-ISE23: 2nd Intl. Workshop on Agile Methods for Information Systems Engineering, June 13, 2023, Zaragoza, Spain
EMAIL: palash.bera@slu.edu (Palash Bera); yves.wautelet@kuleuven.be (Yves Wautelet); geert.poels@ugent.be (Geert Poels)
ORCID: 0000-0002-8364-0972 (Palash Bera); 0000-0002-6560-9787 (Yves Wautelet); 0000-0001-9247-6150 (Geert Poels)



Copyright © 2023 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)

1. Explorative Evaluation of ChatGPT's Ability to Support Agile Projects

Agile software development has been used since the late 90s and some methods implementing its principles like Scrum have now reached a rather high level of maturity and thus stability. Agility came as a solution to the software crisis where the majority of software development projects failed to meet user requirements. Agile development can nevertheless still sound counternatural to some practitioners stuck into their habits. Long and fastidious requirements specifications as well as detailed software design remain practices often used and witnessed in the industry. To get through a successful agile transformation, some support and accompaniment are needed; one cannot become a fully operational agile practitioner overnight because a specific mentality needs to be acquired to master the agile philosophy [1]. In other words, it is not only about applying defined techniques or practices but also to learn to naturally cope with empiricism, value-orientation, quick development, experimentation, alternative testing, etc. Also, there is nothing similar to agile development projects, so teams need to tailor the agile process and custom select which practice to apply. In order to properly sustain novice agile teams, the roles like Scrum master or agile coach progressively appeared. The latter roles are played by a person ensuring that a development team correctly applies Scrum or agile principles within the context of a specific project. These roles are thus not the ones of project managers but more generally specialists facilitating the acquisition of agility by teams into projects and the agile transformation of the organization.

Through the years, knowledge about agile practices has been accumulated because of their application on a wide variety of projects. In parallel, scientific research in the field allowed to formalize the lessons learned within application projects. Kiv et al. [2] proposed a rather formal ontology-based approach to gathering the knowledge accumulated in the scientific literature about agile practices adoption. To support the use of the ontology, an expert system has been developed allowing practitioners to systematically access the knowledge inherited from the scientific literature about agile practices. Structured and validated knowledge fragments can thus serve as a reasoning basis when agile practice adoption decision needs to be made. Very recently, AI-based large language models like ChatGPT appeared and one can legitimately ask if these tools can support practitioners applying agile in a formal fashion. In order to further evaluate if AI can be of any help in this context, we proceeded to a small explorative evaluation.

Prior to presenting the evaluation, we also did an analysis of understanding the Agile practitioners' perceptions of ChatGPT. For this purpose, we analyzed the Twitter feeds of messages that contained the phrases "ChatGPT and Agile methodology" or "ChatGPT and Agile practice." As ChatGPT was introduced in November 2022, it took some time for users to understand the functionalities and capabilities of ChatGPT. Therefore, we analyzed Twitter feeds between January 2023 and March 2023, two months after ChatGPT was launched so that perceptions about ChatGPT were already developed.

Approximately 2500 tweets were identified based on these search criteria. Close to 200 tweets were removed as they were not relevant to Agile software development. The remaining valid tweets were analyzed using text mining software written in R.

On a scale of -1 to +1, the overall sentiment of the tweets was 0.20. Based on this score, the overall sentiment can be treated as neutral which means that those who tweeted with the above-mentioned phrases were neither excited nor disappointed in the context of ChatGPT usage in Agile. The sentiments in each tweet were classified into different categories and the bar chart in Figure 1 shows the distribution of the sentiments. The most frequent sentiment that appeared is "trust." The tweets referred to the issue of trust with respect to ChatGPT. Many tweets mentioned that ChatGPT is not trustworthy. The tweets referred to situations when ChatGPT did not give the correct or accurate results and thus these tweets cautioned the trustworthiness of chatGPT in the context of Agile methodology. It is also important to note that the sentiments- *joy* and *fear* were of similar numbers. An example of a tweet that was related to joy is "ChatGPT can help expedite the software development using Agile method." An example of a tweet that expressed fear is "use of ChatGPT will make the scrum master's job redundant."

able to restate some agile practices even if one could argue for confusion between the agile method (Scrum and Kanban are reported as practices) and the actual agile practices (*test-driven development*, *continuous integration*, *continuous delivery*, *pair programming*, *user stories* and *retrospectives* are reported, see Figure 3). When it comes to asking for more details about a specific agile practice, the tool is able to aggregate information and present it in a meaningful way. The tool provides very structured and rather accurate answers on the overall agile theory and the related practices' use.

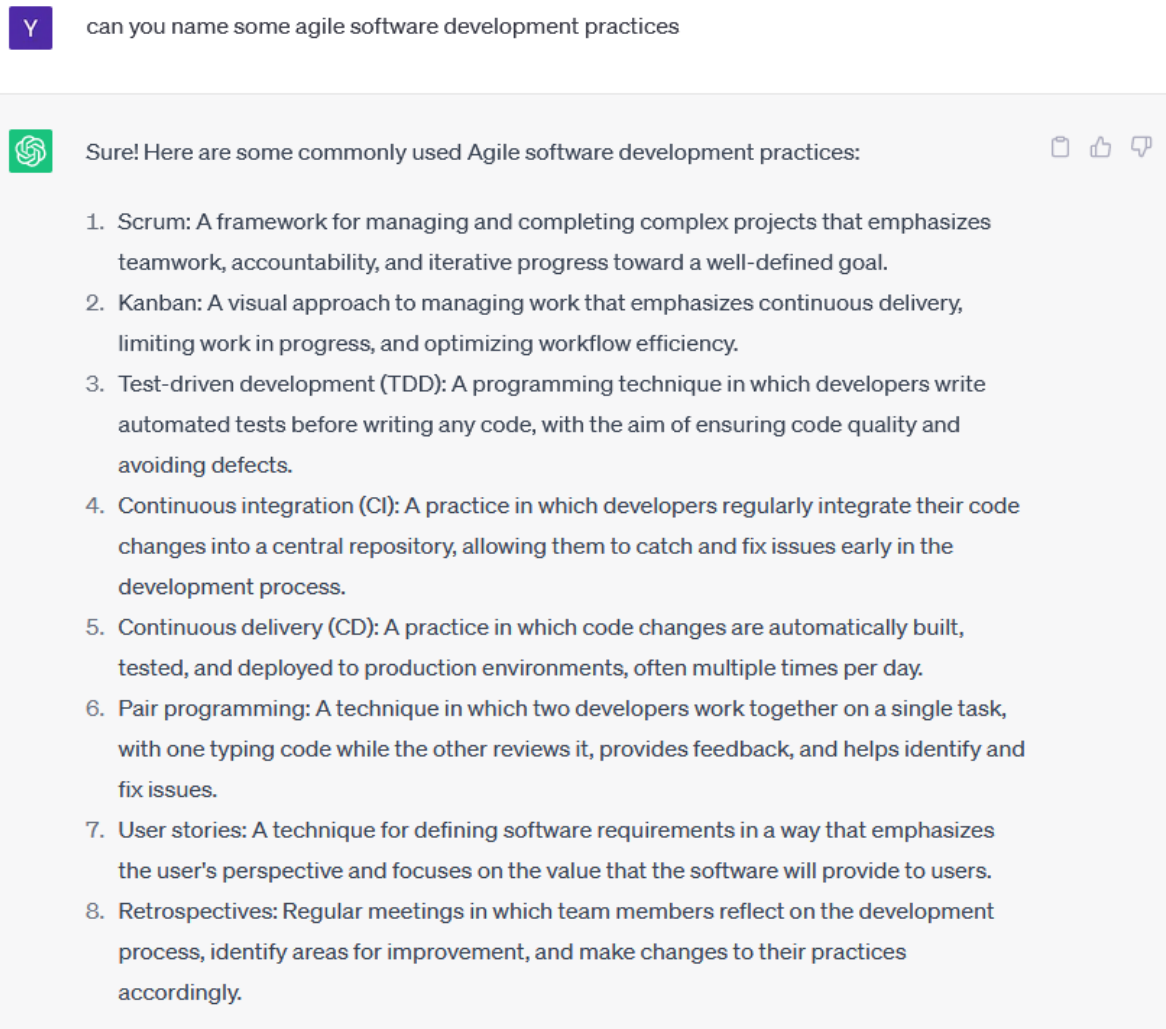


Figure 3: ChatGPT reporting agile development practices

Agile is not only about the understanding and application of pure theory only. First of all, the bridge from theory to practice needs to be covered, which we suggest evaluating here by asking “how to” questions to the tool. Once again, it reports detailed answers, like for the question of *how to evaluate the value of a feature?*, ChatGPT reports *customer/user needs*, *business goals*, *market demand*, *technical feasibility* and *return on investment* as main elements. Asking the (relatively vague) question of *how to organize a scrum team?*, the AI-based chatbot gives some very clear insights. It reports on *define the roles*, *determine the team size*, *establish team norms*, *create the product backlog*, *plan the sprint*, *conduct daily stand-up meetings* and *review and reflect*. Second of all, there are more fundamental philosophical elements to take into account to build up a mentality that makes the developing team act agile when facing a specific situation. As can be seen in Figure 4, ChatGPT is able to answer when asked about the general philosophical elements that drive agility in software development. It indeed reports on *empiricism*, *collaboration*, *flexibility*, *customer focus* and *continuous improvement*. An AI-based tool reporting empiricism is somewhat ironic since it induces that everything cannot be theoretically foreseen or that an input to a situation may not end up in a

deterministic outcome. ChatGPT reports that *empiricism refers to the practice of making decisions based on observed and measurable outcomes rather than relying solely on assumptions, predictions, or plans*. It also further argues that empiricism is made of *transparency, inspection, and adaptation*. It rather accurately reports that ‘what to develop’ decisions need to be taken along the software development rather than with up-front plans in the function of the circumstances and measurable sources of value. So again, it is quite close to the advice we would expect from a dedicated Scrum master or agile coach role played by a human being.

Y what would be philosophical elements of agile methods



The screenshot shows a chat interface with a green OpenAI logo on the left. The user's question is: "what would be philosophical elements of agile methods". The AI's response is: "Agile methods are based on a number of philosophical elements that underpin the approach to software development. Here are some of the key philosophical elements of Agile methods: 1. Empiricism: Agile methods are based on the idea that knowledge comes from experience, and that the best way to learn is through trial and error. Agile teams use feedback and iteration to continuously improve their processes and the quality of their software. 2. Collaboration: Agile methods emphasize collaboration and communication among team members, stakeholders, and customers. This helps ensure that everyone is on the same page and that the software meets the needs of all stakeholders. 3. Flexibility: Agile methods prioritize flexibility and adaptability over rigid processes and documentation. This allows teams to respond to changes in requirements or the market more quickly and effectively. 4. Customer focus: Agile methods are driven by the needs and feedback of the customer. The product owner or customer is heavily involved in the development process, helping to ensure that the software meets their needs and delivers value. 5. Continuous improvement: Agile methods emphasize continuous improvement and learning. Teams are encouraged to reflect on their processes and practices and make changes as needed to improve the quality of their software and the efficiency of their development process. These philosophical elements are reflected in the Agile Manifesto and the 12 principles of Agile development, which provide a foundation for Agile software development practices." At the bottom of the response area, there is a button labeled "Regenerate response".

Figure 4: ChatGPT reporting agile philosophical elements of agile methods

We can now further challenge the tool by furnishing some real project data and asking for the application of some agile practices onto it. Since the tool is essentially text-based, we decided to furnish it with a complex set of user stories (up to 137) coming from a real-life project and ask it to perform a user story map out of these. We made the tool generate an output several times in a row. The first given output was a set of 5 paragraphs each detailing the overall situation described in the given user stories. While strictly speaking it is not a user story map, one can argue that grouping the individual stories around 4 (the fifth paragraph being a conclusion) themes leans toward the structuration of the requirements. Still, no planning could easily be done on that basis. A new generation leads to the creation of general themes (*privacy and control, ease of use, mobility and wearability, assistive*

technology and *communication*). Each of these themes was further divided into one to three epics themselves subdivided into multiple *user activities* further refined in different *tasks*. As can be seen in Figure 5, the presentation by ChatGPT follows an interesting structuring process (theme > epic > user activity > task), even if the quality of the output when compared to the performance of a human domain specialist remains to be evaluated. After that, we wanted to evaluate if the tool is also able to suggest some user stories to realize in priority for the coming sprint (in other words if value could be assigned without more information) so we entered the sentence *with what user stories would you start the development?* Asking why these had been put forward, the tool gives classical answers (*impact on users, dependencies* and *business value*) but with no justification on how value had been associated with the proposed user stories.

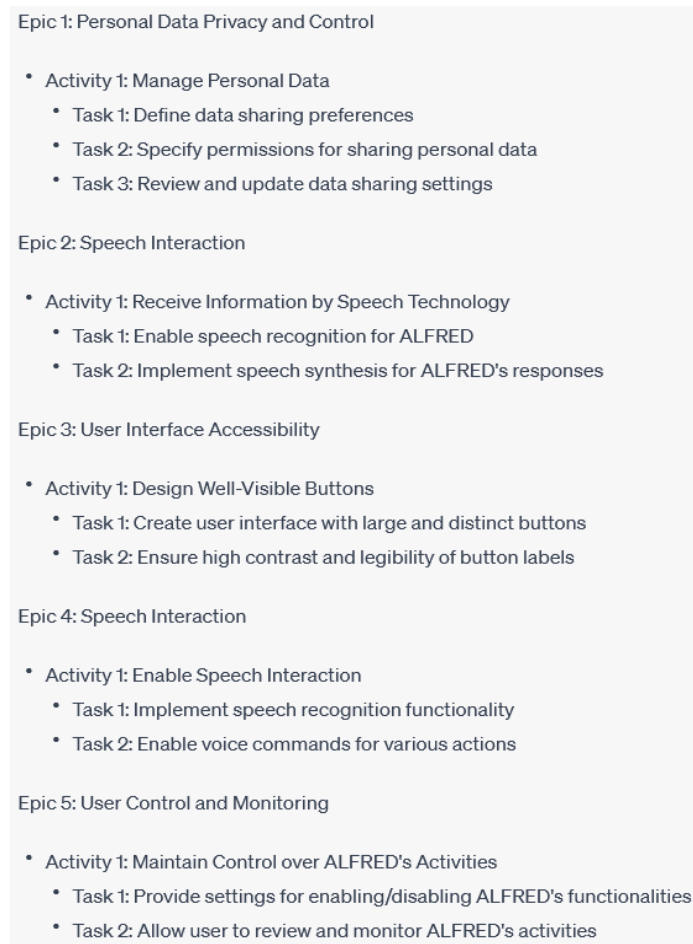


Figure 5: ChatGPT structuring an unstructured use story set

Alternatively, we also asked the tool to relate to some findings validated scientifically in the field of agile software development. For all of the realized experiments, the real source of information that motivated the output for the chatbot is an open issue and we are thus now looking for more formality in the upfront sources of knowledge. We thus asked for particular scientific articles (in scientific journals or conferences) that would summarize agile projects and their success factors through effective identification. ChatGPT generated a few bibliographical sources and gave concrete explanations on each of these sources but, after a cross-checking into the referenced journals, it turned out that none of these articles could effectively be found. In other words, the articles did not really exist or at least not under the given title. This finding also needs more investigation.

All in all, the tool in its version proposed to the public furnishes some interesting features for gathering and structuring generic information on agile but also to help in the requirements engineering and even

possibly within the sprint planning but at the same time we quickly identify a lack of rigor into the information given. When compared to the more formal approach to agile practices like the one of Kiv et al. [2], the given knowledge is certainly less formal, but the possibilities are higher due to the ability to apply some agile techniques directly on datasets. More explorative scientific work should be done to actually compare the responses given by Scrum masters and ChatGPT to requests done by software development teams in real life. This way the accuracy of the output can be more formally evaluated. Also, when providing contextual information, the tool tends to adjust and furnish more specific information. Future research could also incorporate evaluating the ability of the tool to sharpen the quality of its answers (again when compared to a professional in the field) when such information is provided.

2. Summary of the Submitted Papers

This section describes the papers accepted to the second edition of the 2nd International Workshop on Agile Methods for Information Systems Engineering (Agil-ISE'23) and also gives a brief introduction to the keynote speech. The first edition of the workshop was organized in Leuven in 2022. As a whole, the workshop is devoted to scientifically challenging the artifacts and practices adopted in agile software development [3].

The keynote speech by Maya Daneva is entitled “Lessons Learned from 15 Years of Empirical Research on Agile and DevOps”. For the last two decades, an entire Agile industry has been slowly created worldwide. The abundance of certifications, tools, techniques, models, approaches and large-scale frameworks fuels even further the growth of this industry. In the talk, Maya will remove the noise and chaff, unearthing some lessons learned in my 15 years of empirical research on agile and DevOps in national and global companies. She will explain the tacit assumptions behind agile concepts and how in certain contexts – if these assumptions are unrealistic – adopting more agile practices might in fact render an organization more “heavyweight” or waterfall.

In their paper, "Systematic Literature Review of Agile Framework Application for IT System Development in Public Sector", Jolanta Graudone and MāriĶe Kirikova from Riga Technical University in Latvia, discuss the challenges and benefits of applying Agile practices in the public sector. The authors review ten academic papers that explore the application of Agile principles in IT systems development within the context of public sector regulations, extensive documentation requirements, hierarchical organizational structures, and the scale and complexity of IT systems.

The paper identifies several public sector-specific challenges, risks, and areas for improvement in the application of Agile practices, which were organized according to the five phases of the project lifecycle following PMBOK. The authors also identify interrelated aspects across these phases, including the strong interdependency between collaboration and communication, roles and responsibilities, knowledge and skills, and organizational structure.

The authors conclude that while there is still a lack of understanding about how to apply Agile practices in the public sector, cultural change is necessary for successful implementation. They emphasize the importance of Agile practices training and strict role and responsibility definitions to address these challenges. Overall, this paper offers insights into the potential benefits and challenges of applying Agile practices in the public sector and provides recommendations for successful implementation.

The private sector also faces challenges when applying Agile practices to large-scale and distributed IT projects. In their paper "Towards a Solution Proposal to Agile Quality Requirements Challenges in Large-Scale Projects," Wasim Alsaqaf, Maya Daneva, and Roel Wieringa from the University of Twente in the Netherlands focus on challenges related to quality requirements.

The authors argue that software development projects are typically part of larger IT initiatives with relatively stable and predefined goals. To decompose these goals into specific requirements for software

development teams, such as epics and user stories, they propose using Goal-Oriented Requirements Engineering (GORE) techniques. Quality requirements can then be associated with these goals and further decomposed.

To guide this process of goal and quality requirement decomposition, the authors introduce a workshop-based method called Goal-Oriented Agile Requirements Engineering (GOARE). This method uses iStar 2.0 as a modelling language and consists of two key components: the Initiative Owner role and workshop preparation and execution procedures. The paper provides an explanation of these components.

The use of iStar 2.0, and thus integration of GORE and Agile Software Development is also touched upon by Soreangsey Kiv from Tilburg University in the Netherlands, who addresses the selection and adoption of Agile practices in the paper "Socio-Intentional Framework for Agile Methods Tailoring: Past, Present, and Future." The paper outlines the Socio-Intentional Framework For Agile Methods Tailoring that the author previously developed. This framework provides guidance on selecting appropriate practices based on the goals of a project team, their situation, and the dependencies between team members.

To facilitate the selection process, the framework employs a user-friendly tool called Ontology-Based tool for Agile Methods Adoption (OBAMA). OBAMA retrieves information from an ontology that was built from 86 documented case studies on Agile practices adoption. The information is visualized using iStar 2.0.

The paper presents the design of an empirical study to evaluate the effectiveness, perceived ease of use, and usefulness of the method. The study will be performed with MSc students as participants since the method targets teams of novice developers. Data will be collected through semi-structured interviews and a survey instrument.

Students as future practitioners of agile approaches to software development is also central to the paper of Bert de Brock from the University of Groningen in the Netherlands, who highlights the importance of teaching the practice of agile software development in his paper "Teaching and Practising RE for Agility." While students can be taught the principles, methods, and techniques of agile software development, firsthand experience of what an agile approach really means is crucial. For instance, students need to be made aware that requirements based on user expectations change frequently, and results that were previously acceptable might not be acceptable anymore in the next cycle of development.

To provide students with a real sense of agile software development, the author developed a course on requirements analysis and system design. The paper details how this course can be integrated into several ACM curricula, and outlines the expected learning outcomes, course topics, and organization. Additionally, the paper discusses the author's experience of teaching this course online due to the COVID-19 pandemic restrictions imposed by governments.

Finally, the paper "Measuring Agile/DevOps Team Performance" by Thoby Visser from Cap Gemini and Joris Hulstijn from the University of Luxembourg presents a study on improving the performance of Agile/DevOps teams that work remotely at client locations. The paper proposes a data-driven framework for continuous performance improvement of Agile/DevOps teams, based on the DMAIC (define-measure-analyze-improve-control) principles.

To create this framework, the authors had to investigate two research questions: how to measure the performance of Agile/DevOps teams and what factors impact their performance. A list of factors was compiled from a literature review, classified into technical, non-technical, and environmental categories, and ranked by importance based on interviews with nine experienced Agile/DevOps team managers from Cap Gemini.

The authors then conducted a systematic literature review to identify metrics and selected the most cited metrics in two categories: throughput and stability. These metrics balance the concerns of Agility and DevOps. Finally, the authors proposed integrating these metrics and factors in a DMAIC cycle for continuous improvement.

3. Acknowledgements

The workshop chairs thank Fabiano Dalpiaz for his initial involvement in this new edition of the workshop as well as all the authors that have submitted their work. A special thanks to Maya Daneva for accepting to deliver a keynote speech at the workshop.

4. References

- [1] S. Shafiee, Y. Wautelet, S. Poelmans, S. Heng, An empirical evaluation of scrum training's suitability for the model-driven development of knowledge-intensive software systems. *Data & Knowledge Engineering*, Elsevier 146 (2023): 102195.
- [2] 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.
- [3] 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.