Exploring and improving industry-academia communication in software engineering*

Update 2021

Sergio Rico
Lund University
Lund, Sweden
sergio.rico@cs.lth.se

ABSTRACT

Background: Despite the mutual benefit of the industry-academia partnership, the level of joint work is still low. For this reason, the interest in connecting research and practice has increased recently in the academic community. Objective: This research aims to design and apply approaches to improve the knowledge exchange between academic researchers and software engineering practitioners. Methodology: This work can be seen from a design science perspective. Following the design-science paradigm, the knowledge regarding a phenomenon is obtained through the design and evaluation of solutions that apply in a specific context. Consequently, this research work proposes and evaluates approaches to bridge the communication gap. Results: Two approaches have been explored and partially evaluated. The SERP-taxonomy architecture that can be used to describe and link research results and industry challenges, and rapid reviews to foster communication between industry and academia. Conclusion: This thesis will provide empirical evidence of the application of collaborative approaches to improve industry-academia communication and get closer research and practice.

Note: Due to COVID-19, the EASE 2020 conference was postponed to 2021. Thus, this document is an update (on section 5) to the version published in EASE 2020[18].

KEYWORDS

Industry academia communication, industry-academia collaboration, rapid reviews, design-science

1 INTRODUCTION

Researchers and practitioners maintain a symbiotic relationship in software engineering, in which communication is essential. To a large extent, the partnership is voluntary because each party decides when and how to collaborate. However, the ideal of the relationship is a strong interaction that benefits both actors. That level has not yet been achieved in software engineering. While some topics such as knowledge transfer, knowledge translation, and collaboration have caught the attention of researchers, more work is needed to connect academia and practice[2, 8, 9, 12].

This Ph.D. work aims to identify and apply approaches to improve the knowledge exchange between researchers and practitioners. The benefit for academics from this exchange includes to update teaching content, collect empirical data, validate research, and increase the research relevance. Similarly, practitioners could get benefit by taking advantage of research results, and new practitioners could experience a smooth transition from university to industry.

From a methodological perspective, this work can be viewed as design-science research. Under the design-science paradigm, the understanding of a phenomenon is gained through the design of solutions that are valid in a specific context[7]. In this research, different approaches will be implemented in scenarios where researchers and practitioners have the will to work and learn from each other. The work is divided into studies that are described further(Section 3). The overall research question is:

What approaches can be implemented to foster knowledge exchange between researchers and practitioners in software engineering?

The next parts of this report are organized as follows. Section 2 presents an overview of the research using a visual abstract for research-design research. Later on, in Section 3 the finished and under development studies are described. Finally, Section 4 briefly presents the related work.

2 RESEARCH OVERVIEW

To present a general view of the research, Figure 1 depicts a visual abstract according to the proposal from Storey et al.[23] to visualize design-science research. The figure shows three main parts as follows. 1) On top of the figure in a box, the technological rule or main contribution of this research. 2)Below the technological rule, the figure shows the problem-solution relation. This relation includes both, how the solution is designed and how the problem is understood. In the middle of the relation is shown how the solution is validated. 3) The three boxes on the bottom include the evaluation of the research in terms of relevance, rigor, and novelty.

The technological rule captures the main contribution of the research and is expressed in the form *To achieve «effect» in «situation» do «intervention»*. For this work the technological rule has been formulated as follows *"To improve knowledge exchange in a facultative relationship between researchers and practitioners facilitate interactive research approaches"*. Each of the components is described below. The desired effect *«improve knowledge exchange»* supposes that by improving the exchange, each party gets to benefit in their interest. The situation *«facultative relationship between researchers and practitioners»* means that both have the will to work with each other. A facultative relationship means that each actor decides when, how and with who wants to collaborate. The interventions named as *«interactive research approaches»* refer to approaches designed and implemented where researchers and practitioners have a high

*Phd Supervisors: Dr. Emelie Engström, Dr. Martin Höst. Software Engineering Research Group http://serg.cs.lth.se/ Lund University. Sweden

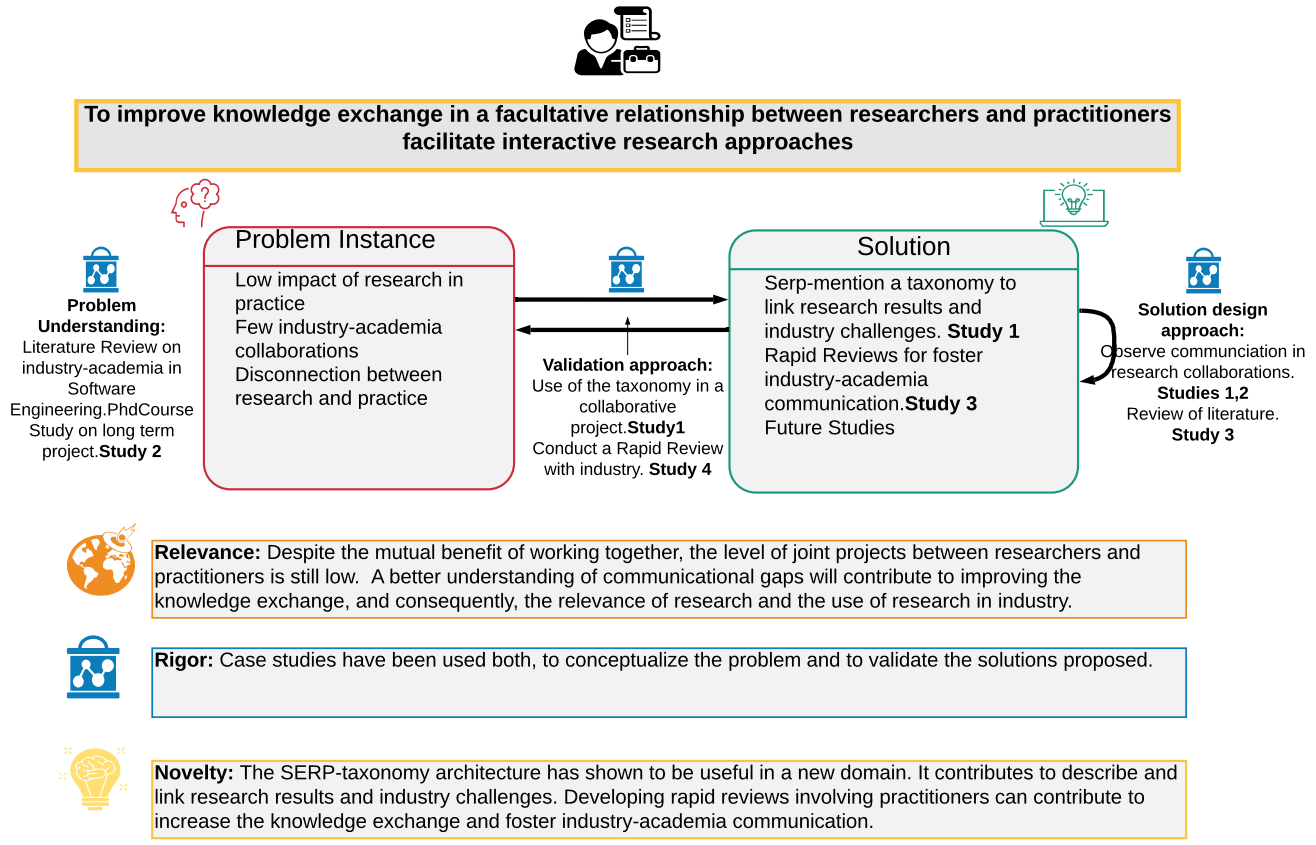

Figure 1: Visual abstract[23] of the Research Work

degree of interaction. In this research, the approaches implemented are the SERP-taxonomy architecture Study 1 and rapid reviews Studies 3 and 4.

The problem-solution pair is reached by studies as follows. The problem instances i.e examples of the problem, are explored through the review of literature and study 2 where the communication in a long term project is investigated. Until now in this research, two different approaches have been designed and validated. The first approach is the development of taxonomies using the SERP-taxonomy architecture to improve communication in software vulnerability management (Study 1). The second approach is inspired by evidence-based medicine and is the use of rapid reviews to foster communication (Study 3). Study 4 validates the proposal from Study 3 in practice. The final part in the visual-abstract summarizes a general view of relevance, rigor, and novelty of this work.

3 STUDIES

This research will be presented as a collection of papers that report studies on a general research topic. In the following subsections, each of the current or finished studies are described. Future studies will continue in the same line of research but focusing more on the design and evaluation of actions to improve communication.

3.1 Study 1: Towards a common language to link challenges and solutions

Study 1 exposes the use of the SERP-architecture in the domain of software vulnerability management [20]. The SERP-architecture supports the development of taxonomies that can be used to link research results and industry challenges in software engineering. Previous experiences have shown the usefulness of the approach in software testing [6]. In this study, SERP-mention, a taxonomy that follows the SERP architecture, was developed and evaluated in a collaborative project whit participants from industry and academia. To build initial versions of the taxonomy, we interviewed researchers and develop a first version. Later, with representatives from companies, a workshop was conducted to validate the result and extend the taxonomy. As the final step, we extracted solutions and challenges from a sample of papers. We showed in this study the usefulness of the taxonomy approach to describe research results and industry challenges. Besides, the potential to link results from academia with problems from industry.

RQ1 *To what extent can the SERP-taxonomy architecture be reused to develop a taxonomy in the area of IoT vulnerability management?*

RQ2 *To what extent is the taxonomy developed useful for improving the communication about vulnerability management between researchers and practitioners?*

3.2 Study 2: Exploring communication in a long-term industry-academia collaboration

Study 2 is a case study. The project under study was a long term collaboration in the area of software testing. The research questions aimed to identify what are the effects of communication, and what facilitates the industry-academia communication. For the data collection, we used a retrospective timeline-based method where information about people, outputs, and activities was presented to project participants. The session was recorded and transcribed for the analysis. We transcribed and coded the transcribed recordings of the sessions by pairs, and followed techniques of thematic analysis. We identified for each instance of communication the following information: parties, the environment where the exchange took place, the content, and the effects. Later on, we identified facilitators for communication. We denoted as facilitators some characteristics of the context that favor communication, for example having a long term relationship, or having regular meetings. In this case study, we followed the guidelines proposed by Runeson et al.[21] to conduct case studies in software engineering.

RQ1 *What are the effects of industry-academia communication?*
RQ2 *What facilitates industry-academia communication?*

3.3 Study 3: A proposal to foster industry-academia communication through Rapid Review

In study 3, we proposed steps to conduct Rapid Reviews in Software engineering based on a literature review in healthcare and the authors' experiences conducting secondary studies. Rapid Reviews are widely used in Evidence-Based medicine. Although evidence-based software engineering is inspired in evidence-based medicine, only few researchers have used rapid reviews in software engineering research[3]. We present a proposal to use rapid reviews to exchange knowledge between academia and practice. To develop the proposal, first, we reviewed papers about Rapid Reviews in healthcare to understand in detail how rapid reviews are conducted. Later on, we synthesized the findings that were relevant to software engineering according to the experience of the authors conducting and reviewing research that involves work with industry. The final result of this study is the method to use rapid reviews in software engineering as a vehicle to exchange between industry and academia. From a methodological perspective, this study can be seen as a literature review. We were inspired by guidelines proposed by Kitchenham et al.[11]. However, the study keep differences with a traditional systematic literature review for the following reasons. 1) We analyzed papers until we reach saturation on the analytic model 2) We searched for articles reflecting on the use of Rapid Reviews instead of primary studies 3) We elaborated on the authors experience a new proposal.

RQ1 *What methodological guidelines for conducting Rapid Reviews in healthcare can we bring to Software Engineering?*

RQ2 *What can we learn in Software Engineering from the interaction between practitioners and researchers when conducting Rapid Reviews in healthcare?*

3.4 Study 4: Evaluating the proposal

Study 4 is in the design phase. The objective is to validate the guidelines proposed in study 3. The next two strategies will be implemented. We will conduct a Rapid Review as such, following the method proposed in study 3. We will follow the guidelines for case studies by Runeson et al.[21]. Similarly, we will collect qualitative data from practitioners and researchers on their views about the rapid review as a mean to exchange knowledge between research and practice. We will probably collect information through a focus group and a survey.

RQ1 *What are the benefits and drawbacks of using rapid reviews to foster communication between researchers and practitioners?*
RQ2 *What are the effects of conduct rapid reviews with practitioners?*

3.5 Future Studies

The following studies in this research will aim to explore other interactive approaches to foster industry-academia exchange. I want to explore how communication differs according to the context, and which approaches fit better according to the scenarios where industry and academia meet.

4 RELATED WORK

Under the view of the triple helix model, communication between industry, university, and the government is at the basis of the knowledge-based economy [13]. Furthermore, communication is a key success factor when companies and universities work together [16], and a condition for knowledge transfer [5].

In software engineering, some of the terms used to describe industry-academia interactions include knowledge and technology transfer, knowledge translation, and industry-academia collaboration. Despite some particularities that make them different, all the relations are based on communication.

The term technology transfer was introduced by Gorshek et al.[10] in software engineering. The authors build a model that make emphasis in cooperation from empirical experience working with industry. Similarly, but adding emphasis to the bi-directional exchange, Mikkonen et al.[15] presented a model for continuous and collaborative technology transfer. A similar term is knowledge transfer. An example of knowledge transfer in software engineering is the work from Cartaxo et al. [4] where the authors present a model to transfer knowledge from academia to companies. In this model, the researcher act as a transfer-agent. Deepika et al. [1] claim that the knowledge needs to be translated to the industry. They propose guidelines to translate knowledge into software engineering practice. Garousi et al. [8] in a systematic literature review present challenges and best practices to run industry-academia collaboration. Communication is included under the perspective of how to deal with meetings and interpersonal communication.

More specific mediums to communicate research results to the industry are the visual abstracts [23] and evidence briefings[4]. The SERP-architecture to link industry challenges and research results is used in this research(Study1).

5 RESEARCH WORK UPDATE 2021

The purpose of this section is to update the status of the research to 2021. We present in the following subsections:

- A summary of the advance and results of the studies conducted.
- Motivation to apply this research in a narrow topic in software engineering.
- A short description of new studies to include.
- An update to Figure 1.

5.1 Summary of advance and results

This research work, including studies 1-3, was defended as a Licentiate thesis[17] at the beginning of 2021. The licentiate thesis resembles a Ph.D. thesis in the structure but is much shorter. Below, it is described the status of the studies described above.

5.1.1 ***Study 1: Towards a common languate to link challenges and solutions.*** The paper entitled "A Taxonomy for Improving Industry-Academia Communication in IoT Vulnerability Management" [20] was published at Euromicro SEAA 2019.

The paper evaluates the potential and usefulness of SERP taxonomies around the topic of software security vulnerability management. SERP taxonomies may be helpful for researchers when defining research questions and searching for relevant literature. For practitioners, SERP taxonomies may help to describe challenges in context and support communication with researchers.

5.1.2 ***Study 2: Exploring communication in an industry academia joint project.*** The paper entitled "A Case Study of Industry-Academia Communication in a Joint Software Engineering Research Project" is under review in a journal.

We identified five facilitators of IA communication and nine project outcomes related to this communication. The facilitators concern the relevance of the research, practitioners' attitude and involvement in research, frequency of communication and longevity of the collaboration. The project outcomes promoted by this communication include, for researchers, changes in teaching and new scientific venues, and for practitioners, increased awareness, changes to practice, and new tools and source code. Besides, both parties gain new knowledge and develop social-networks through IA communication.

Our study presents insights that can provide advice on how to improve communication in IA research projects and thus the cocreation of software engineering knowledge that is anchored in both practice and research.

5.1.3 ***Study 3: A proposal to foster industry-academia communication through Rapid Review.*** We published the technical report "Guidelines for conducting interactive rapid reviews in software engineering – from a focus on technology transfer to knowledge exchange"[19].

We propose guidelines to conduct interactive rapid reviews. An interactive rapid review is a streamlined approach to conduct agile literature reviews in close collaboration between researchers and practitioners in software engineering.

The proposed guidelines will potentially boost knowledge cocreation through active researcher-practitioner interaction by streamlining practitioners' involvement and recognizing the need for an agile process.

5.1.4 ***Study 4: Evaluating the proposal.*** We conducted a multicase study where two groups of researchers and practitioners conducted an interactive rapid review. Currently, we are on the analysis and preparing the diffusion of research results.

5.2 Application area

For the second part of this P.hD, we keep as the main goal to investigate the communication between researchers and practitioners. Furthermore, we have seen the need to focus on a specific topic in software engineering. In the next studies, we will focus on the researchers-practitioners knowledge exchange around the testing of machine learning systems, i.e., software systems that include machine learning components.

Testing machine learning systems seems to be a relevant topic to study researchers-practitioners communication. On the academic side, researchers have found that some testing techniques can be applied as in typical software systems while others require adaption[14, 22]. Practitioners face challenges when developing and testing machine learning systems due to the lack of practical guidelines. Besides, the teams in charge of developing machine learning systems are composed of traditional software developers, data scientists, and a new generation of developers with more scientific backgrounds. In these settings, team management and lack of training in software engineering practices is a known challenge. Thus, the potential benefit for both communities, the need to connect research and practice, and the active work in the field are indications that the topic is suitable to evaluate the approaches proposed in this research work.

5.3 Future studies

The following studies are planned for the second part of this research:

5.3.1 ***Study 5: A literature review on the testing of machine learning systems.*** This review aims to get a sense from academia of how machine learning systems are conceived and how they are tested.

RQ1 *How do researchers conceive systems that integrate machine learning components?*

RQ2 *What testing processes and techniques have been proposed to test these systems?*

RQ3 *What researchers have identified challenging when testing machine learning systems?*

5.3.2 ***Study 6: Exploring how machine learning systems are tested in practice.*** The goal is to explore what practitioners know and how they learn about testing machine learning systems. A first data collection is based on master theses at Lund University related to applying machine learning. After, we plan to conduct interviews with master theses supervisors both at university and in the application domain, e.g. (medical, scientific, industry).

RQ1 *How do practitioners perceive machine learning systems?*

RQ2 *How do practitioners learn about testing machine learning systems?*

RQ3 *What are the common practices to test machine learning systems?*

5.3.3 ***Study 7: SERP-testml linking industry challenges and research solutions in machine learnings systems testing***. Data collected from from studies 5 and 6 is an input to develop a SERP taxonomy of machine learning systems testing. The taxonomy will be used in the following study to support researcher-practitioner communication.

RQ1 *What are the differences and similarities in concepts and practices between researchers and practitioners when testing machine learning systems?*

RQ2 *What are the common challenges in research and practice for testing machine learning systems?*

RQ3 *To what extent the research solutions map with practice challenges?*

5.3.4 ***Study 8: Supporting researches-practitioners knowledge exchange around machine learning systems testing.*** In this study, we evaluate the approaches proposed to foster knowledge exchange between researchers and practitioners. Starting from a real problem around testing machine learning systems, we use SERP-testml to facilitate initial communication and as a starting point to navigate available research and challenges faced by practitioners. Later, we validate the interactive rapid review approach within researchers and practitioners involved.

RQ1 *How interactive rapid reviews may support software engineering practice and knowledge exchange between research and practice?*

RQ2 *What are the learnings for researchers and practitioners when participating in IRR?*

5.4 Visual abstract update

Figure 2. is an update of the visual abstract[23] including the updates mentioned above.

REFERENCES

[1] Deepika Badampudia, Claes Wohlina, and Tony Gorscheka. 2018. Guidelines for Knowledge Translation in Software Engineering. (2018).

[2] Victor Basili, Lionel Briand, Domenico Bianculli, Shiva Nejati, Fabrizio Pastore, and Mehrdad Sabetzadeh. 2018. Software engineering research and industry: a symbiotic relationship to Foster impact. *IEEE Software* 35, 5 (2018), 44–49.

[3] Bruno Cartaxo, Gustavo Pinto, Baldoíno Fonseca, Márcio Ribeiro, Pedro Pinheiro, Maria Teresa Baldassarre, and Sérgio Soares. 2019. Software Engineering Research Community Viewpoints on Rapid Reviews. In *2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. IEEE, 1–12.

[4] Bruno Cartaxo, Gustavo Pinto, Elton Vieira, and Sérgio Soares. 2016. Evidence briefings: Towards a medium to transfer knowledge from systematic reviews to practitioners. In *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. ACM, 57.

[5] Esther de Wit-de Vries, Wilfred A Dolfsma, Henny J van der Windt, and MP Gerkema. 2019. Knowledge transfer in university–industry research partnerships: a review. *The Journal of Technology Transfer* 44, 4 (2019), 1236–1255.

[6] Emelie Engström, Kai Petersen, Nauman bin Ali, and Elizabeth Bjarnason. 2017. SERP-test: a taxonomy for supporting industry–academia communication. *Software Quality Journal* 25, 4 (2017), 1269–1305.

[7] Emelie Engström, Margaret-Anne Storey, Per Runeson, Martin Höst, and Maria Teresa Baldassarre. 2019. A review of software engineering research from a design science perspective. *arXiv preprint arXiv:1904.12742* (2019).

[8] Vahid Garousi, Kai Petersen, and Baris Ozkan. 2016. Challenges and best practices in industry-academia collaborations in software engineering: A systematic literature review. *Information and Software Technology* 79 (2016), 106–127.

[9] Vahid Garousi, Dietmar Pfahl, João M Fernandes, Michael Felderer, Mika V Mäntylä, David Shepherd, Andrea Arcuri, Ahmet Coşkunçay, and Bedir Tekinerdogan. 2019. Characterizing industry-academia collaborations in software engineering: evidence from 101 projects. *Empirical Software Engineering* 24, 4 (2019), 2540–2602.

[10] Tony Gorschek, Per Garre, Stig Larsson, and Claes Wohlin. 2006. A model for technology transfer in practice. *IEEE software* 23, 6 (2006), 88–95.

[11] Barbara Kitchenham and Stuart Charters. 2007. Guidelines for performing systematic literature reviews in software engineering. (2007).

[12] Claire Le Goues, Ciera Jaspan, Ipek Ozkaya, Mary Shaw, and Kathryn T Stolee. 2018. Bridging the gap: From research to practical advice. *IEEE Software* 35, 5 (2018), 50–57.

[13] Loet Leydesdorff and Henry Etzkowitz. 1998. The triple helix as a model for innovation studies. *Science and public policy* 25, 3 (1998), 195–203.

[14] Silverio Martínez-Fernández, Justus Bogner, Xavier Franch, Marc Oriol, Julien Siebert, Adam Trendowicz, Anna Maria Vollmer, and Stefan Wagner. 2021. Software Engineering for AI-Based Systems: A Survey. *arXiv preprint arXiv:2105.01984* (2021).

[15] Tommi Mikkonen, Casper Lassenius, Tomi Männistö, Markku Oivo, and Janne Järvinen. 2018. Continuous and collaborative technology transfer: Software engineering research with real-time industry impact. *Information and Software Technology* 95 (2018), 34–45.

[16] Jakki Mohr and Robert Spekman. 1994. Characteristics of partnership success: partnership attributes, communication behavior, and conflict resolution techniques. *Strategic management journal* 15, 2 (1994), 135–152.

[17] Sergio Rico. 2020. *Approaches to improve researcher-practitioner communication in joint software engineering research.* Ph.D. Dissertation.

[18] Sergio Rico. 2020. Exploring and improving industry-academia communication in software engineering. In *Proceedings of the Evaluation and Assessment in Software Engineering*. 379–382.

[19] Sergio Rico, Nauman Bin Ali, Emelie Engström, and Martin Host. 2020. Guidelines for conducting interactive rapid reviews in software engineering – from a focus on technology transfer to knowledge exchange. https://doi.org/10.5281/zenodo.4327725

[20] Sergio Rico, Emelie Engström, and Martin Höst. 2019. A taxonomy for improving industry-academia communication in IoT vulnerability management. In *2019 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE, 38–45.

[21] Per Runeson and Martin Höst. 2009. Guidelines for conducting and reporting case study research in software engineering. *Empirical software engineering* 14, 2 (2009), 131.

[22] Alex Serban, Koen van der Blom, Holger Hoos, and Joost Visser. 2020. Adoption and effects of software engineering best practices in machine learning. In *Proceedings of the 14th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. 1–12.

[23] Margaret-Anne Storey, Emelie Engström, Martin Höst, Per Runeson, and Elizabeth Bjarnason. 2017. Using a visual abstract as a lens for communicating and promoting design science research in software engineering. In *Proceedings of the 11th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. IEEE Press, 181–186.

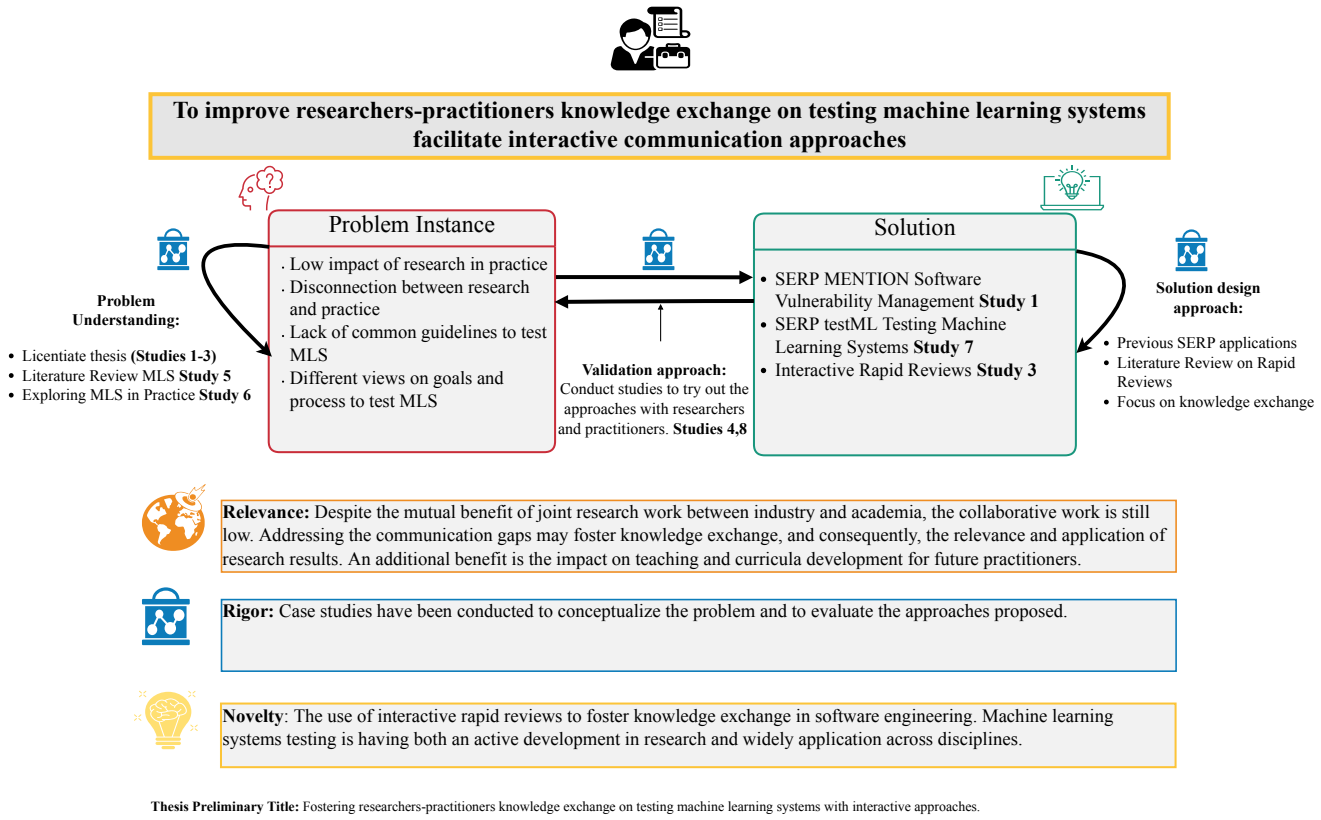

Figure 2: Visual abstract[23] for the Ph.D thesis