Electronic Research Archive of Blekinge Institute of Technology
<http://www.bth.se/fou/>

This is an author produced version of a conference paper. The paper has been peer-reviewed but may not include the final publisher proof-corrections or pagination of the proceedings.

Citation for the published Conference paper:

Title:

Author:

Conference Name:

Conference Year:

Conference Location:

Access to the published version may require subscription.

Published with permission from:

Finding Relevant Research Solutions for Practical Problems - the SERP Taxonomy Architecture

Kai Petersen
Blekinge Institute of Technology
Department of Software Engineering
37140 Karlskrona, Sweden
kai.petersen@bth.se

Emelie Engström
Lund University
Department of Computer Science
Lund, Sweden
emelie.engstrom@cs.lth.se

ABSTRACT

Background: Experience and research indicates that there exist a communication gap between research and industry in software engineering.

Objective: We propose the Software Engineering Research and Practice (SERP) taxonomy architecture to support communication between practitioners and researchers. The taxonomy architecture provides a basis for classifying research from a problem perspective which in turn supports the breaking down of complex practical challenges to researchable units. Thus such taxonomy may support the mapping of challenges in industry to research solutions in the software engineering context.

Method: In this paper we present SERP and exemplifies its usage based on two literature studies in the field of software engineering. Further, we discuss how a taxonomy based on this architecture could have helped us in two past research projects that were conducted in close collaboration with industry. Finally we validate SERP by applying it to the area of software testing, developing SERP-test, and interviewing two industry practitioners and two researchers.

Results: The taxonomy architecture has been applied to two problems in software testing, and has been assessed through interviews with practitioners and researchers. The interviews provided suggestions of how to improve the taxonomy architecture, which have been incorporated. With two examples, we demonstrated how the taxonomy architecture could be used to find solutions for industrial problems, and to find the problems addressed by a particular solution.

Conclusion: SERP may be useful in multiple ways: (1) Given that SERP taxonomies are populated with industrial problems and scientific solutions, we could rapidly identify candidate research solutions for industrial practice. (2) Researchers could benefit from the taxonomy in the reporting of their research to ease the mapping to industrial challenges.

Categories and Subject Descriptors

D.2.5 [Testing and Debugging]: Diagnostics

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

WISE '2014 Västerås, Sweden

Copyright 2014 ACM 978-1-4503-3045-9/14/09 ...\$15.00.

<http://dx.doi.org/10.1145/2647648.2647650>.

General Terms

Practice

Keywords

Evidence-Based Software Engineering; Industry and Research Gap; Software Engineering

1. INTRODUCTION

There is a communication gap between academia and industry. A currently ongoing survey¹ of software engineering publication forums used by practitioners to gain scientific knowledge has shown that practitioners mostly do not access scientific journals. They rather access books, blogs, and social media. Their most useful information is not obtained from scientific publication forums (scientific journals and conferences).

One major initiative to bridge the gap between software engineering practice and research is evidence-based software engineering (EBSE). EBSE aim to support adoption of new software engineering interventions based on sound empirical evidence. Inspired by evidence-based medicine Kitchenham et al. proposed to apply EBSE to software engineering [12]. The EBSE process consists of the steps of: (1) identifying the need for evidence; (2) track down the evidence that satisfies the need; (3) critically appraise the evidence collected; (4) evaluate the evidence based process used.

EBSE requires two general types of information for evidence, problem-focused and solution-focused. Problem-focused EBSE studies start with a practical problem (e.g. how to improve an automotive testing process [11], or how to conduct release planning [23]), while solution-focused studies focus on a solution or group of solutions and investigates its effects in different contexts (e.g. reviews on Test Driven Development [13] or Pair Programming [9]). Mostly, systematic reviews are driven by reviewing literature either from a scientific research problem or solution focus. There is a need to systematically connect the two, as bridging them facilitates closing the gap in communication between researchers and practitioners.

Taxonomies are a good way to start addressing this challenge. The differences in understanding between practitioners and researchers manifests inter alia in terminology, which in turn cement existing differences and the communication gap between different groups of people increase. To bridge those gaps, differences in perspectives need to be identified

¹[http://www.bth.se/tek/aps/kps.nsf/attachments/Sources_pdf/\\$file/Sources.pdf](http://www.bth.se/tek/aps/kps.nsf/attachments/Sources_pdf/$file/Sources.pdf)

and resolved. Investigating and controlling terminology use is one way to start. Many taxonomies in software engineering have been developed to support communication within groups of practitioners or within groups of academics, respectively. Such taxonomies support the common understanding of a specific topic within those groups through classification of for example a group of techniques or process activities. Few taxonomies are used by both researchers and practitioners, maybe because the communication gap has grown too big and it is difficult to develop such taxonomies. To overcome this difficulty we need support the development of SERP taxonomies by guiding the identification terminology differences from a very high-level perspective.

We propose the Software Engineering Research and Practice (SERP) taxonomy architecture to ease mapping between industrial problems and research solutions, and hence improve communication between practitioners and researchers during technology transfer activities and research collaborations. A taxonomy architecture provides a basic building block to built taxonomies in different areas of software engineering e.g. requirements engineering, software testing, etc.) to help bridging the communication gap in these disciplines.

The remainder of the paper is structured as follows: Section 2 presents the related work. Section 3 presents the SERP taxonomy architecture and how to use it. (Section 4) further demonstrates the usage of SERP from the problem and solution perspective utilizing the examples of existing systematic literature reviews. Section 5 presents an initial evaluation of the taxonomy. In Section 6 we discuss implications for research and practice. Section 7 concludes the paper.

2. RELATED WORK

Multiple research works present solutions and studies to facilitate the transfer of research results to industry. As mentioned in Section 1, the EBSE approach proposed by Kitchenham et al. [12] has been widely adopted in the context of software engineering. The main purpose of EBSE is to be able to choose which solution proposed in the software engineering literature to adopt based on sound evidence. Thereby, it is important to assess the quality of studies and ensure the reliability of study conduct (referred to as rigor). However, another important dimension is practical relevance. Practical relevance is given if the evaluation has been conducted under realistic conditions (e.g. sufficient scale, real world systems, and utilization of practitioners) [10]. Though, as pointed out by Rainer et al. [17] empirical evidence is not the main criterion for technology transfer from the industrial perspective. This finding is supported by our currently ongoing survey of which forums are used practitioners when they would like to gain new scientific knowledge; they rely mainly on blogs, social media, and books of major publishers.

Gorschek et al. [8] proposed a process to support the achievement of practical relevance, shown in Figure 1. The process follows an iterative approach, distinguishing between activities conducted in academia and industry. First, an industrial problem is identified, and relevant solutions are gathered from literature. In this step, our taxonomy would provide support as it systematically connects practical problems to research solutions. Based on the findings from the literature, solutions are modified or new solutions are suggested and first piloted in an academic context (e.g. through

Figure 1: Technology Transfer Model [8]

experiments) prior to starting the technology transfer to gain some confidence before making an investment in industry. Thereafter, the solution is iteratively evaluated and improved in a real-world context (e.g. through a series of case studies [19] or action research [2]). Gorschek et al. [8] distinguish between static validation (gathering feedback from practitioners prior to use) and dynamic validation (actual use of the solution in industry).

Wohlin et al. [26] surveyed researchers and company representatives in Sweden and Australia to determine success factors. There was a high level of agreement between academia and industry that the most important factors are buy-in and support from company management, and having a champion at the company. However, all processes for academia-industry transfer are highly relying on the presence of researchers being familiar with evidence-based processes to identify the possible solutions. There is a need to make it easier to link practical problems to research solutions. Therefore, we need support and infrastructures to answer the following questions:

- Which solutions exist to address a specific practical problem observed in the software industry?
- Which problems are addressed by a solution to a software engineering problem relevant to the software industry?

With a sufficient infrastructure, for which the SERP taxonomy architecture should form the baseline, our vision is that practitioners themselves could identify candidate research solutions for their practical problems and challenges.

3. SERP TAXONOMY ARCHITECTURE

3.1 Structure

The SERP taxonomy architecture is shown in Figure 2. Four types of entities are identified as important to classify, namely interventions, context constraints, objective/effects, and scopes. An intervention is constrained by one or more factors in its application context and has one or more objectives or known effects within a certain scope. Each entity of the taxonomy architecture is generic. Thus, the different disciplines in software engineering would have to define

Figure 2: SERP Taxonomy Architecture

their specific categorizations of interventions, contexts, objectives/effects, and scopes.

- **Intervention:** An intervention is defined as an act performed (e.g. use of a technique or a process change) to solve a practical software engineering problem. Many classifications exist already. For example, an overall structure of different areas is presented in Software Engineering Body of Knowledge (SWEBOK) [4].

However, they may not be sufficient if they do not incorporate the problem-focus or classify a sufficient scope as described by the discipline (such as in testing, where many classifications of interventions focus on types of information sources for test case design, and hence only focus on a small part of the over all testing process).

- **Context constraint:** Any intervention proposed will be applied in a context, and may be constrained by that. For example, prerequisites such as specific knowledge and competence required are documented here. Further contextual elements of interest could be obtained from [14] and [6].

However, not all context factors are relevant to classify. SERP suggests a focus on the primary factors preventing applicability of a software engineering intervention or correspondingly, from a problem perspective, the primary factors delimiting the selection of possible interventions.

- **Objective and Effect:** The objectives describe what should be achieved by using the intervention. The effect is a measurable or observable variable quantifying the effect. From a solution perspective focus is on empirical shown effects or assumed effects of an intervention while from the problem perspective focus is on the practitioners objective for applying an intervention.
- **Scope:** The scope of an intervention describes the extent of its intended effect. Correspondingly the scope of a challenge describes which part of the testing process is primarily affected by the challenge or desired improvement.

Each classification of either a challenge or a solution will create links between nodes in the four classification trees. These links may then support activities aimed at bridging

the gap between industry and academia such as for example formulating relevant research questions, identifying relevant points of comparison between interventions, describing industrial challenges with a research perspective and searching for relevant empirical evidence for described challenges.

Given that traceability links exist between the entities of the taxonomy, one can enter the taxonomy at any point (see links between the entities in Figure 2). For example, one could select an objective/effect and find all interventions related to that effect. Knowing the interventions, one could directly trace down the evidence related to them, and also compare different outcomes of interventions that fulfill the same effect or objectives.

3.2 Realization Scenario

We envision a scenario where practitioners as well as researchers provide information based on the SERP taxonomy architecture in their respective disciplines. This is conceptualized in Figure 3.

Practitioners describe their practical software engineering problem qualitatively towards a system (we envision a database that is populated through a survey). After having described their problem, they are asked to classify it according to classification specific to the discipline of interest. Having a common classification is important to be able to make a sufficient match between problems and solutions. The practitioners classify the scope and the objective and effect. From their perspective the objective and effect are what they wish to achieve by solving the problem described.

The researcher follows a similar approach. They describe the solution and also classify it according to scope and objective. Furthermore, they specify the context and constraints that they either assume or observed for their solution. From their perspective the objective and effect documents what they wish to or actually achieved with their solution.

Given that both, researchers and practitioners, utilized the same classification to describe the problem and the solution, now a matchmaking could be made, as is indicated by black nodes in Figure 3. Our proposition is that having a sufficient number of challenges and solutions, practitioners could enter the database to lookup whether their problem is already described and classified, thus being able to rapidly

Figure 3: Realization

identifying solutions. Vice versa, researchers may find challenges related to their solutions, hence being able to confirm their practical relevance. Further, building such a connection may open opportunities for collaboration and evaluation of solutions in an industrial context.

4. EXAMPLE APPLICATION SCENARIOS

The most common scenarios from where practitioners would start in the taxonomy is from the problem/objective or solution/intervention perspective.

4.1 Scenario 1: From Solution/Intervention to Problem/Objective

In the solution based example we start with the intervention (see Figure 4). For the intervention we may identify the primary context constraints (or prerequisites) relevant for its application. In that way we are aware what has to be in place to practically apply the technique. Thereafter, we determine which scope the intervention is relevant for. This can be one or several scopes. Within each scope the use of the intervention fits one or several purposes, which can have one or multiple effects. The effects are what should be empirically observable and measurable.

Example: The example from a practitioner perspective is to decide whether to utilize test driven development (TDD) or test last development (TLD). We formulated the example based on the results in the systematic literature review conducted by Munir et al. [13]. In the example below we have an example where the intervention to be investigated for comparisons is TDD, i.e. the intervention is the starting point. We would first group the interventions according to their scopes (for example we group all TDD studies related to studies on planning, execution, and design). The scope determines what we want to improve, not just the part where the intervention is introduced. As an example, in a TDD student experiment we might just improve the test

Intervention	Scope	Objective /Effect
		Objective /Effect
	Scope	Objective /Effect
		Objective /Effect
	Scope	Objective /Effect
		Objective /Effect
	Scope	Objective /Effect
		Objective /Effect
Context	Scope	Objective /Effect
		Objective /Effect

Figure 4: Scenario 1: From Solution/Intervention to Problem/Objective

design activity, while when introducing TDD in a company the whole development life-cycle may be the scope. On that level we can now compare studies across the same scope. On the next level we look into the purposes that have been investigated, which can for example be to tailor the intervention for a particular context (e.g. using TDD for embedded systems) or improvement (e.g. improving efficiency of TDD) within the scope. The purposes are then evaluated with respect to their effects, e.g. we see which studies provide favorable results for specific outcomes, and which ones do not. Studies that are grouped on that level could, for example, be considered for a meta-analysis [16] given that the statistical data and consistency of quantifying the variables is visible.

4.2 Scenario 2: From Problem/Objective to Solution/Intervention

The practitioner starts specifying the problem, and states what objectives should be achieved by solving that particular problem, and what the desired effects are (see Figure 5). The problem is stated for a specified scope, which further narrows down the set of possible solutions. Given the links in the SERP taxonomy architecture, all interventions made available in the structure are shown, and how they are constrained by the context. This allows to find and compare all solutions relevant for the problem specified.

Example: The scenario discussed in this context is based on the systematic review of how the test selection problem could be solved (cf. [20]). A common practical problem is ever growing regression test suites [1]. Hence, test selection is an essential step for the next test run, given that running the whole suit takes very long time. That is, the objective is to select the test set in a way that testing is improved, for that interventions (test selection strategies) have to be identified. An effect that could be measured is, for example, average percentage of faults detected [25], or time efficiency

Objective /Effect	Interv.	Context
		Context
	Interv.	Context
		Context
	Interv.	Context
		Context
	Interv.	Context
		Context
Scope	Interv.	Context
		Context

Figure 5: Scenario 2: From Problem/Objective to Solution/Intervention

[5]. The measure of effect chosen very much depends on the goal of the organization of what aspect of testing should be improved. The scope is the execution of tests, since an existing test suit is available that has already been designed. Now all types of interventions solving that problem are identified through the linkage of objectives and effects to interventions. A wide range of interventions is presented in [20], such as coverage-based, fault-based, modification-based, requirements-based, genetic-based, and so forth. In many studies it has been identified that context is often not well specified. Hence, this information may not be readily available or deducible.

5. EVALUATION

We evaluated the SERP taxonomy architecture by using it for developing a SERP-test taxonomy, which is not in focus of this paper, and interviewing practitioners and researchers within the field of software testing.

5.1 Method

We conducted interviews with two industry practitioners and two researchers to gather feedback on the taxonomy. Table 1 shortly summarizes the profiles of the interviewees in terms of role and expertise.

We conducted semi-structured interviews [18] while giving a walk-through of the SERP taxonomy architecture and its application on the area of software testing, the SERP-test taxonomy. The interview was structured as follows:

1. Introduction: A short introduction was given to the participants. In particular, it was highlighted that a high amount of solutions is available, though it is hard to identify relevant points of comparisons between the solutions.
2. Feedback on purpose: How would such a taxonomy help you in your testing research (a) or as a practitioner (b) in addition to what we already found the use would be?
3. Presentation and feedback on the SERP taxonomy architecture:
 - Please provide your general reflections about the taxonomy architecture presented.

Table 1: Interviewees' Profiles

ID	Expertise
1	The first interviewee is a researcher in software testing. He is currently a PhD student, who obtained a Licenciante in Technology (a degree half-way to the PhD). Furthermore, he has three years of industrial experience as a software developer.
2	The second interviewee is a researcher in software engineering, focusing on software testing. He received a PhD in 2009. Overall, he has over 10 years of experience in conducting research on testing and software quality in general.
3	The third interviewee is an industrial PhD student and have a licentiate degree in computer engineering. He has worked with software integration, verification, validation, quality control, and improvements for over 10 years
4	The fourth interviewee is an industrial PhD student. He has worked for 20 years with software for embedded systems. He has worked mostly with model based development and encourages and leads implementation of principles for testing and integration at an early stage of development.

- For the given purposes (designing and documenting comparative studies or searching for research to meet a practical challenge) the taxonomy is supporting them well (rate on 1 to 4): Understandability (the terms/structure is easy to comprehend); Completeness (any important dimension/information missing?); Structure (have we connected the entities in a good way?)

The interviews have been conducted by the second author, who also documented them by taking notes during the meeting. The analysis and results are presented in a narrative manner.

Validity threats: The main concern of validity is the number of interviews conducted. Furthermore, the interviewees were working in software testing, hence to further strengthen external validity there is a need to present it to researchers and practitioners in other disciplines as well. All interviewees (also the practitioners) have connections to research and have an awareness of research publications and hence the needs of the research perspective. Therefore, they are not fully representative in their views of the general practitioner, even though they have substantial experience.

There is always a threat that questions of an interview are not understood as intended. Hence, we piloted the interview and iteratively improved it to increase the understandability. With that the risks related to the quality of the research instrument were reduced.

5.2 Results

We present the results from the researchers' and practitioners' perspective, and thereafter compare the two.

5.2.1 Researcher perspective

Both researchers agreed that there is a need to structure information in a systematic way.

The first interviewee provided several concrete suggestions of how to use the taxonomy architecture. That is, to find possible applications of research results and to find related work for his research. He highlighted that he does not use existing classifications, even though he likes them. In many cases, as he pointed out his work does not fit into the existing ones. As a vision forward he would like to see a taxonomy that could be incrementally populated, which is well aligned with the intention of the SERP taxonomy architecture. From a rating perspective, the researcher indicated that he liked the structure, though he found it is not that easily comprehensible. With respect to completeness, he pointed out that he is missing the strength of evidence related to the effect-entity. Thus, it may be useful to indicate in comments whether an effect is confirmed with strong evidence. A rating scheme that could be used for that purpose has been proposed by [10].

The second interviewee highlighted that the structure is good, but has a very broad scope. With the overall structure, this was our intention, given that the SERP taxonomy architecture should be widely applicable. The interviewee also pointed out that the objective could potentially be an intervention itself. This is a question of extracting the essence of the intended classification and stick with that focus for that specific classification. It is true that implementing an intervention or dealing with a context constraint may be described as an objective as well depending on the

situation. Such objective should in turn generate a new classification. A SERP taxonomy and corresponding population strategy need to support this kind of focus keeping. At this point it was not obvious how to create and map substructures in the taxonomy. This hampered understandability. As suggested by the interviewee, more examples will help in making the taxonomy more easily accessible.

5.2.2 Industry perspective

Also, both industry representatives were positive about the relevance of the taxonomy. The third interviewee highlighted that there is a need to take a problem oriented perspective, as the industry knows the problems, but not what solutions to look for. Usually, consultants would sell the solutions. However, as the fourth interviewee pointed out, even though relevant, it is not natural for practitioners to search information like that. Many practitioners, according to him, do not read anything at all. This is however beyond the scope of our taxonomy to solve

Both practitioners pinpointed the need to consider standards and commonly accepted classification schemes when developing SERP taxonomies. For example in the software testing case ISTQB [3] (International Software Testing Qualification Board) is a well known reference in industry. The third interviewee pointed out that he would like to map his own research results using company internal terminology and the fourth interviewee suggested to utilize the Software Engineering Body of Knowledge to structure areas, and to use terminology from standards..

With respect to completeness, the third interviewee would also like to see negative results of an intervention that matches an objective. He liked the possibility of multiple entry points into the taxonomy. Though, the taxonomy should be complemented with a dictionary.

5.2.3 Comparison

Both, from a practitioner as well as a researcher perspective the value of the SERP taxonomy has been supported for the purposes specified.

Even though only a small number of interviews was conducted, two interesting observations could be made when comparing the answers of researchers and practitioners:

- The strength of evidence was only raised from the researcher's perspective, confirming the finding by Rainer et al. that this is initially not a primary concern from the industry perspective.
- The industry perspective highlighted the need of (a) using terminology of standards and (b) capturing negative results in the taxonomy as well.

6. DISCUSSION

6.1 Implications for Researchers

As pointed out in the interviews, researchers could find solutions efficiently when conducting literature studies. With a sufficiently populated taxonomy this could speed up the definition of a valuable solution from an industry perspective.

In order to achieve this, there is a need to focus on developing discipline specific taxonomies for the entities defined earlier (see Figure 2). This should be done in a systematic way

to arrive at an easily expendable and widely accepted taxonomy. Several efforts have been undertaken to systematically develop taxonomies with very well specified purposes. Smite et al. [21] defined a taxonomy to classify software transfers. Unterkalmsteiner [24] defined a taxonomy helping to align testing and requirements engineering. Stricker et al. [22] defined a common terminology for service oriented architectures. Even though the individual approaches of building the taxonomies differ, some common guidelines should be considered that can be learned from them. These should be considered when defining the discipline specific taxonomies, namely:

1. All efforts in building the taxonomies were driven by clearly defined goals (as recommended by Glass and Vessey [7]).
2. A systematic process was followed in taxonomy construction.
3. Taxonomies were built bottom-up, given that relationships were not well understood (cf. [24]).
4. Experts have been involved in the process of taxonomy construction.
5. Taxonomies are validated against their purpose, either through classification based on literature [21], or through industrial case studies [24]. Which strategy for validation to choose highly depends on the purpose of the taxonomy defined in 1.

Practitioners highlighted in the interviews that there is a need to use terminology familiar to them, e.g. based on standards or widely accepted definitions. Examples are the International Requirements Engineering Board [15] and the International Software Testing Qualifications Board [3]. Even though they were not built with a taxonomy in mind that was created for a well defined purpose, the aim of a SERP-taxonomy is to find and extract, as far as possible, a common terminology. The SERP architecture supports that by pinpointing four main components of a problem or solution which needs (but often lacks) a common terminology to enable effective communication between industry and academia. The nodes in a taxonomy represents accepted classifications while leafs represents open issues.

When a taxonomy is well defined, it will not have any value if people do not contribute to populate it with content. There are clear motivations for researchers why they should care. As pointed out in one of the interviews, the researchers may find application areas for their solutions. This facilitates technology transfer as opportunities for collaboration and evaluation become explicit.

A taxonomy based on the SERP-architecture is the interface between the industrial and the academic perspective and as such it forces descriptions of problems and solutions to be relevant from both perspectives. To do so the development and evolution of such taxonomy must be strictly systematic and controlled, e.g. due to that practitioners do not formulate a research but rather a consulting problem.

Considering the technology transfer process presented in Figure 1 the taxonomy directly contributes to the problem formulation by supporting communication between research and practice. Hence, the taxonomy can be well integrated with existing approaches to facilitate the transfer.

6.2 Implications for Practitioners

Researchers have to take the initiative to build the SERP taxonomies using rigorous approaches. However, the taxonomies will be of little value without the practitioners providing their challenges and classifications of these challenges. Having a large number of challenges facilitates the continuous and incremental building of the knowledge base that was suggested by the interviews in Section 5.

The interviews showed that practitioners are very aware of the challenges they are facing, but not necessarily solutions available to them. Through the mapping solutions become readily available for consideration. Furthermore, empirical research of already evaluated solutions may speed up the technology transfer.

7. CONCLUSION

We propose the SERP (Software Engineering Research and Practice) taxonomy architecture as a means to bridge the communication gap between practitioners and researchers in software engineering, which is useful as part of the overall technology transfer process between academia and industry. Our vision is that software engineering researchers develop SERP-taxonomies for the purpose of mapping research results with industrial challenges within their respective areas. Furthermore the SERP-taxonomies should be populated with classifications of known challenges and interventions to create useful traceability links. These traceability links may support several activities to bridge the gap between research and practice, such as (1) definition of relevant research questions, (2) identification of relevant points of comparison between techniques, (3) researchable challenge description and (4) search for relevant research.

The taxonomy structure has been evaluated through four interviews, two with industry practitioners and two with researchers. Both researchers and practitioners agreed on the value of the taxonomy with respect to the specified purposes, namely finding solutions for practical problems, and being able to compare different alternatives to solve those problems. Based on this and our application of SERP to the area of software testing we conclude that the SERP taxonomy architecture is promising as a means to bridge the gap between practical challenges and research solutions.

Though, at this stage the structure of the architecture is not finalized and subject to further improvement. Many practitioners want to describe and analyze their own challenges. To find their own solutions or to show their expertise within an area. They may use a taxonomy under development to structure their thoughts and to look for matching research. While doing so they may also describe what is missing in the taxonomy or why the taxonomy is not supportive in their case. The incentive for adding that information is the possibility to affect research in a way suiting their own needs. The analysis of such information and development of the actual taxonomy need to be done by people with a special interest in the taxonomy as a whole.

Furthermore, future research should focus on building the taxonomies for different disciplines. In our current work we are incrementally developing and populating the SERP-test taxonomy. We use different tools for this, e.g. expert interviews, literature reviews and an online survey. The strength of the taxonomy will show as soon as the taxonomy is populated with a critical mass of information.

8. REFERENCES

- [1] N. B. Ali, K. Petersen, and M. Mäntylä. Testing highly complex system of systems: an industrial case study. In *ESEM*, pages 211–220, 2012.
- [2] D. Baca and K. Petersen. Countermeasure graphs for software security risk assessment: An action research. *Journal of Systems and Software*, 86(9):2411–2428, 2013.
- [3] R. Black, E. v. Veenendaal, and D. Graham. *Foundations of software testing: ISTQB certification*. Cengage Learning, Andover, Hampshire, 3. ed. edition, 2012.
- [4] P. Bourque and R. E. Fairley. *Guide to the software engineering body of knowledge v3*. IEEE Computer Society, 2013.
- [5] H. Do, S. Mirarab, L. Tahvildari, and G. Rothermel. An empirical study of the effect of time constraints on the cost-benefits of regression testing. In *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of Software Engineering (SIGSOFT FSE)*, pages 71–82, 2008.
- [6] T. Dybå. Contextualizing empirical evidence. *IEEE Software*, 30(1):81–83, 2013.
- [7] R. L. Glass, I. Vessey, and V. Ramesh. Resres: The story behind the paper “research in software engineering: An analysis of the literature”. *Information & Software Technology*, 51(1):68–70, 2009.
- [8] T. Gorschek, P. Garre, S. Larsson, and C. Wohlin. A model for technology transfer in practice. *IEEE Software*, 23(6):88–95, 2006.
- [9] J. E. Hannay, T. Dybå, E. Arisholm, and D. I. K. Sjøberg. The effectiveness of pair programming: A meta-analysis. *Information & Software Technology*, 51(7):1110–1122, 2009.
- [10] M. Ivarsson and T. Gorschek. A method for evaluating rigor and industrial relevance of technology evaluations. *Empirical Software Engineering*, 16(3):365–395, 2011.
- [11] A. Kasoju, K. Petersen, and M. Mäntylä. Analyzing an automotive testing process with evidence-based software engineering. *Information & Software Technology*, 55(7):1237–1259, 2013.
- [12] B. A. Kitchenham, T. Dybå, and M. Jørgensen. Evidence-based software engineering. In *26th International Conference on Software Engineering (ICSE 2004)*, pages 273–281, 2004.
- [13] H. Munir, M. Moayyed, and K. Petersen. Considering rigor and relevance when evaluating test driven development: A systematic review. *Information & Software Technology*, 56(4):375–394, 2014.
- [14] K. Petersen and C. Wohlin. Context in industrial software engineering research. In *Proceedings of the Third International Symposium on Empirical Software Engineering and Measurement (ESEM 2009)*, pages 401–404, 2009.
- [15] K. Pohl and C. Rupp. *Requirements engineering fundamentals: a study guide for the Certified Professional for Requirement*. Rocky Nook, 2011.
- [16] Y. Rafique and V. B. Misic. The effects of test-driven development on external quality and productivity: A meta-analysis. *IEEE Trans. Software Eng.*, 39(6):835–856, 2013.

- [17] A. Rainer, D. Jagielska, and T. Hall. Software engineering practice versus evidence-based software engineering research. In *Proceedings of the ACM Workshop on Realising evidence-based software engineering (REBSE '05)*, page 1–5, 2005.
- [18] C. Robson. *Real world research*. Blackwell Oxford, 1997.
- [19] P. Runeson and M. Höst. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, 14(2):131–164, 2009.
- [20] Y. Singh, A. Kaur, B. Suri, and S. Singhal. Systematic literature review on regression test prioritization techniques. *Informatica*, 36(4):379–408, 2012.
- [21] D. Smite, C. Wohlin, Z. Galvina, and R. Prikladnicki. An empirically based terminology and taxonomy for global software engineering. *Empirical Software Engineering*, 19(1):105–153, 2014.
- [22] V. Stricker, A. Heuer, J. M. Zaha, K. Pohl, and S. D. Panfilis. Agreeing upon soa terminology - lessons learned. In *Future Internet Assembly*, pages 345–354, 2009.
- [23] M. Svahnberg, T. Gorschek, R. Feldt, R. Torkar, S. B. Saleem, and M. U. Shafique. A systematic review on strategic release planning models. *Information & Software Technology*, 52(3):237–248, 2010.
- [24] M. Unterkalmsteiner, R. Feldt, and T. Gorschek. A taxonomy for requirements engineering and software test alignment. *ACM Trans. Softw. Eng. Methodol.*, 23(2):16, 2014.
- [25] K. R. Walcott, M. L. Soffa, G. M. Kapfhammer, and R. S. Roos. Timeaware test suite prioritization. In *Proceedings of the ACM/SIGSOFT International Symposium on Software Testing and Analysis (ISSTA 2006)*, pages 1–12, 2006.
- [26] C. Wohlin, A. Aurum, L. Angelis, L. Phillips, Y. Dittrich, T. Gorschek, H. Grahn, K. Henningsson, S. Kågström, G. Low, P. Rovegard, P. Tomaszewski, C. V. Toorn, and J. Winter. The success factors powering industry-academia collaboration. *IEEE Software*, 29(2):67–73, 2012.