An Evidence Profile for Software Engineering Research and Practice

Claes Wohlin

School of Computing, Blekinge Institute of Technology, SE-371 79 Karlskrona, Sweden
Email: Claes.Wohlin@bth.se

Abstract Evidence-based software engineering has emerged as an important part of software engineering. The need for empirical evaluation and hence evidence has grown in the last couple of decades when developing new models, methods, techniques and tools in research. Furthermore, industrial decision-making ought to become more evidence-based. The objective here is to develop and present an evidence-based profile, which could be used to divide pieces of evidence into different types and hence creating an overall picture of evidence in a specific case. The evidence profile is developed in such a way that it allows evidence to be judged in context. The evidence profile consists of five types of evidence, and the profile is illustrated for perspective-based reading. It is shown how pieces of evidence can be classified into the different types. It is concluded that this type of approach may be useful to capture the evidence with respect to a specific topic and in a specific context. The further work includes applying the evidence profile for evidence collected from different type of studies and contexts.

Introduction

Evidence is the basis for decision-making. We daily take decision based on the information available, and hence we practice informal evidence-based decision-making. In science, decision-making should be more formalized. This is done by the introduction of an evidence-based research approach. Evidence-based medicine has been practiced a long time as described by for example [1]. The concept of evidence-based research was introduced into software engineering in 2004 [2] and also presented from a practitioner's point of view by Dybå et al. [3].

A key challenge in all evidence-based research is the synthesis and valuation of the evidence available. In evidence-based medicine the highest level of evidence is based on randomized controlled trials. However, in software engineering the best evidence is definitively context-dependent. This is not only a concern in software engineering, it is also a criticism raised in evidence-based medicine [4].

Synthesis and valuation of evidence can start from two main standpoints: research or practice. In the former case, researchers are trying to synthesize evidence to capture what we know, in some sense, objectively about a specific model, method, technique or tool. In the latter case, the evidence must be re-valued and interpreted in different contexts, for example, different application domains, process models or companies. Something that may be perceived as relevant and useful evidence in research or in one context may not be as highly valued in another context. For example, in a contact with a large telecommunication company, they wondered about the available evidence in relation to productivity and quality changes when moving development of software products from one site to another site. The manager asking the question had a gut feeling, but wanted scientific evidence to better argue "his case" when discussing the challenges with higher level management. Unfortunately, the evidence found was not in the telecommunication domain, and hence perceived as having limited value in the argumentation [5]. This illustrates the need to take context into account when discussing evidence and its value with industry.

The most commonly known use of evidence is probably in law, since it forms the basis for a modern society's juridical system. Thus, a classification of evidence according to the juridical system is here taken as the starting point to introduce levels of evidence for software engineering. A general model is introduced and its use in specific contexts is discussed. To illustrate the model, studies of perspective-based reading are classified into the model first from a "pure" research perspective and then re-valued and interpreted in a specific industrial scenario. The latter is needed given that the evidence must be viewed as being context-dependent as argued above. It is concluded that it is possible to classify evidence and hence package the available evidence using a generic model for research and also into specific cases being relevant for specific industrial contexts.

The remainder of the chapter is structured as follows. Next, related work on synthesis of evidence in software engineering is presented. It is followed by a description of the generic model for valuation of evidence for software engineering. The model is then illustrated for perspective-based reading. After which the chapter is concluded with a summary, including some future work.

Related work

Discussions about the need to synthesize research in software engineering started before the introduction of the concept of evidence-based software engineering. Some examples can be found in literature from the late 1990ies [6, 7, 8]. Pickard et al. discuss combining research results [6]. Miller [7] and Hayes [8] both address the issue of combining research results through meta-analysis. The authors stress the need for systematic combination of research results. They stress the need not only to conduct individual research studies, but also to build knowledge from

combining findings from different studies on a topic. Basili et al. [9] present some early work along these lines when the authors address how to combine a set of research studies and hence knowledge we have in relation to software inspections. Ciolkowski [10] follows up on this line of research by conducting a meta-analysis of perspective-based reading, which is used as a starting point for the illustration later in the chapter.

As a response to the need to collate evidence in software engineering, Endres and Rombach [11] systematized and presented a number of empirical observation, theories and laws in relation to software engineering. The authors have collected a number of recurring phenomena in software and systems engineering.

Despite the increased focus on conducting systematic literature studies [12] in software engineering, there is still too little attention on conducting research synthesis. This needs to change and synthesis needs to be an integral part of systematic reviews to increase their significance and usefulness for research and practice [13]. According to [13], most synthesis is narrative or thematic. Different steps for conducting thematic synthesis are discussed by Cruzes and Dybå [14]. The lack of or at least limited synthesis in software engineering is a challenge for both researcher and practitioners alike. Researchers need proper synthesis to identify research gaps and to be able to generalize research results. Practitioners need synthesis to obtain evidence for decision-making. The latter challenge is also addressed by Pfleeger [15] when discussing the problems for industry to build knowledge from the evidence available in individual research studies. Thus, it can be concluded that there is indeed a need to support both researchers and practitioners in their decision-making process by providing packaged evidence, and not only pieces of evidence.

Different ways of combining evidence have been proposed such as meta-analysis and vote counting [6]. Meta-analysis is primarily a statistical method for combining findings from different studies, while vote counting is more a straightforward count of results pointing in a certain direction. The way to combine evidence is far from straightforward in particular if taking a specific context into account such as a specific industrial context.

The objective here is address the gap in the research literature related in particular to the synthesis and valuation of evidence from both a research perspective and an industrial point of view. This is done by introducing and illustrating a model for valuing evidence of different types and to take the context into account. The combination of evidence is based on a generic model, but the key point is that the evidence must be judged in each specific case. The model should act as a starting point for combining evidence, and should not be perceived as a prescriptive model.

Evidence profile

Based on the above, it is concluded that a model for valuation of evidence is needed. The model presented here is influenced by how evidence is used in criminal law. Thus, the basic conjecture for the work is: Decisions regarding the use of models, methods, techniques and tools in software engineering could be made as law is practiced, although with a lower level of confidence. The latter is added, since the evidence does not have to support the case beyond any reasonable doubt. The key issue is that the evidence is reasonable and it is cost-efficient to act according to the evidence, and the cost may be viewed differently whether representing academia or industry.

Admittedly criminal law is different in different countries, but the basic levels of evidence are still very similar in terms of strength (the levels and interpretations have been discussed with a lawyer with long experience from criminal law in Sweden). If looking at evidence and other sources of information from strong to weak, although the actual order between two items could be argued, we have:

Evidence

1. Physical evidence – for example documentation, digital traces, fingerprints and genetic information (DNA).
2. Eyewitnesses – statements from trustworthy witnesses.
3. Expert witnesses – statements about the accused person from expert witnesses (typically medical doctors or psychologists).
4. Circumstantial evidence – circumstances that indicate guilt, but it is not proof. This could be having a motive or being at the scene of the crime.

Other sources of information

5. Hearsay – second hand information.
6. Self-statements – statements from the accused person (typically with vested interest).
7. Suspicion – A feeling of distrust.

The three lowest levels are not evidence. However, even suspicion is important since it may, as in the case with transferring a software product, be the starting point to try to identify real evidence. Thus, the suspicion is from an industrial point of view very close to what is often referred to as “gut feeling”.

The main research hypothesis in this work is: *It is possible to systematically structure different types of claims and evidence in software engineering to allow for more informed decisions regarding research gaps and the use of models, methods, techniques and tools in specific contextual software development practices*.

A key point here is that the claims and evidence must be evaluated and valued in a specific context whether being in academic research or in specific industrial settings. A structuring of industrial context factors to take into account can be found in [16]. Some examples of contextual factors include: application domain,

size of project and specific technical factors that are deemed to be relevant, for example a specific programing language or use of a specific process model. Evidence from an agile project using Java may not be viewed as relevant for a more plan-driven development environment using C. At the end, each case must be judged separately since it is impossible to state exactly which context factors are important in a specific case.

When it comes to the evidence, several aspects must be taken into account. The aspects are:

- Quality of evidence – an eyewitness may be perceived as very reliable or not. This may be due to exactly what a person remembers about a situation. Reliability of evidence comes from triangulation, i.e. different pieces of evidence corroborates each other.
- Relevance of evidence – given the situation some evidence may be viewed as more relevant than in other cases. For example, a fingerprint in a murder committed in a house is of different importance whether it comes from the homeowner or a burglar.
- Aging of evidence – this relates in the juridical system to memory and time. In software engineering, it is more related to technology change and hence evidence may have aged too much and hence not being perceived as relevant any more.
- Vested interested – evidence given by a person who has a vested interested in the outcome must be viewed differently than if the person is perceived as being objective.
- Strength of evidence – this refers to the strength of the evidence as such, i.e. along the types of evidence listed above. However, the previous four bullets may affect the perceived strength of the evidence.

Inspired by the list of seven types of evidence and other sources of information, a model of five evidence levels is proposed as a general model for handling evidence in software engineering. The reason to collapse it to five levels are that levels 5-7 are not really evidence at all and levels five and six are combined into one level, since their level of trustworthiness may be viewed as quite similar. Hearsay is normally not admissible as evidence in court, and statements from the accused person is normally stated in self-interest unless it is an admittance of guilt. Independently, both these types must be viewed as weak when it comes to trust and basing decision on them, and hence they have been combined into one level in the model. Furthermore, the order of levels 3 and 4 are swapped in the proposed model, since in a specific field, such as software engineering, the people involved are mostly experts and hence external experts do not exist in the same way as in the criminal courts.

In software engineering, evidence may support, for example, a specific tool or provide evidence against it, and hence evidence may be positive or negative in the case of scientific evidence. For example, a new tool may be significantly better or worse than the current tool, or the results may be inconclusive.

The five general levels of evidence suggested are listed in Table 1 and outlined in general guideline terms below. It should be noted that evidence may be for or against a specific model, method, technique or tool and hence in an evaluation case there is actually eleven cells in the table, i.e. five for positive evidence, five for negative evidence and one for inconclusive evidence from a study.

The empty line is for listing papers/studies with evidence of that specific strength after having taken the four additional aspects (quality, relevance aging and vested interest) into account, and combined them with the general descriptions of the area of interest and interpreted in context. For example, if being interested in inspections of research specifications, then inspections of other artefacts are of less importance than if evaluating a specific inspections technique (or reading technique such as perspective-based reading) more generally. This distinction is further elaborated below when discussing the use of the proposed model or evidence profile for perspective-based reading.

Table 1. Types of evidence.

Positive or negative				
Strong evidence	Evidence	Circumstantial evidence	Third party claim	First or second party claim

The following descriptions should be interpreted as guidelines for valuing different types of evidence. It is not intended to define each type of evidence exactly. On the contrary, it is important that each single empirical study is judged based on its own merits and in the context of interests. The key issue is that evidence is put in one of the five types of evidence (positive or negative), and the placement is possible to motivate given the context of interest. Some guidelines for the five types of evidence:

Strong evidence The conditions for judging evidence as strong in general are as follows: well documented controlled experiment with industrial participants, cross company multi case study, and the studies are conducted by researchers who are independent from the inventor of the object of study. The research should be published after peer review.

Evidence: The following are examples of requirements for the evidence level: well documented controlled experiment with non-representative subjects, or series of case studies within a company, published in a peer reviewed conference or journal, and published by independent researcher with respect to having a vested interest.

Circumstantial evidence: Expectations for this level are: well documented controlled experiment by anyone having a vested interest, well documented single case study, cross company survey, and published in a peer review conference or journal.

Third party claim: The expectations for this level of evidence are: experience report, lessons learned, single company survey, and published anywhere but not by anyone having a vested interest
First or second party claim: Finally, the lowest level of "evidence" includes: any information published by the inventor or by anyone else having a vested interest (for example tool developer).

The objective is that Table 1 should be used to place specific studies, and hence obtaining frequency counts of different levels of evidence. Frequency counts allow anyone using the model to generate a bar chart with eleven bars (from strong positive evidence to strong negative evidence with the middle being no evidence, i.e. an inconclusive study. This is referred to as the evidence profile for the object of study in a specific context. The actual studies placed in each level (positive, neutral and negative) should be presented alongside with the evidence profile to enable transparency of the judgments.

The descriptions of each level should be viewed as a guideline for placing studies on the levels. However, all placements of studies should be motivated to make the actual evidence profile transparent and hence possible to accept or challenge. Thus, it is important to provide a characterization of each study together with the motivation for the placement of each study in the table. When presenting the table (and bar chart) all individual studies must be publicly available.

Placement of individual studies may depend on actual usage of the collated evidence. Thus, the placement of studies may change whether studying research in general or a specific application domain or any other specific context. This is illustrated when discussing the actual use of the evidence profile.

Illustration of model

The descriptions of the five types of evidence support the creation of an evidence profile, where the types of evidence as such help providing weights to the different types of evidence. Not all evidence is equally important and different evidence may be viewed as being of different importance depending on the actual usage of the evidence. For example, the evidence in a research situation may be different than if looking at the actual use of a model, method, technique or tool. Furthermore, different pieces of evidence may be viewed differently in different industrial context for example depending on different application domains or different process models being used. Research evidence must be interpreted in context, and hence the evidence profile will become different. Thus, it is only possible here to illustrate the usage of the evidence profile in a specific scenario.

Here, it is chosen to base the illustration on perspective-based reading in software inspections [17]. The main reason being that Ciolkowski published a meta-analysis of 12 sources related to perspective-based reading [10], which means that

the evidence profile obtained here can be compared with the findings from [10]. It should be noted that Ciolkowski has evaluated perspective-based reading in general, i.e. from a general research perspective while here the evidence profile is created for general research scenario and then also discussed for an industry scenario. This is done to both illustrate the differences and how important it is that evidence is interpreted in context when discussing the usage of different models, methods, techniques and tools in industry.

Ciolkowski lists 12 sources of information regarding perspective-based reading. The selection of these sources is further elaborated in [10]. The 12 sources contain 22 studies, which are used to create the illustration of the usage of the evidence profile. Based on the listing in [10], the 12 sources were downloaded, and evaluated. First, a general profile based on PBR is created, then a specific industrial scenario is introduced, which leads to a discussion on the placement of the pieces of evidence to obtain a context-dependent evidence profile for the scenario presented.

General research scenario – The focus is on effectiveness, i.e. ability to identify defects without taking the time into account. Statistically significant results reported in the original study is viewed as positive. When possible this is based on the p-value and hence independent of the chosen significance level in the original study. Here, results with a p-value less than 0.1 are considered as significant from the evidence profile point of view. The type of artefact inspected is not considered relevant. The type of subject is judged according to the basic description provided above with respect to the evidence types. The authors of the first paper on perspective-based reading are viewed as having a vested interest.

Industry scenario – A telecommunication company is considering changing from their current checklist-based inspection method and to start using perspective-based reading with the objective to increase the effectiveness in detecting defects. They are interested in doing so for inspection of requirements, design, code and test documentation. The company is collaborating with an academic partner, and hence ask about their advice based on the available evidence in literature. We have received questions regarding research evidence related to other topics in our close industrial collaboration [18]. Given that they perceive that inspections are not very domain dependent, they do not only want findings from their own domain. However, they still want evidence to be from development of technical systems. They prioritize evidence from industrial usage of perspective-based reading or experiments with industrial participants.

To address the two scenarios, each source of information is gone through with its studies. The sources are discussed in publication order:

Paper by Basili et al. from 1996 [17] – This paper contains four studies with industry participants inspecting requirements documents. The authors are assumed to have a vested interest given that it is the first paper on perspective-based reading. Three studies provide significant results in favour of perspective-based reading. The fourth study is inconclusive.

Report by Ciolkowski et al. from 1997 [19] – This is a technical report and it contains one significant and one insignificant study. The significant study is run with students inspecting a requirements document. The author list contains one researcher involved in the original study, and hence the studies are run with a vested interest.

Paper by Laitenberger and DeBaud from 1997 [20] – The paper does not compare perspective-based reading with any other reading technique, hence the is not viewed as relevant for the objective of finding evidence in relation to perspective-based reading in comparison to other reading techniques.

Report by Ciolkowski from 1999 [21] – The report includes four studies and none of them are significant.

Paper by Biffl from 2000 [22] – The paper contains one study with students. The results are significant, although in favour of checklist-based reading.

Paper by Laitenberger et al. from 2000 [23] – The paper includes one study with industry participants inspecting design documents. The study results in significant results in favour of perspective-based reading. One author has a vested interest as a co-author of the original paper.

Paper by Lanubile and Visaggio from 2000 [24] – The paper presents two studies and none of them has significant results.

Paper by Biffl et al. from 2003 [25] – The paper contains one study with students. The results are significant, although in favour of checklist-based reading. The outcome is very similar to the findings in [22].

Report by Sabaliaukaite from 2004 [26] – The report presents one study and the results are not significant.

Paper by Denger et al. from 2004 [27] – The paper presents one study and the results are not significant.

Paper by Lanubile et al. from 2004 [28] – The paper presents one study and the results are not significant.

Paper by Maldonado et al. from 2006 [29] – The paper presents two studies and the results are not significant. Results are reported for both the combination of the two studies and for the studies separately, and hence the reporting differs slightly between [10] and here. The difference is of little interest given that the results were not significant.

In summary, three studies are removed given that they lacked a comparison with another reading technique. Twelve studies did not provide any significant results. Four studies provide significant results from experiments with industry participants. Three of these are from the original study where requirements document were inspected, and one study is from inspection of design documents. One study with students produces significant results in favour of perspective-based reading. Finally, two studies result in significant results in favour of checklist-based reading in comparison to perspective-based reading. Thus, in reality seven studies have to be evaluated to decide their placement in the evidence profile. Three studies are removed and the other twelve are placed in “neutral” given their lack of significant results.

For the general case, the seven studies are placed as follows:

- The three studies from the original study and the additional significant study with industry participants are placed in "evidence". It is positive that the studies are conducted with industry participants, but the authors have a vested interest, which brings the placement down from "strong evidence".
- The significant study with student participants is placed in "first or second order party claim", since the study is conducted with students including one author also being an author of the original study and the source is a technical report.
- The two significant studies in favour of checklist-based reading over perspective-based reading are placed in "evidence", although on the negative side since the significance is in favour of checklist-based reading.

This results in the evidence profile presented in Figure 1 with eleven classes ranging from positive "strong evidence" being "1" via non-significant results with its bar on "6" to negative "strong evidence" being "11".

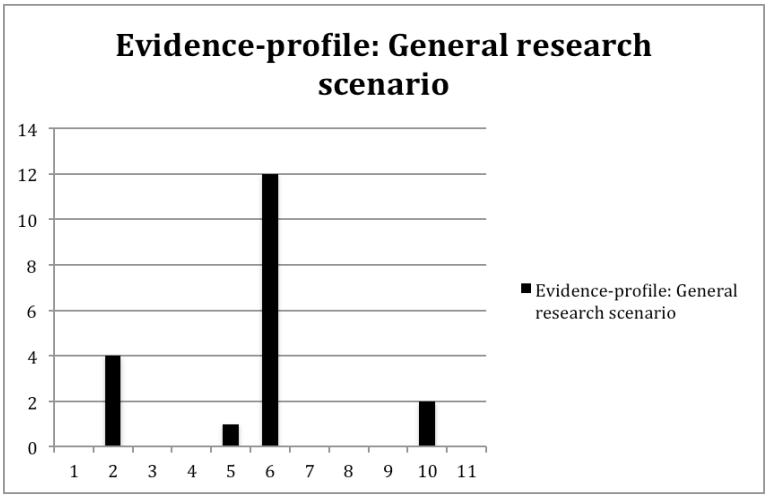

Figure 1. Evidence profile in the general case (General research scenario).

From the research profile in Figure 1 it can be seen that the evidence is quite inconclusive, although more significant results are in favour of perspective-based reading than not. It is noteworthy that all studies with significant results include one or more of the authors of the original paper [17]. Furthermore, both studies significantly in favour of checklist-based reading over perspective-based reading include one joint author [22] and [25]. These observations are aligned with those by Ciolkowski [10].

In this particular case, the industry scenario does not result in any major changes if looking at it generally. However, if looking at the specific phases of interest for the company there is no evidence at all when it comes to inspections of code

and test documentation. Even for design, the evidence is very limited with one significant study. Thus, the company would definitively not change to perspective-based reading for inspections of design, code and test documentation. The results for using perspective-based reading for requirements documents may be considered if the company is prepared to experiment with a new reading technique.

Summary

Evidence-based software engineering has been driven by the need to take informed decision about which models, methods, techniques and tools to use in a specific context. Unfortunately, it is a challenge to synthesize the evidence available. If primarily having controlled experiments, it may be possible to conduct meta-analysis. However, software engineering evidence comes from different types of studies and even synthesizing evidence from controlled experiments is a challenge. Thus, a less formal method is needed. Here an evidence profile has been proposed as a response to the overall research question. It is formulated as a way to capture and visualize evidence. The evidence profile includes five types of evidence, which means that anyone using the approach can classify different pieces of evidence in whichever type found suitable in the specific context. The placement of evidence from different studies must be possible to clearly motivate to ensure that the final evidence profile is found trustworthy and hence useful.

The use of the evidence profile was illustrated with a set of studies of perspective-based reading. This reading technique was used to make the illustration clear. However, the objective is that the evidence profile should be useful for different types of empirical studies including case studies, surveys and other types of empirical studies.

The further research includes evaluating the evidence-based profile approach for other areas than reading techniques, and evaluate its usefulness in relation to synthesis of evidence in systematic literature reviews.

Acknowledgment

The Knowledge Foundation in Sweden supported this work under the grant for BESQ+ (2010-0311).

References

[1] D. L. Sackett, W. M. Rosenberg, J. A. Gray, R. B. Haynes and W. S. Richardson, ”Evidence Based Medicine: What It Is and What It Isn't”, *BMJ* 312 (7023), pp. 71-72, 1996.

[2] B. Kitchenham, T. Dybå and M. Jørgensen, “Evidence-based Software Engineering”, Proc. 26th Int. Conf. on Software Engineering, pp. 273-281, Edinburgh, UK, 2004.

[3] T. Dybå, B. Kitchenham and M. Jørgensen, “Evidence-based Software Engineering for Practitioners”, IEEE Software, Vol. 22, No. 1, pp. 58-65, 2005.
[4] R. E. Upshur, E. G., van den Kerkhof and V. Goel V, ”Meaning and Measurement: An Inclusive Model of Evidence in Health Care”, Journal of Evaluation in Clinical Practice, Vol. 7, No. 2, pp. 91-96, 2001.
[5] C. Wohlin and D. Smite, “Classification of Software Transfers”, Proc. 19th Asia-Pacific Software Engineering Conference (APSEC), pp. 828-837, Hong Kong, 2012.
[6] L. Pickard, B. Kitchenham and P. Jones, “Combining Empirical Results in Software Engineering”, Information and Software Technology, Vol. 40, No. 14, pp. 811–821, 1998.
[7] J. Miller, “Can Results from Software Engineering Experiments be Safely Combined?”, Proc. 6th Int. Symp. on Software Metrics, Boca Raton, Florida, USA, 1999.
[8] W. Hayes, “Research Synthesis in Software Engineering: A Case for Meta-analysis”, Proc. 6th Int. Symposium on Software Metrics, pp. 143–151, Boca Raton, Florida, USA, 1999.
[9] V. R. Basili, F. Shull and F. Lanubile, “Building Knowledge through Families of Experiments”, IEEE Transactions on Software Engineering, Vol. 25, No. 4, pp. 456–473, 1999.
[10] M. Ciolkowski, “What Do We Know About Perspective-Based Reading? An Approach for Quantitative Aggregation in Software Engineering”, Proc. 3rd Int. Symp. on Empirical Software Engineering and Measurement (ESEM), pp. 275-284, Orlando, Florida, USA, 2009.
[11] A. Endres and D. Rombach, “A Handbook of Software and Systems Engineering: Empirical Observations, Laws and Theories”, Pearson/Addison Wesley, 2003.
[12] B. A. Kitchenham and S. Charters, “Guidelines for Performing Systematic Literature Reviews in Software Engineering”, Version 2.3, Technical Report, Software Engineering Group, Keele University and Department of Computer Science University of Durham, 2007.
[13] D. Cruzes and T. Dybå, “Research Synthesis in Software Engineering: A Tertiary Study”, Information and Software Technology, Vol. 53, No. 5, pp. 440-455, 2011.
[14] D. Cruzes and T. Dybå, “Recommended Steps for Thematic Synthesis in Software Engineering”, Proc. 5th Int. Symp. on Empirical Software Engineering and Measurement, pp. 275-284, Banff, Canada, 2011.
[15] S. L. Pfleeger, “Soup or Art? The Role of Evidential Force in Empirical Software Engineering”, IEEE Software, Vol. 22, No. 1, pp. 66-73, 2005.
[16] K. Petersen and C. Wohlin, “Context in Industrial Software Engineering Research”, Proc. 3rd Int. Symp. on Empirical Software Engineering and Measurement (ESEM), pp. 401-404, Orlando, Florida, USA, 2009.
[17] V. R. Basili, S. Green, O. Laitenberger, F. Lanubile, F. Shull, S. Sørumgård and M. V. Zelkowitz, “The Empirical Investigation of Perspective-based Reading,” Empirical Software Engineering, Vol. 1, No. 2, pp. 133–164, 1996.
[18] C. Wohlin, A. Aurum, L. Angelis, L. Phillips, Y. Dittrich, T. Gorschek, H. Grahn, K. Henningsson, S. Kågström, G. Low, P. Rovegård, P. Tomaszewski, C. van Toorn and J. Winter, “Success Factors Powering Industry-Academia Collaboration in Software Research”, IEEE Software, Vol. 29, No. 2, pp. 67-73, 2012.
[19] M. Ciolkowski, C. Differding, O. Laitenberger and J. Münch, “Empirical Investigation of Perspective-based Reading: A Replicated Experiment,” ISERN, Tech. Rep. ISERN-97-13, 1997.
[20] O. Laitenberger and J-M. DeBaud, “Perspective-based Reading of Code Documents at Robert Bosch Gmbh”, Tech. Rep. ISERN-97-14, 1997.
[21] M. Ciolkowski, “Evaluating the Effectiveness of Different Inspection Techniques on Informal Requirements Documents,” Diploma Thesis, University of Kaiserslautern, 1999.
[22] S. Biffl, “Analysis of the Impact of Reading Technique and Inspector Capability on Individual Inspection Performance”, Proc. 7th Asia-Pacific Software Engineering Conference, (APSEC), pp. 136–145, Singapore, 2000.
[23] O. Laitenberger, C. Atkinson, M. Schlich and K. E. Emam, “An Experimental Comparison of Reading Techniques for Defect Detection in UML Design Documents”, Journal of Systems and Software, Vol. 53, No. 2, pp. 183–204, 2000.

[24] F. Lanubile and G. Visaggio, “Evaluating Defect Detection Techniques for Software Requirements Inspections”, ISERN, Tech. Rep. ISERN-00-08, 2000.
[25] S.Biffl, M. Halling and S. Koszegi, “Investigating the Accuracy of Defect Estimation Models for Individuals and Teams based on Inspection Data,” Proc. 2nd Int. Symp. on Empirical Software Engineering (ISESE), pp. 232–243, Rome, Italy, 2003.
[26] G. Sabaliauskaite, “Investigating Defect Detection in Object-oriented Design and Cost-effectiveness of Software Inspection”, Dissertation, Osaka University, 2004.
[27] C. Denger, M. Ciolkowski and F. Lanubile, “Investigating the Active Guidance Factor in Reading Techniques for Defect Detection,” Proc. Int. Symp. on Empirical Software Engineering (ISESE), pp. 219–228, Redondo Beach, California, USA, 2004.
[28] F. Lanubile, T. Mallardo, F. Calefato, C. Denger and M. Ciolkowski, “Assessing the Impact of Active Guidance for Defect Detection: A Replicated Experiment”, Proc. 10th Int. Symp. on Software Metrics, pp. 269–278, Chicago, Illinois, USA, 2004.
[29] J. Maldonado, J. Carver, F. Shull, S. Fabbri, E. Dória, L. Martimiano, M. Mendonça and V. Basili, “Perspective-based Reading: A Replicated Experiment Focused on Individual Reviewer Effectiveness”, Empirical Software Engineering, Vol. 11, No. 1, pp. 119–142, 2006.