2015 IEEE/ACM 3rd International Workshop on Conducting Empirical Studies in Industry

How to Increase the Likelihood of Successful Transfer to Industry

- Going beyond the Empirical

Tony Gorschek
Software Engineering Research Lab Sweden
Blekinge Institute of Technology
Karlskrona, Sweden
tony.gorschek@bth.se

Abstract—The field of Empirical Software Engineering has undergone a much-needed expansion the last decade, and papers of all shapes and sizes are more or less mandated to have an “empirical” part to be published in premiere venues. The positive trend has researchers realizing the benefits, but also the investments needed, inherent to industry collaboration. That is, real practitioners, involved in the development of software intensive product, system, and service development. This paper shortly summarizes lessons learned from over ten years experience of industrial collaboration, and knowledge and technology exchange between applied researchers and industry.

Index Terms—Empirical, Technology and Knowledge transfer, Industry, Applied Software Engineering, Usability, Usefulness, Scalability, Return on Investment, Time to return on Investment, Research co-production.

I. Science and Engineering

Science is the pursuit of truth through the intellectual and applied activity encompassing the systematic study of the structure and behavior of the physical and natural world through observations and experiments (Wikipedia). Thus the fact that software engineering has an empirical *movement* seems to be superfluous. Software engineering should rely on the gathering of empirical results to enable analysis and ultimate synthesis to achieve the discovery of new knowledge.

I was inspired by Glass’ paper The Software-Research Crisis [1]. As a young engineer I wanted to create something that was usable and useful, something that could change how people work, think, and enable them, at least to some small part, in developing better products better.

To actually measure (and I use the term very imprecisely) usability, usefulness and scalability, not to mention return on investment, time-to-return on investment and benefit, of a solution, getting access to real industrial cases is mandatory. Solutions are also an interesting concept. Industry is not primarily interested in research; they are interested in solution (to problems), or new ideas that can make (or save) them money. *Solutions* for the context of this paper can be summarized as any model, framework, technology, tool, process, method, or equivalent that enables organizations to do better products - or do products better. The field of software process improvement has explored this concept widely [2], so has research into technology and knowledge transfer, all identifying many of the same challenges [3]–[5].

This paper presents a straight forward and simple model for industrial collaboration [6][7].

II. Technology and Knowledge Interchange

Experiences from more than ten years, half a dozen research projects, and experiences from working with over twenty companies has accumulated in a common sense, but not always easy to achieve way-of-working (overview Figure 1).

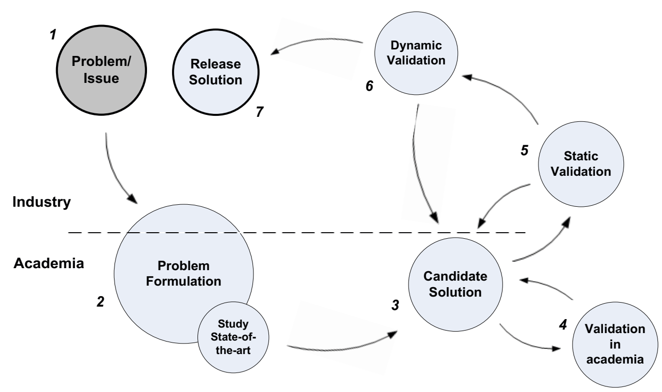

Figure 1. Technology transfer model.

A. Step 1: Problem/Issue

This step serves two main purposes. First, get to know your industrial partner, their process, products and domain to the degree that you can credibly “speak their language” - building trust. Second, you need to objectively assess their product development process to identify challenges. The use of traditional empirical methods like case studies are reasonable, however for enabling deeper understanding a process assessment method is preferable where triangulation of different sources are enabled [8], [9],[10]. In addition the separation of symptoms from actual issues is central and a method like root-cause-analysis can enable this [11]. Light-weight methods are recommended as time and cost are issues.

B. Step 2: Scientific Problem Formulation

As you have a prioritized list of challenges that need to be addressed, based on actual needs identified in Step 1, the challenges need to be packaged into researchable (relevant) “problems”. It is important to check related work and solutions already out there. Reinventing the wheel is not only counterproductive, but also unnecessary as you have plenty of challenges to explore.

The main difficulty of this step is to ensure a continuous dialog with your industrial partner to assure that you stay

978-1-4673-7028-8/15 $31.00 © 2015 IEEE
DOI 10.1109/CESI.2015.9

focused on relevant parts while you are expanding the problem to a researchable and generalizable research topic.

C. Step 3: Candidate Solution

Getting industry to volunteer to do an assessment is not trivial. But it is free, and the worst case is that they get a review. However, you cannot sell them problems, you have to turn them into solutions. A solution is often a package of methods, frameworks and models in combination with processes [12]. A solution does not nessecaraly have to be a completely new concept. There might be concepts out there you can build on, however it is central to focus on the scientific contribution, but more importantly, the probability of being able to create a scalable, usable, and useful solution. Often the two are the same. At this stage finding additional cases (industry partners) is a good investment.

D. Step 4: Validation in Academia

Any solution developed needs to be tested. A controlled test though e.g. experimentation [13] is a good idea to get input for initial refinement. Usually several rounds of academic validation is needed, with continuous refinements based on the results. The main challenge of this step is that a controlled evaluation often demands simplifying the use and application of the solution, as well as working scaled down. Thus this step should be seen as an intermediate step to real validation.

E. Step 5: Static Validation

Static validation involves a “controlled” series of tests, in industry, involving industry practitioners as subjects. It can take the form of workshops, experiments, test-runs where the solution is used and evaluated. Often this step involves training of practitioners and preparation of live and real cases to apply the solution on. This step fills two main purposes. First, the test and collection of feedback for refinement and change of the solution are central. Input in an academic environment (Step 4) is not adequate. More importantly, as you validate the solution in industry, were it is to be used later, you build trust by involving practitioners in the creation of refinements, and indirectly train them in the use of the solution.

F. Step 6 and 7: Dynamic Validation and Release

Given the “evidence” you collected, the trust you build, and the allies you made during the validation steps, the possibility of dynamic validation is real. The previous validation also enabled risk minimization for the industrial partner, as you have tested and re-tested, and involved practitioners in the activities. Without this it would be next to impossible to convince an industrial partner to commit to real dynamic validation.

It is central to realize that this is not “faked” in any way. Nor is it action research [14]. This is handing over the solution to the industry partner, and letting them use it without any interference, help or assistance from the researchers. The practitioners have the option to not use the solution, but also even if they do, they might not use it in a manner you intended.

Dynamic validation does however require a significant investment and preparation. Training of users, support documentation and tools, and other items enabling the use of the solution, have to be developed. Also, a plan for measuring the effects of the solution and how it is used are central for evaluation purposes [15] after the validation. Dynamic validation is one of the few ways to actually test both usability and scalability of any solution.

III. References

[1] R. L. Glass, “The software-research crisis,” *IEEE Softw.*, vol. 11, no. 6, pp. 42–47, 1994.

[2] M. Unterkalmsteiner, T. Gorschek, A. K. M. M. Islam, C. K. Cheng, R. B. Permadi, and R. Feldt, “A conceptual framework for SPI evaluation,” *J. Softw. Evol. Process*, vol. 26, no. 2, pp. 251–279, Feb. 2014.

[3] C. Linz and G. Chroust, “Best Practice Approaches in Know-How and Technology Transfer Methods for Manufacturing SMEs Comparative Description of Best Practice,” *Synthesis (Stuttg).*, 2001.

[4] S. L. Pfleeger and W. Menezes, “Marketing technology to software practitioners,” *IEEE Softw.*, vol. 17, no. 1, pp. 27–33, 2000.

[5] M. Ivarsson and T. Gorschek, “Technology transfer decision support in requirements engineering research: a systematic review of REj,” *Requir. Eng.*, vol. 14, no. 3, pp. 155–175, 2009.

[6] T. Gorschek, P. Garre, S. Larsson, and C. Wohlin, “A Model for Technology Transfer in Practice,” *IEEE Softw.*, vol. 23, no. 6, pp. 88–95, 2006.

[7] C. Wohlin, A. Aurum, P. Rovegård, P. Tomaszewski, C. van Toorn, J. Winter, L. Angelis, L. Phillips, Y. Dittrich, T. Gorschek, and others, “The success factors powering industry-academia collaboration,” *IEEE Softw.*, no. 2, pp. 67–73, 2011.

[8] F. Pettersson, M. Ivarsson, T. Gorschek, and P. Öhman, “A practitioner’s guide to light weight software process assessment and improvement planning,” *J. Syst. Softw.*, vol. 81, no. 6, pp. 972–995, 2008.

[9] M. Svahnberg, T. Gorschek, T. Nguyen, and M. Nguyen, “Uni-REPM: a framework for requirements engineering process assessment,” *Requir. Eng.*, pp. 1–28, 2013.

[10] K. El Emam, W. Melo, and J.-N. Drouin, *SPICE: The theory and practice of software process improvement and capability determination*. IEEE Computer Society Press, 1997.

[11] J. Pernstål, A. Magazinius, and T. Gorschek, “A Study investigating Challenges in the Interface Between Product Development and Manufacturing in the Development of Software-Intensive Automotive Systems,” *Int. J. Softw. Eng. Knowl. Eng.*, vol. 22, no. 07, pp. 965–1004, 2012.

[12] T. Gorschek and C. Wohlin, “Requirements Abstraction Model,” *Requir. Eng.*, vol. 11, no. 1, pp. 79–101, Nov. 2006.

[13] T. Gorschek, M. Svahnberg, A. Borg, A. B. Loconsole, J. Börstler, K. Sandahl, and M. Eriksson, “A controlled empirical evaluation of a requirements abstraction model,” *Inf. Softw. Technol.*, vol. 49, no. 7, pp. 790–805, 2007.

[14] J. U. Ahmed, “Action Research: A New Look,” *Business*, vol. 2, pp. 19–32, 2009.

[15] T. Gorschek, P. Garre, S. B. M. Larsson, and C. Wohlin, “Industry evaluation of the Requirements Abstraction Model,” *Requir. Eng.*, vol. 12, no. 3, pp. 163–190, 2007.