Integration of SE Research and Industry: Reflections, Theories and Illustrative Example

Aiko Yamashita
Mesan AS
Oslo and Akershus University College of Applied Sciences
Email: aikoy@mesan.no

Abstract—Currently, there is limited literature in Software Engineering (SE) that sheds light on the success factors and challenges for knowledge transfer between SE scientists and practitioners. Upon reflections on personal experiences from both academia and industry, this paper attempts to underpin some of the challenges for a successful collaboration, and relate them back to existing theories in the fields of Management, Medicine and Social Sciences. Furthermore, strategies for overcoming some of the challenges are provided and illustrated via a simplified example within the topic of Software Evolution. The intention of this paper is to establish a dialogue for an overall strategy within our field, by providing an illustrative example; and to promote a deeper reflection on the term ‘knowledge transfer’, which has predominantly focused on an unidirectional knowledge flow from academia to industry.

Index Terms—Knowledge transfer, Innovation, Collaboration.

I. INTRODUCTION

Software Engineering (SE), in the same way as many other disciplines, faces challenges to the establishment of successful knowledge transfer schemas with industry. Organisations fail to utilize or capitalize on fundamental findings from scientific literature when building policies or when making decisions. Conversely, practitioners produce new knowledge, but this knowledge is seldom transferred to research, and often remains undisclosed or in the ‘uncharted zone’ of grey literature.

Academic knowledge is based on a model Gibbons et al. [1] describe as Mode 1 knowledge, which is discipline-based. Conversely, Gibbons et al. asserts that Mode 2 knowledge is problem-based, often inter-disciplinary and context-dependent. Mode 2 knowledge is not only becoming predominant in the business environment and commercial R&D, but also has become the the center of attention for policy makers [2]. Although the paper will not debate whether Mode 2 can be equated to the term ‘Applied Research’, it is evident that funding schemas have started to shift progressively under this label, advocating a stronger tie between Modes 1 and 2. Increasing the impact of research on industrial practices has become an important criteria of evaluation for research funding schemas different organizations.

In SE, although collaboration is important for Mode 2 knowledge (e.g., evaluation of techniques/methods in industrial settings), for sub-disciplines as Empirical Software Engineering, collaboration with the industrial sector is deemed as a fundamental driver for knowledge production, much closer to Mode 1 (e.g., empirical evidence from softwareprojects).

Literature in fields such as: Chemistry [3], Medicine [4, 5], Innovation [6, 7], Learning [8], and in particular Management [9–13] have discussed the issue of the existing gap between research and industry. Bloedorn and Stokes [9] asserts that in university-industry collaborations, knowledge transfer is often the exception rather than the rule, and Hansen [6] reports that it is often difficult to implement knowledge coming from outside the companies.

In SE literature on academia-industry collaborations, academia often plays a work force provider role (Beckman et al. [14] and O’Kane et al., [15]) rather than a knowledge producer role. Some examples of work that addresses the issue of actual knowledge transfer between SE research and industry include Kaindl et al., [16] and Rombach and Achatz [17].

Gilbert & Cordey-Hayes [7] propose that knowledge transfer is a dynamic process guided by organizational factors, which may contribute or inhibit its success. Some factors they deemed as important are: time, leadership, and the support of the organizational culture. Despite the fact that many of the factors identified by Gilbert & Cordey-Hayes may apply as well to SE research, there is a need for a deeper discussion of issues and factors within SE. This paper explores some challenges and potential ‘success factors’ based on personal from academia and industry, and attempts to relate them to current theories from more mature fields such as Management and Innovation. Furthermore, the paper discusses a simplified example to illustrate some of the ‘success factors’.

The remainder of this paper is as follows: Section 2 presents the potential challenges and Section 3 describes some potential ‘success factors’ for closing the gap between research and industrial practice, and relates them back to existing theories. Section 4 presents an example to illustrate some of the ‘success factors’. Section 5 concludes with some reflections on the potential role of a ‘collaboration champion’, and challenges the common interpretation of ‘knowledge transfer’.

II. CHALLENGES TO INTEGRATION

The trajectory of my professional career has been industry→academia→industry. Based on personal experiences from working and collaborating in both environments, I describe and explore some of the reasons why integration between academia and industry is be challenging, focusing in
the context of SE. I have seen that mainstream practitioners do not get interested in results from research because:

- They have difficulties understanding it, or do not see any practical use on it
- They do not see any economical potential or have other more immediate priorities
- There are far too many pre-conditions for adoption
- Is perceived as ‘uncool’, ‘too complex’, or ‘too formal’

A. Difficulties understanding it, its implications or usage

I have experienced in many occasions that practitioners have hard time understanding the outcomes from research. Nonaka [18] differentiates between two types of knowledge: the “Know-what” and the “Know-how”. “Know-what” is the more explicit and abstract knowledge such as reports, policies, procedures etc. In contraposition, “Know-how” is tacit knowledge in the sense that it remains within individuals or organizations but is more of a contextual and practical nature. In a way, this distinction is orthogonal to the Modes 1 and 2, described by Gibbons [1]. If we use Nonaka’s classification of knowledge, explicit academic knowledge that is transferred in the form of articles, presentations, reports and books could be turned into tacit knowledge if it is brought into the organization through an employee who is actively using this knowledge in his/her work. One of the problems seems to be the fact that Mode 1 knowledge seems to be the prevalent form in academia, and as such it remains inaccessible from both cognitive and practical perspectives to practitioners.

From a cognitive perspective, Mode 1 remains potentially inaccessible because practitioners do not have the same background, training or preparation in the field as researchers in SE have, and would find these forms of explicit knowledge hard to grasp. Researchers will communicate mostly with their specialty peers. Consequently, they are not required (and thus, potentially not trained either) to simplify and make their work understandable for people outside the field. Partha and David [19] assert that in addition to intellectual joint interests/collaboration within academia, the prevalent reward system (e.g., number of publications) exacerbates this issue. “The curse of knowledge” [20] as a form of cognitive bias amongst researchers can also play a role in this issue. In [11], Starkey cites Leisenring and Todd [p. 74] [21]:

“Because they do not understand the mathematics and statistics that characterise most contemporary research, many articles published in academic research journals today might as well be written in Greek. Until the next generation of practitioners get research training, ways will need to be found for researchers to communicate with the current generation of practitioners”.

From a practical perspective, Mode 1 remains inaccessible because the ‘jump’ between Modes 1 and 2 is not obvious, many times not even for researchers. Researchers may also lack the insider’s perspective that may allow them to formulate the tie between the most urgent and critical needs of an organization and a potential, research-based solution (the Mode 2 knowledge).

The relation between the “What-to” and “How-to” knowledge (or at least, the practical implications of explicit knowledge) plays an intrinsic role on the level of understanding, involvement and sense-making in an organisation. Basically, by facilitating the path to tacit knowledge, researchers will receive less and less the “so what?” response from practitioners. This seems understandable when the system of values (in particular perception on what kinds of knowledge are valuable) differs greatly amongst academics and practitioners [12]. Of course, it could happen sometimes, that there is no foreseeable applicability to Mode 1, but that discussion will be taken in some other occasion.

Another practical perspective that I consider valid is that of physical availability. Yamashita and Moonen [22] report that the two most used sources of information on code smells/antipatterns are Blogs and Internet forums (only 10% claimed to use scientific articles as their source of information). I asked once a developer why practitioners seldom consult scientific articles, and the response was ‘I get the info I need in the forums/blogs’ and ‘I sometimes search for papers, but often I am frustrated cause you have to pay for them, then is no point..’. The first comment alludes to the possibility that although still ‘explicit’, knowledge available on blogs and forms are of more practical, contextualized nature. As for the second comment, a personal belief is that this is one consequence of the prevalence of closed access driven by scientific editorial and publishing bodies. Despite being only an anecdotic example, this might shed a light on the prevalent reluctance of practitioners to look into academic work.

B. Lack of economical incentives / other priorities pop-up

If we rule out start-up initiatives that have spun from research, mainstream practitioners (and in particular managers) do not take great interest on in-house research (the so-called R&D department) nor in the implementation of explicit knowledge stemming from research. As mentioned before, the system of values in industry is highly focused on practical, remunerable applications [12]. Moreover, the sense of time-urgency differs between academia and industry [23], where the focus for profitability stemming from an idea or research result tends to be at the furthest, on a mid-term perspective. Most cases, practitioners’ sense of priority/urgency will be set upon short-term, emergent issues. This behaviour could be related to non-organisational related factors such as competitive pressure, time-to-market, etc.

This situation seems more common in small and mid-sized companies, where they follow a “day-by-day” mantra and they are hesitant on taking long-term economical risks. The same time-urgency factor also often leads to stranded collaborations. Industry wants “readily-made” or “just-in-time” solutions, and they tend to work in a much faster pace than what researchers

1 Many specialized, mature companies have long ago understood the key importance of knowledge production and collaboration with academia, but when I talk about mainstream practitioners, I am mainly speaking about average small to mid-size IT companies, non-specialized companies, or large companies within other domains with an immature IT strategy.
internal business integration to their daily routines or even organisational culture. Kaindl et al. [16] asserts that internal business integration is a necessary condition for a successful transfer initiative. They reported that in some cases, the application of new RE methods interferes with other software development activities, requires the integration of new work processes and tools, and assumes new interaction patterns among stakeholders, including customers. Also, they mention that where certain practices are widely institutionalised, the adoption of new methods is futile without a formal process for changing the status quo. Also, the authors also mention the lack of technical support (from the industrial side) or resources for developing tools (from the academic side), are factors having a negative effect on knowledge transfer. Even if the tools or methods coming from research are promising, they are often too immature to be of use.

The level of intrusiveness also relates to information sharing. Some companies may still operate under a rather conservative view with respect to knowledge sharing and may be reluctant to disclose in-house produced research results or different forms of internal or intellectual data. Collaborations could be viewed as ‘intrusive’ and ‘risky’ to business instead of being viewed as drivers for innovation. Melose et al. [4] reflects upon this attitude, and states that companies need to adopt a new attitude about sharing of information, by adopting “new business strategies that promote value creation through open innovation research networks as opposed to traditional business strategies that promote the development of barriers to competition”. This ‘open strategy’ approach, according to Chesbrough & Appleyard [p. 57-76] [24]: “balances the tenets of traditional business strategy with the promise of open innovation.”

D. Fashion and Perception in IT Industry

Kaindl et al., [16] also discuss that one challenge for technology transfer is the Negative Perception of Formal Techniques. Results coming from an academic settings may face one or other form of prejudice (e.g., “too complicated”; “too theoretical”, “boring”, “not as cool as...”) that needs to be overcome to increase chances of acceptance and adoption. In industry, is not rare that skills or knowledge stemming from academia is dismissed or labelled as ‘theoretical’, where it could be entirely the opposite case. Unfortunately, the current adoption of techniques and methodologies do not necessarily follow empirical evidence or a trial-and-error approach. One example is Lean Software Development [25], whose popularity is increasing in industry despite the fact that yet not enough evidence exists on its viability [26]. Seems as the wave of technology or methodology adoption may obey different rules than our current understanding of what are the necessary conditions for adoption.

Maybe it is worth looking at knowledge adoption process from a sociological perspective and understand better the mechanisms governing it. There is an overly large example of cases where the trends are established by large companies (some of them high-tech, some others from the certifying business) or ‘gurus’, who act as role models driving and dictating perceptions, pretty much like in the Fashion Industry. Kieser [27] explores this issue of Fashion within the context of Management, and explains how many times this could lead to the spread of misconceptions. Jørgensen [28] discusses the phenomenon of myths and oversimplification in Software Engineering, how they occur, and how some of them may potentially have large negative effects within the Software Engineering discipline. One example of oversimplification is the polarized perception of CMMI vs Agile, whiles Turner [29] found that despite some significant differences, the “oil and water” description of CMMI and agile is somewhat overstated.

Moreover, I could not help but to notice that there has been a steady trend on the rebranding or adaptation of already established methodologies/approaches to make them ‘fit’ into IT Industry. One example of rebranding is the Six Sigma [30] based on the traditionally known field of Software Process Improvement. Another example is how much of already known data mining [31] theory and practice is packaged into the overly loaded term of ‘Big Data’...
sacrificing the riguroosity and accuracy that characterizes any scientific discipline. So, we shall not fall into the arms of the “Fashion World” but rather understand better its mechanisms (in particular the possibility that perception may play an enormous role in the success of knowledge transfer and collaborations) if we are to succeed in our endeavours.

III. Ideas on potential ‘Success Factors’

In this section I discuss some factors I deem as important for increasing the chances of successful collaborations and better integration of SE research and industry. It is important to note here that the normative discussion does not point solely to researchers, as other groups like industry, policy makers, universities, and even publishing houses, may play an equally important role.

- Practitioners should be able to get access to research and find it insightful/useful
- The (economical) potential must be outlined and balanced out w.r.t. risks/costs
- Efforts should follow an incremental, well-defined process that harmonizes with mainstream practices or the organization’s culture
- Novel approaches should whenever possible, be accompanied by tool-support that integrates easily to de-facto standards/tools

A. Make it accessible and find the ‘red thread’

Starkey and Madan [11] suggest that more efforts should be placed into improving the communication of research results via more comprehensive abstracts at the beginning of articles, or even through ‘professional adaptation’ sections in journals. Also, Open Access may play a very important role for knowledge transfer and adoption in industry.

Previously discussed issues such as lack of training, and cognitive bias (‘cognition and competencies’ as described by Lam et al., [13]) constitute factors hindering successful collaboration. Lam et al., also comments on the issue of ‘careers and incentives’. There is no well-accepted or established role of “science communicator” or “innovation champion” within the landscape of SE as a scientific discipline or a professional field. In the other hand, pressure on Universities to produce work force for industry may hinder the adequate preparation of practitioners who otherwise would have had better chances to understand research results. Lam et al., suggests restructuring of academic institutions to improve knowledge exchange and dissemination, new incentives and roles within academic or innovation arenas could help making knowledge more accessible and thus increase chances of knowledge adoption.

When it comes to taking a more practical stance on research results, I mentioned early that it is challenging to go from Mode 1 to Mode 2 (to ‘find the red thread’). That is perhaps because an insight on both ‘worlds’ is needed in order to do so. Maybe research institutions need to go the extra-mile and investigate more actively on the challenges and current needs in industry. A good example is the study conducted by Sikora et al., [32] who investigated what are the requirement engineering needs in the production of embedded systems.

B. Increase incentives for practitioners

I believe companies would be see investment in knowledge production and innovation as a much less risky endeavor if appropriate economical incentives are in place. An example in Norway is the User-based Innovation arena or BIA², a programme by in the Norwegian Research Council that provides partial funding to the industrial sector for conducting innovation projects in conjunction with academic institutions. Also, arenas of precompetitive research can reduce the risks and costs for all the involved partners and thus seen with better eyes by managers. But that would also mean that the SE research community would need also view themselves less as a ‘service provider’, and more like enablers for process and product innovation. Another way of incentive, also related to our role of researchers as enablers is to present research results and techniques as means of “professionalizing” the field. A good example is to convey means for analyzing empirical data for decision-making in the context of product or project risk assessments. Increasingly large number of customers in software industry are starting to value and seek objective, evidence-based approaches, rather than subjective approaches.

C. Focus on an Incremental and Harmonic Process

While running a study in Volvo Trucks in Hisingen, Sweden, I asked the Team Lead of a software development group who had successfully adopted a form of Model-Driven Development: “What is the main factor of your success?”, and she replied that an incremental process was key in the adaption and dissemination of their approach. This enables control over risks and enables evaluation of the costs and benefits. Beckman et al., [14] state that measuringness is necessary in any collaborative efforts, and certainly an incremental approach makes it easier to measure the benefits of a collaboration. Fowler et al., [33] states that Trialability (the degree to which an innovation may be experimented with on a limited basis) and Observability (the degree to which the results of the innovation are visible to others) are key characteristics of technology diffusion.

In addition, technology/knowledge transfer needs to be in a certain level of harmony with organizational values and daily pracis. In those respects, Fowler et al., [33] also mentions Compatibility (the degree to which an innovation is perceived as consistent with existing values, past experiences and perceived goals of the user) as a determining factor. For instance, in order to avoid rejection on new approaches rooted in CMMI in a company using Agile, researchers should be prepared to address possible forms of criticism or the “water and oil” perception of CMMI vs. Agile prevalent in industry.

D. The importance of tool-support and degree of automation

When deciding whether or not to change a daily routine, I personally evaluate the feasibility of doing so. If there are far

² www.forskningsradet.no/prognett-bia
to many inconveniences and I do not deem it as critical to make the change, then I would most likely decide not to. I think is the same with companies too. Even there are ulterior motives to propose a innovative change in an organization, complexity (the degree to which an innovation is perceived as being difficult to understand and use) [33] determines in large extent the success or failure of an innovation effort. Many new ideas and approaches are sound and feasible to be adopted in industry, but many times entirely alien factors may define whether they are accepted or not. One such issue is the existence of adequate tool support and degree of automation. If all they see is “too much” manual effort, they would less likely to go for it.

The growing popularity of Open Source platforms and frameworks (if compared to 15 years ago when Open Source was mainly limited to niche or specialized communities) in the mainstream software practice is worth noting. Moreover, the notion of a ‘sandbox’ as an experimentation space that contains documentation, examples and on-the-fly development and execution have become popular, in particular within web development technologies. Microsoft have also started using such strategies (for instance for teaching young kids to program). Maybe is worth learning from some successful Open platforms, such as Docker, Taiga or Kubernetes. Previously, I cited Kaindl [16] who commented on the pressure on academic institutions to transfer everything, anything to industry, and on the lack of resources to exploit results into a more mature form. In that sense, I believe a change in the paradigm is needed to re-think on what resources, evaluation guidelines and incentives are needed in order to increase the competitive value of research outcomes.

IV. ILLUSTRATIVE EXAMPLE

In the current section, I present a simplified example within Software Evolution to illustrate some of the points discussed in previous sections. Let us take the topic of refactoring code smells within the context of Agile development.

The interplay between Agile principles and software quality is an interesting problem: For one side, the high-pace of agile development imposes demands on the code base and the developers, potentially increasing the level of complexity and disorder in the code base. On the other side, the ability to rapidly produce new increments to a product may get hampered by an unmanageable code base.

Thus, empirical studies of code smells and refactoring constitute an challenging issue from an industrial perspective because despite the common usage of refactoring and the growing popularity of code-smell based analysis, we still lack of enough empirical evidence on the cost-effectiveness of refactoring. Such empirical evidence is vital to enable efficient and strategic refactoring decisions.

In [34], the authors found that better support (from a tool automation and process perspectives) is needed for learning, planning and testing activities within the agile flow in order to keep the agile responsiveness. From examining the literature in SE, we could for instance suggest several strategies:

- Semi-automatic code inspections
- Use of indicators for identifying problematic modules
- Evolution monitoring of the system

Which all may sound reasonable, but when it comes to implementing it (and observing these effects) in practice, a well-defined incorporation strategy/process is needed. The example I describe below predisposes an agile maintenance scenario such as the one described in [34]. The example of an implementation process + longitudinal study described here would include: (1) periodical maintainability assessments and (2) periodical refactorings in the development flow.

A. Maintainability assessments

Having slightly longer retrospectives in order to incorporate maintainability assessments could be one way to introduce a more formal measurement and evaluation activity in the development cycle. Normally, retrospectives are meetings held by a project team at the end of an iteration to discuss what was successful about the time period covered by that retrospective, what could be improved, and how to incorporate the successes and improvements in future iterations or projects. During the maintainability assessment, the teams and the architect(s) could identify the difficulties faced during the iteration as well as the problematic modules, and analyze possible reasons for these difficulties. Data drawn from analysis via tool support could provide input for the discussion. This could be for example a code smell detector, or a tool that can analyze the repository and extract which modules were most connected to defects during the last iteration.

B. Periodical refactoring

The abovementioned assessment should produce a maintenance backlog, which will have the function to depict general goals for restructuring and improving the maintainability of the system. The main requirement is that maintenance backlogs need to be enough specific in order to be brake-down into sprint backlogs. A maintenance backlog should resemble a product backlog but focusing on maintainability improvements instead of new functionalities in the product.

In order to incorporate periodical refactoring and constant improvement of design, an additional iteration at the end of each iteration (i.e., mini-iteration) can be implemented, where the maintenance backlog should be used as input for planning and executing the restructuring/refactoring tasks. The mini-iterations should be always shorter than normal iterations. Each of the rules used in normal iterations will apply (e.g., deciding upon the backlog items, planning poker and distribution of tasks amongst the team members). Unit testing, integration testing and system testing should be planned as integral part of the mini-iteration as in a normal iteration. The process is described diagrammatically in Figure refmini-iteration.

However, such a process may need an adequate framework to guide the maintainability evaluations and the restructuring/refactoring during the mini-iterations. The process will
basically rely on two aspects: (1) Tool support and (2) Refactoring knowledge base.

C. Tool support

Semi-automated code inspection is deemed as a feasible approach to control costs and increase the level of automation for improving code quality. Quantitative data (design attributes) such as measures of code and code smells could guide the exploration of extensive code thus may facilitate the identification of problematic areas/modules in a reasonable time frame.

D. Refactoring knowledge base

The result of the previously described analysis, together with what can be called refactoring knowledge base should be used as input for producing the maintenance backlog. A knowledge base should comprise a list of the code smells and design principle violations (and their respective rules, so they can be detected in the source code), which are considered relevant to the company’s context.

The knowledge base should contain also the corresponding refactoring and restructuring strategies for each of the code smells and anti-patterns. Each of the code smells or design principle violations should be assigned a level of Criticality (e.g., high or low) depending of the observed negative consequences these may have in the system (as deemed by the architects or developers). Each of the refactoring strategies should be labeled according to their Cost (time, effort, level of automation etc) and Risk (e.g., high, medium or low risk). This information could be useful for deciding which refactorings to do in case modules with this attribute are identified.

This knowledge base should be stored in a common repository which all the members from the teams and the architects will have access. A simple format like a Wiki could suffice for this purpose, and it is recommended to pursue simplicity in order to make the information more accessible to the members with different levels of experience in the team.

The idea of having mini-iterations is that in the same way as sprint-brundown charts, it may be possible to observe the evolution of the problematic artifacts (e.g., that after refactoring class A was less prone to defects) and given that they are registered in a sprint-backlog, it would be possible (in theory) to keep track of any refactoring/restructuring endeavors.

This is of course a rather simplified idea for implementing a new methodology (e.g., code-smell based quality analysis) in industry, and at the same time studying the actual costs and benefits of refactoring. However, it illustrates some of the points previously discussed: for example, a) it integrates to the organizational culture and working routine (in this context, they were using a form of Agile development), b) it provides a tool alternative to automate some of the activities, and c) it follows an iterative approach that can be periodically assessed, which represents a lower risk for companies, and they can themselves decide if this is giving results or not. Last but not least, by empowering the companies to be in charge of the sense-making process, they could generate new knowledge (and thus, increase their level of professionalism in the field) and feel they are benefiting from the collaboration.

V. Final Remarks

The example provided illustrates several underlying strategies/recommendations for increasing the chances of successful collaborations between research and industry.

A general conclusion that can be made is that more information is needed in order to understand better the system of values in industry; to understand what is the language they speak. Perception of important could be influenced in some cases just by using the right term (as the previously mentioned example where terms as ‘big data’, ‘data scientist’ or ‘data engineer’ have gained traction). Finding the “sweet spots” of research from an industrial is also a challenge. As mentioned previously, ‘selling’ software engineering research as a way of ‘professionalizing’ the field can be one way. In particular for quality evaluation purposes, it has proven to be vital for accreditation. A good example is the Software Improvement Group who has successfully exploited the notion of ‘Linked Scientists’ for providing services such as software due diligence evaluations.

The Linked Scientists, according to Casey et al., [8] constitute part of an emerging form of hybrid work roles and/or careers that support knowledge flows across organizational
boundaries. I relate this hybrid notion to the notion of “collaboration champion” or liaison as a third player in the collaboration picture. Instead of having researchers on one side and practitioners on the other, why not have a third role? This role is in synergy with a growing tendency (such as the example with SIG), where many R&D intensive firms have started to move towards a network model and use network of scientists across organizational boundaries (via their links) to augment their internal knowledge base [13].

Rynes et al. [12] underlines the role of “collaboration champions” when stating that effort towards better intergation should be concerted at all levels, but especially among gatekeepers at organizational levels, in order to actualize the potential of academic-practitioner relationships, and to create a truly improved science of organizations. This “collaboration champions” may help establishing long-term relations and continuous interaction between academia and industry. The first aspect is crucial to the generation of trust, being this last aspect a precondition for successful learning and innovation. The second aspect would lead to higher chances of success, as communication channels would be kept open headint to better understanding and thrust.

Finally, I would like to end this paper by challenging the notion of ‘knowledge transfer’. Many understand ‘knowledge transfer’ as a unidirectional flow of knowledge from academia to industry, knowledge and empirical information from industrial contexts are cornerstones in our field. Thus, we need mechanisms and processes that can foster the generation of knowledge that would be of real and perceived mutual benefit, to reinforce the linkage between academia and industry, ultimately to achieve progress in our field.

REFERENCES