

# On the “Selling” of Academic Research to Industry

Scott Tilley

College of Business  
Florida Institute of Technology  
stilley@fit.edu

Shihong Huang

Computer Science & Engineering  
Florida Atlantic University  
shihong@cse.fau.edu

## ABSTRACT

Technology transfer does not happen automatically. One of the primary reasons that academic research fails to be adopted by industrial practitioners is that the benefits of doing so are not obvious. In contrast, the perceived drawbacks of even attempting to do so are easy to enumerate. This paper argues that academics need to be a better job of “selling” their research to industry in order to facilitate the transfer of their results. One facet of this self-marketing effort is to pro-actively address some of the likely objections that someone from industry might have, such as convincing them that academic research is relevant to their work, that successful prior collaborations are applicable to their situation, and that they will definitely see a short-term return on their investment.

## Categories and Subject Descriptors

K.1 [The Computer Industry]: *markets*. K.6.0 [Management of Computing and Information Systems]: *Miscellaneous – economics*. D.2.9 [Management]: *productivity*.

## General Terms

Management, Economics, Experimentation, Human Factors

## Keywords

adoption, technology transfer, evidence-based software engineering, marketing, research, practice.

## 1. INTRODUCTION

Is it wrong to refer to technology transfer from academic research to industry as “selling”? Not if the goal is the successful adoption of the approach/tool/technique by the broader community. After all, what is technology transfer other than the “buying” of the results (by industry in this case) from the “seller” (academia), using the currency of collaboration, trust, and the potential for further benefits as the means of facilitating the transfer?

Unfortunately, the very mention of “selling” for many academics seems to imply something unclean, something that is for others to do. The truth is quite the opposite: “selling” one’s research is just another aspect of self-marketing that is a positive character trait of all successful professors. In a software engineering sense, selling can

be interpreted as understanding one’s customers and their requirements—something students are taught all the time.

One facet of this self-marketing effort is to pro-actively address some of the likely objections that someone from industry might have when it comes to adopting results from academia [4]. The next three sections of this paper briefly discuss specific barriers that an academic should be prepared to answer right at the start of negotiations: (1) convincing industry that the academic research is relevant to their work; (2) showing industry how successful prior collaborations are applicable to their current situation; and (3) demonstrating how industry will definitely see a short-term return on their investment. By preparing answers to these vexing questions, academics can make the technology transfer process somewhat easier—something that is (or should be) a goal of applied software engineering research

## 2. WE’RE NOT INTERESTED IN THAT

Perhaps the first barrier to adoption is communication. It’s no secret that academic researchers and industrial practitioners speak different languages. Their vocabulary is different, the acronyms they use are different, and their slang is different. Consequently, when an academic approaches someone from industry and attempts to describe their work using academic terminology, the person from industry may have great difficulty understanding its relevance.

For example, during a recent conversation with a representative from a major cell phone manufacturer, the word “redocumentation” was used to describe an area of work that the researcher felt would be directly applicable to the industry representative’s situation. However, the word “redocumentation” proved to be a nearly insurmountable stumbling block to further collaboration. For the industry person, “redocumentation” had no clear meaning, and therefore no clear mapping to what their current needs were. For the academic, the applicability of redocumentation to their problem (which was based on the need to explore legacy code for potential reuse) was obvious. It was only after the term was explained, and related technology was discussed (e.g., reverse engineering, reuse, components), that a common understanding of how redocumentation could be applied to the problem of reuse was arrived at.

The lesson here is that industry *was* interested in redocumentation, just not when it was presented as redocumentation per se. It is not the job of the industrial partner to glean this information from the academic. Rather, it is the job of the academic to sell their research to industry by using vocabulary, examples, and analogies that are understandable by all parties involved.

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. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

TT’06, May 22, 2006, Shanghai, China.

Copyright 2006 ACM 1-59593-085-X/06/0005...\$5.00.

### 3. WE'RE NOT LIKE THEM

Anyone who has studied software engineering process knows that one of the reasons a standardized model has not been universally adopted is that many organizations view themselves as unique. In manufacturing, there are variances in the process due to differences in the product, but overall there are a great many commonalities. In software, this doesn't seem to be the case – or at least that is the perception.

If an academic has a proven track record of successful interaction with industrial collaborators, this should be a selling point when presenting their credentials to a new potential partner. However, if the two industries are not obviously the same, the new partner may not see the relevance of the prior work to their current problem. Again, it is the job of the academic to make this connection; it is not the job of the industrial representative to draw their own conclusions based on their (possibly incomplete or misguided) interpretation of the other industry.

Continuing the redocumentation example from the previous section, one would think that multiple successful previous projects in the automotive sector involving embedded control systems would be relevant experience to another type of embedded control system: cell phones. However, without inside knowledge of the nature of modern automotive control systems, the similarities between two such seemingly different applications may not be obvious. The academic must sell their research in such a manner that shows that, yes, the new industry *is* like the other one (at least to such a degree that a significant part of the results are applicable).

### 4. HOW WILL IT HELP US RIGHT NOW?

In engineering terms, there is often an impedance mismatch between the goals of academic research and industry practice. Academia is supposed to be one of the few places that can afford the luxury of reflective inquiry. This tends to support a longer-term view of projects: initial investigation, development, field trials, and so on. This is proper; academic research should have lofty goals with the potential for significant impact down the road.

On the other hand, industry by and large has a very short-term view of the world. This is necessitated in part by a very practical matter: they need to stay in business. It is true that some large corporations still fund R&D organizations active in basic science and exploratory investigation. However, there is a distinct trend towards applying research results directly towards product development. Whether or not this is a desirable trend is debatable; the fact that it is happening is not.

A direct implication of this somewhat myopic view of things is that most industries look towards academic research as a possible source of immediate solutions to pressing problems. This means academic researchers must be able to convincingly demonstrate that adoption of their work will provide measurable and objective return on investment. Of the three issues discussed in this paper, this is perhaps the one that many academics find the hardest to do—but it need not be this way.

One way of providing such evidence is through empirical studies. There is already a growing realization in software engineering research of the need for evidence-based approaches [1]. Since this method of supporting technology transfer has been successfully used in other fields, it certainly deserves further investigation [2].

Short-term gains need not be in conflict with long-term research either. If one is able to show how the results of past work can help *now*, the argument for further development of the research for even larger gains in the future is made even stronger. From a software engineering point of view, this can be seen as an incremental and iterative approach to applied research. And since even academia advocates incremental and iterative approaches to software engineering, it behooves us to at least try to adopt it ourselves in our own work.

### 5. SUMMARY

This paper outlined three likely objections that someone from industry might have when it comes to adopting research results from academia. There are obviously many more issues that need to be addressed in order to make technology transfer a smoother and more successful endeavor. Nevertheless, for software engineers, the three issues described here offer a starting point.

Academia is not industry and industry is not academia. Each has a part to play in the continuous spiral of information exchange between the two [3]. However, for the exchange to be fruitful there needs to be a common understanding of the roles and responsibilities of each participant. Sabbaticals are historically one of the mechanisms that academics have used to become immersed in the industrial world. The use of reverse sabbaticals (“resident affiliates”) is something that has not been used as much; perhaps it should.

Too often in the past academics view industry simply as funding agencies. Similarly, industry views academia as a type of outsourced R&D center, ready to be exploited for immediate benefit at little cost. As with most things in life, the answer lies somewhere in the middle.

### REFERENCES

- [1] Budgen, D.; Hoffnagle, G.; Müller, M.; Robert, F.; Sellami, A.; and Tilley, S. “Empirical Software Engineering: A Roadmap.” *Proceedings of the 10<sup>th</sup> International Conference on Software Technology and Engineering Practice (STEP 2002: Oct. 6-8, 2002; Montréal, Canada)*, pp. 180-184. Los Alamitos, CA: IEEE Computer Society Press, 2003.
- [2] Huang, Shihong and Tilley, Scott. “On the Challenges in Fostering Adoption via Empirical Studies.” *Proceedings of the 4<sup>th</sup> International Workshop on Adoption-Centric Software Engineering (ACSE 2004: May 25, 2004; Edinburgh, Scotland, UK)*. May 2004.
- [3] Huang, Shihong; Tilley, Scott; and Zhou, Zhiying. “On the Yin and Yang of Academic Research and Industrial Practice.” *Proceedings of the 3<sup>rd</sup> International Workshop on Adoption-Centric Software Engineering (ACSE 2003: May 9, 2003; Portland, OR)*, pp. 19-22. Published as CMU/SEI-2003-SR-004. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, June 2003.
- [4] Tilley, S.; Huang, S.; and Payne, T. “On the Challenges of Adopting ROTS Software.” *Proceedings of the 3<sup>rd</sup> International Workshop on Adoption-Centric Software Engineering (ACSE 2003: May 9, 2003; Portland, OR)*, pp. 3-6. Published as CMU/SEI-2003-SR-004. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, June 2003.