

Technology Transfer and the Tech Broker

Warren Harrison

Technology transfer is the process of transferring an idea from its originator to someone who can use it. Typically, this implies the transfer of research developed in academia to industry.

While there's no dearth of technologies that academic researchers have available and are anxious to transfer, precious little technology transfer involving academia takes place, at least in the software development community. This is of particular concern given the number of journals and conferences, which seems to increase exponentially from year to year.



I performed a Google Scholar search (http://scholar.google.com/advanced_scholar_search) for articles in the "Engineering, Computer Science, and Mathematics" subject areas that were published since January 2000 and have the phrase "software engineering" in the title. Surprisingly (at least to me), Google Scholar returned 73,300 hits. Even if this is an order of magnitude off, we're still talking about literally thousands of pieces of work, yet relatively few examples of successful software development technology transfer exist. The fact that researchers invariably counter this assertion by trotting out the same half-dozen or so examples only confirms this in my opinion.

Some researchers might tell you that their goal is the search for knowledge and that they care little if real software developers use their work. But I can't help but believe that most soft-

ware engineering researchers' ultimate goal is to transfer their results into practice. If this is true, why don't we see more tech transfer in our field?

Selling technology

Selling a new technology isn't all that different from selling any product to a potential customer. You must have

- a product (the technology) that meets the needs of the customer (the production organization),
- a way for the customer to know about the product, and
- a mechanism to reduce the barriers involved in transferring the technology.

I like to call this the Product/Needs/Awareness/Barriers model. The PNAB model can be useful in explaining why the software development community tends to have such a dismal technology transfer record.

A vast number of technology innovations are cooked up by software engineering researchers every year. Each one solves some problem, and (if we're lucky) some small percentage of production organizations know about it. Unfortunately, these innovations don't solve any problems the set of organizations that know about them are facing. Also, some organizations might actually need help with those problems but don't know about the innovation.

Some observers think that researchers should learn about their industrial partners' needs and adapt their work to meet those needs. While

DEPARTMENT EDITORS

On Architecture: Grady Booch,
grady@booch.com

Bookshelf: Warren Keuffel,
wkeuffel@computer.org

Design: Rebecca Wirfs-Brock,
rebecca@wirfs-brock.com

Loyal Opposition: Robert Glass,
rglass@indiana.edu

Open Source: Christof Ebert,
christof.ebert@alcatel.com

Quality Time: Nancy Eickelmann,
nancy.eickelmann@motorola.com,
and Jane Hayes, hayes@cs.uky.edu

Requirements: Neil Maiden,
n.a.m.maiden@city.ac.uk

Tools of the Trade: Diomidis Spinellis,
dds@aueb.gr

STAFF

Senior Lead Editor
Dale C. Strok
dstrok@computer.org

Group Managing Editor
Crystal Shif

Senior Editors

Shani Murray, Dennis Taylor, Linda World
Assistant Editor Editorial Assistant
Brooke Miner Molly Mraz

Magazine Assistant
Hilda Carman, software@computer.org

Art Director
Toni Van Buskirk

Technical Illustrator
Alex Torres

Production Artist
Carmen Flores-Garvey

Executive Director
David Hennage

Publisher
Angela Burgess
aburgess@computer.org

Associate Publisher
Dick Price

Membership/Circulation Marketing Manager
Georgann Carter

Business Development Manager
Sandra Brown

Senior Production Coordinator
Marian Anderson

CONTRIBUTING EDITORS

Robert Glass, Keri Schreiner, Joan Taylor

Editorial: All submissions are subject to editing for clarity, style, and space. Unless otherwise stated, bylined articles and departments, as well as product and service descriptions, reflect the author's or firm's opinion. Inclusion in *IEEE Software* does not necessarily constitute endorsement by the IEEE or the IEEE Computer Society.

To Submit: Access the IEEE Computer Society's Web-based system, Manuscript Central, at <http://cs-ieee.manuscriptcentral.com/index.html>. Be sure to select the right manuscript type when submitting. Articles must be original and not exceed 5,400 words including figures and tables, which count for 200 words each.

there are some examples of this happening successfully, they're rare. From the researcher's perspective, this might not be such a good idea anyway. Research (well, good research) takes a long time, and adapting to the needs of a production organization that measures progress by the quarter can certainly derail anyone's research program.

It's a fact of life that researchers are engineering-driven rather than market-driven. Consequently, when faced with a mismatch between customer needs and a technology, most researchers will start looking for a new customer rather than adapt their work to meet their existing customers' needs (in spite of what they might say). While production organizations might feel frustrated by this, they often fall as short in understanding researchers' motivations and needs as researchers do in understanding theirs.

Even when there's a match between technology and adopter, significant barriers to transfer exist. These range from simply awkward packaging to contractual and intellectual-property restrictions. Neither researchers nor production organizations have much experience (or patience) dealing with technology adoption's administrative barriers, which make it all the more a hurdle.

What the software community needs is an infrastructure that matches researchers with adopters at a specific point in time and helps them overcome the administrative barriers.

The technology broker

At one time, my wife had her heart set on a specific kind of sports car. She had decided on the year, make, model, color, and equipment she wanted. Naturally, no local car dealers had any vehicles on the lot that matched her requirements. However, after a single trip to an auto broker to submit her specifications, she had her dream car in a matter of weeks. All it took was someone who made it his business to know what sellers had to offer and what buyers wanted.

Similarly, having software engineering technology brokers would be a useful thing. Just like the broker who found my wife her dream Corvette, technology bro-

kers would make it their business to know who was doing what software research and which production organizations were looking for what technologies.

Brokers must understand both what the adopter needs as well as what the provider has so that they can arrange the best matches possible. This implies that brokers must have at least enough technical sophistication to understand the technology they're brokering.

To provide the best possible match, brokers must proactively solicit requestors and providers and remain scrupulously independent. One of the quickest ways to ruin such an arrangement is to simply "push" whatever they have in their inventory, regardless of match. This means that brokers must anticipate adopters' needs (perhaps even before they do) and keep track of new research programs at a wide range of research organizations. They couldn't favor their own work—in fact, in a perfect world, brokers wouldn't even belong to a research organization or be researchers themselves.

Brokers must also have a vision to be able to help the parties chart a course. But beyond this, brokers must have the tenacity to see a relationship through to culmination, as well as the people skills necessary to keep everyone on task. Ultimately, brokers will be successful only if a technology gets transferred and both parties are happy with the outcome. So, when one party or the other derails the relationship, whether for good reason or bad, brokers have as much (perhaps more) to lose as the parties.

To make these relationships work, brokers must understand both sides' motivations and translate these for the participants. The motivations will likely differ from situation to situation. For instance, a junior faculty member will be desperate for both research funding and publication, because her tenure case will be measured against these. A technology transfer effort that offers neither might be counterproductive to her career goals. On the other hand, a senior faculty member might be happy to trade technology for data or eventual consulting opportunities, because

the six-year crush for research productivity imposed by the academic tenure system is far behind her. Likewise, some production organizations might be looking for long-term systemic improvements, whereas others might be looking for quick fixes to shore up next quarter's bottom line.

Brokers must also understand the administrative barriers that both parties face. Just as an auto broker can help arrange financing to close a deal, the technology broker's job isn't done until both parties have closed the deal. This might mean smoothing the contractual details, working out intellectual-property agreements, and even providing bridge funding to get a technology transfer effort off the ground.

Finally, brokers must get value from brokering the relationship between provider and adopter. They must somehow be compensated for their efforts; otherwise, they have little motivation to broker someone else's research.

A successful tech-brokering organization

My concept of a tech broker has been greatly influenced by the US National Science Foundation's Industry-University Cooperative Research Center program and in particular the NSF Software Engineering Research Center (www.serc.net). Founded in 1986, the SERC is a consortium of industrial organizations and universities that facilitates the brokering of technology transfer from academia to industry. Although the NSF supplied initial seed money, the SERC has long since become self-sufficient.

The SERC consists of

- a number of industrial affiliates, including large, well-known companies such as Motorola, Northrup-Grumman, and Raytheon and smaller companies such as Baker Hill, Ontario Systems, and Arxan Technologies;
- governmental organizations such as NASA, the Naval Surface Warfare Center, and the National Energy Technology Laboratory; and
- a dozen or so academic affiliates, including my own university.

The industrial affiliates each chip in a modest annual membership fee to help pay the administrative costs of running the SERC. In turn, the SERC helps broker its academic affiliates' research among its industrial affiliates.

But the SERC goes beyond simply matching technology producers to consumers. Each industrial and academic affiliate also agrees to participate in a standardized contractual agreement including intellectual-property and publication rights, indirect costs, and payment schedules. So, once the SERC finds a match between a researcher and a company, a funded technology transfer effort can begin in a matter of weeks, rather than the usual six- to nine-month (or more) negotiation phase following an industrial production organization's decision to do business with a university.


Both researchers and companies have found this valuable enough to keep coming back for more. The success of a 20-year-old self-sustaining technology broker speaks for itself.

Growing technology transfer

Not surprisingly, the issue of technology transfer in software engineering continues to gain interest. Last May at the 28th International Conference on Software Engineering in Shanghai, Roel Wieringa and I helped organize a workshop on Technology Transfer in Software Engineering (www.cs.pdx.edu/~warren/wottse), attended by more than 30 researchers and practitioners. Indeed, the mission of *IEEE Software* itself is to facilitate technology transfer from innovators to production organizations.

Coupled with researchers' growing interest in transferring technology, we must do more to let production organizations know what's available. The growth of technology brokers would help get more of this work into the practitioner community, where it belongs.

What do you think?

Is your organization actively involved in technology transfer as either a provider or a consumer? Do you have any success (or horror) stories you'd like to share? Please write me at warren.harrison@computer.org. 

EDITOR IN CHIEF

Warren Harrison

10662 Los Vaqueros Circle
Los Alamitos, CA 90720-1314
warren.harrison@computer.org

EDITOR IN CHIEF EMERITUS:
Steve McConnell, Construx Software
stevemcc@construx.com

ASSOCIATE EDITORS IN CHIEF

Education and Training: Don Bagert, Rose-Hulman Inst. of Technology; don.bagert@rose-hulman.edu
Design: Philippe Kruchten, University of British Columbia; kruchten@ieee.org
Requirements: Roel Wieringa, University of Twente; roelw@cs.utwente.nl
Quality: Stan Rifkin, Master Systems; sr@master-systems.com
Experience Reports: Wolfgang Strigel, QA Labs; strigel@qalabs.com

EDITORIAL BOARD

Grady Booch, IBM
Christof Ebert, Alcatel
Nancy Eickelmann, Motorola Labs
Jane Hayes, University of Kentucky
Warren Keuffel, independent consultant
Neil Maiden, City University, London
Diomidis Spinellis, Athens Univ. of Economics and Business
Richard H. Thayer, Calif. State Univ. Sacramento
Rebecca Wirfs-Brock, Wirfs-Brock Associates

ADVISORY BOARD

Stephen Mellor, Mentor Graphics (chair)
Maarten Boasson, Quaerendo Invenietis
J. David Blaine, Blaine Consultancy
Robert Cochran, Catalyst Software
Annie Kuntzmann-Combelles, Q-Labs
David Dorenbos, Motorola Labs
Kaoru Hayashi, SRA
Simon Helsen, SAP
Juliana Herbert, ESICenter UNISINOS
Dehua Ju, ASTI Shanghai
Gargi Keeni, Tata Consultancy Services
Karen Mackey, Cisco Systems
Tomoo Matsubara, Matsubara Consulting
Dorothy McKinney, Lockheed Martin Space Systems
Bret Michael, Naval Postgraduate School
Susan Mickel, Lockheed Martin
Ann Miller, University of Missouri, Rolla
Deependra Moitra, Infosys Technologies, India
Melissa Murphy, Sandia National Laboratories
Suzanne Robertson, Atlantic Systems Guild
Grant Rule, Software Measurement Services
Girish Seshagiri, Advanced Information Services
Martyn Thomas, Praxis
Rob Thomsett, The Thomsett Company
Laurence Tratt, King's College London
Jeffrey Voas, SAIC
John Vu, The Boeing Company
Simon Wright, SymTech

CS PUBLICATIONS BOARD

Jon G. Rokne (chair), Michael R. Blaha, Mark Christensen, Frank E. Ferrante, Roger U. Fujii, Phillip Laplante, Sorel Reisman, Bill N. Schilit, Linda Shafer, Steven L. Tanimoto, Wenping Wang

MAGAZINE OPERATIONS COMMITTEE

Bill N. Schilit (chair), Jean Bacon, Pradip Bose, Arnold (Jay) Bragg, Doris L. Carver, Kwang-ting (Tim) Cheng, Norman Chonacky, George Cybenko, John C. Dill, Robert E. Filman, David Alan Grier, Warren Harrison, James Hender, Sethuraman (Panch) Panchanathan, Roy Want