



DOI:10.1145/1461928.1461937

Ashish Arora, Matej Drev, and Chris Forman

# Economic and Business Dimensions

## The Extent of Globalization of Software Innovation

*Will the software development laboratories follow the production mills?*

**H**AS INNOVATIVE ACTIVITY in software become global? The question attracts interest because software has become a global business. For example, exports of business services and computer and information services grew at an average annual rate of 27% in India (1995–2003) and at a rate of 46% in Ireland (1995–2004), with similar rapid growth in Brazil, China, and Israel.<sup>3</sup> Yet, the question remains open. While software production takes place in many countries, innovative software is not created everywhere.

Moreover, there is increasing evidence of growing software development operations in countries such as India. For example, the Microsoft India Development Center (located in Hyderabad) has grown from two products and 20 employees in 1998 to 70 products and over 1,500 employees today. SAP Labs India is now that company's second-largest Research and Development and Global Services and Support center, with 25% of its employees engaged in research and

new product development. IBM's India Software Labs similarly have large operations in Bangalore, Gurgaon, Pune, Hyderabad, and Mumbai. Yet, despite these anecdotes the question is not easy to answer at a general level. In many cases innovation cannot be defined. For that matter, sometimes software cannot be defined.

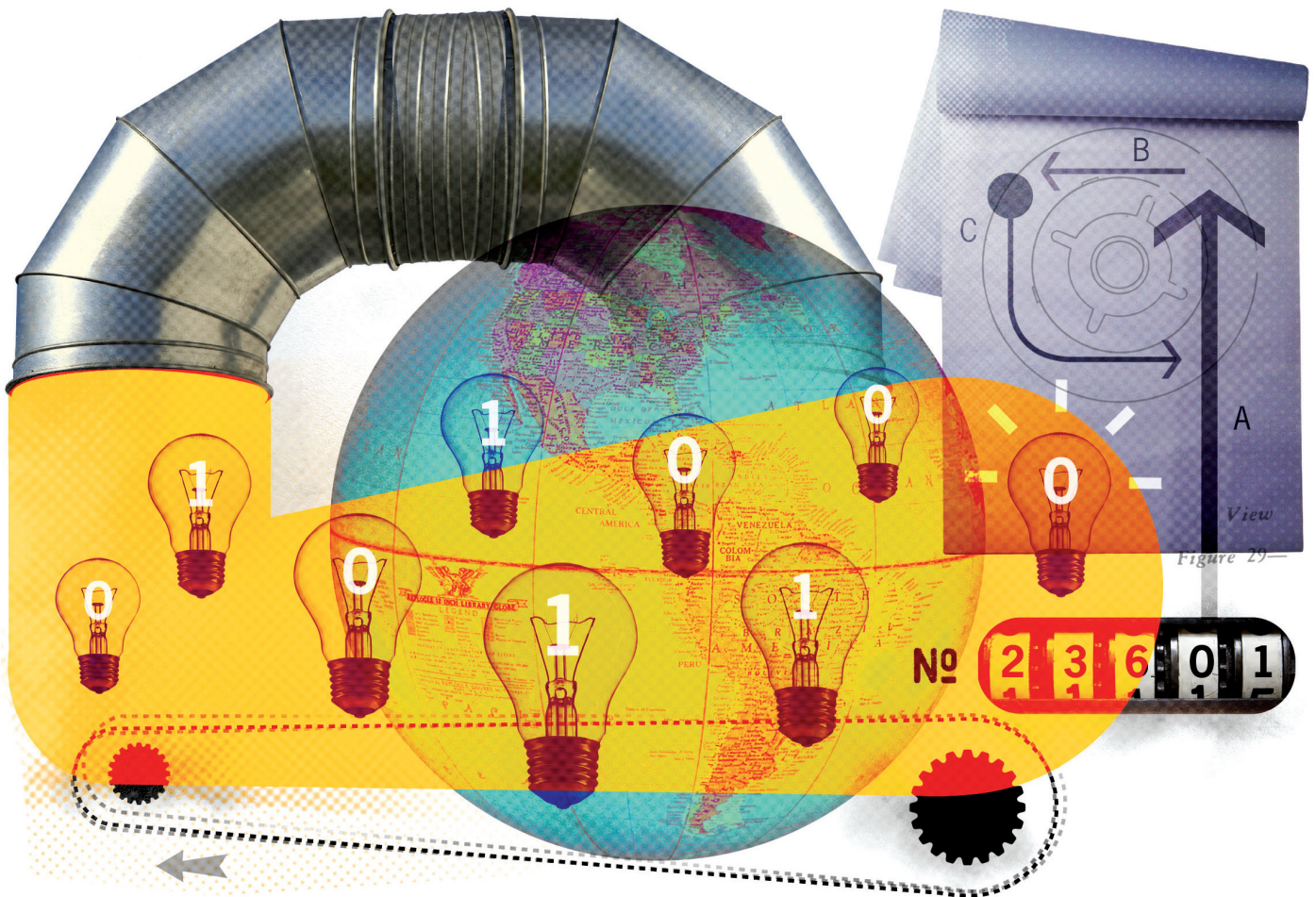
In this column we summarize and extend an investigation into software innovation.<sup>2</sup> We look at software developed to be sold as a standalone product, software developed to be sold as a service (such as software available for use on salesforce.com), and software developed by a user firm. We exclude software for semiconductor chips and also software written for application-specific computers such as some CAD software.

One measure of innovation appears in patent data. To be sure, not all inventions are patented and not all patents are important. However, information in the text of the patents and their technical classes can help identify patents related to software. Moreover, patents

provide a consistent source of data about innovative activity and can provide some useful insights.

In 2007, 12,692 U.S. software patents were issued to inventors in the U.S.—a larger number of patents than all other areas of the world combined (6,397). That is striking because growth in software patenting between 1988 and 2007 in the U.S. was comparable to that of the rest of the world: patenting by U.S. inventors grew at an average annual rate of 32.6%, compared to 34% in the rest of the world. Software innovation in the so-called underdog countries has been growing, but not faster than in the U.S. (see the figure here).

We also interviewed the Indian software research and development labs of several large multinational firms. We found some evidence of increasing autonomy among software development centers. In particular, we found that most organizations started by giving their Indian software labs assigned development or testing work. Yet, it rarely stopped with those assignments.



Increasingly, centers have been given control of key components of software applications and in some cases have been responsible for design and development of entire products.

This was not so for all software. Indian software development centers were most likely to have design control in software projects for which there was a large local market. For example, we found significant autonomy for software projects that involved software development and testing tools and for mobile applications. The Indian market for both of these products is large.

A key determinant of the location of development activities in software is the location of the user. This is particularly true with business software, which is often bundled with a set of business rules and assumptions about business processes.

Growth in emerging economies translates into more lead users, and, thus, more local innovation. Already we have seen this in areas such as software tools, mobile technologies, and

**In many cases innovation cannot be defined. For that matter, sometimes software cannot be defined.**

security software, but there is evidence in other areas as well. For example, i-flex, which began as an offshoot of Citibank developing products for banks in India, has leveraged its success with Indian banks to banks in other emerging markets, and in banks in developed economies as well. In other words, as the distribution of users shifts, so does the locus of software innovation.

This should come as no surprise. Historically, some of the most significant examples of business software—such as IBM's SABRE airline reservation and SAP's ERP software—have arisen through close collaboration between software firms and their users. Since knowledge of business needs is frequently not well codified, such collaboration will often be most effective when performed by firms that are in close proximity to one another.

We see the same pattern today. Conditions for development of innovative new software products (and software firms) are propitious when they occur in proximity to potential lead users. A good example is Israel's longstanding strength in security software, which arose in part due to the advanced needs of the Israeli defense forces.

Our patent data also illustrated the importance of proximity to users. Most software innovation occurring outside the U.S. is found in U.S. multinational firms. One can see this by looking at the country of assignee for patents in-

vented outside the U.S. Among the underdog countries (Brazil, China, India, Israel, and Ireland) most commonly cited as having robust software industries, the fraction of patents assigned to U.S. firms has generally been increasing over time, ranging from 15.4% in 1990 to a high of 64.6% in 2002. This suggests that multinationals—U.S. software firms—can serve as a partial, though highly imperfect, conduit for the needs of lead users.

However, while U.S. software firms appear to be a potential conduit for user needs, they appear to be moving their innovative activity offshore much more slowly than they are moving some programming and maintenance activity. The percentage of U.S.-assigned patents that were invented in the U.S. fell from 94.3% in 1996 to 91.4% in 2007. This decrease in the share of U.S.-assigned patents invented in the U.S. is due in large part to the increase in offshore activity in the underdogs: the percent of U.S.-assigned patents invented in the underdogs rose from 1% in 1996 to 2.6% in 2005.

It will be a while before countries such as India become significant sources of software innovation. There simply are not enough highly talented computer scientists and software engineers in these countries. Moreover, the state of the science and technology infrastructure is weak. The firms we interviewed told us that supplies of fresh engineering graduates were plentiful in India, but

## Conditions for development of innovative new software products (and software firms) are propitious when they occur in proximity to potential lead users.

it was much more difficult to find developers with project management experience and more difficult still to find those with critical business knowledge.

As of 2004 the number of engineering graduates in India and the U.S. were approximately equal, and the number of Indian engineering graduates is growing much faster than that in the U.S.—from 42,000 in 1992 to 128,000 in 2003.<sup>1</sup> Higher education growth in China has been similarly rapid. Yet, do not be misled by these numbers. Despite these improvements in educational infrastructure, R&D as a percent of GDP is only 1.44% in China compared to 2.68% in the U.S. (both 2004 data) and 0.85% in India (2000 data).

What will the future bring? First, it

does appear software product development and testing activities have become increasingly global. Firms continue to experiment with new methods of managing global software development, and such methods will likely increase the share of software work that can be modularized and produced away from the point of product design and architecture.

Second, these trends raise a big question. Entry- and mid-level programming jobs have frequently provided U.S. IT workers with the skills needed to perform more complicated activities such as product design and strategy. In other words, training by U.S. firms has traditionally bestowed an uncompensated benefit to entry-level workers by providing them with certain types of general training and skills, which are very valuable to the workers later in the careers. Indeed, our interviews suggest that lack of these skills has been a significant barrier to innovation in countries outside the U.S.

Many of these entry-level jobs are now going overseas. A declining demand today for entry-level programming jobs in the U.S. and increasing demand elsewhere could make it more difficult for U.S. workers—and relatively easier for those from other countries—to perform complex software design activity in the future. The implications of this would be a more globally dispersed pattern of innovation in software than we see today. ■

### References

1. Arora, A. and Bagde, S. The Indian software industry: The human capital story. Working Paper, Heinz School of Public Policy and Management, Carnegie Mellon University, 2006.
2. Arora, A., Forman, C., and Yoon, J.W. Software. In J.T. Macher and D.C. Mowery, Eds., *Innovation in Global Industries: U.S. Firms Competing in a New World*. National Academies Press, Washington, D.C., 2008, 53–99.
3. Organization for Economic Cooperation and Development. *Information Technology Outlook 2006*. OECD Publications, Paris, 2006.

**Ashish Arora** (ashish@andrew.cmu.edu) is H.J. Heinz Professor in the Heinz School of Public Policy and Management at Carnegie Mellon University, Pittsburgh, PA.

**Matej Drev** (mdrev@andrew.cmu.edu) is a Ph.D. student in the Heinz School of Public Policy and Management at Carnegie Mellon University, Pittsburgh, PA.

**Chris Forman** (chris.forman@mgt.gatech.edu) is the Robert and Stevie Schmidt assistant professor of IT management in the College of Management at Georgia Institute of Technology, Atlanta, GA.

The authors gratefully acknowledge funding from the Software Industry Center at Carnegie Mellon University. Chris Forman acknowledges funding from an Industry Studies Fellowship from the Alfred P. Sloan Foundation.

U.S. software patents invented in the U.S. and other countries.

