

Is the Software Industry's Productivity Declining?

Robert Groth

Software development has seen rapid evolution over the past few decades: software projects are increasing in size and complexity, software companies are merging at a record pace (look at EMC's recent acquisitions of Legato, Documentum, and VMware or the 17 software acquisitions IBM has made since 2001), and open source code is proliferating. Not surprisingly, productivity improvement is a key issue for development organizations.

What is surprising is an article from the 12 January 2004 edition of *Business Week* entitled "Industry Outlook 2004," which declares that over the past five years, software productivity has decreased by 0.9 percent a year in the US. The article discusses productivity "champs" and "slow pokes" based on annual growth rates during that time, and the software industry was at the bottom of the list (see Figure 1). This data begs the following questions:

- Is the software industry really becoming less productive?
- If so, what's being done to improve it?

Explaining the data

To answer these questions, we must first explain the numbers.

"This period was unusual," says James Glen, senior economist at Economy.com. "The productivity measure that we use is created based on both employment numbers and a measure of real output." (Real output is like real gross domestic product by industry—a

measure of value-added output adjusted for price changes.)

Glen says that although real dollar output per worker declined over this five-year period, employment rates rose rapidly for a while and didn't drop as fast as output during the economic downturn. Additionally, software prices actually rose. "Combine it all, and overall software productivity declined," Glen says. "The level of real output decline is probably exaggerated."

Glen acknowledges, however, that software productivity hasn't made major strides over the last several years. "Software remains a very labor-intensive business. It is not like computer hardware, where overall performance and quality have continued to increase dramatically over time." Additionally, outsourcing hasn't appeared to raise productivity.

Questioning the numbers

Not everyone agrees with Glen's numbers, however, or on how to define software productivity.

Eric Schurr, vice president of marketing for IBM's Rational Software Products group, disagrees with how Economy.com measured productivity.

"I believe that software productivity has improved dramatically over the last five years. I question the manner in which these productivity numbers have been calculated. I would look at unit output over time to gauge productivity," Schurr says. "Ten years ago, or even five, the tools available for developers were not nearly as sophisticated. For example, any developer today

can get a fully built application server in the open software domain to use as part of their application. Take the same developer today versus a developer five years ago and give them six months to develop an application. You will find that the developer today can develop a much more sophisticated application.

“Moreover, our customers continue to tell us of dramatic increases in productivity,” Schurr says.

For example, one of Schurr’s IBM customers, Peace Software, went from 80 to 280 developers in two years and effectively managed four million lines of code.

“Our software enabled them to do this more quickly and more efficiently than they ever could have before,” Schurr says. “Were they more productive? Absolutely. We have another case study with Salion, maker of revenue acquisition management software, where we helped increase their productivity by 400 percent.”

Schurr says that productivity hasn’t just increased in IBM’s customer base, but also in the entire software industry.

“Today, much more sophisticated applications are being developed in shorter time spans by fewer people. Ten years ago you could not have a team of four software developers develop a mission-critical, Web-based customer interface in a period of months. You just couldn’t do it.”

Dealing with software complexity

Per Blysa, vice president of product management at Telelogic, had a different take on Economy.com’s productivity numbers. “I’m not surprised by the numbers. For one thing, the complexity of applications has increased dramatically over the last five years, making it harder to be more productive. For example, in the telecommunications industry, a software developer we work with focused on newer technology, and their software complexity increased by a factor of 10. For another, software developers in the late ’90s and early 2000s have been rushing toward pushing development projects without any process. You definitely see exceptions to the rule, but there has been a pervasive attitude that process doesn’t

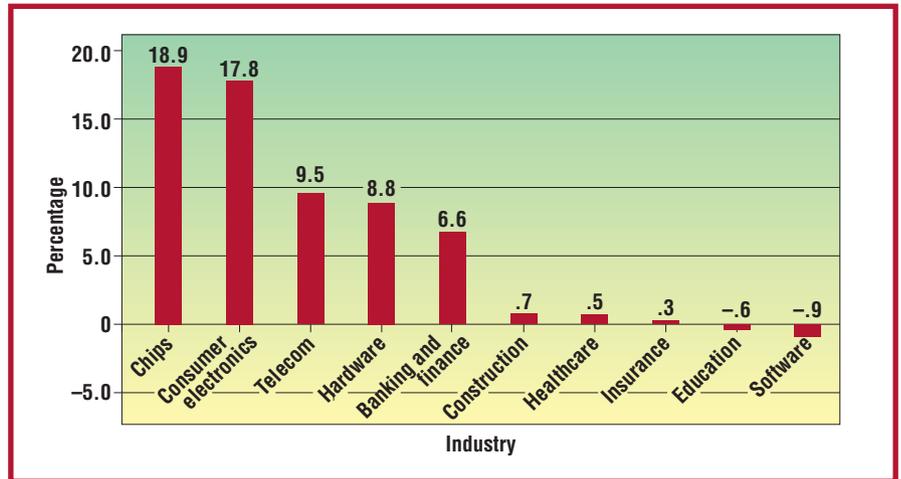


Figure 1. The annual productivity growth rates for industry from 1998 to 2003. (data courtesy of economy.com)

gain you anything. Without process you won’t see productivity improvements.”

Blysa adds, “Almost simultaneously, you had the introduction of Web-based services with Java and a revolution in software outsourcing and simultaneous organizational change. Both of these things increased software development complexity by a wide margin.”

Djenana Campara, chief technology officer at Klocwork, also focused on software complexity and the definition of productivity.

“The numbers supplied by Economy.com do not provide adequate understanding of how to measure productivity,” Campara says. “Certainly we have seen significant advancements in software technology. More complex software applications have provided companies the ability to be more productive. Moreover, more complex software development tools have made it easier to develop these complex applications faster, but at the same time, the complexity of the applications makes it harder for developers to be more productive themselves.”

Campara says development challenges exist today that didn’t five years ago, making it harder to compare developers’ productivity levels. “The complexity of software is such that new developers have a much steeper learning curve when hired on. In addition, existing code has been shifted to developers who were not originally re-

sponsible for it [through offshoring or downsizing].”

Campara says that software applications’ complexity is magnified because development teams are spread out geographically. “Even if you see gains over time from outsourcing overseas, you initially have added complexity in dealing with remote locations, time-zone differentials, and cultural issues,” she says.

Steve A. Stone, director of information work, research, productivity, and business services at Microsoft, says that declining productivity numbers such as Economy.com’s can be misleading. He says Microsoft is currently involved in productivity research.

“Our work also leads us to conclude that we need a better language and mathematics for describing information work.” Although reluctant to comment on whether the software industry’s productivity is in decline, Stone says Microsoft has embarked on an effort to better understand productivity measurement.

If productivity is declining, what are we doing about it?

This effort to discover a better way to measure productivity is Microsoft’s answer to the second question. “The US Bureau of Labor Statistics estimates that nearly 70 percent of the 136 million employees in the US nonfarm workforce were engaged in some form

of information work at the turn of the 21st century,” Stone says. “This implies that there are nearly 100 million information workers in the US in the early 2000s. Microsoft, along with leading members of the information technology industry [Accenture, Cisco, Hewlett-Packard, Intel, SAP, and Xerox] formed the IWPC [Information Work Productivity Council] to research and understand issues around information work, productivity, and measurement.” The IWPC’s work on this is available in the Information Work Productivity Primer (www.iwproductivity.org).

IBM’s approach to increasing productivity, on the other hand, is product-based.

“Rational offers a whole suite of products designed to improve software productivity. It is our mission in life,” Schurr says. “Having said that, discovering what will have the greatest impact for a particular customer varies greatly. One customer may need an automated testing tool to get the greatest impact; one customer may need a configuration management system; another customer could use a set of prebuilt components to help jumpstart them.”

Schurr says that Rational is looking at three main areas for improving software productivity: automating the development process, introducing best practices—including prebuilt components—and using automated testing solutions.

Automating the development process

Bill Shaw, vice president of lifecycle solutions at Telelogic, says the company is concentrating on two areas for going forward: process automation and an industry-specific focus.

“A lot of software vendors have a process, but it is largely manual. The automation of their software development processes will result in large productivity payoffs. We look at the software development life cycle as a whole and try to create a framework that integrates and adds functionality to their existing tool sets,” Shaw says.

He continues, “During a development cycle, there will be at least six to seven different tools that a project may

use to increase productivity at different stages—that is, a CM [configuration management] tool, a design tool, and a requirements tool. If you change more than one tool at a time, the rate of adoption of new tools (and therefore processes) by developers is very low.

“I don’t see anyone else in the industry doing what we are doing to integrate existing tools into a framework that encompasses the entire life cycle,” he says. “Software projects often break down because of the communication gaps between discipline boundaries. We are working to create a framework that integrates existing tools and also allows for integration of new ones in order to most effectively automate process. Where a lot of [our] competitors fall short is in trying to introduce a whole set of new tools and processes at once, which inevitably leads to much lower success rates.”

Shaw says that Telelogic is also developing industry-specific tools and processes in areas such as the military and aerospace, healthcare, telecom, and the automotive industry.

Blysa adds to this, saying, “Increasing software productivity for specific industries has been the theme of our company since its creation in the early 80s. For example, companies that have used our process have seen remarkable gains, like Motorola.”

Detecting defects and improving code

Campara says Klocwork has also been working toward increasing developer productivity. “Klocwork has made major strides in developing static analysis techniques to identify software defects. There is a compelling case to be made that the identification of software defects earlier in the development cycle is the quickest way to make developers more productive.

“We also have been working to [give] developers the ability to use new visualization techniques in order to educate new developers quicker, make code more maintainable, and provide the insight to help developers reuse and refactor existing code. I currently cochair the task force for architecture-

driven modernization at the OMG [Object Management Group]. This is an area that we feel has great potential to drive productivity. The ROI we’ve seen has been remarkable. We have several success stories where developers have reduced development projects from several months to a few weeks.”

Campara says the company isn’t aware of any software tools that really leverage existing code to measure productivity trends build over build, but that it’s developing “a software quality measurement tool that automates the collection of trends on code—even from builds created years ago and only in existence in your configuration tools.”

Common factors

When considering whether software industry productivity is declining, as the *Business Week* article suggests, three factors were common themes among the vendors: the lack of an industry-wide standard definition for software productivity, software applications’ increasing complexity, and the need for more formalized processes in the industry as a whole.

Clearly, no consensus exists as to whether productivity is truly spinning backward and, if so, what to do about it. More software tools and components are available today to help make development easier, even if the job of development is more complex. Yet we can always compare the software industry with the computer chip, telecom, consumer electronics, and computer hardware industries, which have seen their own technologies get more complex year after year but which have also seen amazing productivity growth. The declining productivity numbers the article presented, whether accurate or not, will likely trigger some healthy discussion in the software industry. 

Robert Groth currently works with software developers at Klocwork to improve software quality and productivity. He has worked in high-tech sales, consulting, and development for companies that include Hewlett Packard and Sun Microsystems. Contact him at groth@klocwork.com.