



Presents
An IT Metrics and Productivity Journal Special Edition

Focus on Dr. Michael Cusumano, MIT
A CAI State of the Practice Interview
June 2005

Biography of Dr. Michael A. Cusumano:

Michael A. Cusumano is the Sloan Management Review Distinguished Professor at the Massachusetts Institute of Technology's Sloan School of Management. He specializes in strategy, product development, and entrepreneurship in the software business. He received a B.A. degree from Princeton in 1976 and a Ph.D. from Harvard in 1984, and a postdoctoral fellowship in Production and Operations Management at the Harvard Business School during 1984-86. He is fluent in Japanese and has lived and worked in Japan for seven years. He has been a director of several public and private software companies, and has consulted for approximately 50 major corporations around the world, including Alcatel, AOL, AT&T, Business Objects, Cisco, Ericsson, Fiat, Telecom Italia, Ford, Fujitsu, General Electric, Fidelity, Hitachi, i2 Technologies, IBM, Intel, Lucent, Merrill Lynch, Motorola, NASA, NEC, Nokia, NorTel, Robert Bosch, Schlumberger, Siemens, Texas Instruments, and Toshiba. Professor Cusumano is the co-author of the international best-seller *Microsoft Secrets* (1995, with Richard Selby), which has been translated into 14 languages, as well as the top-10 Business Week book *Competing on Internet Time: Lessons from Netscape and its Battle with Microsoft* (1998, with David Yoffie). He has also published *Platform Leadership: How Intel, Microsoft, and Cisco Drive Industry Innovation* (2002, with Annabelle Gawer); *Thinking Beyond Lean: Multi-Project Management at Toyota and Other Companies* (1998, with Kentaro Nobeoka); *Strategic Thinking for the Next Economy* (2001, with Costas Markides); *Japan's Software Factories* (1991); and *The Japanese Automobile Industry* (1985). His most recent book is *The Business of Software: What Every Manager, Programmer, and Entrepreneur Must Know to Thrive and Survive, in Good Times and Bad*.

◆ ◆ ◆ ◆ ◆ ◆ ◆ ◆

CAI: What, if anything, is going to prevent all IT service operations from being offshored over the next 15 years?

CUSUMANO: I don't believe all software development will ever be offshored for one reason alone; namely, that the best way to develop any kind of creative software is to do it on site with the customer. This is important for both custom-oriented applications as well as product applications.

We have a lot of evidence that dividing projects up too much or separating customers

from their developers and designers just leads to longer, more complex projects that are often very unsatisfactory. There is just too much iteration that is required in good software development, particularly leading-edge software. And this has to be done with your customers. So it's really impossible to offshore everything. There are plenty of customers that actually need to build new functionality and build it quickly, that need to have programmers come in and solve problems that are not so straightforward, and all of this really should be done onsite. What percentage of software development this represents is a difficult question, but generally what we've seen is that it is mainly the more routine development, particularly maintenance operations or enhancements to existing systems, that goes offshore easily. New development on the product side or the custom side is a different story. That's always going to be a mix of offshore and onshore.

Additionally, the cost of going overseas is not such a straightforward subject anymore. We've already seen wages in India rise significantly over the past few years. Wages are going up in India, particularly for experienced managers. At most of the big software companies in India, for instance, experienced managers are now paid internationally competitive salaries. There are also very high turnover rates at the mid and senior level, and this puts additional upward pressure on wages.

There may still be about a three-fold wage differential between US and Indian programmers. However, by the time we figure in all the costs of traveling back and forth, the costs of doing more iterations than we might otherwise have to do, the extra rework, and the correspondingly lengthened requirements and testing phases, the savings of going to India come out around 40%, and that's nowhere near the straightforward wage differential. Moreover, the cost/benefit ratio is actually dropping over time. So, looking out several years into the future, it may very well be the case that the cost of doing new development or even routine development and maintenance in India winds up being higher than doing it in the United States.

Finally, I think you'll find that there are certain classes of projects that will probably never be outsourced, projects for instance that are government-related or defense related. You will also find many small to mid-size companies that, for just a variety of reasons, simply prefer to work with a partner with whom they can see the whites of their eyes. For them, the cost savings will never justify the outsourcing.

CAI: What, if anything, is going to prevent all BPO service operations from being offshored over the next 15 years?

CUSUMANO: This is somewhat of a similar question. I'm not an expert on business process outsourcing; nevertheless, I would say that a lot of these operations are going to be simpler than software development and maintenance so it's possible that we will see a higher percentage of back office service operations offshored.

CAI: What are the critical success factors for US domestic IT service organizations over the next five years?

CUSUMANO: Well that's sort of the flip side of what we already just talked about– how to respond to the Indian onslaught. So that means any domestic IT service organization has to be world class in quality. It has to be very competitive in prices. It has to be able to deliver what it promises in terms of schedule. Not to mention the fact that the Indians are raising the bar in terms of process performance.

Nevertheless, when it comes to quality, the Indians are not actually the best in the world. The best quality routinely comes from Japan. But we also have a number of shops here in the US that can generate software at quality levels that rival and even far surpass the Indians, e.g. defense contractors, companies like Hewlett Packard, etc.

In general, however, it's still far more important now than it was ten or fifteen years ago for US IT organizations to be very knowledgeable about state of the art and state of the practice in software process.

Regarding cost competitiveness– how to achieve this will be a challenge if you come from just being super efficient in development operations. Certainly, some of this cost competitiveness can come from going offshore, but there probably has to be some combination of offshore savings and onshore efficiencies.

Another critical success factor will be the ability to deal with certain technologies more effectively than some of the offshore organizations. There is a lot of invention in software that really does occur in the United States, a lot of it associated with US universities and various partnership programs between universities and US corporations. Whether we're talking about the Java world, or the (.Dot) Net world, or Web Services, or wireless technology, or RFID, most of this technology is coming out of American universities, American research labs, and the minds of American programmers. In this sense, I think US companies have some advantage over the rest of the world in that they see a lot of these technologies first. They can hire the inventors of these technologies. These are people that have all grown up together and gone through the same schools. That's a significant advantage.

CAI: How strongly do you agree or disagree with CAI's use of the Manufacturing Revolution Model as a metaphor for understanding where we are in IT today?

CUSUMANO: Fifteen years ago I wrote a book called *Japan's Software Factories*, which was really about how the Japanese were attempting to transfer what they had learned in conventional industries– production management, quality control, reusability– to software development. And since that time, I've done a couple of surveys and collected a lot of data which indicates that this approach actually works extraordinarily well in certain circumstances. The Japanese routinely show the fewest

bugs per thousand lines of code in any regional survey of software development practices. And some of the best companies in Japan, like Hitachi, have tremendous records with customer satisfaction, on-time delivery, and on-budget performance. In the case of Japan, these were all largely mainframe applications where they had already developed years of experience. They were not trying to invent anything new. They were trying to build large-scale industrial systems that had been built before. They were simply building them in somewhat of a different context, as in the case of bullet train reservation and control software or real-time banking systems.

The point though is that, as long as you set up a process and have a plan for how you're going to do development and figure out ways of leveraging your previous experience on past projects, then you will be able to structure what you are doing more effectively. The software engineering world has accumulated a tremendous amount of knowledge on all aspects of development, from requirements through testing, and if all of this knowledge can be brought together through some sort of standard process approach, we should be able to approximate the level of control seen in manufacturing within a software development environment. Where the manufacturing analogy breaks down, in my opinion, is where half of what you're doing is invention (e.g., creating new functionality or solving business problems that have never been solved before) since this may require the creation of new algorithms or the bringing together of different technologies or various other forms of experimentation. It's probably not the majority of software development but in these circumstances, I don't think that a manufacturing analogy – a pure waterfall style with a sequential manufacturing analogy – works very well.

There is still a structured approach, and it is actually the approach that has been very successfully used by companies ranging from Microsoft to Hewlett Packard, and it's really what I would call a structured/iterative development approach with some very disciplined rules about how to do certain things such as how to coordinate changes that people are making or how to conduct testing operations or how to integrate different components. So it's not a pure manufacturing analogy in a 'Japanese software factory' sense (which originally was really very much a highly structured waterfall process where each phase was very tightly controlled by documents, reviews, and management signoffs) but rather, more of a controlled iterative development process that allows experimentation and that has enough structure to keep development operations under control. And again, we've seen tremendous quality as well as creativity come out of these kinds of development approaches.

I recently headed a study that showed that most world class development organizations today, whether they're in India, Japan, the United States or Europe, use some combination of iterative development techniques with some highly structured approaches to maintaining integration and quality. So you could call that somewhat of a 'manufacturing-like' approach. I've compared some of these practices such as Microsoft's and Hewlett Packard's daily build approach to a Toyota production system approach where the production line stops whenever an assembly worker finds any kind of defect and it's fixed and then the production line continues. That's very much the style of this iterative development model as long as companies are using daily builds and regression testing on those builds and rigorous bug control and bug fixing rules.

CAI: What are the Accentures and CSC's of the world doing right now to remain competitive?

CUSUMANO: Consistent with other things we've addressed in your earlier questions, a lot of these firms are setting up development sites overseas to take advantage of programming talent as well as lower costs. In many cases, their customers are demanding that they have these sites overseas in order to reach certain cost or quality goals.

Another thing that is happening is that some of these companies are spending a lot of time trying to develop particular areas of technical expertise. For example, Accenture is building up a tremendous telecommunications practice. So in many different companies around the world, whether it's Deutsche Telecom, or Nokia, Accenture is trying to develop some of the expertise to do the middleware software linkage. That's a special expertise that gives them some differentiation compared to world class but essentially low cost IT providers.

I think the big companies, to fight the pressure of competing purely on the basis of price, will be more and more focused on the development of technical expertise areas as a differentiation strategy. That's in addition to developing deeper offshore capabilities.

CAI: Regarding the increasing migration of companies to packaged software, how is this going to impact the IT services industry over the next five to ten years?

CUSUMANO: I'm not sure there is data that says there is increased use of packaged software today as opposed to five or ten years ago. But let's talk about it.

There is an impression that the world is moving more towards packaged software. And compared to fifteen or twenty years ago, there is clearly more use of packaged software today in most markets.

But not in all. Japanese corporations, for instance, have not really adopted that much more packaged software. It's sort of been stuck at a fairly low percentage. The Europeans, as well, don't use that much packaged software. These are countries where, traditionally, organizations require a lot of customization.

What you do see these days is more custom systems built that include packaged products. So rather than someone constructing a database or CRM system from scratch or rather than a Japanese company building its own database and CRM system and selling these with their own proprietary operating system, they will now get a license and use packages imported from the United States.

In general, though, even if there are more packages available or free, most of them create lots of business for the IT services industry. Very few customers can use these

packages just by downloading them off the Web and setting them up and running them. Almost all packaged software has to be integrated with other software, and that requires that somebody be there to do the integration. For large companies or even midsize companies, it's extremely rare that any packaged software fills enterprise needs as is.

What I am seeing is shrinkage in product license fees, i.e. the prices product companies can charge for the software. They may sell more of it, and there may be more products out there, but it's the amount of revenue going into services and maintenance that is actually going up. It's actually going up fairly dramatically. In fact, the industry did the crossover in 2001. In 2001, services and maintenance exceeded 50% of all revenues at public software product companies listed on US stock exchanges. This does not even include IT services companies. This is just pre-packaged software companies. These are the Microsofts, Oracles, SAPs, Business Objects, etc. So clearly the world has shifted to services and maintenance. The product side may be becoming cheaper and cheaper, but the more products you have available, the more you are going to need integration, customization, support and maintenance. This happens to be the subject of my latest book: *The Business of Software*.

Questions? Suggestions? Comments? Please contact the IT Metrics and Productivity Journal Editor at michael_milutis@compaid.com.