TUCK SCHOOL OF BUSINESS AT DARTMOUTH
WILLIAM F. ACHTMEYER
CENTER FOR
GLOBAL LEADERSHIP

IBM and the Secure Internet

Just after the opening ceremonies at the Olympic Games in Atlanta in 1996, a behind-the-scenes experiment came to life. Thanks to a partnership of IBM and Olympics organizers, people could learn results from every event immediately via the Internet. At least, that was the plan.

By today's standards, such a project sounds routine. But it was hardly routine in 1996, when the Internet was an experimental frontier. IBM knew it could build a database of Olympics results that was reliable, secure, and up to the minute. The question was, could it make the database accessible to millions of Internet users without a glitch?

The project went well. IBM learned how to build a system that could deliver real-time information to an unprecedented number of users. The experiment began to shine some light on how the company could rebuild itself for the Internet Age.

IBM had excelled in past decades providing *proprietary* stand-alone computing platforms for corporations. As computers became ever more interconnected in the 1990s, however, the company's business results deteriorated. For the Atlanta Olympics project, IBM broke from its tradition of proprietary technologies and built a system based on *open-source* standards. It adapted web-server software developed by the Apache Software Foundation, an open-source software consortium, and strengthened it to operate at high traffic levels.

The Olympics experiment, and several other Internet-related experiments IBM had in progress, suggested that IBM's strength in the new computing environment could be in building software that helped *connect* computers. Specifically, IBM executives believed the company could excel at making connections between the bulletproof systems that could never fail and the wildly exciting but tremendously unpredictable world of the Internet.

This belief became the foundation of IBM's software strategy over the next decade. IBM wrote reliable and secure software—*middleware*—that connected a wide range of systems. IBM's WebSphere brand eventually expanded to a portfolio of more than 100 middleware applications by 2007.

This case was written by Professor Chris Trimble of the Tuck School of Business at Dartmouth. The case was based on research sponsored by the William F. Achtmeyer Center for Global Leadership. It was written for class discussion and not to illustrate effective or ineffective management practices.

To order additional copies please call: 603-646-4055

Transaction Security

Users of personal computers tolerate occasional hiccups: A computer freezes and needs to be restarted. Or normal work needs to be interrupted to install new software. Or a battery dies on a long flight.

Those same users expect better of their banks' computer systems. They expect transactions, say, simple transfers from savings to checking, to be flawless every time.

That expectation puts a great deal of pressure on directors of information technology. A bank's databases are mission critical. They must be secure and reliable 24/7. When there are technology issues, technology teams work around the clock until they are fixed, and they cannot disrupt normal operations while they troubleshoot and repair. Banks have rigid protocols for accessing and manipulating databases, as every customer would naturally expect.

For decades leading up to the era of explosive Internet growth, IT directors trusted IBM for mission-critical systems. The brand was synonymous with "never fails."

At the core of mission-critical systems for large business enterprises in banking and other industries is a tremendously sophisticated, if also somewhat esoteric, piece of computing technology known as a *transaction monitor*. The device's functionality is simple in principle; it ensures that transactions are processed accurately.

For example, airline reservation systems can now be accessed by thousands of people simultaneously. One of the transaction monitor's tasks is to ensure no seat is sold twice. Another is to ensure the system can accurately recover any time a transaction in progress is disrupted. The transaction monitor must also be able to handle tremendous surges in traffic—for example, the surge that brokerage systems face during sudden periods of excitement in financial markets. To function flawlessly, transaction monitors have features like backup processors and logs of every step taken that enable them to recover from faults.

IBM excelled at transaction-processing technology. IBM's senior executives were proud of the company's installed base of mission-critical systems and did not hesitate to make bold claims, such as, "The economy runs reliably because the economy runs on IBM." In fact, before investing in a new transaction system, IBM encouraged companies to stress-test the systems they were considering. Steve Mills, head of IBM's software business, explained,

> The software industry is not well known for the precision of its claims, so we always like to see our systems benchmarked against the competition.

Good tests measured both speed and robustness. Customers stress-tested systems by haphazardly disconnecting network cables and power supplies and noting the results.

In the mid-1990s, corporate IT directors faced new needs. As Internet usage exploded, they began to think through proposals for giving *customers* direct access to *mission-critical data* over the *Internet.* They began to lose sleep—lots of it.

The Internet multiplied the demands on transaction monitors at the heart of mission-critical systems. From the perspective of the IT directors and software engineers who worked on mission-critical systems, the Internet was a dangerous place. Connections were disrupted frequently. Hackers found ways to access systems illicitly. Computer viruses ran rampant.

IBM suffered for its historically steadfast commitment to proprietary technologies. In fact, the company was fighting just to be perceived as *relevant* to the Internet. It was losing the battle for the hearts and minds of the technology industry—losing to younger firms in Silicon Valley. By the time of the Atlanta Olympics, senior executives at IBM perceived that they faced a life-or-death challenge.

Experiments

IBM's Internet strategy, bringing the reliability of legacy systems to the Internet, evolved through a series of experiments and initiatives, of which IBM's partnership with the Atlanta Olympics organizers was only one.

To develop insight into how the Internet would develop and what role IBM could play, the company created an Internet division within its research department in late 1995 under the leadership of Irving Wladawsky-Berger. A 25-year IBM veteran, Mr. Wladawsky-Berger had spent his entire career in research and product development. The Internet division commissioned numerous pilot projects. Beyond the Atlanta Olympics initiative, the group launched a series of six experimental offerings, such as a web application server and an Internet browser. The Internet division was a substantial endeavor. It involved a few hundred full-time people with a variety of skills, plus collaborations with other groups across IBM. Mr. Wladawsky-Berger recalled the environment at that time:

> Many of us were already familiar with underlying Internet technologies, some of which had been around for a while. But what nobody could understand at the time was how the Internet could be used in commercial markets. We knew the Internet was going to be very exciting, but we did not know how IBM, or how anyone, could make money with it.

It was a high-profile endeavor. Customer response to the initiatives was the largest factor shaping the evolution of IBM's Internet strategy. Mr. Wladawsky-Berger saw no obvious single technology IBM should pursue in the labs, and the alternative was to follow trends, launch experiments, monitor customer reactions, and, eventually, figure out how to make money.

He and Chief Executive Lou Gerstner met quarterly to discuss outcomes from the Internet initiatives, trends in the industry, and the strategic implications for IBM. Mr. Wladawsky-Berger recalled the discussions:

> We tracked profit and loss from the initiatives but did not talk much about it, especially in the early stages. Our discussions were more about market strategy and leadership and were longer term in nature. We made some crucial decisions about how to shape our Internet business.

IBM discontinued many nascent initiatives. For example, the company exited the Internet browser market because it was a giveaway, a consumer product, and not related to IBM's core expertise. Mr. Wladawsky-Berger explained,

> Soon, the strategy became clear. Conventional wisdom at the time was that the Internet was so revolutionary that legacy computing, if not legacy businesses, was going out the window. But we saw that you could do incredible things by taking your existing enterprise infrastructure and putting a web front end on it. The features Federal Express and UPS introduced that made it easy for customers to track their shipments online were early successes of that approach. They showed that legacy applications and products could be entirely relevant in the new world.

IBM chose to focus on servers, the computers that served as intermediaries between legacy systems and the Internet. As the strategy crystallized, Mr. Wladawsky-Berger discovered his biggest problem was not developing new technology, it was marketing. Even the internal communications effort was difficult.

By focusing on open-standards middleware, IBM was radically changing direction. Mr. Gerstner's enthusiasm motivated others in the organization. To step up external marketing efforts, Mr. Wladawsky-Berger began hiring senior marketers from other Internet firms to help communicate IBM's message.

The Internet division transferred those initiatives that it did *not* shut down from IBM Research to IBM's business divisions. In Mr. Wladawsky-Berger's view, as soon as the product groups were *interested* in taking over—when there was steady market demand for a new product and certainly by the time there were clear competitors on the horizon—it was time for IBM Research to step out of the picture. Mr. Wladawsky-Berger shut down the Internet division in 1999. The strategy was clear and the division was no longer needed.

Software

As Mr. Wladawsky-Berger explored possible Internet strategies, Steve Mills, another IBM veteran, pursued a different mission, albeit one that ultimately would point the company in a similar direction. Mr. Mills endeavored to rebuild IBM's software business from scratch. The company was still making tremendous profits from software for its proprietary mainframe systems, but Mr. Mills and others worried that the mainframe software business had little growth, possibly even little life, remaining.

In the early 1990s, IBM began to think of software not just as a complement to its proprietary hardware, but as a business in and of itself, one that served multiple platforms, not just IBM platforms. That drive resulted in the development of a PC operating system, OS/2, and investments in other office-work programs. For example, IBM acquired Lotus Development Corporation, the company famous for its spreadsheet Lotus 1-2-3.

Through the latter half of the 1990s, Mr. Mills led a program of radical change, cutting the company's software development budget in half before rebuilding it. It became clear that the Internet and IBM's expertise in building reliable transaction engines were to become the foundation of the company's software strategy. That naturally led Mr. Mills to build and commercialize an e-commerce offering. More broadly, it was clear that in an increasingly interconnected world, IBM's software had to be based on open standards.

The emergence of the Java programming language was an important turning point. Developed by Sun Microsystems, Java enabled browsers to run software loaded from an Internet server, rather than to simply display information. IBM chose to support Java and even played a role in developing a more sophisticated version of the language, Java Enterprise Edition (J2EE).[1] This language incorporated reliability and security features to enable interaction with mission-critical systems. Dr. Danny Sabbah, who reported to Mr. Mills through the late 1990s, recalled,

> Sun marketed Java as a wonderful way to download functionality. They did a great job of promoting it, and the entire industry jumped on the bandwagon. From a technical perspective, there was nothing astounding about Java. It shows the power of marketing and mind share in the industry. We decided to jump on the bandwagon too. It was controversial internally, but we had to build software that was based on open standards.

Java made it possible for corporate IT groups to build Internet servers, later known as *application servers*, that allowed anyone with a browser to access corporate networks. Before Java, an employee or a customer who wanted to gain access to a corporate network needed to install special software for the task.

After shaping the standards for J2EE, IBM began developing a new product line, WebSphere, to make money. WebSphere was a series of software products for programmers who wrote J2EE software for application servers. Some of IBM's mainframe software development groups, still very large businesses, had initiated projects similar to WebSphere. Mr. Mills cut off a number of such investments to move the company in the same direction.

[1] The language was renamed Java EE (Java Enterprise Edition) in 2006.

At first, the products that IBM developed were rudimentary and difficult to use. They were also offered piecemeal. Programmers had to integrate the pieces to gain the desired result. That was painful, but the Internet's potential was so compelling that programmers were willing to put up with a lot of pain. Bringing reliable transaction processing to the Internet was a mammoth challenge.

Between 1998 and 2001, IBM worked on integrating its various programming products into a single, unified WebSphere application server. There were three major components. The first was the transaction engine. IBM acquired Transarc, a company that offered an Internet transaction engine, to jumpstart its own development of a transaction monitor that ran on an Internet server rather than a mainframe. The second was a "component broker," which helped connect internal systems. The third component was a "web mechanism," similar to that built by IBM for the Olympics. Mr. Mills described how he brought the pieces together:

> The three leaders and I agreed that the pieces needed to be put together as one integrated product. It was a challenge to get all of the developers to buy into the big vision and overcome the inevitable parochialism. But we completely stopped development in each of the three buckets and focused entirely on bringing them together as a single design.

IBM hoped the integration would create an application server that made it as easy as possible for developers to build reliable and secure Internet applications. Web developers needed to incorporate a great deal of functionality beyond speedy and reliable transaction processing. For example, banking *customers* needed a different computer interface than did bank *tellers.* Customers needed 24/7 convenience, a very intuitive interface, and the ability to access a wide range of information. Bank tellers needed speed.

IBM built a dedicated services arm to ensure web developers were successful with WebSphere. Dr. Sabbah recalled,

> We'd do everything we had to, to ensure that the software stuck. That meant a lot of face time. We'd fly people around the world to ensure our customers were successful. Today, our technical services team is over 5,000 strong.

Mr. Mills believed IBM's brand reputation was strong for any product that called for transaction security and reliability and, as such, made minimal investments in marketing. Instead, he built a dedicated new sales force for IBM software and trained salespeople to get very comfortable selling "cross-platform" solutions. The transition was difficult for them because for decades they had focused on convincing clients that IBM-only solutions were best. Mr. Mills transferred the majority of the sales team from existing IBM sales forces. He felt WebSphere was a top priority and that shifting the sales resources would help ensure its success.

Training and gaining the support of large and small IT service providers was a priority. Ironically, IBM's own services arm, IBM Global Services, was one of the toughest to persuade because the unit had not adopted a technology-neutral stance. Once WebSphere gained some significant market pull, however, the services team came around.

Mr. Mills reflected on the financial expectations for WebSphere:

> We did not have an extraordinarily detailed financial plan. These investments were justifiable purely on a disaster-avoidance basis. The problem with plans is that they are always built on the past. That doesn't work at critical junctures like this one. Instead, you have to have a view of what the new landscape will look like, and go attack it. We were driven primarily by customers telling us what they needed. Once we had a handful of satisfied customers, we focused on selling those customers more copies. After that, we focused on revenues and, finally, profit. That is the pattern I have seen throughout my career.

The WebSphere application server had gained considerable momentum by 2002. IBM released new versions every year. Standards were evolving quickly, and that created complexity. IBM inevitably created some uniqueness in its software but linked to external standards wherever possible.

E-Commerce

One of the most rapidly growing online commercial applications in the early years of the Internet was e-commerce. Internet software was at the core of such operations. Separate from the WebSphere effort but running parallel to it, Mr. Mills and Mr. Wladawsky-Berger began building a business-to-consumer e-commerce software offering in 1997. Before making the choice to build a software package, they considered a variety of other business models, such as building an Internet "mall" and charging retailers for space in it, and offering a service to Internet retailers who wanted to hand off technology management to a partner.

To lead the venture, Mr. Mills and Mr. Wladawsky-Berger tapped Dave Liederbach, who had been closely involved with the effort to reshape IBM's software operations. Mr. Liederbach was an IBM veteran with experience primarily in sales and service.

One central aspect of e-commerce operations was simply presenting goods for sale over the Internet, and several software companies developed merchandising applications to meet that need. At their simplest, such applications helped retailers build online catalogs and take orders. Retailers would then manually initiate further processing and fulfill those orders.

Naturally, online retailers wanted to automate as much as possible of the order-to-cash process. They wanted an "end-to-end" e-commerce system. They wanted online orders to automatically initiate a sequence of business steps: charging the customers' credit cards, debiting the inventory system, initiating the shipping process, and more. It was in these areas—integrating multiple systems and ensuring reliable, scalable, secure transactions—that IBM could excel.

Adding merchandising functionality was a relative challenge for IBM. The company's earlier transaction-processing systems did little more than process transactions, but the online interaction between a customer and a retailer was far broader and richer than just the transaction. In fact, according to Mr. Mills, less than 2 percent of the information exchanged between retailers and customers online was directly related to the transaction itself.

The e-commerce software market was developing extraordinarily rapidly at the time. As a result, IBM chose to launch a pilot project with longstanding customer L.L.Bean right away. It was unusual for IBM to launch software before it was fully and completed tested.

Because L.L.Bean had always been a catalog operation, the move to e-commerce was a more natural transition for that company than it would have been for a brick-and-mortar retailer. IBM worked with its client to perfect the latter's end-to-end system over a period of several years. Mr. Liederbach described how the market developed:

> Compared to the competition, our offering was extremely secure and rich in functionality in transaction processing. We were not as good at merchandising. It took a while for the market to understand just how critical reliability and security were. What drove the market at the time was hype and cool functionality, but we remained anchored to a five-year plan, wanting to pull our existing database and transaction-processing capabilities into the center of the e-commerce market.

IBM's offering, dubbed NetCommerce, appealed to organizations that valued reliability and security. It also appealed to companies that had an established brand to protect and were unwilling to suffer a major technology breakdown. Many startup Internet retailers took the lead in e-commerce, but almost as many disappeared within just a few years.

Mr. Liederbach described the benefits and challenges of building an e-commerce software startup within IBM:

> We were a well-funded startup, and we were fortunate not to have to prove something to venture investors every few quarters. While some would have focused on quarterly targets, Mr. Mills took a long-term view. Still, we were a startup inside an elephant, and there was a level of bureaucracy that could be difficult. Plus, our brand meant "irrelevant" at that time, and we had to work very hard to get the analysts and the press to pay us any attention.

To build his team, Mr. Liederbach hired extensively from outside IBM. He felt it was important to avoid legacy processes and not to rely on an experience base that was rooted in the past. Because IBM's recruiting engine had atrophied through years of downsizing earlier in the decade, Mr. Liederbach recruited through his own personal networks. While he could offer the opportunity to work with high-caliber people and patient investors, he could not offer competitive stock options or even impressive titles. Nonetheless, he succeeded in hiring nearly 40 percent of his employees from outside the company.

The NetCommerce team spanned three core functions: product development, marketing, and services management. All the functions needed to interact in various ways with other established IBM groups. For example, the product development group worked with IBM hardware teams. IBM was trying to build its Internet server business, at the time dominated by Hewlett-Packard and Sun. The relationship was productive; each side believed the other could help them.

The NetCommerce product development team also collaborated with other software groups, particularly IBM's database group. It was crucial for NetCommerce to work smoothly with IBM's database product, DB2. As a result, DB2 programmers developed parts of the NetCommerce product without the NetCommerce business unit paying for their time. Mr. Liederbach believed it was equally important that NetCommerce work smoothly with competitive offerings, and that created contentiousness. Mr. Liederbach recalled,

> We were a very small fish creating waves. I recall having conversations with IBM general managers running billion-dollar businesses about how I was going to "support" the competition when a client or market required it, such as by integrating with the then leading database, Oracle. Mr. Mills supported an "open strategy" and helped people see the bigger picture.

To save money and take advantage of IBM's existing relationships with clients, Mr. Liederbach chose not to build a dedicated sales force. It was difficult at first to get the sales force to pay much attention to NetCommerce because the dollar value of a NetCommerce sale was in the tens of thousands and the sales force had quotas of several million.

Toward the end of the 1990s, as the e-commerce market became even hotter, IBM created an "e-business solutions team" within the sales force. About 100 members of the team developed expertise in the NetCommerce product. To motivate other salespeople to bring the NetCommerce experts into conversations, IBM changed its sales force compensation structure so that commissions could be shared. Even if one of the NetCommerce specialists completed the sale, the salesperson who initiated the conversation also received a commission. At times, NetCommerce was an asset to salespeople charged with pushing other products. Many companies were working on e-commerce problems, and it was helpful to talk about NetCommerce as one piece of an overall solution.

Mr. Liederbach also worked through IBM's channel development organization to build relationships with IT distributors and IT service providers that served smaller companies. It was relatively new territory for IBM. Microsoft had a much stronger position in such channels. Ultimately, the NetCommerce effort helped establish a foundation for pushing a broader IBM software portfolio.

It was not hard for the sales team to convince customers they needed e-commerce software. However, clients were always concerned about implementation. Mr. Liederbach estimated that at least 70 percent turned to outside service providers for help.

The services management function within the NetCommerce team focused on building the capacity of IBM Global Services and outside IT service providers to implement NetCommerce for clients. Working with outside providers who competed with IBM Global Services made for an awkward relationship between the two IBM teams. This relationship was further complicated because IBM Global Services helped their own clients install software that competed with NetCommerce. Mr. Liederbach recalled,

> It was push and pull with them, but we pushed through the friction and got the relationship off the ground.

Programmers skilled in Java were hard to come by, and that slowed the efforts of clients to implement NetCommerce. To ensure every client was successful, the NetCommerce team at one point assigned 40 developers, about one-fourth of their development team, to work directly with customers, helping them implement NetCommerce in their own IT environment. In the process, the NetCommerce team did everything they could to transfer skills and knowledge to clients.

In 2000, Mr. Liederbach was promoted to work directly for Mr. Gerstner. NetCommerce was a successful offering. Soon thereafter, NetCommerce was redesigned atop the WebSphere application server and re-branded WebSphere Commerce, so that the two packages were fully integrated and fully compatible. Some of the WebSphere Commerce features migrated to the application server because they had broader applicability. The application server and the e-commerce platform became the foundation for further expansion in the middleware category.

Middleware

As of 2002, IBM's strategic objective for WebSphere was straightforward: to offer the most capable application server on the market. The key criteria buyers would evaluate were the number of systems with which the server could connect (e.g., IBM databases, IBM mainframes, Oracle databases, SAP enterprise software, Siebel customer-relationship-management software, and so forth) and their security, speed, scalability, and reliability. Also, buyers evaluated how easy it was to develop new applications to run on the application server. Could you hire someone out of college to write applications? Were training materials readily available? There were plenty of reference books available that described the features of J2EE, and that was an advantage for IBM.

By 2004, the ambitions for WebSphere had expanded. IBM had begun to view it as a reliable and secure foundation on which other IBM products could be built. The company proceeded to build a family of WebSphere products. Other IBM software divisions besides WebSphere, including Lotus, Tivoli, and Rational, joined in the effort, building new products for WebSphere and converting existing products to be compatible with the WebSphere foundation. IBM consciously stopped short of building end-user applications. WebSphere Chief Technology Officer Jerry Cuomo explained,

> We are really not in the applications business; we are in the applications-enabling business. We want to make it as easy as possible for corporations or their IT service providers to design and deploy their own applications, or for independent software providers to develop applications based on WebSphere technology, thereby creating a vibrant ecosystem of WebSphere application vendors.

Small independent software developers could not practically develop their own middleware modules, such as a transaction monitor. It was beyond their resource base. Thus, the WebSphere family of products enabled small companies to build software for corporations.

As WebSphere expanded from its core product, the WebSphere application server, to a broader tool set for programmers, the WebSphere brand also expanded, from a certain set of features to a philosophy of how modern corporate IT systems should be built and managed. One core tenet of the philosophy was that applications developed using WebSphere should be interoperable with most any system and easy to reuse. A related tenet was that corporations should resist the urge to make wholesale changes to their existing systems to take advantage of the newest technologies. As Mr. Mills put it, “The best code is the code never written.” WebSphere was intended to facilitate building links among existing systems and newly written code. One product, WebSphere Enterprise Service Bus, was designed to be a single junction to which many applications could be connected, eliminating the need to create separate point-to-point connections every time two applications needed to interoperate.

The notion of reusable interoperable applications was becoming popular in the industry. An industry analyst coined the term *service-oriented architecture* (SOA) to capture the approach, and IBM helped to popularize the idea, linking it to the WebSphere brand. According to marketing executives in the WebSphere group, SOA was an approach to breaking up applications into small component pieces and making each piece accessible. *Services* were small applications that could easily be called upon by other applications. For example, a company could create a service that verified zip codes matched cities in addresses and make it available for use in accounts payable applications, order verification applications, and shipping systems. Companies could readily make their services available across organizational boundaries. Some organizations that had done so were surprised by how frequently the applications were used and learned just how crucial the scalability of services was.

Another core tenet of the WebSphere philosophy was that IT systems should not solidify a company's work processes in concrete. Business conditions were always in flux. Processes needed to change. New partnerships were formed and old partnerships were dissolved.

IBM attacked the need for greater process agility with a product called WebSphere Process Server. The software presented managers with an environment that allowed them to define or modify work processes visually, using diagrams and flowcharts. Once managers defined processes in this manner, WebSphere Process Server could create some of the necessary software code automatically. Then, programmers could complete the job. In the idealized SOA world, redesigning a work process required not much more than disconnecting and reconnecting existing services.

Corporations typically found it hard to redesign processes that crossed organizational boundaries—from company to supplier, partner, or customer. Inevitably, partnering organizations had developed unique ways to format their business data. Thus, to build a new process that crossed organizational boundaries, one that allowed applications to interact directly without any human intervention, inevitably required some expensive process redesign and systems reprogramming. Such efforts tended to be complicated by the tensions that accompany any partnership, rooted in power struggles and uncertainties about how much information should be shared with business partners.

IBM worked to develop products that made it easier to connect a company's computer systems to those of its business partners. Some meta-standards, such as Web Services Description Language, were evolving that enabled one computer to communicate to another, "Here is how my information is formatted," before sending the information itself. Nonetheless, eliminating humans from routine organization-to-organization interactions remained challenging. Some forward-thinking organizations began to try to measure process agility to assess their progress.

Because the WebSphere family of products was expanding rapidly and the WebSphere development team completed new releases frequently, the WebSphere marketing team chose to "launch" WebSphere twice per year, introducing all the new products and functionality in one consolidated event focused on a single theme. The first such event was in September 2005, when IBM formally embraced the notion of services-oriented architecture and introduced the Enterprise Service Bus. A subsequent launch focused on the theme of business process management and introduced WebSphere Process Server. Sandy Carter, by 2007 WebSphere's head of marketing, described her approach:

> Prior to 2005, studies showed that the WebSphere brand was perceived as old. We needed to make WebSphere appear supercool. SOA is the new approach to integration, so we needed to make WebSphere the platform of choice for SOA. We talked to every influencer we could to make the September 2005 announcement as powerful as possible. The results were dramatic. We turned up the volume around SOA, and now competitors have tried to emulate it.

Ms. Carter's marketing team included a group that focused exclusively on SOA. A corporation that chose to embrace SOA embarked on an agenda to transform the information architecture for the entire enterprise. A decision to make such a move was one that a company would likely live with for 10 years or more, so directors of IT wanted to be sure they made the right choice. As such, IBM expected perfection from its software. Far from the experimental releases of the late 1990s, by 2007, IBM expected WebSphere software to work perfectly every time.

The sales cycle was long, and IBM prepared a lot of demonstrations of the technology, often customized for prospective buyers. Customers considered IT transformations that involved much more than just changing systems. The transformations also required redesigning processes and changing the way people worked. As part of its marketing effort, IBM published books advising companies on how to make the transition.

WebSphere General Manager Tom Rosamilia's group was organized functionally, with heads of product development, marketing, and sales, among others. The product development group was further organized by WebSphere product. Over 6,000 developers worked on expanding the WebSphere family. IBM had made an aggressive sequence of acquisitions to expand WebSphere's capabilities and the capabilities of the development team. Leaders of each product development road map were tied to rigorous product road maps and schedules. Missing a deadline could mean missing one of the twice-per-year launch events and, thus, going to market with limited marketing support.

An architecture review board, which included Mr. Cuomo and others, ensured that the WebSphere family of products was as internally consistent and compatible as possible. WebSphere leadership endeavored to give developers autonomy and room for creativity within that constraint, while avoiding duplications of effort.

WebSphere had expanded from a series of experiments, starting in the mid-1990s, to an application-server and e-commerce offering in 2002, to a family of over 100 products by late 2007, by which time it had delivered revenue growth for 37 consecutive quarters. The WebSphere team tracked its sales pipeline closely and knew the lead-to-sale ratios across all its products. Based on this data, the team made hard projections of sales one to two quarters out. When there was a shortfall in an industry or a country, the marketing team dispatched extra support.

Mr. Mills assessed WebSphere's progress:

> Today, recognition is high, nearly to the point that people think they have to look at WebSphere because everyone else is using it. We have a lot of references. We'll continue to do well if we continue to keep our eye on how value is shifting and keep moving there aggressively.