



**Rational**® software

## **Building the On Demand Business: Three Imperatives for Software Development**

*A Whitepaper from the Rational software division of IBM*

## Executive Summary

One of the major factors in determining a company's success in the on demand era is its software development capability. The new business demands and consumer expectations that define e-business on demand™ will require a higher level of responsiveness and agility from software development teams. And the software applications these teams maintain and develop must be ever-more innovative to keep businesses ahead of the competition.

To support the on demand business and its operating environment, business software applications must:

- Adapt quickly to changing business needs.
- Capture and maintain a strategic advantage.
- Be reliable and scalable for the growing on demand business.

To ensure these advantages, businesses must rely on software development organizations to extend and upgrade legacy applications, customize and extend commercially available applications, and develop new applications that offer competitive differentiation in the marketplace. Yet, to leverage maximum value from software through customization, extension, and new development, most organizations must improve their software development capability with techniques suitable to the on demand world.

Specifically, they must embrace three software development imperatives:

- 1. Develop iteratively** – Teams must use a results-oriented process that yields increasingly improved iterations of a software system until it is ready for deployment. This reduces project risk, increases predictability, allows proper scope, and reduces design flaws.
- 2. Focus on architecture** – They must base all software design on reusable components and a service-oriented model, which can be maintained, upgraded, or replaced without compromising overall system function. This allows applications to be designed for change, with reduced complexity and higher quality and integrity.

**3. Manage change and assets** – Teams must track all changes to the software under development, manage the team’s activities, and protect the development assets that are strategic and unique to the business. This shortens development lifecycles by allowing multiple teams to develop in parallel, protects critical assets of the development process, and improves confidence in deployed software.

IBM Rational, formerly an independent company and now one of the IBM software brands, offers a comprehensive software development solution based on the three imperatives above. The IBM Rational platform combines software engineering best practices, market-leading tools, and expert professional services, all of which drive rapid and continuous improvement in software development capability for on demand businesses.

In addition, IBM Rational offers more than 20 years experience in promoting and delivering *integrated* and *open* software systems, both of which are key characteristics of the on demand operating environment:

**Integrated** – IBM Rational has contributed considerable thought leadership and expertise in the areas of Service-Oriented Architecture (SOA), enterprise and software architecture, and heterogeneous platform support.

**Open** – IBM Rational has a long history in developing and supporting the goals of open computing. This includes development of the Unified Modeling Language (UML), now a standard for modeling applications, database design, and business processes. IBM Rational has promoted and participated in the development of a wide variety of open computing standards. It offers support for major programming languages and operating platforms, and it provides an extensive set of application programming interfaces for third-party tool inter-operation.

Thousands of companies around the world have realized the benefits of the approach advocated by IBM Rational. Their processes are results-oriented, the artifacts they produce are well-designed and reusable, and they are working at higher levels of capability now required by the on demand era.

---

**Contents**


---

- 2** *Executive Summary*
- 4** *Introduction*
- 6** *Creating the on demand e-business*
- 9** *Deriving business value from software applications: the need for software development*
- 10** *Three imperatives for improving software development capability*
- 11** *The development process for on demand software applications*
- 20** *Supporting the on demand operating environment*
- 24** *Software development solutions from IBM Rational*
- 30** *Conclusion*

## Introduction

Remember the frenetic pace of the mid-1990s, when nearly every business was adopting an Internet strategy? The possibilities seemed endless. Websites sprung up overnight, promoting every conceivable good, service, and category of information. Banks began offering corporations and individuals access to account information and soon were giving customers the ability to transfer funds between accounts and pay bills. Airlines started moving from simple online flight display to online booking. Businesses of all sorts were busy augmenting their sales channels with secure, interactive storefronts.

Since the dawn of the World Wide Web, companies seeking to exploit its business value have had a lot to keep up with. They've not only had to adopt – and adapt to – rapidly changing technologies, they've also had to be innovative in order to outshine the competition. In making the first step to basic Internet access, businesses hired Web-savvy teams who could help them quickly respond to a international wave of enthusiasm for the Web. The second step, integration of internal systems to enable Web-based transactions, presented even more challenges. Companies had to automate essential business process that would take customers beyond the hype of virtual billboards and into the realm of real value exchange.

What transformations will the next few years bring? A new era of business automation has begun, and more and more companies are racing to become a part of it. As one of the pioneers of “e-business,” IBM has a vision for this next phase. It's called e-business on demand.<sup>™</sup>

In the same way that businesses made the transition from simple Internet access to integrated, Web-enabled business process integration, companies must take a critical third step, creating an enterprise based on business processes that are integrated, end-to-end, across the company and across multiple enterprises, linking businesses with key partners, suppliers, and customers. The result? On demand e-businesses that will be able to respond with flexibility and speed to any customer demand, market opportunity, or external threat.

***Software development teams must be able to extend existing applications, customize and deploy packages, and build and deploy new applications more quickly than ever before.***

To get there, organizations are leveraging the power of software. They must be able to extend existing applications, customize and deploy packages, and build and deploy new applications more quickly than ever before. For this reason, a company's software development capability is one of the major determining factors in how successfully it makes the leap to on demand computing. Their software development tools and techniques must support the new business demands and consumer expectations that define e-business on demand™. And the applications they use must be ever-more innovative to keep their business ahead of the competition.

This paper will explain how a business that wants to become an on demand business can improve its software development capability. We will examine 1) the relationship between the defining characteristics of e-business on demand™ and the applications that automate and integrate business processes; 2) the software requirements that support the essential operating environment of on demand computing; and 3) how software development is essential in creating strategic advantage. We will consider the characteristics of successful software architectures for organizations that thrive in this new era. Finally, we will explain the unique benefits of using the IBM Rational solution: the software development platform for an on demand world.

*The on demand era requires leaders to see and manage their business as an integrated whole.*

## I. Creating the on demand e-business

What is an on demand e-business? IBM defines it as one whose leaders can see and manage their company as an integrated whole. This means that all sectors of the business must engage each other in a dynamic transformation of formerly isolated departmental operations into full business processes integrated across the company and outside to their customers.

An on demand business has four essential characteristics<sup>1</sup>. It is:

**Responsive** – intuitively responsive to dynamic, unpredictable changes in demand, supply, pricing, labor, and competition.

**Variable** – flexible in adapting to variable cost structures and processes associated with productivity, capital, and finance.

**Focused** – concentrated on core competency, differentiated tasks and assets, with tightly integrated strategic partners.

**Resilient** – capable of managing changes and threats with consistent availability and security.

To attain these characteristics, companies need an operational environment that ties their business processes together.

### Building the On Demand Operating Environment

This broad integration and connectivity is accomplished through what IBM calls the On Demand Operating Environment, which helps businesses deliver greater value to their customer more quickly. The on demand operating environment has its own four characteristics:

**Integrated** – Far more than merely connecting disparate computing assets, the on demand environment must enable the integration of core processes and systems so that business can flow inside companies and across multiple enterprises.

**Open** – As the basis for integration, open technical interfaces and agreed-upon standards are essential.

**Virtualized** – Through an interconnected consolidation of computing infrastructure, grid computing will allow distributed computing resources to be shared and managed as a single large, virtual computer.

<sup>1</sup> For more information from IBM about e-business on demand,<sup>™</sup> go to:  
[http://www-3.ibm.com/e-business/index\\_fl.html](http://www-3.ibm.com/e-business/index_fl.html)

**Autonomic** – Similar to the way the human autonomic system manages our vital functions, the enormous complexity of tomorrow’s computing systems must rely on technology that manages itself.

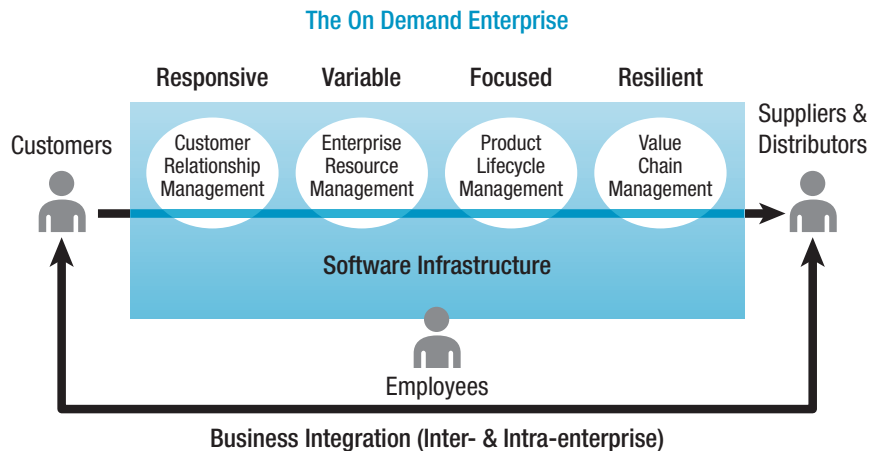
The benefits of this on demand operating environment are enormous. It facilitates the interoperation of systems, because business applications can be *integrated* according to *open* standards. And as businesses prepare their internal systems to support processes across the company, they can take advantage of a growing *virtualized* grid of computing resources that allows them to reach more customers farther and faster. Finally, the complexity of these systems will be managed *autonomically*, thus relieving much of the human-intensive activity traditionally associated with large-scale business computing.

**Software Applications and the On Demand Operating Environment**

The most common types of business applications at work in today’s companies drive a variety of management capabilities, including customer relationship management (CRM), enterprise resource management (ERM), product lifecycle management (PLM), and value chain management (VCM). These remain valuable resources in the on demand business. However, in the on demand era, these applications are not only integrated end-to-end across the company; they are also integrated with key suppliers and/or distributors outside the company, delivering greater business value to customers. (See Figure 1.)

***A company’s software applications are a key element in becoming an on demand business.***

**Figure 1:** In the on demand business, today’s most common types of applications are integrated end-to-end across the company, connecting both distributors and suppliers, as well as key customers.



This paper focuses on one part of the operating environment – the software applications that integrate and automate the business. While the on demand operating environment consists of much more than the software applications themselves, it is the application layer that ultimately integrates and automates the business; therefore, a company’s software applications are a key element in becoming an on demand business.

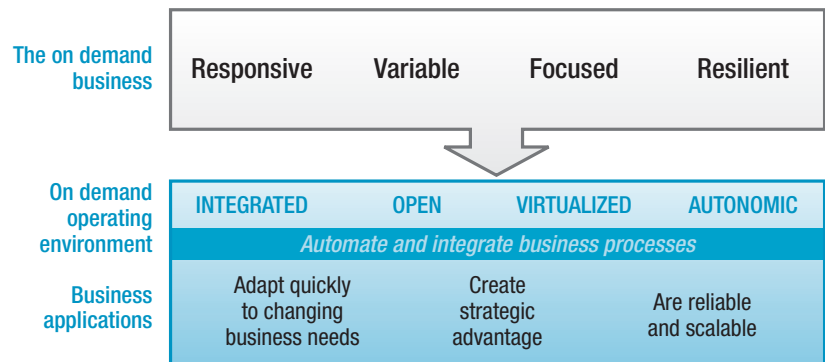
In the on demand era, what does a business need from its software applications? Business software applications must meet three requirements:

**Adapt quickly to changing business needs.** In the on-demand world, business requirements change more rapidly than in traditional businesses. As business conditions change – new opportunities emerge or new threats appear – a company must be able to modify their business applications even more rapidly to exploit the opportunity or defend against the threat.

**Capture and maintain a strategic advantage.** A business must leverage the unique qualities that differentiate itself in its market. To fully leverage those differentiators, on demand companies are leveraging the power of software.

**Are reliable and scalable.** An application only delivers value when it’s running and performing well. Critical applications must function continuously and flawlessly, even while the business is growing.

**Figure 2:** A key element in meeting the challenges of on demand business, the application layer integrates and automates business processes.





## II. Deriving business value from software applications: the need for software development

In the on demand era, businesses need to leverage all forms of applications to create competitive advantage. They need to maximize the value of their existing legacy systems, customize and deploy commercially available packaged applications, and build new, custom software and applications:

**Legacy applications** – Businesses can't afford to "rip and replace" legacy systems that still feature unique business rules and valuable custom logic. Their IT departments need to leverage what they have today by extending it, upgrading it, and exposing it to other applications – including today's ever-more-powerful database management technologies, middleware, and client-side applications.

**Packaged applications** – If businesses purchase commercially available software, they must ensure the applications support the essential, strategic demands of the business. While purchasing a packaged application can save a company time and resources, it is rarely the case that a purchased application can be deployed "out of the box" for any key business function. Businesses must customize and extend these off-the-shelf applications to support the core competencies of their business in competitive ways.

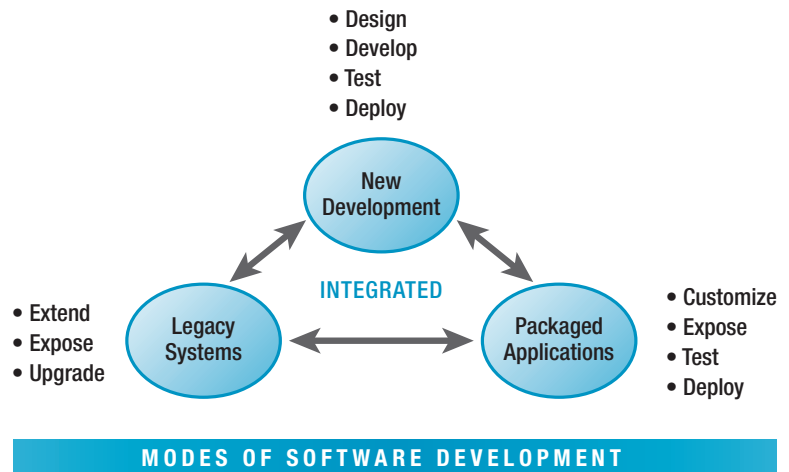
**New development** – Companies seeking true differentiation in the marketplace – especially those operating as on demand e-businesses – will need to develop new software systems that are unique to their business. That software must be designed, built, tested, and deployed, and it must drive essential business strengths.

All of these activities – extending legacy applications, modifying packages, and creating new applications – are forms of software development. In the on demand era, successful organizations will integrate and automate their business processes by integrating all of their applications. And an organization's ability to integrate their applications and thus leverage their value through customization, extension, and new development and deployment is a direct function of their software development capability.

***An organization's ability to integrate their applications is a direct function of their software development capability.***

In other words, for a business to succeed in the on demand era, a strong commitment to software development – including the skills associated with integration, customization, and new development – is essential.

**Figure 3:** To ensure that all applications are strategically valuable, adaptive, and reliable, businesses must customize, extend, and build them. Each of these types of applications require different modes of software development activity.



### III. Three imperatives for improving software development capability

For an organization’s software development capability to meet the needs of e-business on demand™, software development teams must embrace three specific “software development imperatives”:

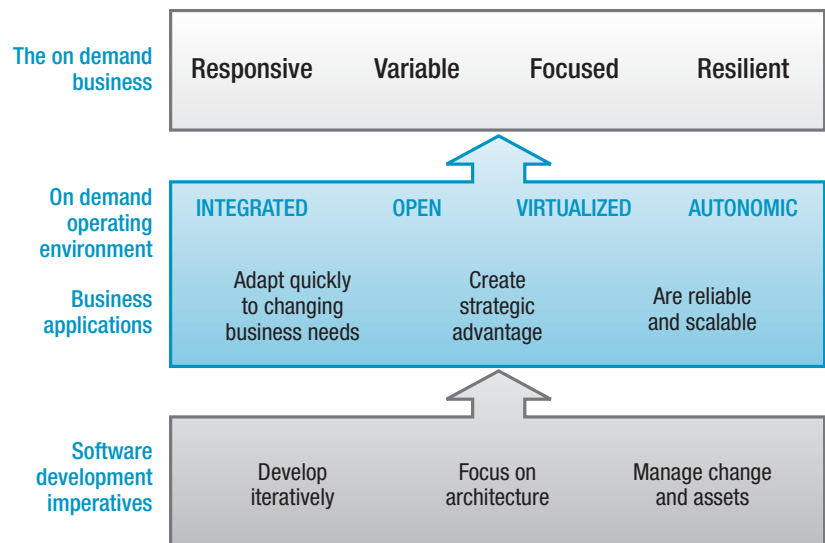
**Develop iteratively** – use a results-oriented process that yields increasingly improved iterations of a software system until it is ready for deployment.

**Focus on architecture** – base all software design on reusable components and a service-oriented model, which can be maintained, upgraded, or replaced without compromising overall system function.

**Manage change and assets** – track all changes to the software under development, manage the team’s activities, and protect the development assets that are strategic and unique to the business.

The relationship of these three software development imperatives to the applications and operating environment they support is shown in Figure 4:

**Figure 4:** Software development imperatives support adaptive, strategic, and reliable applications within the on demand operating environment.



#### IV. The development process for on demand software applications

As shown in Figure 4, e-business on demand™ is based on a logical relationship of 1) business fundamentals, supported by 2) an operating environment and business applications, supported in turn by 3) software development imperatives that drive the business applications. To express this relationship more simply:

To improve your business... ➡ exploit the power of software... ➡ ...by improving software development capability

But as a means toward improved software development capability, what is entailed in our three software development imperatives? If an organization is developing software, what, specifically, is required from development teams and processes? Let's examine each of the software development imperatives outlined in the previous section in more detail, and explore the benefits for the software which supports an on demand business.

***Each iteration verifies the system architecture, application requirements, and software quality.***

### **1. Develop Iteratively**

An iterative development process yields increasingly improved versions of a software system until it is ready for deployment. Each iteration includes a mix of analysis, design, construction, and testing, resulting in a demonstrable form of the software that can be validated and refined. Development teams start by addressing a limited set of functions pertaining to the high-risk areas of the project. With each iteration, teams add functionality and produce a working version of the application. Each iteration verifies the system architecture, the application's ability to satisfy its requirements, and the quality of the software.

By adopting an iterative development process, software development teams are able to keep the ultimate users of the application apprised of progress, since a working version of the system under development is always available (demonstrable) for inspection. This is an important concept: Software teams stay focused on results. Moreover, teams using a mature iterative development framework, such as the Rational Unified Process<sup>®</sup>, or RUP<sup>®</sup>, employ a flexible process – one that is customizable to any size project. This flexibility means teams can adjust their development process quickly as business needs change and requirements for new applications arise.

An iterative, “results-oriented” process also represents a significant advance over “activity-oriented” processes, such as the traditional “waterfall” method, which mandates that each phase of the development process – e.g., design – is finished before the subsequent phase – coding – begins, so the output of one phase “washes down” to the next. This traditional process also requires teams to work separately on different parts of the system until the final phases of assembly and system testing – the phases when mistakes are discovered, code must be scrapped and reworked, and disappointment becomes inevitable. (see Figure 5)

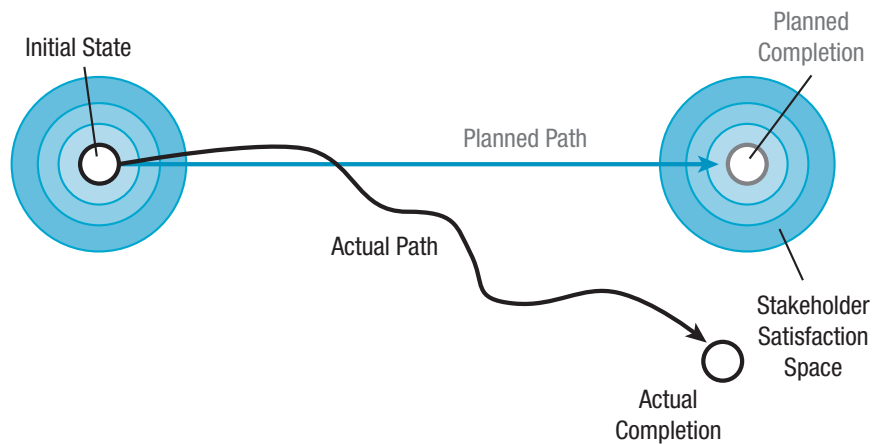
By contrast, an iterative development process offers the following primary benefits for building applications for the on demand operating environment:

**Reduces project risk** – i.e., the most difficult aspects of system performance and meeting end-user requirements are tackled early. Teams start with a limited set of requirements at the outset, which allows them to focus on the hard things first. With all major risks addressed and mitigated early on, fewer “show-stopper” mistakes are likely in the later phases of the project, and

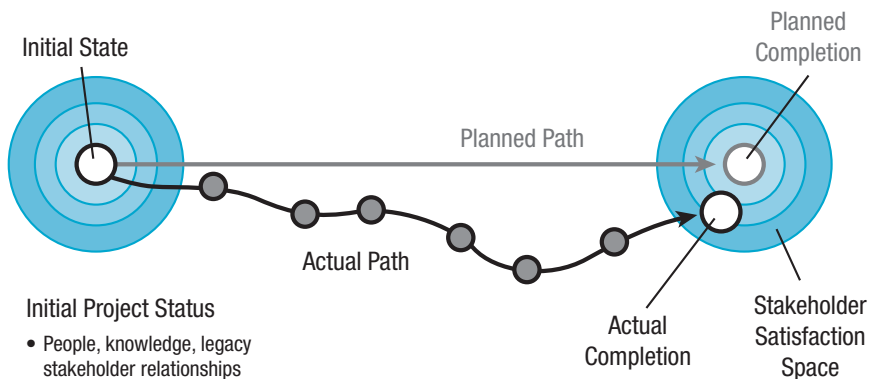
deadlines – which are no longer negotiable in on demand business – are more easily met.

**Increases predictability** – Using a traditional “waterfall” process, software development teams cannot fully assess product quality until the final stages of assembly, when a working version of the software becomes available. If errors are discovered at that late stage – and they always are – teams must either rework the application (incurring high costs and missing customer deadlines in the process), or deliver an application that does not meet customer expectations. With iterative development, teams can more accurately predict outcome because each iteration is designed, coded, and tested, and the team “zeros-in” on the correct result. If the project strays off-course at any point in the process, teams are better able to make adjustments in the next iteration. (See Figure 6)

**Figure 5:** In a traditional waterfall project, quality can only be assessed according to details of the plan, not product functionality, because the software isn't assembled and tested until the end of the project. At that point, teams typically start over and rework code, which means missed deadlines and high costs, or they deliver sub-optimal applications.



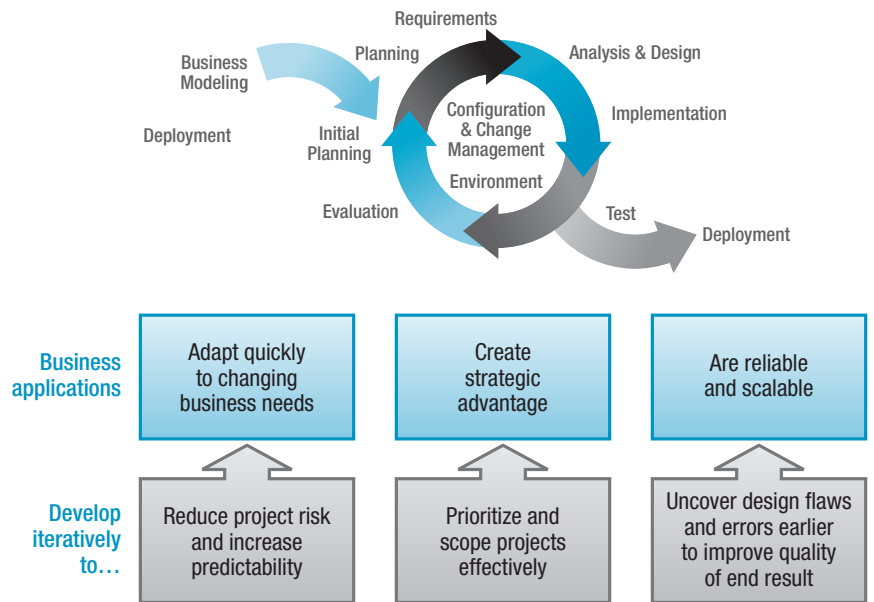
**Figure 6:** With an iterative process, each iteration (dark gray dot) is a working version of the software under development, which allows the development team to understand “where they are” in the process and make corrections as needed.



**Allows proper scope** – Iterative development allows project managers to scope each new software iteration with its incremental improvements, instead of scoping the entire project from its inception to completion. This ensures that the teams focus on the most important aspects of the application and ensure the end-result is a high-quality system.

**Reduces design flaws** – By addressing and mitigating the highest risks first, and by continuous verification of software quality through iterative testing, software projects – including integration efforts, new applications, and extensions to legacy systems – yield higher quality applications that are more reliable.

**Figure 7:** An iterative development process supports the major objectives of applications running in the on demand operating environment.



### **Iterative Development: The Benefits to Business Applications**

An iterative development process is essential in meeting the major objectives of applications running in the on demand operating environment. Because iterative development reduces risk and increases predictability, software development teams are better able to quickly adapt applications – whether these are pre-existing, pre-packaged, or new applications created by the business – to meet rapidly changing business needs. Thus return on application investment is more rapidly achieved. Because an iterative process allows teams to better manage the scope of their project, organizations can ensure they are delivering applications that meet the most strategic, high-leverage aspects of their business. And with reduced design flaws at the end of an iterative project, IT managers can be much more confident that the applications they deploy will remain reliable and scalable for growth over time. Figure 7 shows the relationship between iterative development and the major objectives of applications running in the on demand operating environment.

***For an on demand business, software architecture becomes doubly important as the business adapts its applications to meet changing needs.***

### **2. Focus on Architecture**

An application's architecture is the most critical determinant of its success or failure. A properly designed architecture ensures the application will meet the business needs, perform acceptably, scale with the business, and be adaptable over time. A poor architecture results in applications that are inflexible, fragile, unreliable, and expensive.

For an on demand business, software architecture becomes doubly important as the business adapts its applications to meet changing needs. A solid, well-understood architecture provides the foundation for the critical twenty percent of all artifacts (requirements, components, etc.) that will drive the overall success of a new project. It allows on demand development teams to address the high-risk requirements before worrying about the complete breadth and depth of the software details. This results in far less scrap and rework over the project life cycle, which means faster response times to on demand business challenges.

A proper architectural focus has three key dimensions: component-based design, software-oriented architectures, and visual modeling.

A component is a cohesive set of pre-existing lines of code, either in source or executable format, with a defined interface and behavior. A software architecture based on components with well-designed interfaces allows more rapid change, because any of its components can be modified without affecting the rest of the application. A rapidly growing trend in software architecture design is the “Service-Oriented Architecture” (SOA) model. SOAs not only allow companies to integrate their internal systems according to component-based design principles, but also allow outside customers access to selectively exposed functionality. SOAs will help companies build and integrate the software that transforms them into on demand businesses.

Designing an effective architecture can be a challenging task that typically involves a number of participants. Effective communication, unambiguous definition, and a reliable means for capturing and modifying the design are critical. The industry has produced a standard – the Unified Modeling Language (UML) – to satisfy these needs. Using the UML, teams can produce an accurate, graphical representation of an application’s architecture. These models can be shared with others who will immediately understand them because of the precise definitions of the UML. The quickest and most effective way to produce these models is to use a visual modeling tool. In today’s complex world, a visual modeling tool is essential to building a quality architecture. They are far more than just drawing tools; they can directly generate code and other artifacts from visual models, thus eliminating human error and increasing productivity.

For the on demand era, a focus on architecture yields the following important benefits:

**Design for change** – Component-based architectures, including emerging service-based architectures (SOA) designed to support Web Services and grid computing, allow the software to be changed quickly, because components can be switched readily, or modified, without compromising overall system integrity. This is a fundamental requirement to support responsiveness and resiliency for the on demand business.

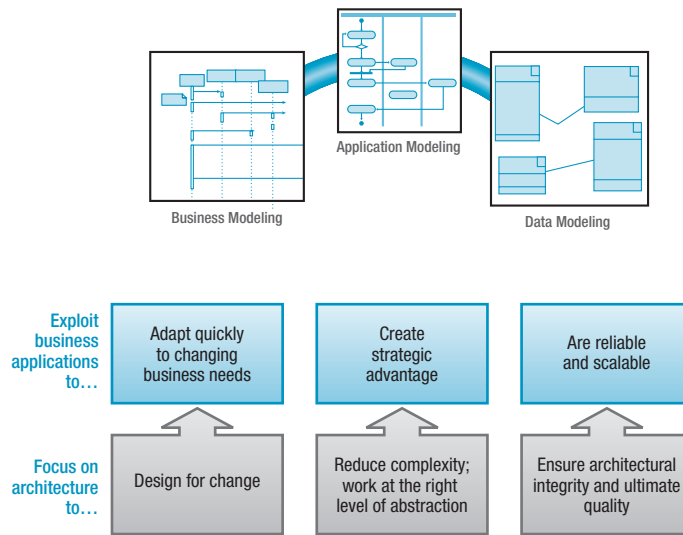
**Reduced complexity** – Visual modeling allows project managers to focus on the core requirements of a software application, thus leaving the functional details to the coding specialists on the team. Component-based architectures allow interactions to occur between the various parts of a system without



requiring the exposure of methods and details between all functional “chunks of code.” Together, visual modeling and component-based design allow all teams to work at the right level of “abstraction” – i.e., the level of complexity best suited to their role in the project.

**Integrity and quality** – Experience shows that an application’s architecture is the single largest determinant of its quality. A good architecture allows the application to be modified over time; applications with poor architectures will decay as they are modified due to their brittle and rigid designs.

**Figure 8:** A focus on architecture drives key benefits for applications running in the on demand operating environment.



**Architectural Focus: The Benefits to Business Applications**

An architectural focus supports the major objectives of applications running in the on demand operating environment. Because a well-designed architecture allows applications to be designed for change, they are able to adapt more quickly to the rapidly evolving needs of the business. Applications that can be rapidly changed with high confidence allow the organization to focus on its strategic differentiators as they evolve in the competitive business arena, ensuring that organizations get the highest ROI from their business applications. And because good architecture improves an application’s integrity and quality, they naturally become more reliable and scalable. Figure 8 shows the relationships between a strong architectural focus and the major objectives of applications running in the on demand operating environment.

***A managed response to change enables successive iterations to run smoothly.***

### **3. Manage Change and Assets**

Embedded in the very concept of e-business on demand™ is the notion of change. Quick response to new opportunities, customer demands, or threats to security or overall business stability is an essential, defining capability in the on demand era. But “responsiveness” in terms of on demand software development does not mean “reactionary.” Instead, a managed response to change, which also guards against corruption of assets created during the development process, enables project managers to keep the process of successive iterations running smoothly. Software configuration management (SCM) is a key capability in modern software development practice. It allows teams to carefully trace requirements over the project lifecycle, during which numerous changes – including changes to the requirements themselves – occur.

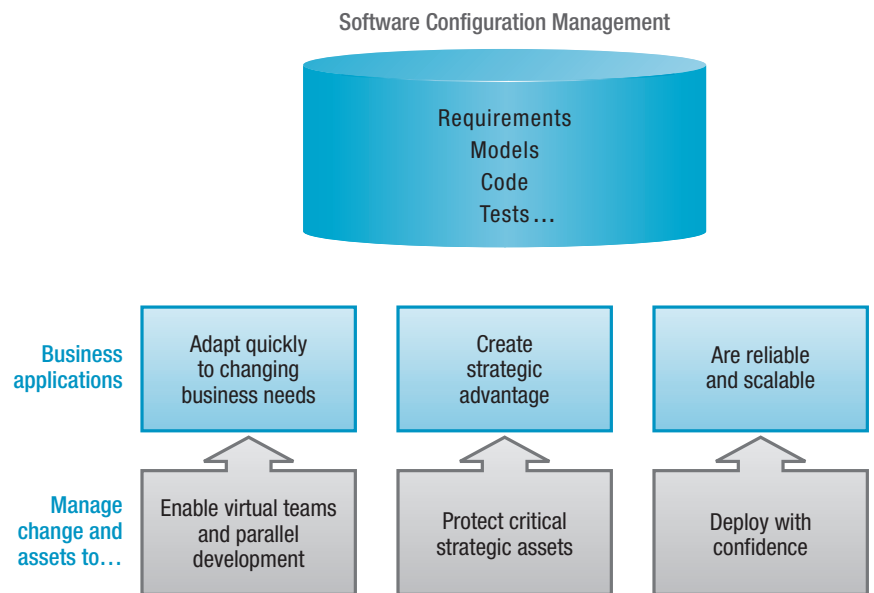
Change and assets management offer the following key advantages for on demand software development:

**Enables virtual teams and parallel development** – Advanced SCM systems allow multiple, sometimes overlapping, branches of a project to be worked on by different development teams simultaneously, so more work can be accomplished faster, on demand, without sacrificing quality.

**Protects critical assets** – A company’s software development assets – requirements documents, design models, source code, automated test suites, etc. – are unique, strategic resources that cannot be purchased or recreated from outside sources. Just as valuable as a corporation’s business assets, these software development artifacts must be managed and protected. Effective change management systems ensure that no unit of code or component under development is ever lost or over-written. This affords an important safeguard against the threat of security breaches or disaster.

**Allows confidence in software deployment** – Change and assets management ensures that teams who are building and maintaining complex systems remain in sync as they combine multiple versions and various pieces of software code. Change management systems also allow all requirements to be traced throughout the project lifecycle, so that the high-level architecture translates to a software system focused on user expectations.

**Figure 9:** Managing changes and assets created during the software development process supports the major objectives of applications running in the on demand operating environment.



**Change and Assets Management: The Benefits to Business Applications**

Managing changes and assets during the software development process is required to meet the major objectives of applications running in the on demand operating environment. Virtual teams and parallel development capability means faster project turn-around times to meet rapidly changing business needs. Because the software development assets are as strategic to the business as the applications they create, protecting and managing them is central to delivering strategic value. And greater team confidence in deployed software speaks for itself. Figure 9 shows the relationships between change and asset management and the major objectives of applications running in the on demand operating environment.

## V. Supporting the on demand operating environment

As described in Section I., IBM defines the four characteristics of the on demand operating environment as *integrated*, *open*, *virtualized*, and *autonomic*. Software development teams need to target their efforts to building this new environment, to help businesses deliver greater value to their customer more quickly.

Of these four characteristics, effective software development will enable development teams to deliver *integrated* and *open* environments. As *virtualized* (grid computing based) and *autonomic* (self-managing) systems become common aspects of tomorrow's business computing platforms, organizations whose operating environments are integrated and open will be best prepared for the advantages of the complete, on demand environment.

This section will focus on the importance of integrated business applications and open standards for on demand customers and businesses.

### 1. Integrated

A major step in the evolution of e-business was the integration of disparate computing capabilities with the Internet, so that customers could begin conducting real business via the Web. That has been, and remains for many companies, a relatively straightforward form of software integration to enable basic transactions with their customers. The integration requirements to support on demand computing are considerably more complex. But the benefits are enormous.

***Service-Oriented Architectures will ultimately reduce the high cost of integration efforts.***

Today, forty percent of all IT spending is targeted at integration efforts – simply making things work together. This is the result of years of accumulated assets, both legacy systems and custom applications, all of which must be choreographed to enable increasing consumer and intra-business demands. For on demand computing, software developers must adopt new modes of integration in order to reach customers faster and more cost effectively. These will include new techniques such as Web Services and Service-Oriented Architectures that will ultimately reduce the high percentage of integration effort, and redirect development energy toward delivering greater business value.

There are three major areas of integration where software development organizations must increase their expertise: service-oriented architecture, enterprise architecture, and heterogeneous integration.

**Service-oriented architectures (SOA)** – SOA is the next step in component-based development. As a key enabler of the growing Web Services trend supporting integration efforts, SOAs not only allow companies to integrate their internal systems according to component-based design principles, but also allow outside customers access to selectively exposed functionality. SOAs will help companies build and integrate the software that transforms them into on demand businesses.

**Enterprise architecture** – An enterprise architecture (systems architecture, etc.) offers companies an understanding of what they have, where redundancies exist, and how best to share information over an optimized architecture. The systems architect's job is to understand how the pieces of a system interact, to know its stress points, to understand its boundaries and the business problems that the system is designed to address. Ideally, the systems architecture determines the organization of the teams charged with building and maintaining the system. In turn, development project teams become more motivated as they are assigned greater responsibility and accountability for specific portions of the architecture.

Supporting large systems integration projects, systems architecture integrates software design with the relevant part of the business. This allows software development organizations to add new value to their efforts through a disciplined, engineering approach to systems development.

**Integration in a heterogeneous world** – As noted earlier, businesses in the on demand era must integrate their systems to share data across the enterprise, and with their partners and key customers. This requires more than the integration of software applications, but a new understanding of an end-to-end architecture which integrates the business appropriately with the outside world. To do this:

- *Requirements must be well-understood and managed* – In many cases, systems integration projects present the most demanding requirements that development teams can face, since individual systems were designed for different purposes, via different tools and languages. Understanding

the high-level business value of the integration as well as the technical means for accomplishing it is typically a significant challenge.

- *Integrations must be based on a sound architecture* – Just as single applications should be built for responsiveness and resiliency in the face of change, integrations must be designed for future changes.
- *Results* must be of consistently high quality, verified by automated testing.
- *System integrations must be based on standards* that support heterogeneous environments (see the “Open” section below).

***Even technologies for strategic differentiation must be designed for future integration and interaction with other systems.***

## **2. Open**

The adoption of open standards has grown exponentially since the explosion of the World Wide Web required, for instance, every desktop to run some form of browser displaying HTML. As more companies embrace the era of on demand computing, open computing standards will provide the oxygen for all forms of integration (discussed in the previous section) to move forward. Successful on demand businesses will depend on open software development standards as a means for internal and external integration, aimed at attracting and retaining customers in the global marketplace.

The use of open standards does not mean that every company uses the same technology. Far from it. It *does* mean that customers will demand that transactions and data exchange over e-business networks can be processed and shared as expected by other systems. Even the technologies developed for strategic differentiation must be designed with future integration and interaction with other systems in mind.

Three trends indicate the growing importance of open computing standards, and point the way for companies seeking to adopt an on demand operating environment:

**The Unified Modeling Language** – A widely used standard, the UML is a method of visual notation for capturing strategic and tactical decisions in ways that human beings and software development tools can interpret. As a standard language understandable to both humans and machines, the UML eliminates the confusion that is inevitable among team members from different cultures, backgrounds, and spoken languages. The result? Software teams work more efficiently, and design decisions can be made in concert with customer input with greatly reduced ambiguity.

**Service-Oriented Architecture** – Noted earlier for its growing role in the Web Services arena, SOA is equally important as a structure which embraces multiple standards, including XML, Java, SOAP, .NET, WSDL, etc. All of these standards allow businesses to create Web Services for internal, and eventually external, integrations and interoperability among vendors, partners, and their customers.

**Component-Based Design** – Long considered an important feature of object-oriented software design, components and component reuse will remain the cornerstones of an on demand software architecture. Standard component models facilitate application design and interoperations.

## VI. Software development solutions from IBM Rational

The three software development imperatives described in Section IV – develop iteratively, focus on architecture, and manage change and assets – are the foundation of a multifaceted, integrated solution from Rational software, formerly an independent organization and now one of the IBM software brands. IBM Rational has maintained a consistent mission over the past 20+ years: “to ensure the success of customers whose businesses depend on developing and deploying software.” Because of the essential role software development plays in an on demand business, IBM acquired Rational in February 2003.

### Customers using tools, best practices, and services from IBM Rational report extraordinary improvement in ROI:

- 33% reduction in development costs
- \$9.5M Net Quantifiable Benefit
- 1440% ROI
- \$409K Net Quantifiable Benefit
- 222% ROI

From architectural modeling languages to emerging Web services standards, IBM Rational software plays an active role in shaping the future of software development practices and technology. IBM Rational participates in numerous standards committees, employees have authored over 50 publications, and thought leadership is showcased at dozens of industry conferences each year where IBM Rational employees are frequent keynote speakers.

This section describes the major offerings in software engineering best practices, development tools, and professional services that Rational brings to the IBM software portfolio. It also describes the unique capabilities of these offerings in supporting the integrated and open requirements of the on demand operating environment.

### The Software Development Platform for an On Demand World

The Rational platform for software development is the industry’s leading integrated solution for designing, developing, testing, and deploying software. The Rational platform combines software engineering best practices, market-leading tools, and expert professional services. By combining all three, software development teams can drive rapid and continuous improvement in software development capability.



**1. Best Practices and the Rational Unified Process®: Process Made Practical**

At the core of all IBM Rational software development tools are best practices harvested over the course of 20 years of collaboration with software development organizations. By applying these disciplines, software development teams establish a predictable development process that is relevant, targeted, and up-to-date. These best practices are:

**Develop Iteratively** – to identify and eliminate risks before they threaten your project

**Manage Requirements** – to ensure resilience in the face of inevitable change

**Use Component Architectures** – to make your architecture tangible to all practitioners

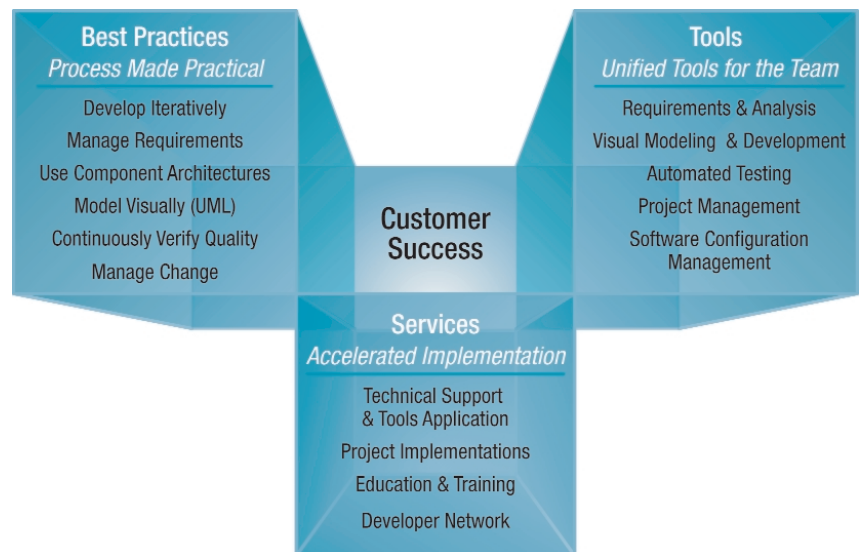
**Design and Develop Visually** – to attain and preserve a high-quality architecture

**Continuously Verify Quality** – to ensure quality throughout the development life cycle

**Manage Change** – to enable efficient parallel development within teams and across the enterprise.

These best practices are captured in the Rational Unified Process, which has rapidly become an industry de facto process framework standard.

**Figure 10:** The Rational platform combines software engineering best practices, market-leading tools, and expert professional services.



**Customers claim reductions in time to value:**

- 66% reduction of development cycle time (9 to 3 months)
- 2 month manual testing cycle reduced to 2 days
- Test time reduced from 6-8 testers taking several days to 1 tester taking a few hours
- 66% reduction in release cycles

**2. Automating the Development Life Cycle: Unified Tools for the Project Team.**

Rational Tools provide full life-cycle support for both individual contributors and the extended development team. These innovative, award-winning tools feature unprecedented integration to automate tedious tasks, accelerate workflow, and speed development cycles. Rational's technologies map to five major areas of the software development life cycle, as described below:

**Requirements and Analysis** – For defining a system under development, and maintaining those requirements throughout the project lifecycle.

**Visual Modeling and Development** – For understanding and designing components and supporting architecture that will define the application under development.

**Automated Testing** – For quality assurance at every phase of the development life cycle.

**Project Management** – For keeping high-quality products on time and on budget, including process guidance, project reporting, and progress tracking.

**Software Configuration Management** – For managing the production and modification of files, directories, components, and systems, plus tracking software defects and corrections.

### **3. Services: Accelerating Implementation**

As expressed in a variety of ways throughout this paper, software development expertise is critical to the success of on demand computing. To make sure development teams have the skills to stay competitive, teams need technical support, consulting, training, and access to a network of expert advice. With an emphasis on knowledge and skills transfer, services from IBM Rational encompass over 1,000 technical professionals, more than 20 years of experience, and a wealth of online training and knowledge resources.

**Rational Technical Support** – For teams to get fast answers to immediate problems with one set of support contacts and one phone number to call anytime.

**Rational Professional Services** – For accelerating any phase of a development project, with customized projects tailored by Rational consultants who focus on knowledge and skills transfer to improve on organizations's software development capability and speed their proficiency.

**Rational University** – For training at dozens of worldwide campuses or onsite at customer locations, including more than 60 courses on the full software development life-cycle.

**Rational Developer Network** – For online technical information, including online training, licensed and original content, and an online community of peers.

**Customers report improved project management capabilities:**

- Effectively managed 125 project builds in one year
- Enabled growth from 80 to 280 developers in 2 years
- Successfully developed 3+ product versions in parallel
- Effectively managed projects spanning 3 countries, 4 internal sites, and more than 300 developers and testers

**4. Supporting the On Demand Operating Environment**

*How does IBM Rational help on demand businesses achieve an Integrated Operating Environment?*

The three software development imperatives described in this paper – develop iteratively, focus on architecture, and manage change and assets – serve as critical guides for organizations embarking on the integration of their business systems across the company.

**Service-oriented architecture** – IBM Rational has contributed considerable thought leadership in this area and other areas associated with the growing trend in Web Services implementation. Of particular note is IBM Rational's contribution to several standards bodies, including the Web Services Interoperability (WSI) organization; the Universal Description, Discovery and Integration (UDDI) project; and the Reusable Asset Specification Consortium.

**Enterprise and software architectures** – IBM Rational's leadership and capabilities regarding enterprise architectures helps all customers plan and better architect their systems. Rational Software was the key driving force behind the creation of the industry standard UML. Today, the IBM Rational approach to system architectures is holistic – it provides a unified process along with integrated tools – and spans several areas previously discussed in this paper: requirements management; visual modeling and the Unified Modeling Language (UML); the Rational Unified Process; automated testing; and configuration and change management. In each of these areas IBM Rational offers advanced tools and consulting capabilities.

**Support for a heterogeneous world** – IBM Rational maintains strategic partnerships with all major platform vendors and supports a variety of technologies and standards so customers can integrate – and share information – across systems. IBM Rational provides support for Java, J2EE, .NET, Linux, SOAP, XML, WSDL, embedded operating systems and languages, C, C++, etc.

**Customers claim tremendous increases in productivity:**

- 90% reduction in bug backlog
- 300% productivity increase
- 100% productivity increase
- 90% reduction in bug backlog
- 400% productivity increase
- 97% productivity increase
- 1200% increase in developer productivity

*How does IBM Rational help on demand businesses achieve an Open Operating Environment?*

With its long history in developing and supporting the goals of open computing, IBM Rational has earned worldwide leadership in the field of software development. IBM Rational has partnered for years with the major players in computing platforms and computing standards, as well as a wide variety of consortiums who maintain standards specifications. In addition, IBM Rational serves as an active member in numerous standards bodies.

**The UML**, developed by IBM Rational, is now a standard language maintained by the Object Management Group (OMG), a not-for-profit consortium that produces and maintains computer industry specifications for interoperable enterprise applications.

**Promotion and development of open standards** – IBM Rational not only supports and drives the usage of open standards in its products, but actively participates in a variety of standards bodies. Standards body participation includes the Business Process Management Initiative; the Eclipse.org Consortium; the Institute of Electrical and Electronics Engineers, Inc.; the Internet Engineering Task Force; the Java Community Process; the Organization for the Advancement of Structured Information Standards; Object Management Group; Reusable Asset Specification Consortium; Universal Description, Discovery, and Integration project; World Wide Web Consortium; and the Web Services Interoperability Organization.

**Support for Service Oriented Architecture** is a natural extension of IBM Rational's long-held commitment to component-based design. Now, service-oriented architectures are offering a more economical alternative to cross-system integrations than any single component model architecture can provide, especially as more business seek to provide access to customers and partners outside the firewall. This means that software development organizations committed to one component model or another can leverage their existing software, even continue using a given model format for component development, and still reap the benefits of increasingly popular SOAs as they expose selected functionality within their business systems.

**In all areas of the software development life cycle, our customers claim improvements:**

- Automatically generated 750,000 lines of code
- 80% fewer bugs
- Improved visibility, traceability, and predictability
- Overall testing time reduced by 50%-75%
- Effectively managed 4 million lines of code
- 30% reduction in required documentation
- Improved communication and collaboration of 7000 consultants across 70 countries

*For details on IBM Rational customers reporting these results, go to [www.rational.com/success](http://www.rational.com/success).*

Over its 20 year history, IBM Rational has designed support into its tools for most major standard languages, from Ada to Visual Basic. An extensive set of Application Programming Interfaces (APIs) allows third-party and customer-built tools to interoperate with tools from IBM Rational. And support for multiple operating platforms allows heterogeneous software development organizations to use open standards for connecting applications across the enterprise.

## Conclusion

For a radical transformation of business systems to succeed, technology managers need a clear understanding of the business goals their applications and operating environment must support. The IBM on demand vision is compelling, and its achievement will transform not only business systems, but also the ways in which customers and partners across the globe interact with corporations and with each other. Every industry will benefit, with higher quality, more affordable products and services available worldwide.

To leverage the full potential of e-business on demand,<sup>™</sup> the leaders of software development organizations must incorporate proven principles that will ensure their success in the on demand era. That means they must focus on three imperatives for successful software development:

- **Develop iteratively**
- **Focus on architecture**
- **Manage change and assets**

Thousands of companies around the world have realized the benefits of this approach. Their processes are results-oriented, the artifacts they produce are well-designed and reusable, and they are working at higher levels of capability than ever before. These companies are delivering the promise of the on demand era.



### IBM software integrated solutions

IBM Rational supports a wealth of other offerings from IBM software. IBM software solutions can give you the power to achieve your priority business and IT goals.

- *DB2® software helps you leverage information with solutions for data enablement, data management, and data distribution.*
- *Lotus® software helps your staff be productive with solutions for authoring, managing, communicating, and sharing knowledge.*
- *Tivoli® software helps you manage the technology that runs your e-business infrastructure.*
- *WebSphere® software helps you extend your existing business-critical processes to the Web.*
- *Rational® software helps you improve your software development capability with tools, services, and best practices.*

### Rational software from IBM

Rational software from IBM helps organizations create business value by improving their software development capability. The Rational software development platform integrates software engineering best practices, tools, and services. With it, organizations thrive in an on demand world by being more responsive, resilient, and focused. Rational's standards-based, cross-platform solution helps software development teams create and extend business applications, embedded systems and software products. Ninety-eight of the Fortune 100 rely on Rational tools to build better software, faster. Additional information is available at [www.rational.com](http://www.rational.com) and [www.therationaledge.com](http://www.therationaledge.com), the monthly e-zine for the Rational community.

© Copyright IBM Corporation 2003

IBM Corporation  
Software Group  
Route 100  
Somers, NY 10589  
U.S.A.

Printed in the United States of America  
01-03  
All Rights Reserved

IBM, Rational, the e-business logo, e-business on demand, the Rational Unified Process, RUP, the IBM logo, and the Rational logo are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Other company, product and service names may be the trademarks or service marks of others.

The IBM home page on the Internet can be found at **[ibm.com](http://ibm.com)**

♻️ Printed in the United States on recycled paper containing 10% recovered post-consumer fiber.