



**Building the  
On Demand Business:  
Four Imperatives for  
Improved Software  
Development**

Mike Perrow  
IBM Rational Communications  
IBM Software Group

## Executive Summary

One of the major factors in determining a company's success in the on demand era is its software development capability. New business needs and consumer expectations that define e-business on demand™ will require a higher level of responsiveness and agility from software development teams. 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, scalable, and manageable 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 four 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** – Teams must create a sustainable architecture that enables development to react to new business requirements without breaking the systems that are already in place. This requires reusable components and service-oriented architectures that can be maintained, upgraded, and replaced without compromising overall system function. A sustainable architecture allows applications to be designed for change, with reduced complexity and integrity.

**3. Continuously ensure quality** – Teams must ensure application fitness at each stage of the development lifecycle. Responsibilities include a sensitivity to the validity and accuracy of evolving requirements, a rigorous commitment to architectural and code quality, and thorough test planning and test execution. Once applications are deployed, teams must continue their vigilance to ensure rapid reaction and proactive, responsive customer service.

**4. 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 offers a comprehensive software development solution based on these four imperatives. This solution 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 extensive experience in promoting and delivering *integrated* and *open* software systems, both of which are key characteristics of the on demand operating environment, -- the means by which business increase their flexibility and IT strengths.

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 Driving business value through software  
applications: The need for software  
development**

**10 Four imperatives for improving software  
development capability**

**11 The development process for on demand  
software applications**

**22 Software development solutions from IBM  
Rational**

**28 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 processes 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.

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. They must create 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 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 build new applications, integrate, modernize, and extend existing ones, and deploy these applications more quickly than ever before.***

To get there, organizations are leveraging the power of software. They must be able to build new applications, integrate, modernize, and extend existing ones, and deploy these 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 an emerging on demand organization 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 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, a means to integrate and automate their business processes. But to differentiate your business from those you compete with, these processes cannot simply be based on “out of the box” capabilities provided by packaged applications. Instead, each business must customize their systems in concert with their unique competitive strengths. The benefits of this customization are many. Primarily, it is the key to achieving strategic, competitive advantage over companies locked into vertical, siloed business processes.

---

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

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

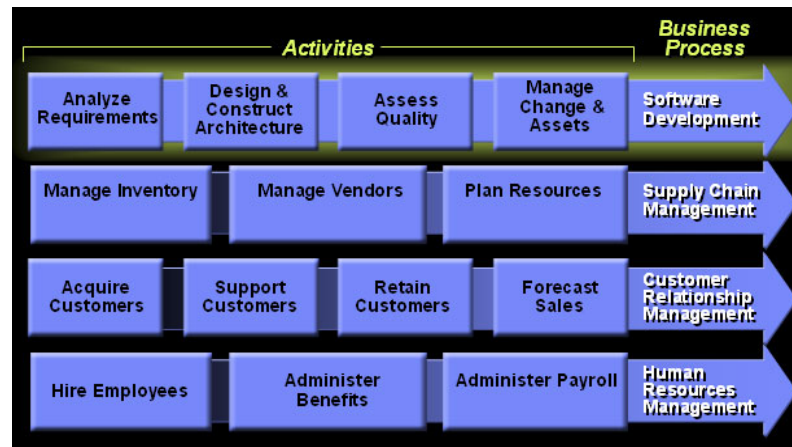
**Software development: Essential to the on demand enterprise**

The most common types of business applications at work in today's companies drive a variety of strategic capabilities, including customer relationship management (CRM), supply chain management (SCM), and human resource management (HR). These remain valuable resources in the on demand business. In fact, in the on demand era, these applications are 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. And the key to creating these valuable, customized integrations is *the capability of a business's software development organization.*

In fact, not only is software development key to integrating and automating business processes, but it is a strategic business process in itself. As such, the process of software development benefits from the same type of horizontal integration long applied to supply chain management, customer relationship management, and human resources management processes.

When we compare software development with other business processes, we see that each is composed of activities which were once considered distinct, and supported by different IT system. Over time, business integration has evolved to the point where these activities are now recognized as components of a single, horizontally integrated business process supported by a single integrated application: HR, CRM, etc. (See Figure 1.)

**Figure 1:** Just as important to running a competitive business as CRM, SCM, and HR systems, software development itself is a strategic business process -- "A collection of activities that takes in one or more kind of input and creates an output that is of value to the customer."<sup>2</sup>



<sup>2</sup> Source: Hammer & Champy, *reengineering the Corporation - A Manifesto for Business Revolution*, p. 93

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

Software development is at the beginning of this transformation. Companies are increasingly realizing the value of an integrated software development platform in improving the efficiency of interrelated software development activities. With the introduction of IBM® Rational Suite® software development tools in 1999, IBM was the first to provide this integrated software development solution, and we remain the industry leader.

Let's examine the role that software applications play in integrating and automating a competitive business. While the on demand business consists of much more than its software applications, the applications ultimately integrate and automate essential business processes. How, then, do a company's software applications help it to become an on demand business? Essentially, business software applications must meet the following requirements:

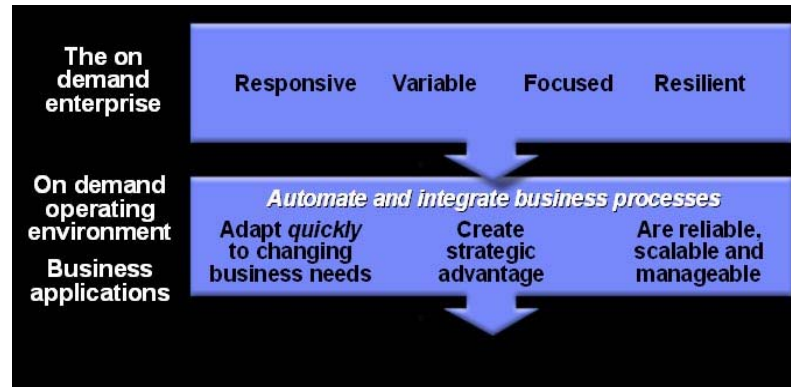
**Adapt quickly to changing business needs.** In the on-demand world, business requirements change more rapidly than in traditional businesses. As business conditions change — new opportunities emerge or new threats appear — a company must be able to modify their business applications even more rapidly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

**Remain reliable, scalable, and manageable.** 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, and they must allow for routine maintenance -- such as software upgrades and updates -- as technology advances continue.



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



## II. Driving business value through software applications: The need for software development

What kinds of software systems are we talking about? 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 offer unique business rules and valuable custom logic. Their IT departments need to leverage what they have today by modernizing, extending, upgrading, and exposing tried and true business logic to other applications — including today's ever-more-powerful database management technologies, middleware, and client-side applications.

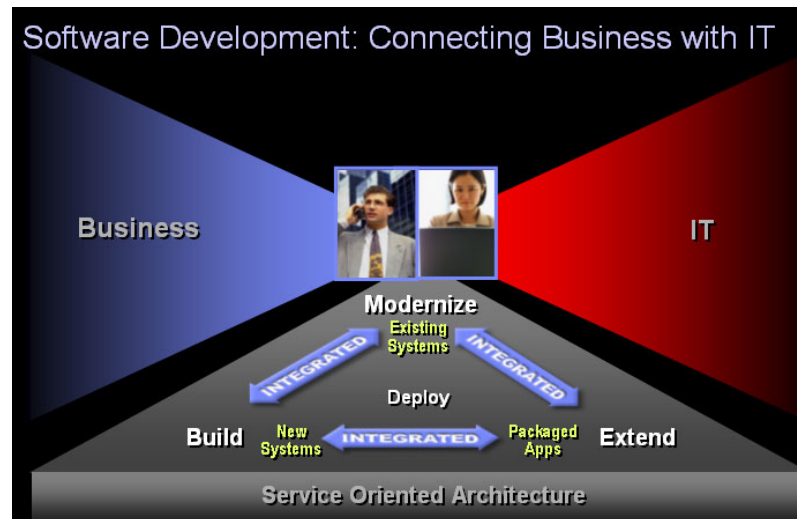
**Packaged applications** — If businesses purchase commercially available software, they must ensure those applications support the essential, strategic demands of the business. While purchasing a packaged application can save a company time and resources, it is rarely the case that a purchased application 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 automate and integrate essential business strengths.

All of these activities — modernizing legacy applications, extending packages, and building 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 modernization, extension, and new development and deployment 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 connected to business value, software development organizations must modernize, extend, and build them. Each of these types of applications require different modes of software development activity.



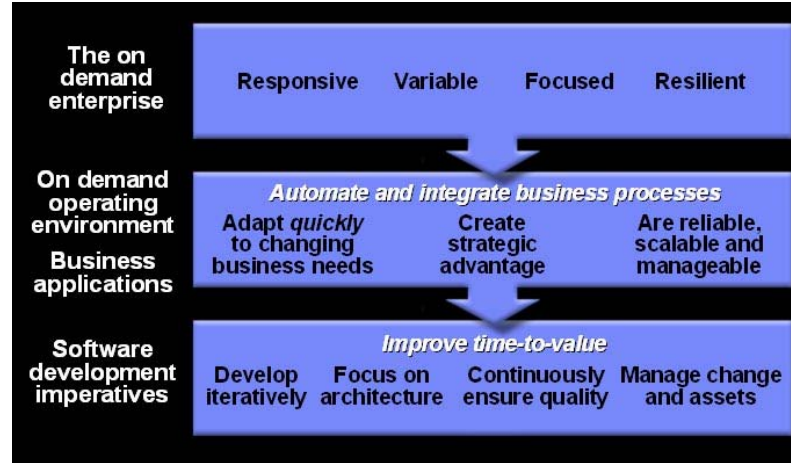
### **III. Four imperatives for improving software development capability**

For an organization's software development capability to meet the needs of e-business on demand, software development teams must embrace four specific "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** — Teams must create a sustainable architecture that enables development to react to new business requirements without breaking the systems that are already in place. This requires reusable components and service-oriented architectures that can be maintained, upgraded, and replaced without compromising overall system function. A sustainable architecture allows applications to be designed for change, with reduced complexity and integrity.
- 3. Continuously ensure quality** — Teams must ensure application fitness at each stage of the development lifecycle. Responsibilities include a sensitivity to the validity and accuracy of evolving requirements, a rigorous commitment to architectural and code quality, and thorough test planning and test execution. Once applications are deployed, teams must continue their vigilance to ensure rapid reaction and proactive, responsive customer service.
- 4. 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.

The relationship of these 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 capability.

But as a means toward improved software development capability, what is entailed in these 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, which results 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® methodology, or RUP®, employ a flexible process — one that is customizable to any size project. This flexibility means that 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 — and these phases are when mistakes are discovered, code must be scrapped and reworked, and disappointment becomes inevitable (see Figure 5).

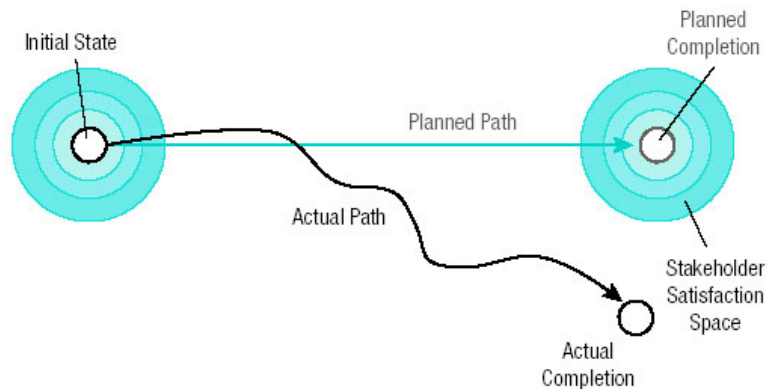
By contrast, an iterative development process offers significant benefits for building applications for the on demand operating environment, as follows:

**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

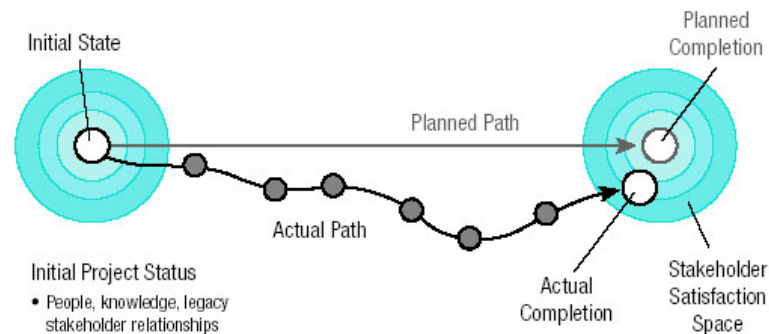
risks addressed and mitigated early on, fewer “show-stopper” mistakes are likely in the later phases of the project, and deadlines — which are no longer negotiable in on demand business — are more easily met.

**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.



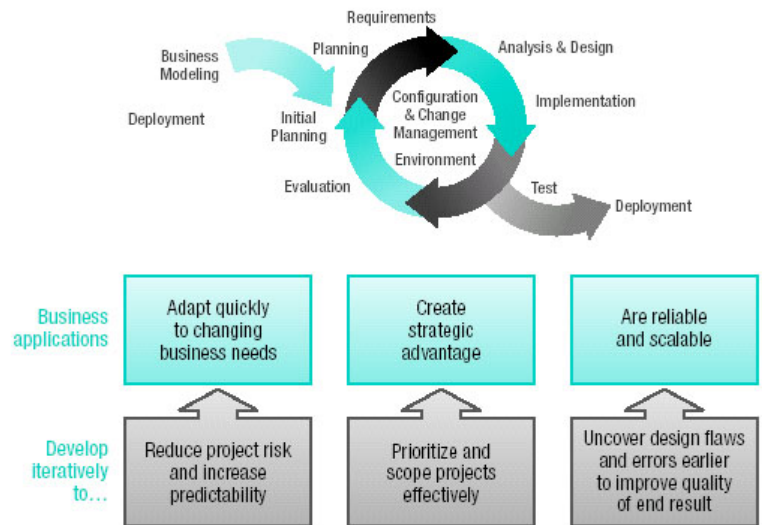
**Figure 6:** With an iterative process, each iteration (dark gray dot) is a working version of the software under development, which allows the development team to understand “where they are” in the process and make corrections as needed.



**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.



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

### **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.

### **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 to maintain.

Software architecture becomes doubly important as a company seeking to become an on demand 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 project. It affords a stable development environment so that teams are able to add new functionality over time without risking breakage of the overall system. It also allows for changing various components within the development environment as new technologies need to be added and as old ones need to be replaced.

For software development teams, a proper architectural focus allows the benefits of component-based design, including service-oriented architectures, which helps teams achieve higher levels of code reuse, project



after project, than ever before. Moreover, they can make better use of rapid application development techniques within a more structured production environment.

How does sound architecture afford these benefits?

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 allow companies to integrate their internal systems according to component-based design principles. As these same companies design and build systems for their customers and partners, SOAs allow existing components to be reused with selected functionality exposed to inside and outside parties.

Without a sound architecture to back them up, teams using rapid application development (RAD) techniques typically produce limited results. However, an “architected” RAD (ARAD) approach provides a unified environment for all team members that allows integration of legacy and enterprise systems and development of scalable n-tiered applications. Teams are able to take advantage of templates and patterns for industry based solutions, as well as foundation classes that help jump start development projects.

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 precision in UML notation. The quickest and most effective way to produce these models is to use a visual modeling tool, which is essential to building a quality architecture within complex technology environments. They are far more than just drawing tools; they provide semantic language properties to enforce rules providing consistency and functional integrity of the models. They can also

directly generate code and other artifacts from visual models, thus eliminating human error and increasing productivity.

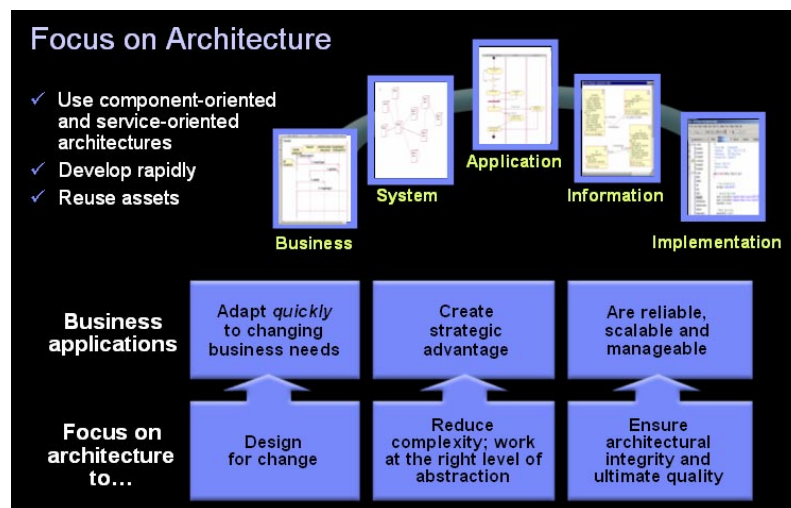
For the on demand business itself, a focus on architecture yields the following important benefits:

**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, more than anything else, an application’s architecture determines its quality. A good architecture allows the application to be modified over time. Thus, for businesses competing in the on demand era, the ability to deliver quality applications with the flexibility to handle rapid change will help ensure long-term business success.

**Figure 8:** A focus on architecture drives key benefits for applications driving the on demand business.



**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 and integrated 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.

**3. Continuously ensure quality**

Software testing accounts for at least 30 percent of total software development costs, but despite this high expenditure, software bugs -- mistakes discovered after product deployment -- cost the US economy an estimated \$59.9 billion annually.<sup>3</sup> One reason for this high "cost of quality" is the tendency for development organizations to conduct testing late in the project lifecycle, often after isolated teams have completed their work, modules are assembled, and testing teams begin the long process of discovering defects or noncompliance issues. This eleventh-hour testing strategy usually leads to code being scrapped, reworked, recompiled, then retested, and it naturally causes projects to run over time and over budget. Another reason for the high cost is the difficulty organizations have tracking and reproducing errors. Often, finding a bug itself is quite simple, but communicating all of the details necessary to reproduce and fix that bug can take hours or days. Sometimes, a developer cannot reproduce the bug based on the information provided, which can lead to operational downtime -- including unproductive finger-pointing between quality assurance and development teams.

---

<sup>3</sup> Source: National Institute of Standards and Technology (NIST) news release, "Software Errors Cost U.S. Economy \$59.5 Billion Annually," June 28, 2002

To remedy this, project teams must work to prevent, detect, diagnose, and remove defects throughout the application development and deployment lifecycle, not just at the end of it. We refer to this as the need to continuously ensure quality. The emphasis here is to address quality concerns earlier in the project life cycle -- when errors are significantly less costly to fix -- and to close the loop between the operations team that finds errors, and the software development team that diagnoses and repairs them.

Continuously ensuring quality also means stopping defects before they are coded. Many software applications fail to satisfy the client's requirements. Validation of requirements before coding even begins helps prevent significant rework that often occurs when the client and development teams realize that requirements were misstated or misunderstood. Also, after software is deployed for production, the need to ensure that it continues to function as intended in areas like performance is significant to the project's success. In other words, continuously ensuring quality means "even after the software is up and running."

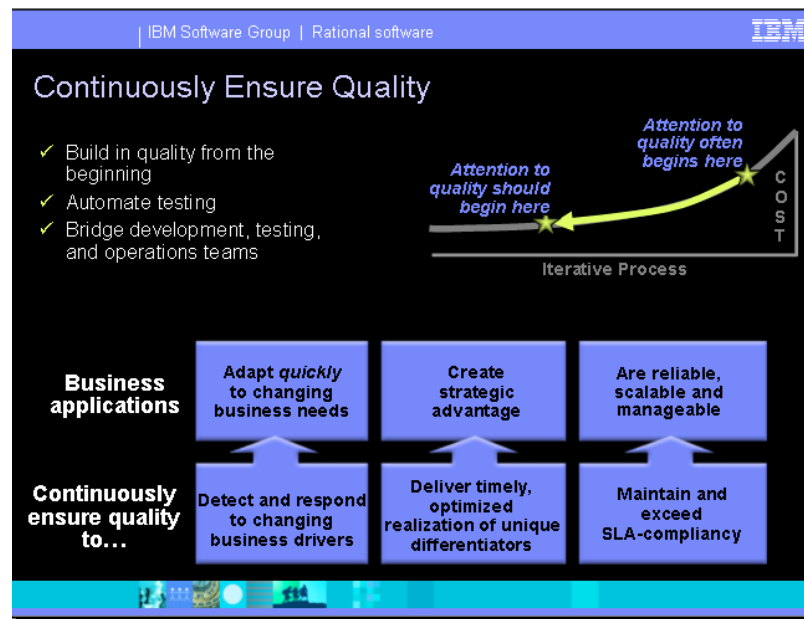
Commercially available tools and techniques can help teams discover and track errors, improve test coverage, and conduct regression tests (test that ensure bug fixes or other improvements to code under development do not themselves introduce errors). Yet the tools themselves are only effective when put to use by a team that has adopted a philosophy of quality, one which is shared and practiced across the entire development organization. The more software development teams work to bridge the gap between analysis, development, testing, and operations, the more it will function as a quality-aware organization capable of reducing organizational downtime, rapidly detecting faults, and automating subsequent rebuilding and redeployment.

By continuously ensuring quality, an on demand business derives the following benefits:

- **Detect and respond to changing business drivers** – Through iterative reassessment of the quality of customer use cases, requirements, and architectural models as well as code, teams remain sensitive to evolving customer needs and thus can rapidly respond. This ensures that the software not only functions as specified, but also continues to solve the right problem.

- **Delivers timely, optimized realization of unique differentiators** – Because systems are continuously changing, a process of continuous validation is required to ensure that software continues to differentiate effectively. The more component-based and interdependent our systems become, the more important it becomes to continuously validate functionality through an automated regression testing program.
- **Maintains and exceeds SLA-compliancey (“service-level agreement” established with customer)** – Many of the factors that influence user experience can and do change once the application has been deployed -- for example, external Web services, operating environments, and partner and supplier integrations. Only by verifying application performance post-deployment can we ensure the quality of the end user’s experience over time.

**Figure 9:** Too often, software development organizations do not test for software quality until late in the application lifecycle. By continuously ensuring quality from the beginning, teams are better able to deliver expected results to internal and external customers.



**Continuously ensuring quality: The benefits to business applications**

Essentially, continuously ensuring quality means that organizations must strive to build quality into their products and IT systems from the beginning of the iterative lifecycle. This effort pays off at the highest levels of the business, because business systems that are tested “early and often” are more likely to satisfy customer needs, and function as intended.

At the technical level, software development organizations dedicated to continuously ensuring quality experience better communication and understanding across development, testing, and operational teams, which speeds overall time to value for business applications. As a result, all business applications benefit from improved architectural models, a better understanding of user requirements, and more sound and reliable code. Figure 9 shows the relationship between continuously ensuring quality and on demand business applications.

#### 4. 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:

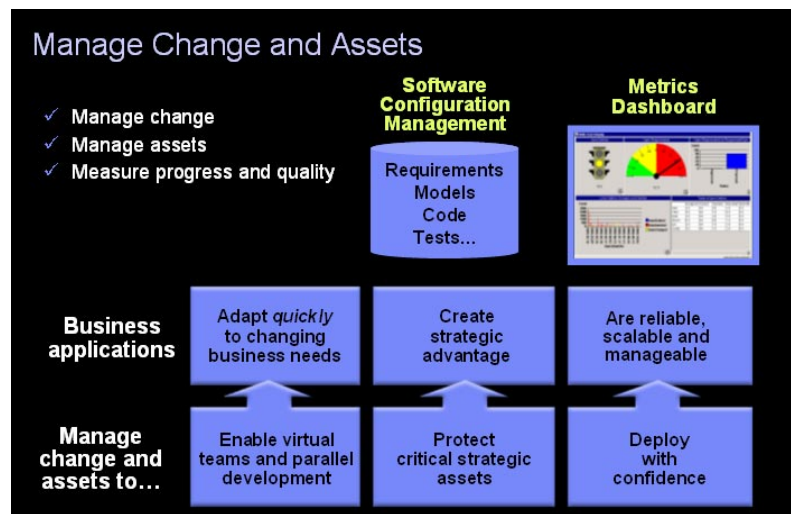
**Enables virtual teams and parallel development** — Advanced SCM systems allow multiple, sometimes overlapping, branches of a project to be worked on by different development teams simultaneously, so more work can be accomplished faster, on demand, without sacrificing quality.

**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.

***In all areas of the software development life cycle, our customers claim improvements. Customers claim reductions in time to value, report improved project management capabilities, and claim tremendous increases in productivity.***

## 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, 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 four imperatives for successful software development:

- **Develop iteratively**
- **Focus on architecture**
- **Continuously ensure quality**
- **Manage change and assets**

Deeply committed to the IBM Software Development Platform, IBM Rational delivers software engineering best practices, development tools, and professional services to today's organizations seeking the software development capability that these four imperatives represent.. 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 realizing 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.*

© Copyright IBM Corporation 2003, 2004 . All Rights Reserved.

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

Printed in the United States of America  
01-03

IBM, the IBM logo, the e-business logo, e-business on demand, Rational, Rational Unified Process, RUP, Rational Suite, , DB2, Lotus, Tivoli, and WebSphere 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)



Part No. TP077