Global communication,
global development, global success
October 2004

Rational. software

Geographically distributed development with the IBM Software Development Platform: A unified life cycle approach

Brenda Cammarano
Market Manager for Enterprise Transformation
Offering and Solutions Marketing
IBM Software Group

Contents

Introduction

You can't get through the day without reading or hearing about outsourcing. A trend that began in manufacturing and retail production, outsourcing is now gaining momentum in IT and everyone is talking about it—even the 2004 presidential election candidates. The intent of this paper is not to solve all of the issues that outsourcing—also termed geographically distributed development (GDD)—has created, but to take a look at some of the business requirements addressed by GDD that are driving innovative changes in business today.

In this paper, we'll take a look at some of these business requirements, why they are important to your business strategy, how GDD can have an impact on these business drivers, how to effectively address the risks of GDD and the tools necessary for a successful GDD strategy.

Realigning business and IT

Agility. Globalization. Cost containment. Tightening competition. These are just some of the fundamental business requirements currently challenging many organizations. Today's business world forces corporate information officers to examine these complex economic factors and reevaluate their IT strategies, to better align development projects with the organization's business drivers.

Let's consider each of these in turn.

Agility. These days, when anyone uses the word *agility* in the context of software development, most people think of a flexible system architecture based on component-based development. Although agility is still a benefit of component-based development, its definition has expanded. Increasingly, many believe businesses need agility to adapt quickly to volatile market conditions and economic fluctuations—that is, to accommodate changes in staffing structure, programming, system architecture, software development processes and in how systems conform to business governance and budgeting structures. As the common denominator in these areas, agility also allows organizations to adapt to never-ending local, state and federal mandates

requiring compliance, and even to cope in the aftermath of a terrorist attack. The entire organization has to be ready to adapt to all of these changes at a moment's notice.

Globalization. Globalization is the new frontier in business growth. The need to build or manage business concerns across international boundaries arises from many conditions of the modern age, and successful businesses adapt to its many requirements. For instance, a company may face a pending merger or acquisition that leads to teams being dispersed throughout the world. Or a business might need to expand by establishing a market presence in a foreign country, and afterward, it may need to localize a product offering (reengineering the product to accommodate local languages and customs). In all cases, the key to success is an ability to understand and work within different cultures.

Cost containment. Learning how to do more with less becomes necessary for all businesses at some point. The state of the economy will always affect the health of every organization. Gone are the high-rolling days of the dot-com era—the world as we know it today must not only conform to tighter and more stringent budgets, but organizations must also learn how to do more with fewer people. They must increase staff productivity to implement a greater number of processes, systems and products that are more effective, more efficient and more profitable.

Tightening competition. A market leader in any industry must have the mindshare of the customer and prospect base and be known as an innovator that offers high-quality products and services. A leading organization's products must be available at all times, and its support teams must offer the highest levels of customer service. To meet these demanding standards, organizations require specialized, highly differentiated software systems that surpass the capabilities of the competition, including systems to expedite internal processes—sales processing, inventory tracking and product development for example. Organizations also need systems designed to enhance the customer experience—Web sites that promote and advertise products and services and e-commerce features that allow customers to order products and execute transactions online.

Introducing GDD

To address these four business requirements, many companies are turning to a GDD model as part of their IT strategy for software development projects. For the purpose of this paper, *geographically distributed development* refers to the practice of managing a software development team beyond the traditional bounds of a development staff singularly located in a building or office structure. In a GDD model, development staffing may be distributed across town, in another state or province or even overseas. Some companies are engaged in distributed development within their geographically distributed corporate boundaries, which may involve multiple sites for a single software development project, as well as service suppliers (outsourced companies) that assume part of the software development life cycle.

GDD benefits companies by providing a diverse and agile organization that is characterized by the capacity and capability to act on a moment's notice.

GDD allows companies to not only augment their operations worldwide, but also benefit from reduced labor rates and variable staffing, along with a reduced time-to-market using a development staff available around the clock. GDD benefits companies by providing a diverse and agile organization that is characterized by the capacity and capability to act on a moment's notice.

But GDD is not an easy model to adopt. As great as GDD may sound, the keys to success lie in its execution and implementation.

Facing the risks of geographical distribution

As companies move to GDD, they face certain inherent risks that must be attacked head-on. The greatest risk centers on communication: a dispersed project team needs to communicate accurately, precisely and unambiguously in an environment in which team members may speak different languages. Communication is also hampered by the barriers of distance and time zones, cultural differences, knowledge- and work-transfer issues, conflicting in-house processes and ownership of software artifacts. All of these barriers are further complicated by the need to manage change and assets across geographic boundaries in a secure environment.

Affecting all dimensions of the GDD life cycle, language and cultural differences challenge a project manager's ability to clearly state project requirements and system functionality. Managers need to communicate architectural design and process flow so that all team members understand project requirements, thereby ensuring that the right application is developed.

Knowledge- and work-transfer methods may be routine in a co-located work environment. But in a GDD environment, they cannot be taken for granted. Does each GDD team member clearly understand the scope of the project? Are individual responsibilities clear? And do all team members know how to communicate work status and detailed data relating to the next task? Is there a standard unified process that provides best practices and consistent process guidelines? Managing change and assets so that teams can develop and integrate software changes across all locations in a seamless environment to keep the project moving forward becomes equally complicated.

To address these challenges, companies are turning to software development tools that provide the infrastructure for a unified approach to process, communication, software development and project metrics.

A solution that knows no boundaries

The IBM® Software Development Platform provides a life cycle solution that knows no boundaries. It can help distributed teams communicate, develop and manage software projects effectively by optimizing the internal infrastructure. And the IBM Software Development Platform helps ensure project success because it is based on industry-proven best practices and tools. The IBM Software Development Platform provides best-of-breed support for individual practitioners performing requirements management, visual modeling, analysis and design, testing, project management, configuration management and change management, all based on a proven process.

Additionally, IBM Rational has integrated the products within this solution to automate workflow, team communications, the sharing and reuse of information and life cycle traceability. These integrations unite a team and contribute to project success. This automation leads to project predictability, improved communication, higher quality and reduced time-to-market.

Deployment considerations for distributed teams

To realize the benefits of an integrated product set such as the IBM Software Development Platform, new implementation strategies may need to be considered and combined. One such strategy is to incorporate Citrix technology into the business plan. This technology provides an application server environment that centralizes the execution and administration of applications on a server and allows multiple users to access them over a network. This model essentially turns a device into a thin client that only needs the ability to display and manipulate the user interface.

Additional techniques to consider in your distributed strategy include providing Web-based access to shared repositories and reliably replicating and synchronizing repositories across geographies.

A unified process

A unified process is critical to a distributed team's success because it helps establish a highly repeatable procedure, a set of software development best practices and standard methods of execution.

A unified process is critical to a distributed team's success because it helps establish a highly repeatable procedure, a set of software development best practices and standard methods of execution. Such a process provides a common vocabulary and clear definitions of responsibilities that unite dispersed teams promoting a common vision and culture. Once the process is established, tools can be used to automate each discipline.

Team-based tools that support a unified process and rely on shared information repositories can be adversely affected by physical distances, limited bandwidth and inconsistent wide area network (WAN) reliability and performance.

Additional factors that can complicate the deployment of tools are the company's network structure and security policies. To establish the ways in which your organization can effectively implement a GDD structure, you must first understand and define your development infrastructure and tool usage model for each discipline throughout the development life cycle. For example, you should consider these questions:

- *Where is your central intelligence located for each software development discipline—requirements, testing, project administration and management for example?*
- *Who creates project requirements and which team members need access to these requirements? Who needs to be able to modify the requirements?*
- *How do team members access the network? Are they limited by firewall issues? Is virtual private network technology an option?*
- *Are your developers working with models and code?*
- *What members on your development team need access to defect records submitted by your testing team? Where are these artifacts located on the network? Will team members need read/write access to the defect records so they can record information about when the defect was fixed, or will they only need the ability to read where the defect was found?*
- *Is Citrix currently a part of your infrastructure? If so, what members on your team can benefit from tool usage through Citrix?*
- *What is your existing network configuration?*
- *How will your organization's security policy affect GDD tool usage?*
- *How do you measure the status of your project when your team spans the globe? How do you effectively measure the number of outstanding high-priority defects? Have your critical requirements been addressed and developed as part of the application?*

As you begin to assess the location of your team members, the layout of your networking infrastructure and the current process that defines your development workflow, you will begin to establish the foundation for how process and tools need to be implemented or augmented to accommodate your specific needs.

Let's take a closer look at some of these questions to see why they are so important. And let's pay particular attention to how they will affect process and tool deployment, and how they will affect your development life cycle.

Process and tool deployment

Critical to a distributed team's success throughout the development life cycle are highly repeatable processes and standards of execution that provide a common vocabulary and clear definition of responsibilities. Because of the critical need to understand and follow the process through each discipline in the development workflow, ease of use and accessibility to the process is paramount.

To address this need, we turn to the IBM Rational Unified Process® (RUP®) framework. RUP is a software development process framework based on proven best practices. It offers valuable guidance and common processes throughout the development life cycle. By combining best practices from many disciplines—such as project management, business modeling, requirements management, analysis, design, testing and change management—into one consistent, comprehensive process, RUP promotes a common vision and culture throughout development. This approach helps to strengthen the relationship between dispersed sites by providing online documentation of project-specific process information. This shared process also improves communication, enabling development teams to collaborate effectively, work more efficiently and decrease time-to-market.

As a Web-based solution, RUP extends easily to support your development team at all locations. The Web site is created using dynamically generated HTML pages, which are published in the form of multiple RUP Web sites, each representing a configured and tailored process definition. RUP browser applets enable the RUP Web site to be dynamically accessed through a number of standard Web browsers. For easy navigation through the site, additional navigation applets are executed.

To address your specific project needs, the RUP platform provides a flexible process framework with powerful configuration tools that help you select and deploy a set of process components precisely tailored for your project. As a project is initiated, your project lead can configure the base process framework using RUP Builder and process Plug-Ins that are relevant to your project. These assets can be customized for the project size and the specific technology, tool or domain.

Once you have tailored your process, RUP Builder helps you quickly and easily deploy it to the team as a Web-based project view. Team members install the custom process from a central location directly onto their own machine. Each team member may then use a shared view of the entire project, which details workflow diagrams, process documents and specification documents relevant to your industry. Furthermore, team members can individually refine their project view using the MyRUP tool so that only the activities, artifacts and documents relevant to their work are displayed.

Communicating requirements

Poorly established and defined requirements are often the primary reason for project failure in a distributed environment. Verbal communication or loosely defined requirements used as a foundation for project development can lead to bad assumptions, errors and, ultimately, the wrong solution. The diverse cultures that are usually represented in large-scale, global GDD scenarios present communication challenges; therefore, anything other than concise and clearly defined and documented requirements will likely lead to unexpected and costly results.

IBM Rational® RequisitePro® software, a powerful and easy-to-use requirements management solution, provides an effective requirements management process based on the combined power of a database and the ease of Microsoft® Word. Word documents used for recording requirements provide context or supplementary requirement information. Rational RequisitePro uses a database to assign attributes such as priority, difficulty and status as well as to organize and track requirements. Related requirements can be linked, which helps the analysis team establish and analyze the impact of change; in other words, as a given requirement changes over time, it is easy to see the impact of the change on other related requirements. Having this type of real-time change visibility lets you pinpoint its effect across the project, which helps you make quick, informed decisions for scope management or resource reallocation. This requirements solution promotes better team communication, enhances collaboration and reduces project risk.

RequisitePro offers several configuration options for the distributed team. The best configuration for your team will depend on your team infrastructure. For instance, the location of central intelligence in your group for requirements definition will determine where best to set up the host server for RequisitePro; generally, the host server housing RequisitePro should be located local to your central intelligence, or *core users*. You must also determine which members of your team need access to project requirements. Will they be updating requirements? Or will they simply need to review requirements? Once you have mapped out your requirements usage model, you can evaluate the options available for tool deployment.

The native Microsoft Windows® user interface provides access to RequisitePro, which allows full-feature functionality, including project administrator functionality. Remote users benefit from IBM Rational RequisiteWeb software, which allows users to create, display and delete documents, edit requirement properties, engage in discussion groups and create requirements directly in

the RequisitePro project database. To mark requirements, delete existing requirements and edit requirements documents, RequisiteWeb users can take the documents offline, edit them in Word and bring them back online from RequisiteWeb. Once the document is back online, their changes are synchronized with the repository and made visible to other project members.

RequisitePro supports Citrix/Windows Terminal Server technology. You should consider this option if your existing infrastructure already supports Citrix technology or if Citrix is being considered as part of your distributed deployment plan. Given the resource requirement, this option should be considered only for a reasonably small distributed team. It can also benefit select members of a larger team who require the capabilities of the full native RequisitePro interface, including access to tool-to-tool integrations.

Visually model

Diversity of languages, cultures and time zones demands that we clearly communicate a vision of the application—from usage models to class and activity models—so that application functionality can be clearly understood. You need to be sure each team member envisions the same workflow, that key points or words used in every conference call are understood and that no cultural or language differences cause misinterpretation of project goals. You need to be sure that all members on your team are thinking of the same overall design at inception rather than discovering miscommunication at deployment.

With the Unified Modeling Language (UML), software professionals can visually model their analysis and design activities in a uniform and consistent manner, giving teams a common method with which to communicate and document a project's architecture and design.

The Unified Modeling Language (UML) has become the software development industry's standard notation for software architecture and design. With UML, software professionals can visually model their analysis and design activities in a uniform and consistent manner, so teams have a common method with which to communicate and document a project's architecture and design.

To make UML easy to work with, IBM Rational created the industry-leading and award-winning IBM Rational Rose® XDE™ family of visual modeling and development tools. Rational Rose XDE software provides tools for creating and maintaining UML models that depict project architecture, process flow, logical component relationships and database design. This software can also generate code directly from your models and generate models from your code, which gives you full control over when and how models and code are synchronized.

Rational Rose XDE tools use a file-based method for storing UML models and associated information. To provide repository and team development support, you need underlying configuration management like that offered by IBM Rational ClearCase MultiSite® software to help distributed teams add, update and view models and merge all updates on a distinct schedule.

For team members who only need to view models, Rational Rose XDE offers a Web publishing feature that allows users to share architecture and designs with all stakeholders, whether or not they have Rational Rose XDE installed. Models are converted to HTML and published on your intranet, where anyone with a Web browser can view them.

Defect tracking

Multiple teams, locations, languages and disciplines can cause mass confusion when work is transferred between disciplines within the software development life cycle. A number of questions arise:

- *Has the development team fixed the defect?*
- *Has it been passed to quality assurance for testing?*
- *Who has been assigned the latest change request introduced in the triage meeting?*

Left unanswered, these types of questions can lead to delays in project delivery and poor quality.

IBM Rational ClearQuest® MultiSite software meets the needs of most organizations with a proven change request management process. It also can be customized for your specific needs. Its robust, flexible workflow support includes e-mail notification and submission options, so your team members can be informed as change requests are updated. You can define unique workflows for any type of change request. To address team members at all locations, automatic synchronization provides up-to-date access to replicated defect and change tracking information so the entire team stays in synch.

Tool deployment considerations are contingent on your usage model. As described earlier, your initial assessment should determine:

- *Which users will need access to change requests such as defects*
- *The workflow of the change requests*
- *What users will be involved throughout the workflow*
- *Where the administration of the tool can be executed*

ClearQuest utilizes a two-tier, client/server architecture. Enhancements, defects and other change request records managed by ClearQuest are stored in a relational user database. ClearQuest stores information about a project's change management policies and practices (records, field definitions, valid states and transitions, data-entry forms, hooks and triggers) in a separate relational database referred to as the *schema repository*.

ClearQuest MultiSite enables the replication and synchronization of schema repositories and user databases across multiple locations. As with ClearCase MultiSite, this is a good solution for teams that consist of discrete sub-teams, each hosting team members on a central local area network. Like ClearCase MultiSite, ClearQuest MultiSite provides a failsafe mechanism in the event of a server outage.

As with ClearCase MultiSite, replication and synchronization capabilities support primary users in dispersed geographies. So whether users are developers who need to update defects or change requests with bug fix information or testers who need to create new defect records, each can use the native interface to access the local copy of data.

For remote users who do not have an extremely reliable and fast connection to the servers, a Web interface is available to allow users to add, update and query change request records. Alternatively, ClearQuest software supports Citrix/ Windows Terminal Server for remote users and administrators who require access to ClearQuest administrative tools not supported through the Web interface.

Application testing

Application testing and continuous quality involves collaboration and communication across the entire development team. You need a test plan that covers all dimensions of your application to ensure total application coverage. Especially critical in a geographically distributed environment is full communication among the team defining requirements, the testing team and the development team fixing application failures detected through the test plan.

To assist teams with this collaborative discipline, IBM Rational provides a comprehensive system testing family. This includes IBM Rational TestManager software to help you administer a complete test plan including manual testing workflows, test logs and automated reports of test execution, as well as IBM Rational Performance Tester software and IBM Rational Functional Tester software.

At the core of test planning is TestManager software, which utilizes a two-tier, client/server architecture. Information about the test artifacts and test execution results are stored in a series of associated files know as a *test datastore*, and an associated relational database stores pointers and promotes easy data access to and through the datastore. For distributed teams, familiar Internet protocols facilitate inter-team communications for day-to-day testing.

Asset management

A system for managing assets serves as the backbone to distributed development. An effective infrastructure must scale with team size and number of locations; it must provide automatic, reliable replication of software assets between sites, which provide local access to all software assets to support a 24x7 software development work cycle.

Distributed development teams may consist of multiple office environments in different geographic locations, each capable of hosting a repository and multiple database servers. IBM Rational ClearCase MultiSite offers an effective asset management solution in this situation. As an add-on product for IBM ClearCase® software, ClearCase MultiSite provides reliable replication of repositories storing any file system object across geographies, whether binary or textual. Each central sub-team location works with a local copy of the repository for optimal performance and can bypass any issues related to WAN or Internet performance. MultiSite leverages ClearCase support for branching and merging, so any changes made at one location can be synchronized and merged with changes at other locations. Synchronization and replication intervals are controlled by the software change management project administrator. During synchronization, only the changes are moved between locations, not entire versioned object base (an IBM Rational ClearCase database repository for storing multiple, sequential versions of project artifacts under development) copies. Having multiple replicas also provides a failsafe mechanism to minimize project downtime and rework in the event of a server outage.

If your topology includes a substantial number of remote developers working at home or at small satellite offices, ClearCase offers a Web interface that supports many of the standard ClearCase features. For those requiring full ClearCase functionality, Citrix/ Windows Terminal Server is available.

Project status and measured success

Determining project status can be an especially daunting task for distributed teams. Cultural differences tend to add complexities; for example, the norm in some regions may be to assert success when the project is actually weeks behind schedule. IBM Rational software places all project- related data at your fingertips so that you can accurately measure all factors contributing to the success or failure of your project—the number of top priority defects or the number of requirements that have not yet been addressed in the development cycle, for example. These types of detailed statistics can enable you to objectively measure project status and progress without requiring you to decipher what your team 10,000 miles away really meant by "almost on target."

In addition to having your finger on the pulse of project status, it is equally important to determine if expected returns were realized during a GDD project. Were the project requirements all met? If not, will the whole project need to be scrapped or require major reworking? At the time of deployment, were a significant number of priority bugs still unresolved? Did the need for development resources spike at the end of the project or did it level off? A dramatic rise prior to project deployment could be the result of significant problems leading to an unstable product. Questions like these will help you evaluate the success of the project and pinpoint how you can make the next distributed project more successful.

The IBM Rational ProjectConsole™ tool, a capability within the IBM Rational Team Unifying Platform™, collects actual development data from products within the IBM Software Development Platform along with third-party

products, presenting the results graphically so that you can easily and quickly assess project progress and quality. ProjectConsole allows you to objectively measure and better predict those areas that will require special attention and helps answer various types of questions:

- *Where should I focus scarce resources in order to stay on schedule?*
- *What trends could affect cost and schedule?*
- *Is the project stable and ready for deployment?*

ProjectConsole can help keep your projects on schedule and on budget so that you can recognize the benefits a distributed team can bring.

Project managers and all designated team members can easily access the project Web site dashboard via a browser. Visibility of reports and views can be limited to certain team members based on access control lists. Project managers can view pertinent data across each discipline to quantitatively and objectively measure project progress and quality. Only project administrators or users that need to design or modify ProjectConsole templates require a local client to be installed. To collect data from all development sites so that the dashboard reflects comprehensive project status, you can collect the data from remote sites using the ProjectConsole collector server, and then send it to the ProjectConsole server site using remote method invocation, which is Java's application programming interface (API) for process-to-process communication, or you can collect data directly from the central location (server site).

Product integrations in a distributed environment

Product integration within the IBM Software Development Platform provides the power to automate and integrate the business process of software development, so organizations can become more focused, responsive and resilient. This integration begins with project requirements traced through

application design, functionality testing, defect tracking and change management. As the topography of your development team extends to a distributed environment, the integration points will require attention. Of course, your more robust data repositories and interfaces will require less attention than those which were formerly designed and deployed for limited, local access.

The integration between the products discussed in this article occurs at the client level. The clients exchange information and create links between data in their respective repositories, resulting in data in one product repository being referenced by a different repository. Consequently, integrations will perform better when the repositories are co-located. In addition to product dependencies, the IBM Rational Administrator tool—the tool used to define the set of product repositories for a particular project—must be considered when establishing an infrastructure to support products within a distributed environment.

It is therefore important to define integration points that you want to use within the life cycle. Understanding your complete usage model as defined in your initial life cycle assessment will help you weigh integration options. Factors affecting integration usability include:

- *Hosting location of product repositories*
- *Data access through Web interfaces*
- *Use of terminal server technology*
- *Data replication of product artifacts*

Summary

Today, many companies are turning to geographically distributed development to achieve more agility and better cost containment while improving their competitive position in an increasingly global marketplace. However, introducing a GDD model into your IT strategy poses substantial risks—namely, the uncertainty surrounding a project team's ability to communicate accurately, precisely and unambiguously. Communication is difficult in an environment in which team members may use different languages; people are unable to communicate in real time due to distance and time-zone barriers; and there are conflicting in-house processes, knowledge- and work-transfer issues and issues involving the ownership of project artifacts. Common processes, tools and reporting help overcome this risk.

Achieving expected results with a GDD model requires an up-front investment in people, process and tools. Companies must first thoroughly assess their development community and supporting infrastructure in order to implement a development process and tools that will support a distributed team. The IBM Software Development Platform offers a life cycle solution that helps in all areas critical to GDD success. With it, distributed teams communicate, develop and manage software projects more effectively and efficiently.

IBM Corporation
Software Group
Route 100
Somers, NY 10589

Produced in the United States of America

11-04

The IBM home page on the Internet can be found at **ibm.com**

GC34-2474-00