

Why adopt a formal QA and Testing Methodology?

Background

Many organizations have concerns about their internal IT development maturity and capabilities. Corporate priorities are focussed on delivering more projects using new technologies and techniques with reduced time to market. A priority for many organizations is to ensure the quality of their development practices by improving their Testing and Quality Assurance practices. In examining their Testing/QA practices, organizations should ask the following questions:

- Are our development processes productive?
- Do we have a quality-focussed approach?
- Are we producing too many bugs in our projects?
- Are we testing our systems effectively?
- Are we testing the integration of our enterprise systems?
- Do we concentrate on doing things right the first time rather than attempting to test in quality at the end?
- What changes can be made to improve our testing process?

Our consultants have worked with numerous clients on their development and testing practices and have compiled this white paper to summarize some of those experiences and their advice for answering difficult questions such as those posed above.

What are the barriers to good Testing practices?

There are a number of barriers that may impede the progress of testing practices in an organization. Some of these barriers are described in this section.

Lack of Established Testing Practices

Many organizations do not have established testing practices. Organizations require a reliable, repeatable testing process which includes creating Test Plans, and Test Cases, establishing Test Environments, executing Test Cases according to plans and validating actual Test Results against Expected Results.

Project Driven vs. Process Driven

Testing is often viewed as a "phase" squished between the coding and the project target date, rather than as a set of tasks that are performed throughout the development life cycle. Consequently, inadequate time is allowed for preparing and executing tests.

Testing is typically viewed from a project viewpoint, where the goal is delivery of a specific product at a specific target date, rather than from an organizational process viewpoint, where the goal is to improve the overall development process and quality of products. This is the difference between project quality control and organizational quality assurance.

Quality is everyone's responsibility

Program code which has undergone minimal unit testing is often "thrown over the wall" to the Test/QA group, if there is one, or to the support/maintenance groups. These groups are then faced with attempting to remove defects in testing or in production. A better approach is to maintain a quality focus throughout the development with every project team member being responsible for the quality of their products.

Developers vs. Testers

Developers are not usually interested in testing, preferring instead to work on the more glamorous aspects of software development such as web and GUI development. In addition, Testing specifically requires enthusiastic testers; developers may not have the required attitude or aptitude to be good testers.

New Technologies

Technological changes require new or revised testing techniques. Client/server, object-oriented, web-based, message-based and other architectures require changes to some testing techniques, but the underlying core quality assurance principles must be retained regardless of the technology used.

Test Tools and Environments

Many testing tools are still shelfware. Despite the fact that tests must be executed multiple times both during and after a project delivery, automated test tools have not caught the interest of most organizations.

A significant challenge is the lack of testing environments in which applications can be tested in isolation from the production environment. This is most acute in the client/server environments where funds must be allocated to acquire and support appropriate testing server environments.

The most critical barriers to testing and quality assurance are organizational and cultural in nature. The technical issues can be solved if the right organizational priorities are set and if resources are committed to testing and QA.

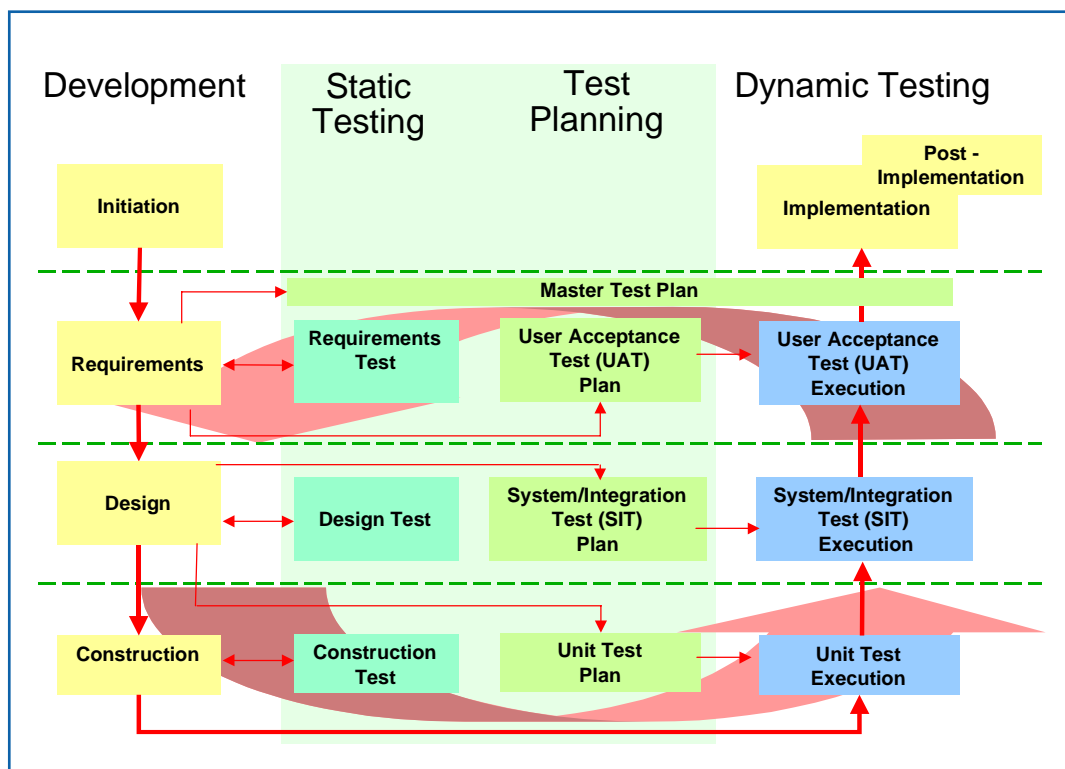
Why a Testing Methodology?

The Testing process is different and separate from the Development process, but the two processes remain integrated. Development and Testing have different goals:

- The goal of Development is to create or upgrade system components
- The goal of Testing is to ensure the conformance of the new or upgraded components to specifications and standards.

A Testing Methodology is the cornerstone of the testing and quality assurance activities in the organization. A Testing Methodology clearly identifies:

- the processes to be followed
- the products to be delivered and
- the practices for ensuring the high quality of all delivered software development products.



The diagram above shows a typical Development life cycle and the separate Testing processes. Static Testing, Test Planning and Dynamic Testing steps are clearly identified. By defining Test Plans at the appropriate stage of development, the quality of Test Plans can be assured since sufficient time is available to create the plans when the application requirements are clear in testers' and business users' minds.

By integrating a Testing Methodology with a quality process incorporating inspections/walkthroughs, shown as Static Testing on the diagram above, quality can be engineered into each development product during delivery. Test execution or Dynamic Testing then acts as a quality assurance mechanism to validate the quality of the delivered applications.

What are the optimal characteristics of a Testing Methodology

Fit with Existing Methodologies

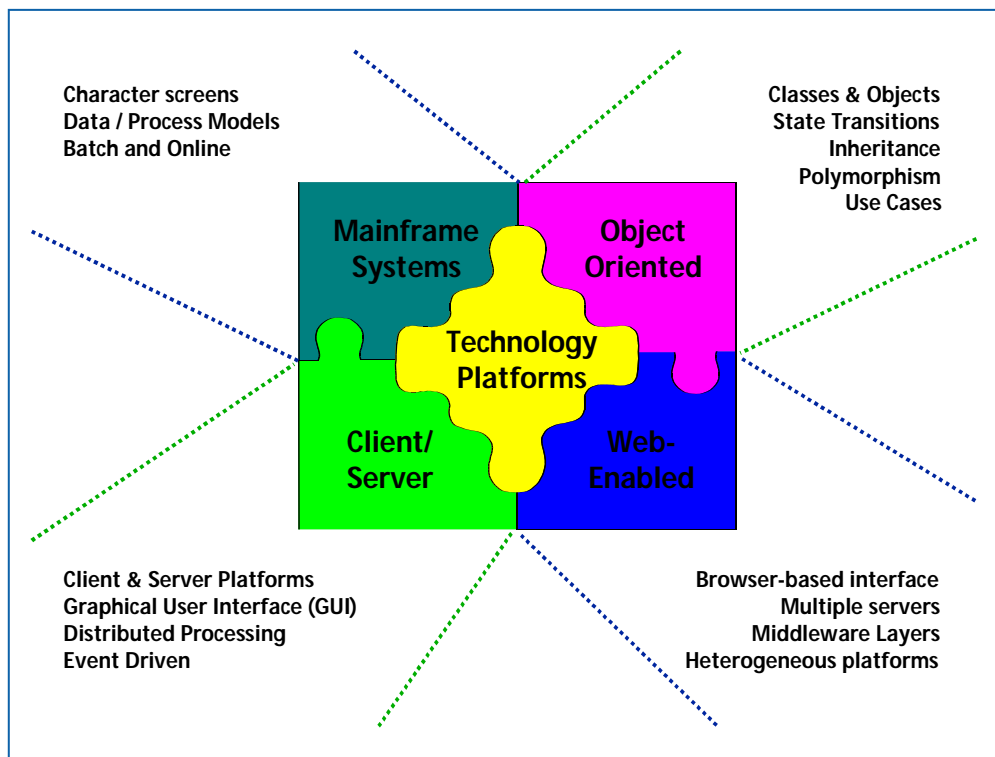
A Testing Methodology must integrate with your existing Systems Development methodologies. The goal is to integrate testing and quality assurance practices into the existing development methodology without changing all of your existing development processes. This requires an extremely flexible Testing Methodology, since an organization may utilize several methodologies, such as:

- An iterative approach for client/server and web-based systems
- An object-oriented approach for OO systems
- A waterfall approach for mainframe DB2 systems

The required flexibility is the ability to adjust the testing deliverables and tasks to match the development methodology.

Support for Multiple Technologies

A Testing Methodology must be adaptable to support various types of technology platforms as depicted in the diagram below. See the following section which specifically deals with the issues of Web-enabled, e-Commerce testing.



Support for Multiple Project Types

A Testing Methodology must be adaptable to various types and sizes of projects, including:

- New development projects
- Enhancement/Upgrade projects
- Maintenance projects or quick fixes
- Outsourced projects
- Software Package installations

Focus on Testing as a Process, not a Phase

Testing is not a "phase", but a process that is performed throughout the development life cycle. The development life cycle should contain steps that prevent defects from travelling further into the project. In this way, testing is a method of confirming quality rather than the primary method of preventing defects.

Throughout the testing process, various tools, techniques and environments will be utilized to achieve the desired level of testing. Standards, guidelines and best practices must be developed, documented, shared and promoted across all groups within each company involved in Testing. The continuous improvement of these standards, guidelines and best practices should continue through all stages of the testing process.

Risk-Based Testing

We cannot do 100% testing. The level of effort devoted to testing should reflect the importance of the software to the organization that maintains or uses it. Software critical to the organization must undergo more stringent testing than software that plays a lesser role. For example, establishing that a software system is used in mission-critical situations would dictate very stringent test criteria and assurance that these criteria were met. On the other hand, an assessment that the target software plays only an incidental role in the ability of the organization to perform its functions would change the nature of the risk exposure and of the test requirements to minimize that exposure.

These optimal features are required to ensure the long-term success of a Testing Methodology.

e-Commerce and Web-based Testing

One of the hottest areas of technology development today is e-Business or e-Commerce.

The Gartner Group estimates that through 2000, driven by the complexities of multilayer architectures and component/plugin technology, Internet-based applications have the potential to deliver 2 to 5 times as many defects on average, compared to traditional, two-tier client/server applications.

This is a heads-up that development is getting more complex and more error-prone. As well, e-Commerce applications are available to a much wider audience than your internal applications and therefore require more testing and validation to ensure usability by your customers and business partners.

There are a number of key aspects involved in testing e-Commerce applications:

Large, Unpredictable Transaction Volumes

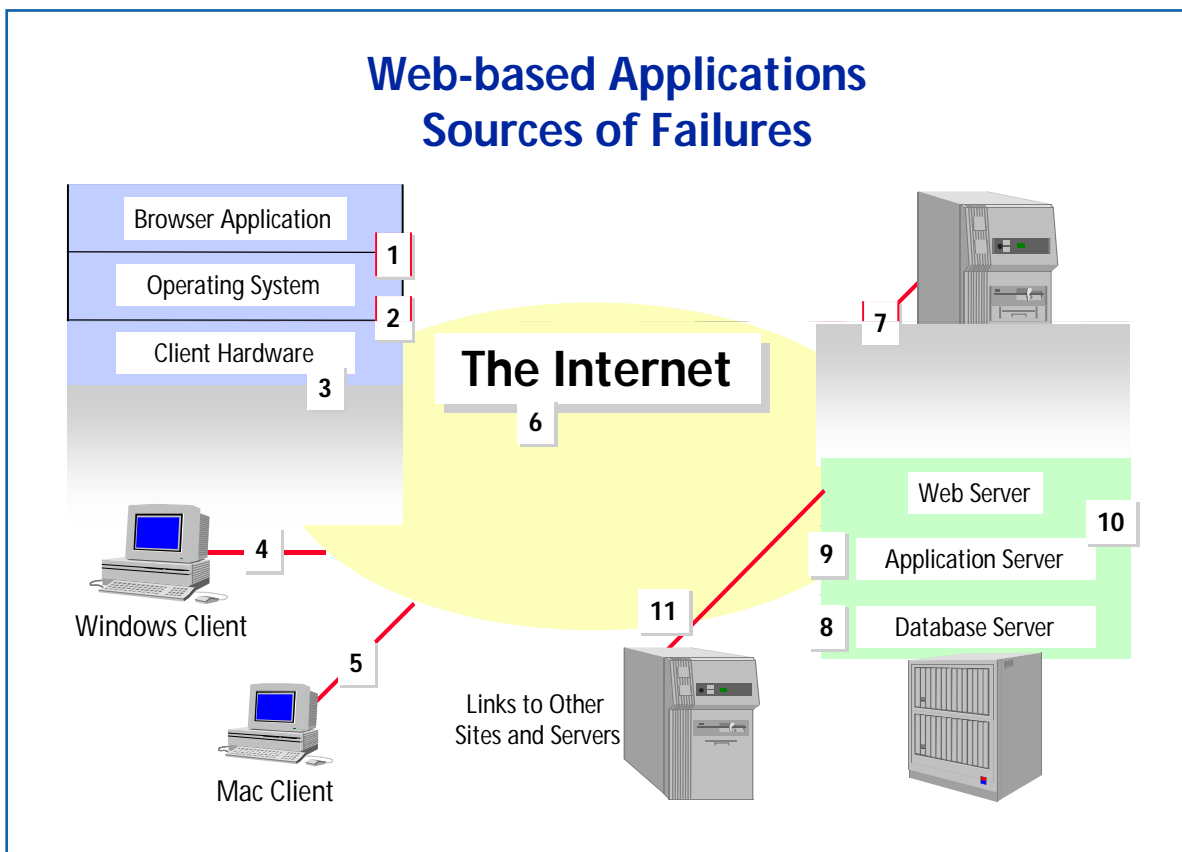
The internet environment has the potential to swamp your system with transactions. Performance, load and stress testing must be utilized to effectively simulate the real-world volumes and conditions that will be present on the internet. A testing methodology must deal with volume and performance as one of its key quality factors for web-enabled systems.

Multi-platform technology environment

The internet, extranet and intranet environments are inherently heterogeneous from a technology viewpoint. Test plans and test case designs must include configuration testing for various browser, operating system and hardware configurations.

Regression testing to ensure application consistency

The internet is a very dynamic environment. Changes such as broken links to partner or internal applications can leave a customer with a bad impression. Regression testing is required to ensure that applications retain their consistency and accuracy.



As with most new technologies, the basic testing and quality concepts remain the same, but the methods and techniques must be upgraded for the increased complexities of the new e-Commerce and web-based application environments.

What are the Organizational and Cultural issues related to Testing?

Management Commitment

Both IT and Business management must be committed to the goal of producing quality products. With management commitment, anything is possible; without it, little will be achieved since priorities will be set in favour of other organizational directions.

Development Organization

We see a number of organizational approaches, with varying degrees of success:

Organizational Description	Assessment
Development Group only	<ul style="list-style-type: none"> · Developers also perform testing · Primary focus on development, not testing · Testing is compromised to complete on target · Focus only on project, not on process improvement · Little or no commitment to testing/quality
Development Group + Test/QA Group	<ul style="list-style-type: none"> · Developers perform unit test and then "throw over the wall" to Test/QA group · Test/QA viewed as a bottleneck, slowing down the process and causing projects to be late · Test/QA is focussed on testing, but is often overwhelmed by the volume of testing
Project Team assembled from Developers and Test/QA members	<ul style="list-style-type: none"> · Test/QA staff assigned to perform testing/QA functions within project teams · Test/QA staff have dual responsibilities: to project and to Test/QA group · The entire project team has a goal to improve the quality of products
Same as previous + QA Practice Improvement Group	<ul style="list-style-type: none"> · All the advantages of previous, plus a separate group with a focus on process/practice improvement.

As described above, the development organization is critical to the quality of the products delivered.

Quality Focus

The organization requires a focus on the quality of every deliverable. Inspections and walkthroughs can provide a quick start to this quality focus, with additional automation being provided as required. The goal is defect prevention - to eliminate the defects early in the development cycle before they can become serious problems that require significant time and expense to remove.

Recognition of QA/Testing Career Paths

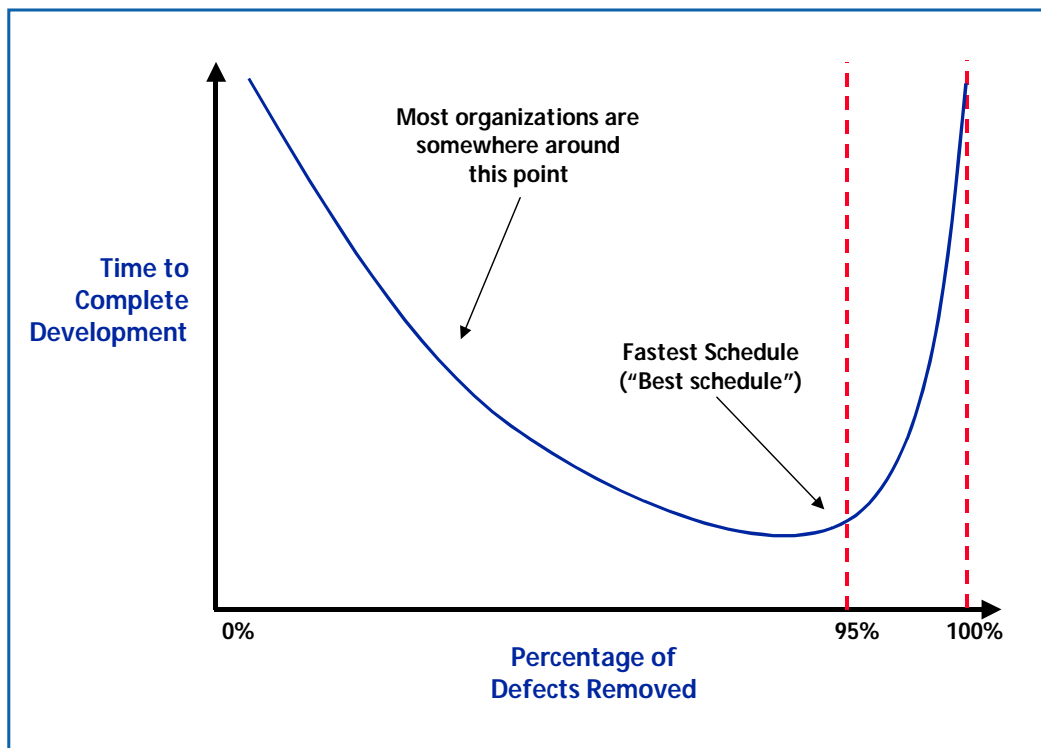
The lack of enthusiasm for testing and QA must be met by a commitment to recognize and reward QA professionals. Testing and QA should be recognized career paths. We should also recognize that the way we recruit/assess developers and testers may need to be altered: Development managers should recruit developers and QA managers should recruit testers.

Selling IT management on the benefits of a quality process

Savings from a Quality Approach

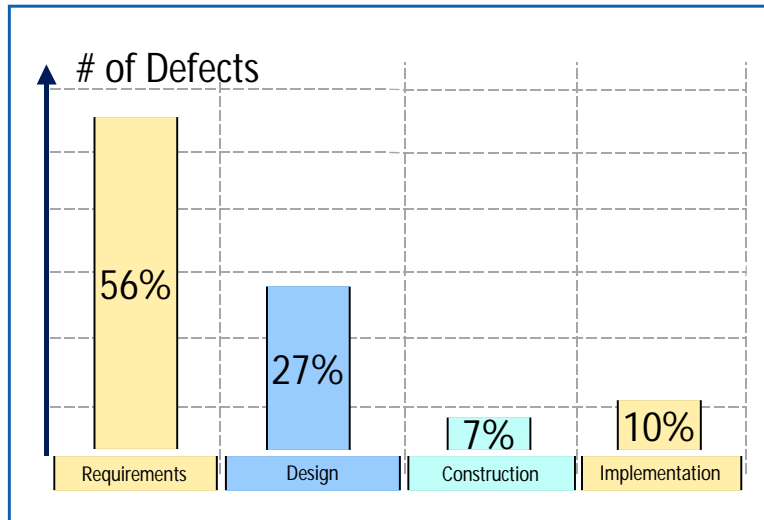
There are numerous studies available that confirm the fact that a quality-based approach to development and testing saves money by:

- Eliminating defects earlier
- Reducing re-work within a project
- Reducing production problems after release.



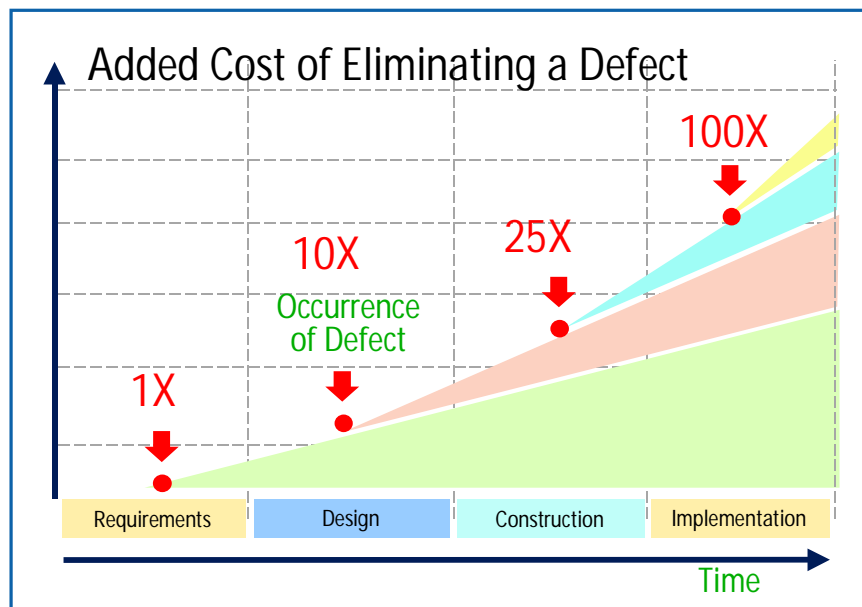
As shown in the above chart, the projects that achieve the lowest defect rates also achieve the shortest delivery schedules (McConnell 1996). Some organizations have very low defect rates, shown on the far right of the curve. Decreasing defect rates beyond a certain point increases development time and is only justified in safety critical applications.

Defects in Development Life Cycle



As shown in the above chart, most defects are introduced early in the development life cycle (Kit 1995). A requirements-based testing approach, with a focus on defect prevention and early deflection is the key to project success.

The Cost of Defects



As shown in the above chart, the relative cost to fix a defect escalates as we move toward production release. It costs 10 times as much to remove a defect at Construction as compared to at Requirements; it costs 100+ times as much to remove the defect once it is in production.

Some Hard Facts

- Each hour spent on quality assurance activities saves 3 to 10 hours in downstream costs (McConnell 1996).
- A requirements defect that is left undetected until construction or maintenance will cost 50 to 200 times as much to fix as it would have cost to fix at requirements time (McConnell 1996).
- Between 40% and 70% of initial software development time and resources can be devoted to error detection and removal (Kit 1995).

The Key Results

What are the key results from a Testing Methodology and a quality focus:

- Reduced risk
- Reduced costs
- Reduced delivery time; fewer late projects
- Increased customer satisfaction due to fewer post-release problems.

Better, Cheaper, Faster - the three magic words for IT Management. A solid Testing approach, integrated with sound quality assurance processes will allow projects to be delivered faster, with less re-work, fewer defects and higher quality.

Bibliography

- 1) McConnell, Steve, Rapid Development, Microsoft Press, 1996.
- 2) Gilb, Tom, and Dorothy Graham, Software Inspections, Addison-Wesley, 1993.
- 3) Kit, Edward, Software Testing in the Real World, Addison Wesley, 1995.

Contacts:

E-Mail Contacts can be sent to
infobal@infobal.com Or in
North America call toll free:
1-800-482-7770

Corporate Headquarters
22 St. Clair Avenue East 4th Floor,
Toronto, ON Canada M4T 2S3

Tel: (416) 962-2049
Fax: (416) 962-8020

Ottawa Office
99 Metcalfe Street, Suite 401,
Ottawa, Ontario K1P 6L7

Tel: (613) 569-5785
Fax: (613) 569-4711

E-mail: infobal@infobal.com
Internet: www.infobal.com