BEYOND DEFECT TRACKING:

Evaluating Defect and Change Tracking Solutions

IBM Rational ClearQuest v2003.06.00

Evaluators' Guide

Preface

Welcome to *Beyond Defect Tracking: Evaluating Defect and Change Tracking Solutions* from IBM Rational software®. We've prepared this guide to address the many issues you and your development team will likely face as you evaluate alternatives for defect and change tracking.

This guide was prepared for managers, administrators and developers who want to benefit from a scalable, robust defect and change tracking solution that can support any size team, regardless of platform or location. If you are committed to doing something about defect and change tracking for your team, we think you'll find this guide useful.

Please feel free to contact us if you have any questions or comments, or if you would like to arrange a demonstration or evaluation of any of IBM Rational software's solutions.

About IBM Rational and the Full Life-Cycle Solution

IBM Rational ClearQuest® is supported and maintained by IBM Rational software, which helps organizations create business value by improving their software development capability. Partnering with a leader like IBM ensures you will receive high quality support and mentoring for your projects – a key ingredient of successful software development.

IBM Rational software is unique at providing not just software tools but also a set of software engineering best practices (the Rational Unified Process®); an integrated set of software tools spanning the entire development lifecycle; and services to help individuals, teams, and organizations accelerate their software development by leveraging lessons learned.

As you review IBM Rational ClearQuest, keep in mind that while IBM Rational ClearQuest addresses your immediate defect and change tracking needs, it is also part of a broader solution, the Rational Team Unifying Platform, an open and integrated set of collaborative tools for software teams.

Aside from IBM Rational ClearQuest, IBM Rational Team Unifying Platform includes:

IBM Rational RequisitePro	A requirements management tool combining the simplicity of Microsoft Word® with the power of a database.
IBM Rational ClearCase-LT	A configuration management tool to version and baseline all project artifacts
IBM Rational TestManager	A test management tool to link test cases to requirements and ensure your testing efforts stay in sync with changing requirements
IBM Rational Unified Process	Browser-based process guidelines that improve team performance through industry-proven best practices
IBM Rational SoDA	A software documentation tool to generate live reports from all domain (requirements, test, design, change requests, defects)
IBM Rational ProjectConsole	A project status tool to provide quick and reliable reports on project status
IBM Rational Developer Network	An online community of more than 80,000 members with access to over 2500 targeted artifacts, whitepapers, articles, best practices, training, and topical discussions.

The IBM Rational Team Unifying Platform provides tight integrations between these products. Depending on your needs, you should consider either IBM Rational ClearQuest or IBM Rational Team Unifying Platform.

Introduction

With the right tools, any organization's chance of success increases dramatically. Nowhere is this more true than in software development, where the pressure to deliver high quality applications in the least amount of time has become a make or break factor for thousands of modern corporations.

Defect and change tracking systems provide a centralized, secure, reliable system for capturing, storing, tracking, and analyzing defects and other change requests. They provide important information about the nature of change requests, why requests were made and by whom. Defect and change tracking systems also follow change requests through to completion, identifying actions taken to address change requests and ensuring no requests are lost. With effective defect and change tracking, managers have instant insight into the status of specific change requests, as well the ability to view overall project status and analyze trends – critical information for planning, allocating and prioritizing resources.

Chances are you're either replacing a homegrown system you've outgrown, replacing a commercial system that isn't meeting your needs, or you're outfitting a new team with development tools for the first time. Regardless of your situation, the basic steps you need to follow to find the tool that's right for you and your team are the same. They involve the following:

Define and document your requirements

Meet with stakeholders and gather requirements. Defect and change tracking tools are used by various teams throughout the organization, including development, testing, tech support, documentation, product management, and so on. It's important to have representation from all roles during the requirements gathering phase to be sure you have a balanced perspective. Some of the questions to be asked during this phase include:

- What do you like about your current defect and change tracking tool, and what don't you like?
- What are "must-have" features in your next tool? Prioritize all your requirements.
- What's your timeline? You need to balance the risk of moving too quickly and selecting the wrong solution, or wasting time trying to evaluate too many products. Here's where a defined listing of requirements can help you limit your search.
- What platforms does the tool need to support?
- Will you want to grant access to extended team members, such as beta customers or consultants?

Narrow down the choices

There are a lot of defect and change tracking tools in the market place – which one is right for your team? With your documented requirements you can quickly eliminate vendors who don't meet your needs, and develop a manageable list of contenders. In addition to analyzing the product itself, you also need to consider some other critical factors that will determine whether or not you have a successful deployment, such as:

- Services. How helpful and responsive is the support organization? Are they technically savvy? How important is it to them that you're successful? Are services provided worldwide, 7x24? Is there adequate training available? What if you want onsite implementation assistance – is it available?
- Company stability. Is this a startup that could go under tomorrow, or are they established? What's their level of expertise and experience with defect and change tracking?

Evaluate and select

Now you've got your list down to 2 or 3 tools. It's time to take a closer look and make a decision. What can they REALLY do? What's just hype? Here are some ways to find out:

- Demo. Ask for an onsite demo or an evaluation copy. See for yourself how the product works.
- Third party reviews, references. Are there any reviews you can look at? Any references you can call? Any successful product should have both.

Key Requirements for Defect and Change Tracking

Key requirements for defect and change tracking products fall into several major categories, including capabilities for capturing all types of change, automating workflow, customizing and enforcing process, generating useful metrics, and integrating with other development tools. Security and scalability are also important. Finally, the capabilities, stability, expertise, and track record of the vendor must be taken into consideration.

Capturing Change

To maximize productivity and quality, any stakeholder should be able to submit a change request quickly and easily. The DCT system should support the submission and tracking of different types of change requests, with the ability to develop unique workflows and business rules for those different types.

Out-of-the-box deployment

While the ability to customize and tailor a system to fit your needs is critical, it is just as important that you're provided with an easily implemented out-of-the-box solution so your team can start taking advantage of a solid defect and change tracking process right away. This out-of-the-box configuration should provide all necessary functionality for capturing, managing, and reporting on change requests, with all the needed forms, fields, state transition models, reports and charts, to ensure you can begin using the product immediately. As your needs evolve, or experience is gained, you customize the product to accommodate procedural or operational changes.

Multi-platform client interface options

The DCT system should provide access to the system on all major platforms, including Windows, Linux, UNIX, and the Web. It's unlikely that all members of your development team will be centrally located. It is also unlikely that everyone will be operating on the same platform. Your DCT solution should enable all contributors to submit, modify, and track change requests through one, centralized system, regardless of what platform they're working on or where they're located.

Limited system access for non-licensed users

There will inevitably be extended members of your team, such as beta customers and consultants, who do not require daily access to the system, nor will they require full DCT functionality, but who should be able to submit change requests and run limited queries from time to time. Your DCT solution should provide a limited interface option, ideally through the Web or via Email, to address these needs without requiring a license.

Ease-of-use

In addition to offering users convenient options for accessing the system, it's also critical that the interfaces be designed to maximize ease-of-use. Confusing, inconsistent GUI's will frustrate team members, and discourage use of the system. Conversely, well-designed, intuitive interfaces will reduce the learning curve and encourage all stakeholders to contribute to the change request process. In addition, interfaces on the different platforms should offer a consistent look and feel, so users can easily move back and forth from one platform to another.

Submit and modify change request records via email

For extended team members who don't require regular access to the system, or for core team members who might be on the road at a customer site, the ability to submit and modify change request records via email saves time and helps ensure new defects or enhancement requests get immediate attention.

Support for multiple change request types

Tracking bugs is not enough. The system should enable submission, modification, and tracking of other types of change requests, such as enhancement requests, documentation updates, specification revisions, and so on. Because your process for fixing a bug is likely different from your process for implementing an enhancement request, the system should support a unique set of processes and workflow rules for different change request types.

File attachments

Supplemental files often help convey critical information relevant to the particular record being managed. For example, you might want to attach a screenshot showing a particular error message or dialog box. The system should support the attachment of files to change request records.

Support for internationalization and language localization

With today's multinational, distributed organizations, it's important to have a system that offers operating system support and native client support for as many non-English languages as possible.

Workflow Automation and Process Enforcement

In order to maximize efficiency, your DCT tool should automate procedures wherever possible. It should also help to enforce a common, consistent process for submitting, assigning, resolving and verifying change request resolutions throughout the development lifecycle. A robust DCT solution can help streamline communications across different roles on your team, reducing the seemingly endless string of emails, meetings, and conference calls.

Automated, configurable email notification

To accelerate the flow of change requests through the development lifecycle, and to enforce workflow processes, the system should support automated email notification. You might want to trigger email notification based on events such as:

- Changes to a field's value
- Some action being taken to modify a change request record
- A change request record showing up in the result set of a specific query

For example, you might want to set an email rule that sends email to the quality assurance manager whenever a defect is transitioned to the Resolved state, indicating the defect fix is ready to be verified by QA. The system should provide an intuitive, GUI-based mechanism for configuration of email rules, such as defining the parameters under which email notification occurs and who emails should be sent to.

Mandatory fields

Users of the system should be able to easily distinguish mandatory fields from optional ones, and the system should enforce the filling in of the mandatory fields. The administrator should be able to dictate which fields are mandatory, optional or read-only through a simple point and click operation.

Bi-directional linking of related records

Your DCT system should support the ability to establish logical two-way links between different change requests, such as in parent-child relationships, and enable business rules to be applied to the associated records. Use cases for these associations might include:

- The ability to create dependent fields, so that when the value of parent field changes the value of the child field also changes.
- The ability to establish a rule such that a parent record can not be closed until all the constituent child records are closed.
- The ability to link records through reference fields, to enable easy and accurate entry of data common across multiple change request records.

Automated update of multiple records

When similar modifications need to be applied to multiple records, your DCT system should support automatic update of those records, rather than having to update each record manually.

Configurable process enforcement through scripting

A process that can not be enforced will not be followed, yet no two organizations follow the exact same process. Accordingly, your DCT system should provide a built-in scripting mechanism to refine your process and go beyond what can be enforced through GUI-based operations. The system should support industry standard scripting languages that work across platforms, so scripts need only be written once. The system should provide commonly-used scripts out of the box, rather than requiring administrators to write them from scratch. Scripts can automate workflow and help enforce process in important ways, such as:

- Setting initial field value
- Validating contents of a field
- Creating a dependent field based on user input
- Checking for duplicate records
- Controlling user access for certain change request modifications

Customizability

Many tools claim to be customizable, but the degree of customizability, as well as the ease with which those customizations can be implemented, is a critical factor that varies widely from one product to the next. Perhaps you have a small team with relatively simple needs, with no real need to tailor your system. In that case, make sure the out-of-the-box deployment fits your needs. However, if you're managing a medium to large-scale project, or are expecting significant growth, customizability is a must.

GUI-based customization module

The system should enable most customizations to be implemented through a GUI-based mechanism rather than requiring programming. This should include the ability to add, modify, and delete fields, customize workflow rules and state transition lifecycles, modify user interface layout, and design public and private reports, charts, and queries.

Customizable forms and fields

The system should provide a complete, proven set of forms and fields out of the box, so you can start work right away. However, you should also be able to add, delete, and modify forms and fields as needed. For example, you might want to model the forms in your new system based on your legacy system, to make the transition to the new system smoother and easier.

Customizable state transition flow

If the out-of-the-box state transition lifecycle does not match your requirements, the system should make it easy for you to add, delete, and modify states as needed. Perhaps you want to customize the lifecycle states to match your exact process, or implement different state transitions for different types of change requests. The system should provide a GUI-based, intuitive mechanism for defining the process life-cycle states, actions and rules associated with each change request type.

Test changes before deployment

The ability to test changes before going live, without interrupting work in progress, is a critical productivity enhancer, as well as a way to ensure the quality of customizations implemented. In many cases, defect and change tracking tools require all users to log out while changes are being made to the system. In addition, changes made go live immediately, with no chance for review. For obvious reasons, this is very disruptive to the team, and reduces productivity. In addition to simply making it easier for the administrator to make changes at all, the ability to test changes before going live ensures that the customizations work as planned, and do not disable or otherwise damage the existing system.

Open, documented API

Some customizations can not be implemented through a graphical interface, so your DCT system should provide an open API based on an industry standard such as COM. The API needs to be well-documented, with detailed descriptions of the properties and methods associated with each of the API objects. The API should enable administrators to write code that accesses specific database elements, and well as enable external programs to access the database when desired. This gives you the freedom to extend the capabilities of the tool almost without limits, enabling the system to be tailored directly to the needs of a specific project or team.

Automatically propagate customizations to all platforms

Customizations should deploy automatically to clients on all platforms, rather than requiring separate work to be done for each. For example, a new field added by the administrator using the customization module should be visible to all users upon login, regardless of whether they're using the Windows, Web, Linux or UNIX client.

Easily apply customizations to multiple projects

The metadata or attributes that define the customizations should be stored separately from the user or project database where the actual defect and change request records are stored. The system should support the ability to easily apply those customizations to other user or project databases, rather than having to tailor multiple systems. For example, you might want to apply an existing set of customizations to a new database, and then make some modifications, rather than building an entirely new set of attributes.

Metrics: Queries, Charts, and Reports

It's not enough to simply capture and track detailed change request data – you also need to be able to query on that data to produce accurate, meaningful metrics. Project managers will want to know if resources are allocated appropriately, if change requests are taking too long to be resolved, if the number of high severity defects is threatening to shift a release date, and so on. Individual contributors, such as developers and testers, will want to know what change requests they should be working on.

Configurable reporting, charting and querying support

The system should provide comprehensive and customizable reporting, charting, and querying capabilities. The system should automate the process of generating reports, charts and queries through wizards and other intuitive, GUI-based mechanisms.

Out-of-the-box queries, charts and reports

The system should provide a comprehensive set of pre-defined queries, charts and reports as part of a standard install to enable team members to view useful metrics immediately.

Easily export query and report data

The record data derived from queries and reports should be suitable for export to industry standard report formats. For example, you might want to take some query results and export them for use in Microsoft Excel spreadsheet.

Metrics across change request types

In order to have true project-wide data for analysis, the DCT system should enable queries and reports to be run across different change request types. For example, you could run a query to show the status of all unresolved defects, enhancement requests and documentation updates in one report, rather than having to run separate reports for each change request type.

Management metrics

The system should provide managers with objective, accurate data about the project-wide impact of change requests throughout the system. Managers should have access both to instant snapshots of current project status, as well as access to distribution, trend, and aging data for more in-depth analysis. These metrics should enable managers to instantly answer questions such as:

- "How many urgent defects are still outstanding?"
- "Are assigned change requests distributed evenly throughout the development group?"
- "Has the submission rate increased or decreased over the last 6 weeks?"

Personal metrics

While management will want to generate project wide metrics, individual team members should be able to generate metrics that spotlight workloads and enhance personal productivity. For example, the system should support the ability to automatically run a personal to-do list each time a user logs into a particular project.

Integrations

Software development is a team sport. You've got various roles on the team - developers, testers, managers, release engineers, and so on, all of whom have separate responsibilities, but all of whom are working on related issues. Communication is a key factor, and the right tools, integrated with one another, can enhance communication and efficiency in important ways.

Integration with software configuration management tools

Software assets must be changed in order to implement change requests. These changes, however, do not exist in isolation. Everyone involved in the software development process needs to understand not only the individual changes but also how the changes fit into the overall development process. The integration of defect and change tracking tools with software configuration management products ensures that everyone involved in the development effort understands the changes others have made, and enables development teams to:

- Query across releases to determine which requests were implemented where
- Identify the actual source code changes implemented for a particular change request
- Determine which change requests have been implemented in one release but not another

Your DCT system should enable you to associate change requests with the specific, versioned software assets used to implement those changes. Ideally, the integration should provide the ability to do activity-based change management, where work is done based on activities, rather than individual files and directories.

Integration with requirements management tools

The integration of DCT and requirements management tools enables managers to understand how a given change request will affect the ability to deliver on certain requirements, or to understand the origins of particular requirements. For example, an analyst reviewing project requirements can view the enhancement requests that drove the requirements, and can even modify the enhancement request's properties from within the DCT tool. This comprehensive mapping between DCT and requirements management enables users to have quick access to information about a requirement's origin, or to analyze the potential impact of a given change request, enhancing their ability to make informed decisions and weigh tradeoffs.

Your DCT system should enable users to associate change request records with documented requirements and support viewing of association details in both tools, for a fully bi-directional integration.

Integration with automated testing tools

When issues are discovered using automated testing solutions, which include functional, runtime, and performance testing tools, those issues often result in requests for change to the system. Perhaps a bug needs to be fixed, or an idea for performance enhancements in the next release needs to be submitted. In order to ensure those change requests are securely captured and addressed, they need to be submitted into a reliable, centralized DCT system. Integrations between DCT and testing tools help improve productivity and quality, by streamlining and automating the submission, and ensuring accurate, reproducible test data is associated with a particular change request.

With an integration between DCT and testing tools, users should be able to submit change requests directly from the testing tools, without having to launch the DCT application separately. And when viewing change request records from within the DCT tool, test data such as test scripts and logs should be accessible through a simple point and click.

Integration with project management tools

Integration between DCT tools and project management applications enable managers to create a comprehensive "closed loop" project tracking system. It's an important two way association between the project plan schedule in the project management application, and the actual work being done using the DCT tool. As individual contributors work on defects and other types of change requests in the DCT tool, the progress of work is fed into the project management application, thereby updating the project plan with the most up-to-date information available. At the same time, information from new and updated project plans is exported into the DCT tool, enabling individual contributors and managers to adjust their priorities and schedules based on the most up-to-date project plan dates.

Users of the DCT system should be able to enter project plan information, such as estimated time of completion, or number of hours worked, directly into the DCT tool. Defects and other types of change requests should be directly linked with project plan tasks in the project management application, so they can be easily updated and synchronized to reflect any changes made in either system.

Scalability

There are many defect and change tracking solutions on the market, but will the system scale to your needs? You want a system that meets your requirements today, but will grow and adapt as your organization evolves.

Database(s) provided out-of-the-box

For smaller projects that don't need an industrial size database, or organizations wanting to do a test deployment, the DCT system should provide at least one database out of the box.

Support for multiple databases

Different organizations and different projects require different databases. Your DCT tool should offer a choice of industry-standard databases that scales from small projects all the way up to large-scale enterprise-level projects.

Flexible migration options

The system should make it easy to move from one database system to another, with comprehensive import and export utilities. For example, you might do a test deployment in Microsoft Access, and then migrate to Oracle when the project begins.

Support for distributed databases

For smaller distributed teams, or remote users who only interact with the DCT system occasionally, Web access may very well be sufficient. However, for larger distributed teams, or very active remote users, network infrastructure problems and Web server issues can hinder productivity. For these teams, the DCT tool needs to support the use of distributed databases. This helps geographically distributed teams to work together more efficiently by providing local access to replicated, up-to-date defect and change tracking data, and enabling automatic synchronization of those databases at any time, When combined with a configuration management solution that also supports distributed databases, this provides a complete, distributed change management solution.

User Administration and Security

Administration of the tool should be intuitive, data should be secure, access should be controlled, and you should be able to generate audit trails of what was changed, when, and by who.

Project level security

The system should support the ability to restrict access to project databases through user login and password authentication.

Field level security

The system should support the ability to control changes made to fields within a change request record. For example, you should be able to designate a specific field as read-only, optional, or mandatory. In addition, the system should support extended scripting ability for more sophisticated field access controls.

Ability to hide change request records

Your DCT system should allow certain defects and other types of change request records to be hidden from specific users based on what group those users belong to. For example, if you want to grant a major customer access to your DCT system, in order to be able to check on defects they've submitted, you would grant them access only to those specific records, and hide all other defect records in the system.

Flexible user administration

You have different roles on the team, all members of different groups, such as testing, development, tech support, product management, and so on. Your DCT system should support the ability to set up user groups that make sense for your organization, and assign access restrictions specific to those groups.

Tracking change request history

Your DCT system should maintain a permanent, auditable record of all changes made to the state of a change request. At a glance, you should be able to see who made the change, what was changed, and when.

Vendor Assessment

The decision to implement a new defect and change tracking solution is a long-term one. Because you select a vendor as much as a product, it is critically important that you choose the right partner: One with a history for making development teams successful. The things to look for during a vendor audit are:

Experience and expertise

Consider the backgrounds of the development group, support team, consulting and training staff, and senior management. Look for proven expertise, professional credentials, and a track record of success in building and supporting commercial products.

Customer references

No evaluation is complete without talking to or visiting current users of the product. Ask for reference accounts that are similar to your team, and gather opinions of the product's functionality, stability and growth path, as well as the quality of customer support. Understand the scope and range of each vendor's installed base, and ask vendors to provide evidence that their customers are expanding their use of the product.

Comprehensive worldwide support

Evaluate the vendor's ability to support your organization on a local or global basis. Fully explore all support options, and evaluate the vendor's ability to provide training, consulting, and deployment expertise where and when you need it. The ideal vendor is one that provides a full complement of global support services, allowing you to fast-track deployment and get fast answers to critical questions when they arise. This should include support for internationalization and language localization. This would include native client support for non-English languages such as French, German, Japanese, and other European and Asian languages.

Solid track record and stability

As quick as you can say the word “Java,” development environments evolve and change. Make sure your vendor demonstrates the necessary focus and R&D resources required to stay one step ahead of the change curve. Explore the financial stability, revenue growth and “staying power” of the company as well as its track record of product line expansion and innovation.

A “big picture” perspective

Look for a vendor with first-hand experience integrating defect and change tracking into the complete development lifecycle. An integrated solutions provider can provide important insights into how other areas of your software development process impact your organization's overall productivity. By providing a ready-made path to additional development automation, an integrated solutions provider can help you leverage and extend your investments.

Evaluation Checklist

The following checklists identify specific features and functions that development teams should consider as they evaluate defect and change tracking solutions. The identified features correspond to the key requirements listed in this guide. Space is provided for adding your own team- or site-specific requirements.

Feature or Function	Evaluation Product 1	Evaluation Product 2	Rational ClearQuest
Capturing Change			
Out-of-the-box deployment			✓
Client interface options on Windows, UNIX, and Web			✓
Limited Web-based access for non-licensed users			✓
Intuitive, user-friendly interface design			✓
Email-based change request submission			✓
Email-based change request modification			✓
Attachment of files to change request records			✓
Internationalization/Localization			✓
Workflow Automation and Process Enforcement			
Automated email notification			✓
GUI-based configuration of email notification rules			✓
Mandatory, optional, and read-only fields			✓
Logical bi-directional linking of related records			✓
Automated update of multiple records			✓
Configurable script-based process enforcement			✓

Customizability			
GUI-based customization module			✓
Customizable forms and fields			✓
Customizable state transition flow			✓
Ability to test customizations before deployment			✓
Open, documented API			✓
Automatic propagation of customizations to all client platforms			✓
Easily apply customizations to multiple projects			✓
Metrics			
Configurable queries, charts, and reports			✓
Out-of-the-box queries, charts, and reports			✓
Easily export query and report result sets			✓
Run metrics across change request types			✓
Management metrics			✓
Personal metrics			✓
Integrations			
Integration with software configuration management tools			✓
Integration with requirements management tools			✓
Integration with automated testing tools			✓
Integration with project management tools			✓
Integration with process guidance tools			✓
Scalability			
Database(s) provided out-of-the-box			✓
Support for multiple, industry-standard databases			✓
Flexible data import/export utility			✓
Support for distributed databases			✓
User Administration and Security			
Project-level security			✓
Field-level security			✓
Ability to hide change request records			✓
Flexible user administration			✓
Tracking change request history			✓
Vendor Assessment			
Worldwide, 7x24 technical support			✓
Onsite consulting services			✓
Comprehensive training programs			✓
Company stability			✓
Product references			✓

Conclusion

In this guide, we have sought to provide a perspective on defect and change tracking that is relevant, up-to-date, and helpful for software development teams. We have identified the need for these teams to move beyond basic defect tracking capabilities in the face of growing development complexity. Because we believe defect and change tracking is a long-term investment for your development team, we have stressed the importance of a requirements-based process and hands-on team evaluations in selecting a product. We have also identified some specific product and vendor issues to address as you consider alternative solutions. Regardless of where you are in the evaluation process, we encourage you to consider our product and vendor selection criteria.

About IBM Rational software

Rational software from IBM helps organizations create business value by improving their software development capability. The Rational software development platform integrates software engineering best practices, tools, and services. With it, organizations thrive in an on demand world by being more responsive, resilient, and focused. Rational software standards-based, cross-platform solution helps software development teams create and extend business applications, embedded systems and software products. Ninety-eight of the Fortune 100 rely on Rational software tools to build better software, faster.

And our worldwide network of IBM solutions and services professionals translates these advanced technologies into business value for our customers. Additional information is available on the Internet at www.ibm.com .