

INDEX

IBM Rational Team Concert	Page 4
Success Story 1	Page 5
Success Story 2	Page 6
Success Story 3	Page 7
Success Story 4	Page 8
Success Story 5	Page 9
IBM Rational Focal Point	Page 10
Success Story 6	Page 11
IBM Rational Doors	Page 12
Success Story 7	Page 13
Success Story 8	Page 14
IBM Rational Software Architect	Page 15
Success Story 9	Page 16
Success Story 10	Page 17
IBM Rational ClearCase, Rational ClearQuest & Rational BuildForge	Page 18
Success Story 11	Page 19
Success Story 12	Page 20
Success Story 13	Page 21
IBM Rational Change	Page 22
Success Story 14	Page 23
Success Story 15	Page 24
Success Story 16	Page 25
IBM Rational Quality Manager	Page 26
Success Story 17	Page 27
IBM Rational Functional Tester	Page 28
Success Story 18	Page 29
IBM Rational Performance Tester	Page 30
Success Story 19	Page 31



Dear Rational Customer,

In the following pages, you will find several success stories of how teams in Rational India and outside have benefited immensely from deploying Rational products. We are confident that you and your teams will derive similar benefits by following a similar approach.

We welcome you to visit <http://jazz.net> to follow and participate in the exciting journey of building a new generation of Jazz based products to help make software development more collaborative, productive and enjoyable.

Please contact Kurian John (kurianjo@in.ibm.com) if you need further help on Rational Products. Thank you for joining us and we look forward to building success together.

Warm regards,

Pawan Rewari
Director, Rational India Lab
and the IBM Rational India Team



IBM Rational Team Concert

A collaborative software development environment developed on the Jazz technology platform, IBM® Rational® Team Concert (RTC) enables developers to collaborate together using integrated Planning, Source Control, Work Items, Build, Dashboards, Reports, and Process Support.

- Improve developer productivity with real-time visibility into individual and team related work
- Built in project dashboards to track the progress, troubleshoot project problems and take corrective action earlier in the project lifecycle.
- Improved cycle times through automated and reusable patterns, services and architectures
- Earlier detection and accelerated resolution of defects
- High predictability of project outcomes
- Customizable work items and workflow
- Rational ClearCase® and ClearQuest® connectors

Success Stories

Development teams such as Corporate Tools, Rational Software Architect, Rational Application Developer, Focal Point, Build Forge are successfully using RTC for planning and scheduling work, managing source code and builds, and tracking defects.



Success Story #1: Rational Team Concert

Team

Rational Focal Point for Project Management

Challenges

A geographically distributed team, changing requirements, requirement traceability, time involved in learning different tools for development, version control system, defect tracking, builds, and most importantly conforming to agile planning while tracking the development work are some of the challenges of this team.

Adoption process

To address these challenges and implement a suitable solution, the team initially did a research on overall work-flow where each work item is completed in four-week iterations, and the software being delivered at the end of each iteration. The flow starts from the high level direction provided by the Project Management Committee (PMC) team and the Product Manager and distributed to the component teams, namely Development, Functional Verification Team (FVT), System Verification Testing (SVT), Documentation. The Work items identified for the iteration are provided with estimates in the meeting. The “planned for” attribute of the other work items are left as unassigned (backlog).

Implementation

The progress on the work items are tracked daily with regular updates by work item owners. The team lead or the manager always gets the latest status of the project. Builds are scheduled twice a day and also on demand. After each iteration, the team retrospects and captures the result in RTC.

Benefits

- Lesser status meetings as stake holders can view the iteration plan for various components in RTC.
- Reduction in development time because of integrated IDE, Requirement and Configuration Management.
- Reduction in the number of defects due to continuous build and integration.
- Better collaboration with discussion (and the history) available in the work items itself and the retrievability of tasks through user stories and work items.



Success Story #2: Rational Team Concert

Team

The Corporate Tools Team is involved in support and enhancement of products comprising of Software Configuration Management (SCM) and Change Management application (C/C++ server, Java APIs), Documentation Review Tool (Java server, swing/Web clients) and Defect Synchronization tools(Java) to synchronize data between external customer facing system with internal SCM and change management system. Efforts are also ongoing on testing automation through Rational Functional Tester (RFT) to reduce significant manual and repetitive testing effort.

Successful adoption results

- RTC server has been easy to set up and maintain. Platform has been very stable with zero outage in the past six months.
- Significant improvement in team communications:
 - Clarity on allocated tasks with expected outcome and improved accountability.
 - Regular updates on work items keep everyone in synch, reducing number of status meetings/follow ups.
 - Non project-related tasks coordinated through work items requires very less follow up due to single place holder (My Work View) for all the allocated tasks.
 - Integrated SCM and Build tool kit reduces code management (regular check-in, working builds) overheads to great extent. Activities requiring more than half a day with legacy SCM tool now are completed within minutes. Despite the learning curve with a new SCM, productivity improvements are excellent.
 - Common Server Workspace ensures speedy startup for a new team member or on any new system. It also keeps the workspace configuration consistent removing any local ambiguities.
 - Agile Planning and tracking is very intuitive.
- Information Retention:
 - All related information is consolidated in a single place holder.
 - Information is not distributed and limited in email (especially technical information) and can be searched easily whenever required. The data history is maintained allowing any new person in the team start up fast.
- Geographic Distributed Development:
 - Very easy to start team members in remote locations due to common workspace and ready availability of all information.
 - Regular updates on work items integrated team members across locations.

Benefits

With RTC, the team has experienced a productivity improvement of 20-30%. Familiarity with RTC is improving day by day. With reduction in overhead tasks (communication, code management), focus on main development tasks has increased, improving the quality of our deliverables.



Success Story #3: Rational Team Concert

Team

The Rational Team Concert Client for Microsoft® Visual Studio® brings the Jazz vision to .NET developers. With this flavor of the RTC client, the team can use Jazz to collaborate, track projects, manage your source code, defects, build components and so on – all from within the Visual Studio IDE.

Adoption process

Following the Jazz team's model of "self-hosting", the team has developed the RTC Client for Visual Studio using the RTC Client for Visual Studio, RTC Eclipse and the RTC Web Client. The team follows a develop-build-test-self-host-develop cycle – making incremental enhancements and improvements to the client at every iteration. This makes it very easy to share the same iteration cycles and plans, follow the same process and rules, use the common build infrastructure and also have the same server/client upgrade cycle. This has reduced redundant infrastructure and maintenance costs and cycles.

Successful adoption results

- **Planning:** The team plans the iterations using the iteration-planning feature in the RTC Web client. Every iteration plan has a set of planned work items and one can easily see the overall status of iteration, with the workload on individual team members. Reports on progress are viewed directly from the Web dashboard and no one needs to provide any explicit status, making the developers happy and the reports always current.
- **Configuration Management:** The development teams share the same stream for source code deliveries. The component-based Jazz model makes it architecturally easy for the client and server pieces to be shared as different Jazz components, while at the same time belonging to a single stream. Client components are shared, developed and delivered to the stream using the RTC Client for Visual Studio, while the server team uses the RTC Eclipse client to develop server components and deliver them to the same stream.
- **Work Items:** The development and test teams collaborate among themselves and with each other by sharing work items. Developers have design discussions around "Enhancement" or "Task" work items. Not only does this bridge geographical distance, but also captures such discussions for posterity. The team logs and tracks defects using "Defect" work items, which capture discussions around defects, resolution notes, and also code fixes through associated change sets. Code reviews, approvals, effort estimates, actual estimates etc are captured in each work item, and the planning views update themselves automatically based on these values.
- **Builds:** The teams share builds. Even though the server and the client components are developed out of different IDEs, the .NET developer can trigger a build, check build status, or download build results for server side components, without having Eclipse installed, and viceversa. This has significantly improved the integration testing process.



Success Story #4: Rational Team Concert

Team

The Websphere Business Services Fabric (WBSF) team adopted Agile methodology for the product development and conducted training on Agile with the help of the Quality Software Engineering team. The Agile model was a challenge for the team which had earlier worked in the waterfall model.

Adoption process

- Each iteration was six weeks. Scrum was used as a method to track potential blockers, dependency and issues. As a best practice the team gathered feedback from the team for each iteration, through surveys. The feedback was taken as lessons learnt and was addressed in the next iteration.
- Daily builds, BVT, working through the user stories, and collaboration were some of the best practices which were improvised with every new release.
- Rational Team Concert was used for project planning, estimation, collaboration, tracking and control etc.
- Complexity of an Epic and Story was identified in the planning meeting through the Poker
- Estimate of each task was defined by the individual developer after considering all the aspects.
- Used interlock meetings between Development, Quality Engineering and Documentation teams to understand the scope of work to be done in each iteration.
- Any architectural issues and impediments were addressed through RTC where the team could create the architectural issues or impediments for the Architects.
- Exit criteria for each sprint were defined. This helped contain the blocker issues in the same sprint and did not let it spread to the next cycle.
- Different charts like Burndown chart and team velocity charts were used to understand how the team was performing on a weekly basis.

Benefits

As the WBSF team is spread across India, USA and Canada, RTC played a key role in planning, collaborating across teams and giving a complete picture of the product development at any point of time from any part of the globe.



Success Story #5: Rational Team Concert

Team

The Build Forge team uses RTC for all software development activities, namely project planning, tracking, source code management, and defect tracking.

Adoption process

- Planning activities: Planning meeting at every iteration to discuss and plan the user stories for that particular iteration. These user stories are then created in RTC and assigned to individual users. The individual users then create appropriate (sub-stories, if required and) work items and put their estimates.
- Source Code Management: Use the RTC source repository to store all our source code and the test suite artifacts.
- Tracking defects: The work items in RTC are utilized to track defects and enhancements. RTC provides a seamless integration of work items with source code (changesets) and also builds.
- Build Management: RTC has its own build engine. Utilized this to do the builds, both developer builds and the integration stream builds. Build artifacts are created for every build which contain the status of the build.
- Project Management: The stakeholders can monitor the health of the project through the dashboards in Jazz. During iteration planning, the estimates to various tasks are assigned. The individual developers then update the work items periodically with the actual time spent on the work item (and the revised estimate, if any).

Successful adoption results

- **Transparency**: All users can see the list of pending user stories in RTC, before the planning starts
- **Traceability**: RTC provides traceability between user stories and actual work items. Planning is done for user stories, but users work on the work items. These user stories and work items (along with the estimates) help in tracking the progress of the project.
- **Process**: RTC allows us to customize our SDLC process. For example, we can have a rule that the developers must associate an open work item to a change set before delivery. Such customization can be done specific to our project alone.
- **Integration**: Seamless integration between source code (activity/change set), work items (defects/enhancements) and builds. Users can associate work items to change sets.
- **Team Approval/Reviews**: RTC provides a feature, where we can request team/peer approval/review of any work item or activity. This helps us in offline code reviews
- **One click build**: Users can fire a build (local workspace or integration stream) in just one click
- **Integration**: The builds are integrated with work items. On even of build failure, a defect can be filed from within RTC that automatically integrates the work item with the build.
- **Dashboard**: The dashboard gives comprehensive information about the project in general to all the stakeholders. It also gives a clear picture of what all user stories are complete and how much work is done.



IBM Rational Focal Point

A one-stop Web-based solution for product and portfolio management, Rational Focal Point's unique prioritization and visualization capabilities help organizations make better decisions faster, and plan and monitor the activities effectively in a dynamic market.



Success Story #6: Rational Focal Point

Team

The Focal Point lab uses Focal Point for a collaborative decision-making approach for every release of the Focal Point product.

The Focal Point product development lab is located in Sweden and India. The development and quality assurance teams faced several challenges in meeting their objectives.

Challenges

- The evolving business needs and requirements of customers and stakeholders had to be collated into a single repository.
- Enhancements and feature requests had to be prioritized for implementation after considering the market potential and other parameters.
- The planning, execution, monitoring, and control of release iteration were done manually.
- Test and defect lifecycle management activities used disparate systems.
- Traceability of release requirements to test cases to defects was ineffective.

Successful adoption results

The team found answers to their problems in the homegrown product IBM Rational Focal Point itself and the product lab team decided to lead by example. This not only resolved the internal problems but also ensured that the product was robust and comprehensive enough to serve the needs of product and portfolio management.

Benefits

- Prioritization and visualization of requirements enabled the lab teams to map the requirements to releases, and automate the management of these requirements, thereby improving the productivity and consistency.
- Facilitation of release lifecycle management helped in assigning tasks to resources, tracking progress, monitoring issues and risks, and delivering the product.
- The configurable workflow enabled necessary customization and email notifications, and was used for completely configuring the test management and defect management lifecycles.
- Generating traceability matrix from user requirements to software requirements to test cases to open defects boosted the performing impact analysis for implementation of new features and sizing a release.
- The easily accessible product-suite dashboard helped to take informed decisions at a strategic level.



IBM Rational Doors

IBM Rational Doors®, a family of requirements management solutions, improves quality by optimizing communication and collaboration and by promoting compliance and verification. The market and technology leader in requirements management - Doors - demonstrates core strengths in capturing, linking, analyzing and managing changes to requirements during the project lifecycle.

Successful deployments

- The Doors development team themselves have adopted the tool and has helped them to effectively manage changing requirements. It has also helped them to formalize the internal processes and obtain ISO 9001 certification. It has improved the confidence of different stakeholders and thus enabling more sales.
- The cross-product test team adopted Rational Doors for capturing the requirements for common client installer and for VMware demo image.
- The Configuration and Change Management (CCM) Software Quality Engineering (SQE) team customized Rational Doors for data synchronization between different Doors servers.



Success Story #7: Rational Doors

Team

Rational Doors development team

Requirements

- Required traceability to navigate from the user requirements to system requirements and then to the test requirements.
- Flagging of changes in upstream requirements to track the changes.
- Baselines to capture a state of a document at any given point in time.
- Support for formal and informal reviews of user requirements, design documents and test documents.
- Customization to extend the capabilities of the current features.
- Centralized information system to maintain project related information as a repository holding different documents.
- Access for geographically distributed teams.

Successful adoption results

- Traceability: For the last nine years, all the user requirements, system requirements for Doors and its integrations with Doors XT and Doors Web Access are stored in Doors and are used to drive the development. With the traceability feature, all the test requirements, test cases and results via integration with QC and now with RQM are all maintained and managed in Doors.
- Rework costs reduced: Using the suspect link feature in Rational Doors, the team was able to prevent any changes from being overlooked. This helped the team to reduce rework costs.
- Baselines: Used the baseline feature to capture the read-only copies of documents at any given point in time.
- Reviews/Approvals: Formal reviews were implemented using the Change Proposal System in Doors while discussions have been used for informal reviews. The review process was automated and the document could be reviewed electronically.
- Customization: The team used DXL in various scenarios to customize and develop additional utilities other than what is provided in Standard Doors. The team managed change control using the Change Logs in Doors.
- Information Repository: Planned for different projects using task list in the form of modules with different attributes in Doors. Lessons learnt and other knowledge base materials are shared in the Doors repository. Doors could support various file formats and also import from different sources thus making it a flexible repository of information.
- Support for globally distributed development: The database supports close to 100 globally distributed



Success Story #8: Rational Doors

Team

The Rational CCM SQE team uses Doors to manage the test artifacts and to link the Functional Specifications with test artifacts for better traceability.

Challenges

The SQE team experienced problems in accessing the Irvine Doors database using remote access.

- The connection was slow.
- Could not access documents when Irvine Doors server was not available.
- Any modifications to the FS were not traceable as that had to be handled outside the tool.

Successful adoption results

- The solution was to have two databases so that data was transferred across for ease of access.
- Doors DXL utility was written to transfer data between the two Doors servers.
- Data was synchronized on a daily basis to get modified changes across servers.

Benefits

- All the changes are available to the team locally.
- No remote connection is required and hence faster access.
- Traceability: The team works locally and creates high-level scenarios and links that with the functional specifications.
- Coverage: Provides mechanism to track tested features with scenarios and test cases mapped to functional specifications. Gives a clear overview of the requirement and the associated test scripts.



IBM Rational Software Architect

Gives clear overview of the requirement and the associated test scripts. IBM Rational Software Architect is an advanced Unified Modeling Language (UML) 2.0 based model-driven development tool that enables architects, systems analysts and designers to specify and communicate project development information from various perspectives, to several stakeholders. It leverages the power of model-driven development and helps development teams use UML 2 to create well-architected, model-driven applications and services.

Successful deployments

- The UML AL team has achieved significant improvements in unit-test coverage using the Code Coverage facility available in Rational Software Architect.
- SOMA-ME leverages the Rational Software Architect platform.



Success Story #9: Rational Software Architect

Team

UML Action Language (UAL)

Challenges

When a product evolves with various requirements, refined at each iteration, testing may sometimes become highly time consuming because it involves lots of manual work and error-prone analysis tasks. Besides, lack of quality of the manual tests can compromise the ability to test problem in the code being checked in. UML Action Language (UAL) team faced a similar problem.

Adoption process

The team used Rational Software Architect (RSA) for significantly improving code coverage of the JUnits being run as part of nightly build.

Benefits

This not only helped the team improving line-level code coverage (more than 2500 JUnits that cover more than 80% of various UAL components) but also provided regression detection by the time of the nightly build. Moreover, running Junit requires minimal effort thus allowing developer and code reviewer do other productive work while the test suite runs.



Success Story #10: Rational Software Architect

Team

SOMA - ME (Service-Oriented Modeling and Architecture - Modeling Environment)

Challenges

IBM products and tools did not provide support for service modeling activities as defined in the Service-Oriented Modeling and Architecture (SOMA) methodology of IBM. Service modelling:

- Remained conceptual
- Lacked traceability and change propagation.
- Lacked the benefits of modeling such as visualization capability, structural specification, template-based guidance and capturing of decisions.
- Could not be directly used for downstream activities such as SCA Module, Component Modeling and Object Modeling.

Successful adoption results

SOMA-ME has been developed as a UML based SOA modeling tool for the SOMA methodology of IBM. It is an extension of IBM Rational Software Architect and leverages various capabilities of IBM Rational® Software Architect platform - including UML Profiles, UML Model Templates, Transformations, Patterns, Model Validation, Views and Report Generation Framework.

Benefits

- Increased practitioner productivity and delivery quality using predefined templates, automation, streamlined tooling
- Faster learning of SOA methodology with tool support and online guidance
- Consistent design of components based on uniform reference model facilitate reuse and integration



IBM Rational ClearCase[®], Rational ClearQuest[®] and Rational BuildForge[®]

Rational ClearCase[®]

IBM Rational ClearCase[®] is an industry-leading solution that provides sophisticated version control, workspace management, parallel development support and build auditing to improve productivity. Rational ClearCase connects diverse teams by providing cross platform support and integrating with leading IDEs to work in your preferred environment.

Rational ClearQuest[®]

IBM Rational ClearQuest[®] provides change tracking, process automation, reporting and life cycle traceability for better visibility and control of the software development life cycle. Rational ClearQuest provides out-of-the-box Application Lifecycle Management (ALM) template that provides a firm foundation for ALM best practices.

Rational Build Forge[®]

IBM Rational Build Forge[®] is an adaptive process execution framework that automates, orchestrates, manages, and tracks all the processes between each handoff within the assembly line of software development, creating an automated software factory. Build Forge integrates into your current environment while adding valuable capabilities around process automation, acceleration, notification, and scheduling.

Success Stories

- Global Distributed Development of IBM DB2[®] using IBM Rational ClearCase, IBM Rational ClearQuest and IBM Rational Build Forge
- Global Distributed Development of IBM DB2 using IBM Rational ClearCase, IBM Rational ClearQuest and IBM Rational Build Forge
- Build Forge setup in IBM Rational CCM group



Success Story #11: Rational ClearCase[®], Rational ClearQuest[®] and Rational BuildForge[®]

Team

IBM DB2 product development team

Challenges

IBM DB2[®] product is being developed and maintained by a worldwide team of more than 3000 engineers (including developers, testers, build/release engineers, install engineers) spread across seven global locations. With support for multiple versions, on various platforms, by distributed teams, the challenges in this product development environment are great.

Successful adoption results

Flexibility and parallel development: IBM DB2 product development teams have been using Rational ClearCase for almost seven years now. The DB2 team easily managed to achieve global development and deployment of source code with ClearCase multisite.

- The base version of Rational ClearCase has great deal of flexibility and supports a wide range of platforms.
- With Rational ClearCase, the DB2 product development team created parallel projects for development. This way they developed and mastered individual components of a product remotely. The code was easily and seamlessly integrated.
- Tracking code changes to a particular project, branch, developer, and workspace can be easily deciphered with Rational ClearCase and changes made are visible across the sites within minutes using the multisite functionality of Rational ClearCase.
- With Rational ClearQuest integration, the DB2 product development team could easily manage its source code.
- Rational Build Forge integration follows robust process management architecture. It deploys builds faster and can restart a build in case of build failures. This helps products with a huge code base like DB2.

Customer quotes

“Rational ClearCase - The fact that a particular build level can be easily recreated without much of a hassle is fantastic.”

“With Rational ClearQuest integration, we added a very good defect management product making release management tasks much simpler.”



Success Story #12: Rational ClearCase[®], Rational ClearQuest[®] and Rational BuildForge[®]

Team

Configuration and Change Management (CCM) team

Challenges

ClearCase and ClearQuest builds involve interfacing through huge source code repository stored in CC spanning across multiple Versioned Object Base (VOB) with more than 1.5TB data. Before adopting Build Forge, all CC builds, CQ builds, spins, and tests were executed and created as scheduled jobs or cron jobs. In most cases the scripts had to be run manually.

Rational Build Forge setup in IBM Rational CCM group

- Two console on IBM X-series 336 with 4 CPU and 8GB RAM. One at ISL and the other at Lexington.
- MySQL DB hosted on Network Appliance – 6GB.
- 11 different platforms supported with the agent on more than 140 hosts.
- More than 20 users logged in to the console at a given time.
- 22 streams scheduled to build each night.
- 450-500 jobs run each week.
- More than 500 projects or libraries within a single console.

Build Forge adoption results

With Rational Build Forge to manage build orchestration, it allows complete build process to be managed via its Web console UI and all the build activities are now automated. You can also restart builds in case of any build errors.

Benefits

- 50-100% increase in productivity of RE team due to less time spent on managing the builds.
- Turnaround time for builds spruced up by 800%.
- No broken builds due to asynchronous problems.
- Testable images get to development and QE faster saving countless hours.
- Manual errors are minimized.
- Extended solution to development teams to provide full support for feature streams.



Success Story #13: Rational ClearCase[®], Rational ClearQuest[®] and Rational BuildForge[®]

Team

Rational Labs

IBM Rational ClearCase and ClearQuest are the market leaders in the configuration management and change management tools respectively. Both tools are capable of supporting large scale distributed development involving thousands of developers working in a 'follow-the-sun' mode.

The Rational development labs spread across the globe use ClearCase and ClearQuest for their product development. This is a testament to the capabilities of these two products as enterprise solutions for distributed development. These labs are spread across 8 locations in almost all continents and engage 4000 engineers developing, on an average one major release for each of 40 odd products every year. Major releases, patch and mod releases need to be churned out at a greater frequency. This indicates the degree of parallel development that these tools support.

Benefits

- ClearCase and ClearQuest support a scalable, centralized deployment model as also a complementary solution based on replication of data across multiple sites. This leads to a hub-and-spoke deployment where a few sites with geographical proximity share a centralized server and teams across continents use multi-site deployment for replicating data.
- ClearCase and ClearQuest can be easily configured for high availability and backup-and-restore, and supports a highly secure mode of operations (including access through SSL). These capabilities are leveraged in our internal deployment too.
- ClearCase and ClearQuest are integrated in our environment to ensure that source-code changes made by developers are tied to project artifacts like enhancements, defects, etc.
- Rational Labs play the role of the first customer of any release of ClearCase and ClearQuest, by deploying in-house the beta version of a release in this massively distributed environment.



IBM Rational Change

IBM Rational Change is an integrated Web-based change control and management solution that can help increase quality, improve accountability and provide consistent change management for software delivery. Change Provides powerful change management reporting capabilities, including traceability, trend analysis, metrics, and graphs for CMMI compliance.

Success Stories

- Cross product test team used Rational Change for defect management.
- CCM Software Quality Engineering team used IBM Rational Change as project tracking tool, managing document review cycle, managing physical and virtual infrastructure and defect management.



Success Story #14: Rational Change

Team

Software Quality Engineering (SQE) Team

The SQE team uses IBM Rational Change as a project tracking tool.

Challenges

The plan for validating IBM Rational Synergy on various platforms, locales and languages is maintained in a Test Matrix (excel format). This indicates where (in which platform), what (type of test) and when (time frame) to validate and track the currently available builds of Rational Synergy. A change life cycle was customized to depict the appropriate business flow.

- Decision-making during scrums was based on the updated test matrix.
- Task selection consumed a lot of time depending on the number of test cases.
- Scrum updates were shared through emails.

Adoption

- Tool reads a pre-formatted test matrix to create the planned activity list for the given sprint.
- During scrum you can pick tasks. After you choose your tasks in the system, the scrum master can trigger a system generated email that would contain the details of the scrum.
- You can update your status in the tool at the end of day.
- Backlogs are calculated automatically and reflected in the test matrix, after which an email is sent to the team with the project details.
- The tool publishes this updated test matrix to the project server shared across labs.

Results

- Quick selection of unassigned tasks.
- Automatically recalculated the estimates based on risk and the status of the project.
- Tool generates appropriate status reports, and the automated workflow forwards the reports to the appropriate parties.
- Provides easy normative analysis of tasks.



Success Story #15: Rational Change

Team

Software Quality Engineering (SQE) Team

The SQE team uses IBM Rational Change for managing document review cycle.

Challenges

Documentation is essential for successful delivery of the project. Any changes to the documents would result in changes to other artifacts.

- Documents were shared through emails.
- Poor traceability led to ineffective review cycles and documentation.
- Several copies of the documents were circulated before the document freeze.
- Documents were placed as controlled objects under Rational Synergy, thus; reverting to previous changes was an arduous task.

Adoption

- Submit a RCR (review-change-request) for documents or test scripts under review.
- Select the documents or test scripts for review based on the remaining project schedule.
- Easy to incorporate changes made in the previous review cycle.
- Enhanced traceability.
- Supports multiple review cycles, resulting in effective documentation.



Success Story #16: Rational Change

Team

Software Quality Engineering (SQE) Team

The SQE team uses IBM Rational Change for managing physical and virtual infrastructure

Challenges

Problems such as co-ordination on usage, maintenance and ensuring security updates have to be overcome while managing infrastructure comprising of 100 client machines that is used by 17 members.

- Send requests for new machines or updates to the IT team.
- No synchronous way of finding out available machines.
- Lack of coordination resulted in pre-emption of test machines by new users, thus increasing the risk of overwriting or duplicating the setup.

Adoption results

Rational Change provides flexibility to define customized life cycles. This life cycle has enabled the SQE team to focus on testing efficiently without investing effort on infrastructure management.

- Query for freely available machines for testing at any point of time in the change tool.
- Quickly figure out which machine is available, reserve it and start using it.
- Notify IT if security patches or updates must be done on the machine.
- Automatically notify users, administrators and IT teams when new machines are reserved or sent for updates through triggers.



IBM Rational Quality Manager

IBM Rational Quality Manager is a collaborative, Web-based, extendible test management framework solution that supports the full test lifecycle. (Planning, Setup, Construction, Execution, Analysis and Reporting).



Success Story #17: Rational Quality Manager

Rational Functional Tester System Verification Testing (SVT) team used Rational Quality Manager (RQM) and is now able to continuously evolve test management practices, enhance the product quality and reduce the test time.

Team

Rational Functional Tester SVT team

The team has used RQM for two consecutive releases.

Successful adoption results

- Team collaboration and project tracking: Every team member in SVT logs in to the LDAP integrated RQM Server. RQM customizable dashboard gives a comprehensive and real-time status. Team members track and update the individual work assignments. Team members are better informed and aware of prioritized list of tasks to be completed. The Web-based system and in-built reports makes tracking an easy task as team members and globally distributed teams are able to collaborate better. They are able to track passed and failed test cases in each iteration. Rich querying options enables better tracking and status reporting.
- Reuse artifacts: SVT team reuses the test artifacts, which has increased product-quality and reduced human error. SVT currently uses manual test option.
- Agile testing: RQM supports agile testing and the SVT team groups the test cases by category and executes milestone based testing. Inbuilt reports by groups/category ensure easy re-run of failed test cases.

Benefits

RQM adoption has benefitted SVT team overall in terms of better team collaboration, increased product quality, real-time status and improved test lifecycle practices.



IBM Rational Functional Tester

IBM Rational Functional Tester validates the application quality with automated GUI testing, functional testing and regression testing. It provides robust testing support for testing Windows®, .Net, Java™, HTML, Siebel, SAP, AJAX, PowerBuilder, Flex, Dojo, and GEF applications. It can also be used for testing Adobe® PDF documents, and zSeries®, iSeries®, and pSeries® applications.

Successful deployments

- Improved the productivity of the Rational Focal Point for Project Management team and the Rational Team Concert (RTC) SVT team for GUI automated testing
- Rational Functional Tester's ability to leverage Java has helped the ClearCase, ClearQuest, Rational Quality Manager, Rational Team Concert, Insight, and Rational Requirements Composer automation teams to build robust automated testing solutions.
- Enterprise Application Training Services (eATS) team from LTC has automation of more than 3270 console operator functions that has resulted in an easy and efficient test execution.
- IBM Global Business Services (GBS) BFDP team has adopted Rational Functional Tester for SAP automation for test execution and reporting.
- Rational Application Developer and Rational Software Architect Quality Engineering team has developed a robust automation suite for some of the components of Rational Software Architect / Rational Application Developer (Page designer and Web site designer) using Rational Functional Tester. The GEF support in RFT 8.0 has enabled significant automation of the modeling tools testing.



Success Story #18: Rational Functional Tester

Team

Rational Focal Point for Project Management FVT team

The Functional Verification Testing (FVT) team had a few specific requirements for automating the product. They leveraged the Business Logic Utilities Environment (BLUE) framework based on RFT to proceed with the automation.

Challenges

- Automating an application with frequent object properties change
- Handling object without unique properties
- Sharing of object repository
- Support for Dojo Objects

Adoption

Rational Functional Tester provides Find APIs that can be used to find objects of any application. BLUE (Business Logic Utilities Environment) is a test automation framework developed within IBM. It leverages the capabilities of the Find API's provided by RFT to identify the objects of various applications. It provides reusable libraries supporting specific GUI domains such as HTML and Dojo.

Rational Focal Point for Project Management has been built using Dojo controls. The BLUE framework provides extendable methods for Dojo object recognition. Object repositories can be maintained as a script file in BLUE enabling easy sharing of object repositories in a team and easy merging with source control systems.

BLUE provides a means to separate the “what” from the “how”. The framework provides built-in methods that can be used to recognize objects with and without unique properties. It also provides xml logger methods that can be used to generate the reports of the automation run.

Setup of the framework over RFT required minimal effort; hence one can get started on it with ease.

Benefits

- Ready to use methods available
- Provides an extendable Dojo library
- Helps follow modular approach for script development.



IBM Rational Performance Tester

IBM Rational Performance Tester (RPT) tests the performance of Web servers and network solutions running a variety of protocols, such as HTTP, Citrix, SAP, and SIP.

- It identifies the presence and cause of system performance bottlenecks.
- Users can create code free tests quickly without programming knowledge.
- It provides a rich, tree-based test editor that delivers both high-level and detailed views of tests.
- Automates test data variation and enables insertion of custom Java code for flexible test customization.



Success Story #19: Rational Performance Tester

Team

Rational Focal Point for Project Management SVT team

The Functional Verification Testing (FVT) team had a few specific requirements for automating the product. They leveraged the Business Logic Utilities Environment (BLUE) framework based on RFT to proceed with the automation.

Challenges

- Performance testing was one of the primary charters in SVT. The teams had to execute performance, reliability, long run and volume/stress testing on complex topologies and provide a detailed report.
- The team had to provide performance benchmark and baseline data for each milestone and tracking them manually was very tedious.
- The biggest challenge was to create huge amount of test data to mimic real world customer scenario.

Successful adoption results

The team adopted RPT and has successfully exercised the following test scenarios:

- Volume Testing: SVT team had to generate large volume of data into system before performing any SVT test. They used RPT to populate data in system, which was very fast and efficient in data population. Using this data the SVT team could detect good performance and scalability issues.
- Performance Testing: Successfully performed
 - Long run reliability testing,
 - Multi user concurrency performance testing and stress testing.
 - Created Performance Baseline and Benchmarking (PBB) and most importantly finding Break Even Points (BEP) of various product components.
 - Found many performance issues at early phase of product development.

Reporting

The generated detailed report by RPT on each product component and scenarios proved to be very useful for all the stakeholders to drill down on specific component code base. It helped them to provide detailed performance and reliability assessment report. The team could understand the regression in performance in each milestone.





© Copyright IBM Corporation 2009

IBM Corporation

Software Group

Route 100 Somers,

NY 10589

U.S.A.

All Rights Reserved

IBM, the IBM logo, ibm.com, Rational, Build Forge, ClearCase, ClearQuest, Doors, are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries.

A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at ibm.com/legal/copytrade.shtml. Other company, product, or service names may be trademarks or service marks of others.

References in this publication to IBM products or services do not imply that IBM intends to make them available in all countries in which IBM operates. The information contained in this document is provided for informational purposes only and provided "as is" without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. Without limiting the foregoing, all statements regarding IBM future direction or intent are subject to change or withdrawal without notice and represent goals and objectives only. Nothing contained in this documentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM (or its suppliers or licensors), or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

This document illustrates how one organization uses IBM products. Many factors may have contributed to the results and benefits described; IBM does not guarantee comparable results elsewhere.