



Rational software

Smarter approaches to enterprise modernization with collaborative development solutions from IBM.

Proof points in enterprise application development solutions

Contents

- 2 *Challenges of managed systems delivery*
- 4 *Improving the success of handoffs*
- 5 *Using real-time collaboration instead of assumptions*
- 6 *Relying on automation and governance instead of guesswork*
- 10 *Synchronizing and better managing deployments*
- 11 *Using enterprise CI/ALM to manage business systems delivery*
- 13 *Future-proofing your software infrastructure with Rational Team Concert for System z software*

Challenges of managed systems delivery

Since the first program was written, business and technology trends have been driving new business rules and interaction models for production applications. These models require increasingly sophisticated interdependent solutions. In other words, managed delivery of mixed-workload solutions has become the norm, and businesses need to adapt and apply enterprise modernization techniques to remain competitive.

In the pre-Internet era, large IT shops normally employed multiple autonomous development and production environments with parallel release cycles. Any interaction between the mainframe and the distributed environments would have been simple, and separate teams would perform similar and sometimes overlapping functions using multiple sets of tools. The teams operated efficiently but independently of one another. When there was little or no interdependency between the host and the distributed environments and little demand for the managed delivery of heterogeneous, highly interdependent — or “close-coupled” — mixed-workload solutions, this method was usually sufficient.

Now, many years into the age of the Internet, you may be discovering that your existing tools and methods — which were suitable for managing handoffs and promotions within their respective environments — are less able to support the level of collaboration and synchronization necessary to ensure smooth handoffs of cross-platform software solutions. To try and mitigate the risk of malfunctioning deployments, teams may resort to manual methods based on spreadsheets, e-mail or daily meetings to meet the collaboration, governance and communication needs of mission-critical, mixed-workload production environments.

Highlights

Even midsize businesses can lose tens of thousands of dollars per hour when mixed-workload production systems go down as a result of a failed handoff.

Any tooling investments should support the collaboration and governance needed to manage the delivery of mixed-workload solutions as well as simplify reporting of the effectiveness of the system.

Now consider the possible effects of a failed handoff on the business. Even mid-size installations can incur losses of tens of thousands of dollars per hour when production systems are down.

In addition to the increasing difficulty and risk of handoffs, the trend toward geographically distributed development adds another layer of complexity to the problems of collaboration, governance and communication. And all this is set against the backdrop of demands for increased responsiveness to new business opportunities and for higher returns on mainframe investments, as well as of the decreasing availability of skills necessary to meet these demands.

Tooling investments must be justified, particularly in an economic downturn or rebound. You must show that tooling can produce a measurable and demonstrable improvement in your teams' abilities to evolve the application portfolio. A solution should not only facilitate the collaboration and governance needed to manage the delivery of mixed-workload solutions, but also simplify reporting of the effectiveness of the system. Additionally, it should help teams and management quickly identify and rectify potential problems before reaching production.

In this paper we will discuss how using IBM Rational® Team Concert™ for System z® software in conjunction with collaborative application lifecycle management (C/ALM) can help your teams meet the challenges of reliably delivering mixed-workload solutions.

Highlights

Solutions that provide a high degree of collaboration and that can automate governance and workflow management can transform your software delivery activities into a strategic advantage.

IBM Rational Team Concert for System z software provides the infrastructure for collaboration, communication, governance and reporting with substantially less effort and cost than using many single-purpose tools.

Improving the success of handoffs

You can easily and cost-efficiently configure a modern software delivery system to help teams collaborate and communicate effectively, thereby reducing the risk of failed handoffs. Solutions that provide a high degree of collaboration and that can automate governance and workflow management — which is necessary to minimize the risk of unplanned downtime — can transform your software delivery activities into a strategic advantage.

With modern cross-platform lifecycle management solutions such as IBM Rational Team Concert for System z software, it is now possible for teams to provide the infrastructure support necessary to enable implementation of whatever level of collaboration, communication, governance and reporting is appropriate and to do so with substantially less effort and cost.

In the absence of adequate guidance and controls to prevent missteps, the complex interactions of modern mixed-workload solutions present many opportunities for the handoff to go wrong. Insufficient guidance or controls are symptoms of a software delivery system that lacks the infrastructure elements necessary to support cost-effective automation and governance.

Highlights

Modern software delivery systems such as Rational Team Concert for System z software feature collaborative development technologies to facilitate practically instant access to any team member who is online – regardless of their physical location.

Using real-time collaboration instead of assumptions

What is the top priority of production teams? For most, it's preserving the integrity of the production environment. The nightmare scenario for production staff is an unexplained failure and undocumented remediation procedures, leading to an indefinite period of downtime while the team runs ad hoc experiments to get production back online.

In some cases, investigation and remediation of production environment failures reveal that the flow of information among stakeholders was hindered. For example, a key piece of information — such as a requirements dependency or an undocumented assumption — may not have been communicated to a team member. In the absence of timely information, a team member made assumptions and took action in order to get the package into production. However, one of the assumptions was incorrect, resulting in a failure.

If you are part of a geographically distributed team, you're not able to have an informal hallway chat or mention a quick change while at the water cooler. With so many organizations relying on teams dispersed across different buildings, cities and time zones, it is easy to imagine the above scenario.

Modern software delivery systems such as Rational Team Concert for System z software feature collaborative development technologies to facilitate practically instant access to any team member who is online. Colleagues can chat as well as share and annotate their work, without having to leave their development environments. Team members can easily find out when colleagues are available for consultation.

Highlights

Even the leanest organizations can design and implement robust, automated software development and delivery processes with effective and appropriate controls and governance.

As work is performed, Rational Team Concert for System z software annotates work items. It preserves written discussions, changes, authors and build results in the context of the active work item. The platform enables you to quickly connect with the individuals responsible for that work item — without leaving the screen and with only a few clicks of a mouse — so you can confer, brainstorm or troubleshoot. With this detailed work item tracking capability, globally distributed teams can communicate and collaborate effectively.

Relying on automation and governance instead of guesswork

Inadequate guidance and reliance on poorly documented or manual procedures can also delay projects. The same scenario described above can also result when, in the absence of automated guidance and procedures, team members use improvisation, guesswork or undocumented steps to deliver a work item to production.

The standard features of Rational Team Concert for System z software enable even the leanest organizations to design and implement robust, automated software development and delivery processes with effective and appropriate controls and governance. Enabling a team to use agile software development,¹ Scrum² or even a traditional iterative process is simply a matter of selecting one of the many preconfigured process templates available in Rational Team Concert for System z software when you establish the project. Each process model is built on experience gained during the Rational software team's more than 25 years of assisting clients with delivering successful business software solutions. The templates are configured with customizable workflows, roles and role-based permissions.

Highlights

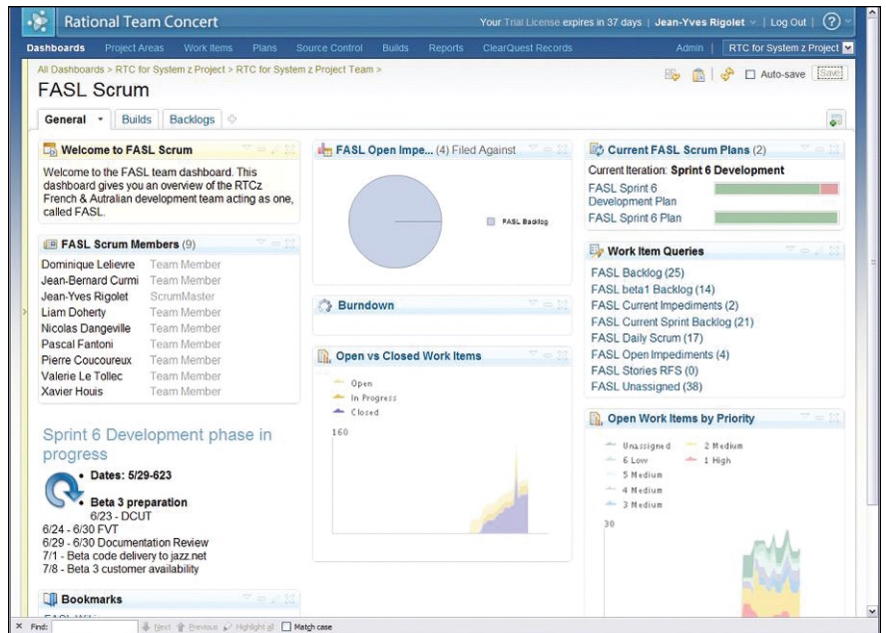


Figure 1: Rational Team Concert software enables you to customize dashboards so you can effectively govern your project from a single view.

Rational Team Concert for System z software helps make it easy for users to initiate projects with predefined models, rules and roles.

Initiating a project with Rational Team Concert for System z software is as simple as choosing a predefined model that best aligns with the team's desired work style and then reviewing and modifying predefined rules to accommodate the organization's policies and practices. Finally, simply assign staff to predefined roles and connect Rational Team Concert for System z software to the existing component repository, such as IBM Rational ClearCase®, Subversion or your mainframe software configuration management (SCM) solution. If you don't have a preexisting repository, you can populate the integrated Rational Team Concert for System z SCM facility, which enables a fully parallel software development environment using streams.³

Highlights

Using Rational Team Concert for System z software, you perform all work on components in the context of a work item, which follows an automated workflow.

The process engine provides transparent guidance and enforcement to help team members ensure that they are doing the right work on the right version of the right components at the right time.

Once a Rational Team Concert for System z project is created, you can start work by planning project iterations, assigning work items to the iterations and delegating staff to work items. Using Rational Team Concert for System z software, you perform all work on components in the context of a work item. When team members change a component in the course of implementing a work item, the change is attached to the work item as part of a change set, facilitating traceability and improving governance. When work is complete and signed off on as appropriate, the automated workflow capability of Rational Team Concert for System z software delivers the work item and its change set to a predefined target stream, also known as a staging area, for validation. Assuming a successful outcome, the software promotes the work item to the next stage in the workflow.

Each team member works in a private Web-based workspace, selecting an SCM stream to use as the “parent” and destination for the finished work.

The process engine built into Rational Team Concert for System z software provides transparent guidance and enforcement to help team members ensure that they are doing the right work on the right version of the right components at the right time. When they are finished with the component and ready to deliver the work, the platform automates the process of reconciling and annotating the differences between the original and the new versions of the component and then records the delivery.

Highlights

Automating administrative tasks with software frees application teams to devote more time to developing and delivering new and innovative business solutions.

By adopting the automated governance features of Rational Team Concert for System z software, you can reduce or even eliminate the need for time-consuming manual procedures and their inherent cost and risk. Automating administrative tasks with the software frees application teams to devote more time to developing and delivering new and innovative business solutions.

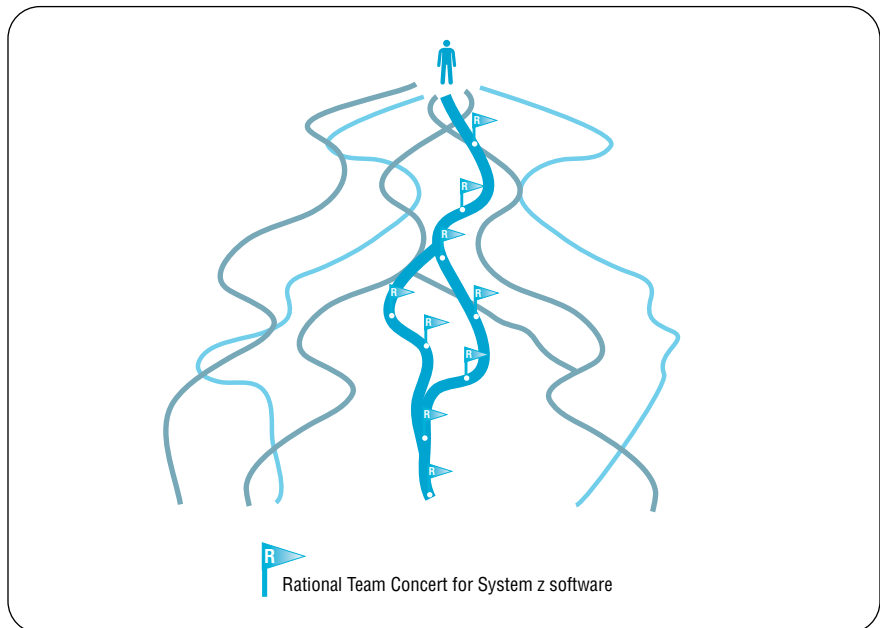


Figure 2: In the same way that signs on a ski trail safely guide skiers to their destination, Rational Team Concert for System z automated governance guides teams to the successful delivery of business software solutions.

Highlights

Rational Team Concert for System z software is highly accessible, enabling team members to initiate builds from any client on their organization's intranet.

Synchronizing and better managing deployments

With the rapid evolution of business requirements demanding quick, routine delivery of cross-platform solutions, software delivery systems need to be easily and cost-effectively configurable to perform this task. Once a luxury, this is becoming an indispensable requirement of the business.

A Web-based retailer is one example of a company that must constantly evolve its business rules and processes. Suppliers, inventory, pricing and product availability change continuously, and those changes must be reflected in the company's online catalog. A change in a state tax rate, for example, may require changes to host database tables or a customer information management application, as well as the JavaServer Pages technology that presents this information to the customer. A robust software delivery system can manage the deployment of this kind of closely coupled solution as a routine automated operation—and be no more taxing than deploying a single static Web page.

With Rational Team Concert for System z software, authorized team members can initiate builds from any client on their organization's intranet for any work item or set of work items—including IBM z/OS® components. Rational Team Concert for System z software includes a robust build management feature, which supports mixed-workload applications via an intuitive interface that simplifies the creation of build instructions.

Highlights

The Rational solution offers specialized features that simplify the creation of mainframe build scripts.

Successful software delivery depends on the right delivery infrastructure as well as the right methodology and processes.

At build time, Rational Team Concert for System z software delivers the change set and its build definition to a build agent running in the target environment in which it is executed. The Rational solution offers specialized features that simplify the creation of mainframe build scripts. You can specify and invoke mainframe commands—such as compile, link-edit, promote and bind—as part of an integrated, cross-platform build procedure, along with similar compilation steps for corresponding distributed components.

The Rational Team Concert for System z repository stores the build results, which include the work items and change sets that compose the build. It also includes a snapshot of the state of the source code base when the build was performed. Together, these artifacts form a comprehensive record of the build, making it easy to analyze or repeat if needed. With this information documented and easily accessible, teams can quickly diagnose, repair and back out of malfunctioning changes.

Using enterprise C/ALM to manage business systems delivery

Even the best software delivery system cannot compensate for a weak or nonexistent software delivery methodology. Although Rational Team Concert for System z software contributes the infrastructure for a software delivery system, success also depends on adopting the right methodology and processes.

Highlights

A process methodology can help an organization align requirements, development and validation across the enterprise.

C/ALM is a process automation blueprint that helps streamline and improve software delivery systems.

With the demand for rapid turnaround in delivery of new business solutions, effective communication among all stakeholders — from business owners to production staff—is essential to reduce the risk of missed deadlines and failed deployments. Implementing a process methodology can help by deploying common tools, shared terminology and a standardized usage model to align requirements, development and validation across the enterprise. The standardization and alignment of work practices and terminology improve communication among stakeholders, simplifying collaboration.

IBM developed C/ALM in response to new development organization challenges such as geographically distributed teams, an increased focus on business value and reliance on disparate packaged solutions as a result of rapid delivery expectations. C/ALM is a process automation blueprint that helps streamline and improve software delivery systems. It frames the interactions, or handoff points, among roles — such as business owner, project manager and developer — to define the workflow through the enterprise independently of the software tools used to implement and automate the workflow.

Highlights

Rational Team Concert for System z software offers a C/ALM process template — your key to IBM's experience and best practices and to a better application lifecycle management process.

Web-based Rational Team Concert for System z software provides users with robust communication tools regardless of where they are located.

You can implement a C/ALM framework by using a C/ALM process template offered by Rational Team Concert for System z software and as a preconfigured process, or database schema, offered by the IBM Rational ClearQuest® enterprise change and defect tracking solution. By adopting C/ALM as the framework for your application lifecycle management methodology, you can take advantage of IBM's experience and best practices, helping you define and implement your application lifecycle management process more quickly.

Future-proofing your software infrastructure with Rational Team Concert for System z software

The days of routinely building monolithic applications in house and from scratch are largely gone. The more usual scenario comprises multiple teams creating composite solutions based on legacy code, distributed solutions and third-party packaged solutions.

Rational Team Concert for System z software is part of a new suite of IBM software engineering and automation solutions built on the IBM Jazz™ open collaboration platform. Rational Team Concert for System z software is a Web-based solution, which means that users have the same communication tools available to them regardless of where they are located. With transparent, real-time access to colleagues and the ability to quickly confer with them in the context of the activity or work item concerned, communicating becomes easier than making assumptions and reigns as the new default behavior.

Highlights

Rational Team Concert for System z software enables a flexible, automated continuous build that can free your teams from administrative tasks and allow them to devote more effort to higher-value business objectives.

An enterprise delivery platform such as IBM Jazz technology can help keep you from hitting the “technology wall” as new deployment options or targets are identified.

The Rational Team Concert for System z software asset repository can be hosted on IBM z/OS, Linux[®], Sun Solaris, UNIX[®] or Microsoft[®] Windows[®] operating systems, and it includes local build agents for the supported environments. Communications among the components are sent via established Internet protocols. Users access the repository using a Web client or an Eclipse-based client, such as the IBM Rational Developer for System z mainframe-based integrated development environment. The collaboration and governance infrastructure, the sophisticated features for synchronized mixed-workload builds and the user-configurable clients deliver a flexible, automated build management solution that can free your teams from lower-value administrative tasks and allow them to devote more effort to the higher-value tasks of timely delivery of new business solutions.

An enterprise delivery platform that is built on open protocols and has the capability to deliver to any combination of target environments can help keep you from hitting the “technology wall” as new deployment options or targets are identified. Rational Team Concert for System z software and the Jazz platform are part of IBM’s strategic direction, and the open architecture can help you future-proof your investment by supporting easy addition of new features and customization as the needs of your organization evolve.

Highlights

Explore www.jazz.net to find out about upcoming enhancements to Rational Team Concert for System z software and to find out more about C/ALM.

For more information

To learn more about how you can improve the delivery of mixed-workload solutions using IBM Rational Team Concert for System z software, visit:

ibm.com/software/awdtools/rtcz



-
- 1 Agile software development refers to a group of software development methodologies based on iterative development, in which requirements and solutions evolve through collaboration among self-organizing, cross-functional teams.
 - 2 Scrum is an iterative, incremental framework for managing complex work, such as new product development, commonly used with agile software development.
 - 3 Streams are persistent groupings of identified versions of components that form a significant version of an enterprise application. Consider streams to be like staging areas that teams use to form a baseline or known starting point for new work. They can also act as the destination or target environments to which work is delivered. For example, you can have a revision 3.2 production stream, a revision 3.3 user acceptance testing stream and a revision 4.0 development stream. Work delivered to a stream is visible only in that stream. You can use multiple, separate streams to isolate incompatible, risky or unpredictable activities so that they can be performed without compromising the integrity of other work.

© Copyright IBM Corporation 2009

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

Produced in the United States of America
November 2009
All Rights Reserved

IBM, the IBM logo, ibm.com, Rational, Rational Team Concert, and System z 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

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

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 documentation is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this documentation, it is 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. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this documentation or any other documentation. 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.

IBM customers are responsible for ensuring their own compliance with legal requirements. It is the customer's sole responsibility to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws.