Building a New Rational Web Site with Rational Suite

by Christina Howe
Director of Internet Services
Rational Software

In April of last year, Rational Software determined that its Web site no longer measured up to the high standards the company had set for itself. Obviously, what was cutting edge technology in 1997 had become insufficient for a company whose identity centered on industry-leading tools and services for e-development! Based on a proprietary infrastructure that the original vendor was no longer supporting, the site could not scale to keep up with Rational’s rapid growth and long-term business needs. We soon concluded that it was time to rebuild from the ground up.

Fortunately, when it came to securing the right tools for my team to do the job, we had to look no farther than our own Rational Suite. In fact, we viewed the complete site overhaul as an opportunity to showcase Rational Suite as an end-to-end e-development solution in a demanding implementation environment.

Using the Suite enabled us to launch the new and improved company Web site (www.rational.com) right on schedule, after two and-a-half months of intensive development. The site includes more than 5,895 Java Server Pages, 8,000 pages of static content, and 14 Enterprise Java Beans supporting 14 new application subsystems. We relied on more than 100 reusable design patterns and Java components for the design and implementation. The site was also more tightly integrated with existing back office systems within the same timeframe, dramatically improving both customer self-service and other e-business capabilities.
The account that follows not only illustrates how Rational Suite helped our Web development team build a reliable site very quickly; it also explains how each Suite product supports specific activities during the development lifecycle.

RUP Unifies Efforts of Distributed Team

Why did we have such a tight time frame for this project? Rational has two major software releases each year that occur in our first and third financial quarters, respectively, and we wanted to have the new site up and running well before the second release cycle. In addition, we wanted to have the site up in time to support Rational's annual User Conference, scheduled for the end of August 2000. As the leader of this effort, I knew from experience that if we hoped to make that tight deadline, we'd need to observe equally tight organizational practices.

We made the decision to supplement our in-house team of Web developers and engineers, who specialize in Java development, by engaging an outside vendor to assist in the systems integration for the Web site. We chose Context Integration, a Burlington, Massachusetts-based Internet professional services firm and a member of Rational's Unified Partner Program who uses the Rational Unified Process (RUP®). RUP is a software engineering process that enhances team productivity and puts the experiences of thousands of projects into the hands of each project member through its embedded best practices. Because it's totally Web-enabled, everyone could access what it had to offer through a browser.

Together, the Context and Rational teams used RUP to develop guidelines, templates, and examples as we moved through each of the RUP's critical phases: Inception, Elaboration, Construction, and Transition (see Figure 1). Because RUP is tightly integrated with other Rational tools, the entire combined development team could reap the benefits of using Unified
Modeling Language (UML), software automation, and other industry best practices.

As Matthew Burnett, Context's project manager for the Rational Web site redesign put it, "When developing software this fast, it is essential that you follow the Rational Unified Process principles. Because we were working from a shared vision and using the Rational Unified Process, the Inception and Elaboration phases went smoothly."

In fact, throughout every phase, RUP enhanced communication among all the developers and engineers by providing the distributed team with one knowledge base, one modeling language, and one view of how to develop the content and infrastructure software to support the Web site.

**Rational Requisite Pro and Rational Rose: From Inception Through Construction**

During the Inception phase of the new Web site development, Context performed a comprehensive evaluation of the application and a thorough technology assessment of personalization servers currently on the market. We selected a server and then headed into a four-week Elaboration phase. This included developing a prototype of the site and using Rational Rose to...
Availability (IBM Hosting) 99.999 percent

<table>
<thead>
<tr>
<th>Availability (IBM Hosting)</th>
<th>99.999 percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java Server Pages</td>
<td>5,895</td>
</tr>
<tr>
<td>Static Content Pages</td>
<td>8,000</td>
</tr>
<tr>
<td>Enterprise JavaBeans</td>
<td>14</td>
</tr>
<tr>
<td>Application Subsystems</td>
<td>14</td>
</tr>
<tr>
<td>Reusable Design Patterns and Java Components</td>
<td>100</td>
</tr>
<tr>
<td>Page Views/Month</td>
<td>4,708,508</td>
</tr>
</tbody>
</table>

The Rational Unified Process (RUP) is used throughout to guide development, unify team efforts.

Using the UML enabled the entire Context team to comprehend and provide feedback on the overall system architecture. By modeling, we created artifacts that the entire team understood and that were not subject to individual interpretation. This, in turn, enabled us to generate whatever other documents and deliverables we needed via Rational SoDA, in less time than it might otherwise have taken. During the Construction phase, Rational Rose saved the team a significant amount of time by generating code automatically, based on the architecture. In addition, throughout these phases, the rapid prototyping methodology embodied in the RUP and the Suite tools was invaluable when it came to tackling high-risk design challenges, such as the back-end integration of the Web site’s e-commerce capabilities with our Rational legacy infrastructure.

Rational ClearCase and Rational ClearQuest: Tight Controls for Code and Content
Both Context and our in-house team used Rational ClearCase as the project's CM (Configuration Management) solution and Rational ClearQuest for defect tracking and workflow management. Because we had so many publishers, Rational ClearCase's versioning functionality was particularly useful, and it gave us the ability to store every file connected with each version, not just source code. Together, Rational ClearQuest and Rational ClearCase gave us a single, comprehensive platform for efficient Web content and code management.

Of course, although the design and content development team knew a lot about manipulating and storing HTML, GIFs, and JPEGs, ClearCase and ClearQuest were new tools to them. So we brought in our own Rational consultants to provide on-the-job training and classroom sessions to help the entire team achieve a core competency with these tools. As the development effort progressed, everyone appreciated the structure the tools provided.

Rational ClearCase stores all project work in a shared Versioned Object Database (VOB). With access to this centralized repository, team members could work on related development activities simultaneously. They'd check out parts of the project and work on them in their own private, highly flexible workspaces: Content publishers can either view their work in relation to changes made by other team members or keep it isolated until all changes are complete and checked in.

This was a great advantage because our team was cross-functional and widely distributed, with people on both the East and West Coasts. We had content developers, designers, Web developers, and software engineers all working in concert, and every one of them had access to the assets stored in the Rational ClearCase database. In addition, because Rational ClearQuest is Web-enabled, everyone had access to browser-based defect and change tracking. This was invaluable for coordinating the team's efforts.

<table>
<thead>
<tr>
<th>Tool</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rational Rose</td>
<td>Inception through end of Construction to visually model site prototype, and then create detailed analysis and system design</td>
</tr>
<tr>
<td>Rational ClearCase</td>
<td>Throughout for configuration and code/content management</td>
</tr>
<tr>
<td>Rational ClearQuest</td>
<td>Construction, Transition, and Post-Deployment for defect tracking and change management</td>
</tr>
<tr>
<td>Rational Suite TestStudio</td>
<td>Construction for GUI functional testing, load and performance testing; Transition and Post-Deployment to test for and resolve problems</td>
</tr>
</tbody>
</table>
Because ClearCase and ClearQuest are also integrated with Rational RequisitePro, our team managers always had a clear understanding of what they needed to control. We had instant access to precise specifications about what would or would not be in the Web site, and ClearQuest updated them automatically whenever a change was made.

**Testing the Site**

Rational Suite TestStudio supplied invaluable support for testing the new Web site. As soon as Context generated the system requirement with Rational RequisitePro, our testing team began creating its own test requirements and plans based on those specifications.

Then, as we moved into the load-testing phase, we used Rational Suite TestStudio tools to record basic functional testing for our GUI (Graphical User Interface) and to create virtual users for load and performance testing. We were able to record a wide variety of test cases that could be rapidly executed at any time during the development cycle. Integration of this kind can be frustrating to achieve because it involves a highly repetitive "test and fix" process, and each time you fix a bug you must do regression testing to ensure that you didn't break the rest of the site. We had to repeat our tests many times to verify integration with the back-end systems. But each time we corrected an error, Rational Suite TestStudio's regression-testing capabilities allowed us to quickly regenerate and run all the scenarios to ensure that the bug fix didn't break the rest of the site.

Because of our schedule constraints, the team started load and performance testing in our staging environment, even before the production environment was completed. Based on Rational's previous Web site statistics, we were able to profile the site's projected user base as well as usage patterns: Our old site had received 19.5 million hits and provided 2.2 million page views on a monthly basis. The average page views per day were 75,838.

With user profiles and patterns clearly defined, the team was able to generate a set of virtual user groups that performed various virtual activities. One group, for example, simply browsed the site, while another used its e-commerce capabilities. This testing enabled us to discover and address performance bottlenecks in a controlled and iterative fashion. That, in turn, helped us to move into production far ahead of where we would normally have been, given how complex the development process was.

Even after the site went live, the development team continued to use Rational Suite TestStudio to test and resolve problems that cropped up. It was very helpful to have all our virtual users scripts already created to load test the site, identify any problems, and be confident that the fixes did not cause further problems.

**Services Were Essential**

Without Rational e-development Services we could not have made our deadlines. Having training and support from day one of the development
lifecycle saved a significant amount of development time and resulted in a much smoother implementation. As specific issues came up during the course of the project, my team turned to Rational consultants and Technical Support for expert help and advice.

Throughout, Rational's training and support organizations transferred knowledge to the entire team, including invaluable lessons on using testing tools. We not only enjoyed the short-term benefit of accelerating this particular implementation, but also gained a broad set of skills that we can apply in future development projects.

**Better Quality Despite the Tight Timeframe**

Our new Web site offers Rational customers a far better user experience than our old one: greater ease-of-use, faster response time, and a high-level of personalization. Visitors can now obtain individualized content on support issues, information on appropriate courses and schedules at Rational University, and more.

Site availability is also much better. The site is run out of an IBM data center in Southbury, Connecticut, and IBM hosting provides us with 99.999 percent availability as well as the latest in disaster recovery and security capabilities.

Since the launch of the new www.rational.com, 86,000 people have registered and established profiles to receive personalized content. In addition, the new site currently generates 4,708,508 page views per month, which represents a 76 percent increase over the previous version. My team takes great pride in these numbers.

As for me, I'm really pleased I had the opportunity to lead my team in this effort, which demonstrated that Rational tools really do help you get the job done faster without sacrificing quality. Next time around, I look forward to addressing more of the content management issues using the new Rational Suite ContentStudio, which is built on Rational ClearCase and also includes Vignette's V/5 Content Management Server. The integrations between these tools give you a unified system for managing Web code and content -- and that will make our job even easier.

For more information on the products or services discussed in this article, please click [here](#) and follow the instructions provided. Thank you!