IBM

Michigan's largest health benefits plan provider doubles development capacity while increasing on time, on budget projects with IBM Rational.

Overview

■ **The Challenge**
Facing increased competition from for-profit health insurance companies, Michigan's largest health benefits plan provider sought to deliver new services to members, providers, and group agents. The company needed a proven development process supported by an integrated software development platform.

■ **The Solution**
The company implemented IBM Rational Unified Process, or RUP, and IBM Rational solutions for requirements management, visual modeling, and performance testing.

■ **The Benefit**
Since adopting the IBM Rational Software Development Platform, the company's development team has doubled the number of projects it can complete in a year, while increasing the rate of delivering projects on time and on budget to 90 percent.

As a private, nonprofit corporation, Michigan's largest health benefits plan provider holds a unique position within the state's health insurance industry. Yet, it still faces growing competition from for-profit companies in the market. To meet these competitive challenges while continuing to assure its subscribers reasonable cost and access to quality health care, the company sought to improve the efficiency of its information services (IS) initiatives. Specifically, it wanted to significantly increase the number of IS projects completed each year and better align IS initiatives with business needs. At the time, it needed to maintain the high standards for quality and compliance required by the Health Insurance Portability and Accountability Act (HIPAA) and other government regulations.

"The healthcare market has become significantly more competitive in recent years," explains Anthony Callan, the company's methodology coach. "As an organization, we realized that to stay competitive with these smaller organizations we would need to improve our services. "For information services, there was a big push to complete more projects every year. Our goal was to double the number of projects we had done the previous year without adding staff. At the same time, we wanted to increase the number of projects we delivered on time and on budget."

To help the information services organization achieve these goals and better govern its software development efforts, the company adopted the IBM Rational Unified Process® (RUP®) methodology supported by solutions from the IBM Rational® Software Development Platform, including IBM Rational RequisitePro® for requirements management, IBM Rational Rose® Developer for visual modeling, and IBM Rational Robot for load testing.

New development process
As a health insurance provider, the company relies heavily on its IS organization to develop and maintain its core claims processing systems.

These systems are based largely on mainframe software, but are also accessed through new Web-based systems that provide information and services to members, health care providers, and group administrators.

In this environment, the IS team recognized that implementing a consistent and repeatable development process was essential to achieve the productivity increases they sought. "To be more productive, we needed to find an optimal way of completing projects, and we needed to establish a repeatable way of developing systems. In the past, our development processes were generally waterfall, but we had multiple ways of handling artifacts and documentation on different projects. We saw the RUP methodology as a more consistent and repeatable way of developing systems," says Callan.

The iterative approach of RUP also provided several advantages over the team's waterfall methodology, in which each phase of the project must wait for the previous phase to complete before it can begin. "Iterative development has been very beneficial in the Web space for scheduling the projects, identifying and addressing the risks we have, and getting big projects to move," Callan explains. "For some projects, we find we do not have all the information we need when the project starts. With an iterative approach, we can move forward on several of the use cases, and push others down the line. In a waterfall project, if we didn't have all the information up front, we couldn't move the project."

The IS team configured RUP to meet the specific needs of its Web development teams. In addition, the team has created a second configuration to help its mainframe developers rapidly adopt many of the best practices of RUP. "In the mainframe space, it is more difficult to schedule projects because there are resources, such as test environments, that are not available at all times. To simplify planning across multiple projects, we created a hybrid version of RUP that does not require iterations. In this way, the mainframe teams are getting up to speed on the Unified Modeling Language (UML), use cases, and other RUP artifacts, so that as we increase the number of mainframe projects using iterative development, the transition will be easier," says Callan.

Managing requirements and use cases
In the inception phase of projects, the requirements team has started to use Rational RequisitePro to define requirements and trace them throughout development. Naren Pallaseni, a business application designer at the company, works with project stakeholders in the initial phases of a project to capture business requirements, identify use cases, and develop use case specifications in Rational RequisitePro.

"Instead of using a cumbersome spreadsheet to manage requirements, I have an effective tool to capture all the business requirements and link them to functional requirements, which I derive from the use case specifications. With Rational RequisitePro, we are better able to deliver applications that meet business requirements and enable the business to achieve its objectives. Going forward, we will expand our use of Rational RequisitePro and link our test cases to the functional requirements, so we can track requirements throughout development and testing," says Pallaseni

Pallaseni's team is also responsible for creating use case diagrams in the Unified Modeling Language using IBM Rational Rose Developer for Java™. These diagrams provide a graphical representation of the use case specification used by the design, development and testing teams as a basis for developing the system design and test cases. "Rational Rose Developer really helps us to efficiently capture the use cases, document them, and communicate them to the team," he adds.

Design and construction

Like the requirements team, the design team is also involved in the earliest stage of each project. Designers use Rational Rose Developer to model the conceptual design based on an overview of the project. "Our conceptual design is like a high-level package diagram," explains Tim Barnette, a senior application developer on the design team. "The design is done in sketch mode in the beginning and, over iterations, it becomes a UML blueprint model of our application architecture."

Later, when the design team receives the use case specification and use case models from the requirements team, designers begin to develop additional UML diagrams, drilling down a from high-level description to very specific design details. On Web and client server projects, the team uses Rational Rose Developer to develop package diagrams, as well as class diagrams and sequence diagrams. Mainframe projects typically involve more activity diagrams. According to Barnette, visual modeling provides far reaching benefits to the entire development organization. "With Rational Rose Developer, we are getting a clearer picture of the enterprise architecturally, and that is enabling us to make better decisions," he notes.

Streamlined quality reviews

IS teams conduct two major quality reviews during the development of each project,

one after the design is complete and one following the completion of the code. Barnette cites the design quality review process as a key area that has been improved through visual modeling. "When our applications are modeled correctly, we have much more reviewable artifacts than we have had traditionally. In the past, teams have brought in code for a quality review of the design. Of course, at that point it is no longer a design, it is an implementation. The person performing the review was not always familiar enough with the code to make a proper assessment of the design decisions made in the process," says Barnette.

"When we use Rational Rose Developer to model the design, we can easily see if the design decisions are correct. In addition, when we model consistently, it is easy to see dependencies by looking at the model. When we change one piece, we know what impact that will have on other pieces, and without modeling, that is just not the case," he adds.

Rajiv Das, an application development manager at the company, agrees and notes that the models are also an advantage during the subsequent quality review conducted as development completes and testing is about to begin. "Rational Rose Developer is really helpful at the development

quality review as well, because it makes it easy to verify our Java methods and classes by comparing them to our models."

Accelerating development with new solutions

The IS team is in the process of expanding its use of the IBM Rational Software Development Platform in a number of areas to further increase productivity and accelerate development. The design team is transitioning from IBM Rational Rose Developer to Rational Software Architect. Likewise, the Web development team is in the process of migrating from IBM WebSphere® Studio Application Developer to another Eclipse-based integrated development environment, IBM Rational Application Developer.

The team plans to leverage Rational modeling solutions to automatically generate Java code and jumpstart development. They are also planning to reverse engineer models from existing code to further streamline quality reviews. "One of our goals in the next six months is to use Rational products to automatically generate code from our technical designs," Das notes. "We also have a significant amount of code that was written before we began using modeling tools. We need to reconcile that code with our designs. We plan on using Rational modeling products

to reverse engineer this code and ensure our design consistently reflects the implemented code."

As in the past, if the team needs additional help as they implement Rational solutions, they can rely on IBM Rational consulting services for support. Callan notes, "IBM and IBM Rational consultants have been excellent with training, installation, and deployment. We have used them quite a bit and they have always been able to answer our questions and come up with ideas on how we can use the products more effectively. They have helped us in all aspects of what we need to do."

20,000 virtual testers
One of the advantages of iterative development is the ability to start the testing process early in the development life cycle. "We get started in the inception phase," says Das. "A typical deliverable at inception is the test strategy document that ensures that everyone is on the same page— the business and IS." Developing a comprehensive test strategy is crucial to ensuring quality, because the company's systems operate in a complex IS environment, comprised of Web front-ends, IBM WebSphere MQ middleware, and

mainframe applications, as well as IBM DB2® and other databases.

With this environment, performance testing is imperative to ensure that applications spanning multiple systems will continue to respond in a timely manner under heavy user loads. The testing team uses IBM Rational Robot to conduct extensive load testing of systems—identifying and resolving bottlenecks before the applications are released. The tests provide insight into how the system will perform under projected user loads, and at what loads system performance begins to slow. "We use Rational Robot for scripting, execution, and monitoring of our performance tests. We simulate usage with virtual users. For systems used by providers, for example, we can simulate the number of users we currently have, as well as those projected in growth estimates provided by the business. In some cases, we have measured performance by simulating up to 20,000 simultaneous users," says Das.

The team's use of the IBM Rational Software Development Platform is expanding in this area as well. In addition to linking test cases to functional requirements in Rational

RequisitePro, plans are underway to minimize manual testing processes by using IBM Rational testing solutions to automate functional tests.

Capacity doubled, and re-doubled
Substantial improvements in efficiency and productivity are needed for a development organization to complete twice as many projects in the same amount of time without significantly adding staff. Since adopting the IBM Rational Development Platform, the company has done just that. "Last year, our goal was to double our capacity, to complete 60 projects in one year. With the help of IBM Rational solutions, we achieved that goal. This year our goal is to double that again—to quadruple our capacity of two years ago—and we are getting close to that mark as well." says Callan.

Callan adds that increasing capacity did not come at the expense of budget overruns or schedule slips. "At the same time, 90 percent of our projects that ran under RUP were completed on time, on budget. The rate for projects not using RUP was 75 percent," he reports.

Not resting on these accomplish-
ments, the development team
continues to drive improvements
in efficiency and governance that
will enable the company to gain a
competitive advantage as it fulfills its
unique responsibility to the people of
Michigan. "With RUP, we now know
that we have a repeatable process.
We are now continuing to optimize
our use of Rational products, to
leverage their full value by linking
requirements, visual modeling, and
testing together to ensure that we, as
a project team, can further enforce
and support the methodology we
have in place."

For more information, visit:
ibm.com/rational

IBM®

G210-2131-00