

▶ **The RUP: An Industry-wide Platform for Best Practices**

by **[Per Kroll](#)**

Director, Rational Unified Process Development and Product Management Teams

The Rational Unified Process® (RUP®) is a comprehensive, Web-enabled set of software engineering best practices that provide customers with guidance for streamlining their teams' development activities. Earlier this month, Rational announced its collaboration with a broad range of key industry leaders, including IBM, Microsoft Corporation, and Sun Microsystems, to expand the Rational Unified Process into an industry-wide process platform. This article presents the Rational Unified Process as an evolving platform that facilitates software development, and it examines the new capabilities in the latest version of the RUP.

The Rational Unified Process product has evolved from a software engineering process deployed as a Web site providing Rational's guidance on best practices, to an industry-wide platform for best practices. But what do we mean by an industry-wide platform? Essentially, we mean a platform that facilitates software development for any size organization, and which provides -- or allows the addition of -- specialized content for a wide variety of software markets, without overwhelming a practitioner with irrelevant procedures. Today, the software industry's most innovative pioneers including IBM, Microsoft, and HP use the RUP process platform to capture and document their know-how and best practices. They rely on the RUP platform as a distribution mechanism for these best practices; it is a vehicle through which thousands of projects and end users consume and adopt them. In this article we will:

- Discuss the business and technology drivers that have defined an accepted framework of best practices.
- Look at the emergence and evolution of the RUP.
- Examine the underlying components of the RUP platform.



- ▶ [subscribe](#)
- ▶ [contact us](#)
- ▶ [submit an article](#)
- ▶ [rational.com](#)
- ▶ [issue contents](#)
- ▶ [archives](#)
- ▶ [mission statement](#)
- ▶ [editorial staff](#)

Business and Technology Drivers

Several industry trends have driven the development of Rational's industry-wide platform for best practices. Some of the most influential are:

- Rapid introduction of new technology and evolution of existing technology.
- Standardization and commercialization of tools, methods, and process.
- The trend by consulting companies to no longer include a proprietary process as a competitive differentiator.
- Industry skepticism regarding traditional and heavy-weight processes that are dogmatic and manager focused.

These trends are discussed in the following sections.

Rapid Evolution of Technology

It is very difficult for developers to keep up with constantly evolving technology, and lack of knowledge is becoming a major impediment for the adoption of new technology. To address this issue, vendors have increased their focus on guidance and best practices in the marketplace, with the goal of effectively distributing their advanced tools and technologies onto a developer's desktop. Examples of such solutions include MSDN, Oracle Technology Network, and Vignette Global Marketplace. The goal is to put the knowledge that developers need just a mouse click away on their desktop.

Standardization and Commercialization of Tools, Methods, and Process

Over the last 30 years, the software engineering industry has continuously moved toward standardization and commercialization of technology and knowledge. This meant that in the 1960s and 1970s, commercial compilers become available; in the 1980s, CASE tools and databases; and in the 1990s, advanced configuration management systems and IDEs. In the mid 90s, we also saw standardization in the methods and modeling language area. The Unified Modeling Language (UML) was originally developed by Rational and its partners, and later adopted and managed by OMG as an industry standard. By the late 90s, companies were ready for standardization of process, and many companies abandoned their homegrown process and began looking for a commercially available one. This allowed the large investments necessary to build an enterprise-wide process, such as the Rational Unified Process, to be shared by many companies. Companies that have standardized on the RUP, for example, include CGE&Y and Merrill Lynch.

Consulting Companies Less Reliant on Process as a Competitive Differentiator

Similarly, while standardization has resulted from the commercial availability of computing technology, the competitive differentiation based on proprietary processes that consulting companies once relied on has also become much less

significant as a business driver. Process used to be a selling factor for any consulting company worth its name, but as customers got tired of project overruns and the poor quality of delivered applications, they started to demand some assurance that the practices used by system integrators were proven. Since it is extremely difficult to assess the applicability and value of a homegrown process, customers started to demand commercially available processes with a proven track record, and which could be evaluated more easily by independent reviewers. As a result, there has been a rapid move among system integrators away from homegrown processes to commercially available processes. The RUP has been the process of choice for many of these companies, including CGE&Y, Deloitte Consulting, and IconMedialab.

Industry Skepticism Regarding Existing Processes

At the same time, the software development industry became skeptical toward traditional and heavyweight processes that are prescriptive and have a strong primary focus on the needs of managers. These processes typically promoted a "waterfall"¹ sequence of development, functional decomposition, and a document-centric approach. These processes had been popular in the late 80s and the 90s, and proved to be ineffective for several reasons:

- *Processes were of little value to developers.* Since the guidelines primarily focused on describing *what* should be done, and not *how* it should be done, developers saw little value in them. Since it did not help them do a better job, developers resented this type of process.
- *The waterfall approach did not allow effective risk management.* Even worse, since these processes typically followed a waterfall approach, which is ineffective in addressing key risks early in the project (regardless of whether the risks are technical or business related), the project success rate declined as technology evolved and projects became increasing complex and risky. The industry was ready for a change.
- *The process was hard to access.* Processes were often published in thick binders, and the content was not integrated with desktop tools. This made it hard to find the information being sought, so process binders simply collected dust.
- *The process did not focus on delivering value to customers and other stakeholders.* The waterfall-based and dogmatic nature of the processes focused on many artifacts, on heavy documents and activities that did not help deliver real business value to end users and other stakeholders.

The Emergence and Evolution of the RUP

It was in this business climate that the Rational Unified Process gained popularity. From a product development standpoint, the RUP has gone through two complete product phases and is now moving into its third phase (see Figure 1).

Phase 1: 1996-1999

The RUP process development team² was launched in February 1996. It was

essential that the process should be iterative, architecture-centric, and use-case driven, since these were some of the main best practices that Rational had identified from its work with its customers and partners. Because Rational has always also had an extremely strong focus on the needs of the practitioner, the RUP process development team believed that the process had to be nonintrusive to the practitioner, and that the average analyst, developer, and tester had to find it was easier to do their jobs with the RUP than without it. These benefits may seem obvious, but prior to the RUP's introduction, most processes could not be described in these terms. The RUP team also realized that the technical process had to be anchored by a solid project management (macro) process to guarantee project success. The macro process needed to ensure that risks were addressed early, that projects were run cost effectively, and that overall business objectives were met.

Having set these objectives, we then took on the task of harvesting and documenting Rational's know-how into one consistent knowledge base during the years 1996 through 1999. Rational already had a (mainly unpublished) process called the Rational Approach, and through acquisitions of other companies we had gained access to the Objectory Process³ as well as additional processes centered on requirements management, testing, and other areas.

By 1999, the Rational Unified Process covered the full lifecycle, from Business Modeling through Deployment, as well as areas such as Configuration Management and Project Management. Furthermore, the RUP combined a technical process that added value for practitioners -- rather than impeding their ability to do their jobs -- with a macro process that ensured that overall business objectives were met.

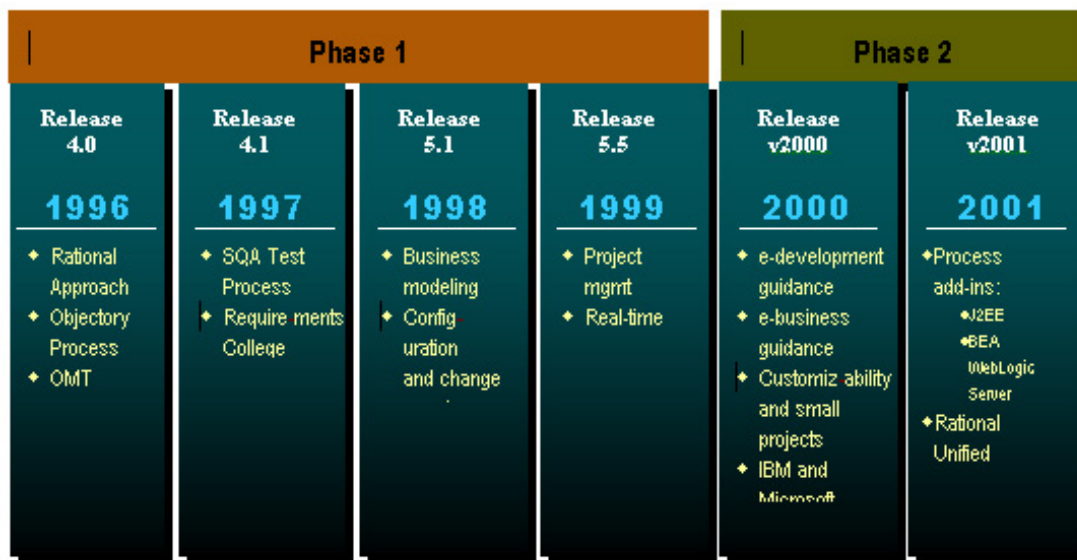


Figure 1. RUP Product Development Has Gone Through Two Distinct Phases. *In Phase 1, we integrated content within Rational to create a consistent process covering the full lifecycle. In Phase 2, we collaborated with a variety of industry leaders to provide in-depth guidance around technology such as leading development platforms.*

To ensure practitioner value, the RUP was also tightly integrated with practitioner tools. Sales soared, and the RUP became the most widely used process for iterative and component-based development. Its success could be attributed to the fact that the RUP was Web-enabled, easy-to-use, and

nonintrusive.

Phase 2: 2000-2001

By 2000, the Rational process development team had captured all its internal know-how and had started to look elsewhere for more specialized expertise. We already had a full-lifecycle process and were looking to add more targeted, specific value for developers. After discussions with many developers, it was obvious that we needed to provide more detailed guidance on developing specific types of applications. We found that developers wanted to know how to develop applications on the J2EE or WinDNA platforms. They wanted to know how to use the RUP to develop e-business applications. But we needed outside help to provide best practices in these areas.

The solution was to collaborate with other companies. We worked with IBM, Sun, and BEA to gain expertise on how best to build applications on the Java platform and build e-business applications. We worked with Microsoft and AIS (Applied Information Sciences) to understand how best to build applications on the WinDNA platform.

Dilemma and Opportunity: The Move Toward Phase 3

RUP sales continued to accelerate, and the rate of adoption was greater than we ever had dreamed possible. But along with our success, we had a problem accommodating the enormous interest the RUP was generating. After one year into Phase 2, we had more partners and customers wanting to collaborate with us on developing new content than we had the bandwidth to handle.

At the same time, two other issues arose. On one hand, RUP users were looking for a variety of very high-level specialized knowledge and wanted Rational to add more content in areas such as commercial, off-the-shelf software development, creative design, content management -- and the list just got longer as we added new content. On the other hand, as we added more content, it became harder for some users to know where to start, given the volume of material now included in the RUP. And the architecture of RUP did not allow us to address both of these problems at the same time.

To resolve this dilemma, we started plans late in 2000 for Phase 3, which would develop the RUP into an industry-wide platform for best practices -- a process that could meet the needs of even the most specialized software development organizations, yet remain accessible to the many types of practitioners working on various phases of the software development life cycle.

A Closer Look at the Industry-wide Best Practices Platform

The initiative to make the RUP a broadly accepted process platform had the following objectives:

- *Right-sizing*: Allow projects to "right-size" the process they use; that is, allow projects to use "just enough" process. Users should be able to start with a small process, and as project needs expand, to grow the

process to address those needs.

- *Process guidance*: Make a wide array of process guidance available to RUP users, including specific guidance on how to develop software for a variety of architectures, target devices, and hardware/software platforms. The best way to achieve this is to make it easy for platform and tool vendors, system integrators, and other industry leaders, to make their know-how available in RUP format.

Today's RUP platform consists of four major components:

1. An infrastructure that allows Rational partners and customers to build and package best practices into process components
2. A distribution and marketing channel for process components
3. Deployment mechanisms for process components allowing you to "right-size" the process you use
4. A framework for accessing best practices, and which is tightly integrated with practitioner tools

Let's take a closer look at each of these components.

An Infrastructure for Building and Packaging Best Practices

One of our objectives is to increase the amount of specialized content available to RUP customers. To accomplish this, we are taking advantage of two phenomena that we have observed over the RUP's six-year history. In the first place, Rational partners typically have an interest in developing specialized content to make their customers more successful in using their tools, or to advance their thought leadership in a certain technology or domain of expertise. Normally, it is in their interest to distribute this content at no cost, but some may want to charge for it. Second, RUP customers typically want to develop specialized content to address their specific, internal needs. Some of this content may be of special interest only to a given company; but in some cases the content may be of interest to other companies as well, and they may choose to distribute it to RUP users outside the company either free of charge or for a fee.

Componentizing the RUP

To enable partners and customers to develop RUP content independent from one another, we developed a component-based architecture for the RUP. To describe this architecture, we needed to introduce some new concepts:

- A **RUP Base** contains process elements that are likely to be useful for all projects, and which capture some of the fundamental principles of the RUP, such as iterative, use-case driven, and architecture-centric development. A RUP Base can be extended with RUP Plug-ins.
- A **RUP Plug-in** is the deployable unit for one or several process components that can be readily "dropped" onto a RUP Base to extend it.
- The **RUP Process Framework** is an extensible architecture for process

definition. It provides:

- A systematic means for decomposing and capturing process knowledge into well-defined (typed) process-definition elements, such as role, artifact, activity, guidelines, concepts, and so on.
- A set of rules and policies to assemble and organize these elements into a cohesive customized process.
- An extensible process template to serve as a basis for process authoring.

The architecture of the RUP process framework is based on the Software Process Engineering Meta-model (SPEM, an OMG specification and UML domain model⁴), and its extension is supported by a set of tools: Rational Process Workbench⁵ and RUP Builder. It includes a RUP Base.

- The **RUP process platform** comprises the RUP Process Framework, the supporting toolset, and a set of ready-made process plug-ins.
- **Core RUP** consists of the RUP Process Framework plus the RUP Plug-ins that are fundamental to software engineering and within the expertise of Rational Software. Core RUP is what we typically ship on the RUP product CD.

This division of the RUP into a *RUP Process Framework* (containing a *RUP Base*) and a number of *RUP Plug-ins* makes the product far more flexible. The very nature of component-based architecture, along with the well-defined guidelines provided by the RUP Process Framework, allows a variety of companies to develop RUP content independently of each other.

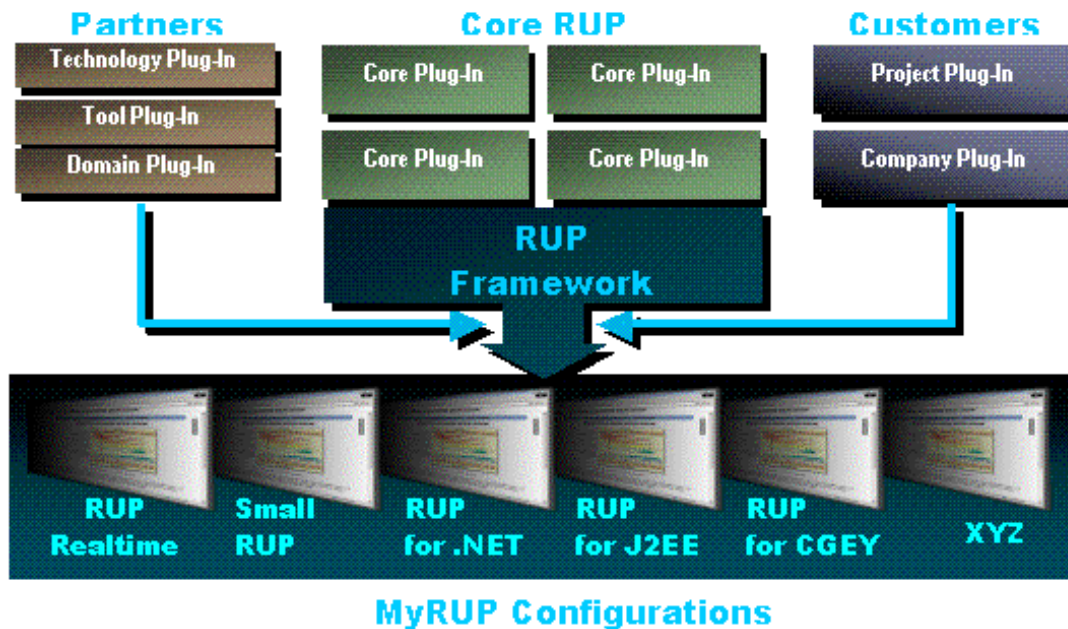


Figure 2. RUP's Component-based Architecture Facilitates Independent Plug-in Development. The component-based architecture of the RUP allows partners and customers to independently package their know-how into plug-ins. Customers can now choose from a wide variety of plug-ins and deploy those that are appropriate for their project.

As shown in Figure 2, partners can now package their know-how of a certain technology, tool, or domain into a RUP Plug-in. Customers can also take advantage of this technology to produce plug-ins that are specific for a project, a division, or their entire organization, thus further leveraging their investments in .NET, J2EE, or other development and deployment platforms. This allows companies with large numbers of software developers to put their vast know-how at the fingertips of all their software engineers.

Later we will see how a RUP user can determine which plug-ins to deploy for a certain project.

An Industry Standard for Process Authoring

About two years ago, we started to work with IBM on a standard for process authoring. The starting point was the meta-model for RUP and IBM Global Service's processes. We later brought this work to the Object Management Group (OMG), a standards body that owns, among other things, the Unified Modeling Language (UML). Through close collaboration with many other companies, this work evolved by June 2001 into the SPEM OMG standard for process authoring noted earlier.

SPEM is supported by the RUP and Rational Process Workbench, our own process-authoring tool. SPEM provides an intellectual foundation for the capabilities of Rational Process Workbench. And as an industry standard, SPEM offers users a common, underlying structure and terminology for processes, making it easier for people to build RUP Plug-ins.

Building RUP Plug-ins

With SPEM as the theoretical foundation and Rational Process Workbench as the process-authoring tool, partners and customers can now package their know-how into RUP Plug-ins. Rational Process Workbench is a Rational Rose add-in that allows you to take advantage of the power of UML and visual modeling to model the various process elements you have. New capabilities in the Process Workbench allow you to define how your process elements should extend or override process elements in the RUP Framework. The Process Workbench also allows you to package your process elements, and the definition of how they extend and override process elements in the RUP Framework, into a RUP Plug-in.

Process engineers who use Rational Process Workbench appreciate its power, but it should be made clear that process authoring and using the Process Workbench require a certain level of expertise. You need to be familiar with Object-Oriented modeling, Rational Rose, and the RUP to build a RUP Plug-in.

To assist the process engineer in building RUP Plug-ins, we offer tutorials and training material, as well as referrals to partners willing to build specialized Plug-ins for a fee.⁵

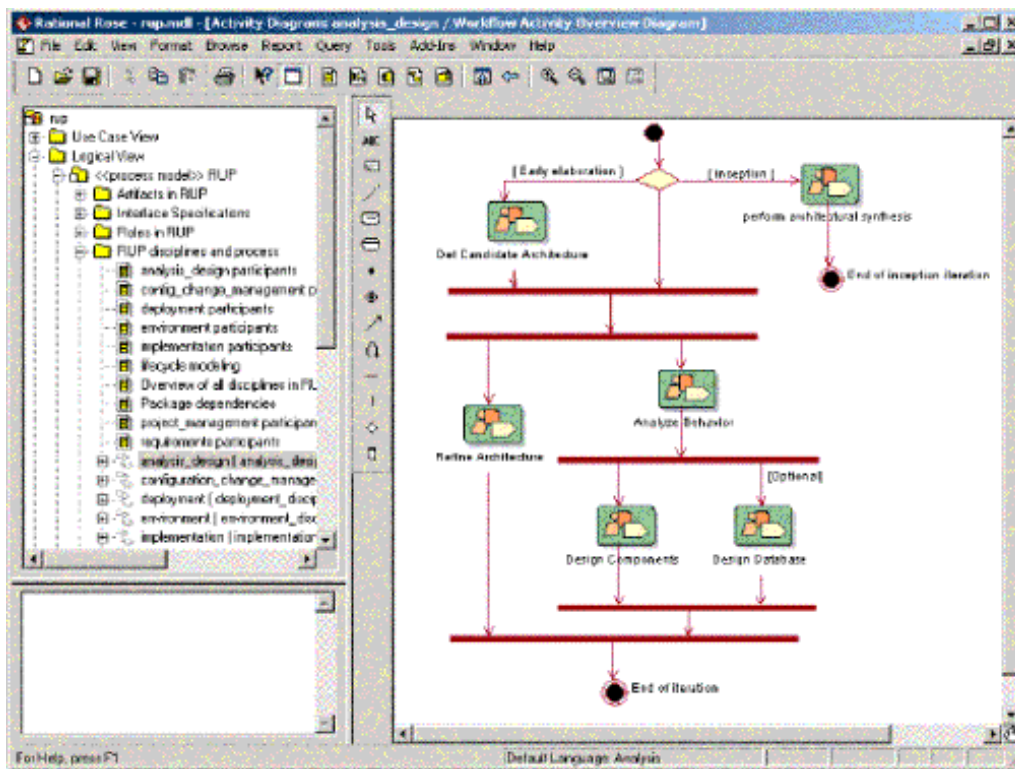


Figure 3. Rational Process Workbench Exploits the Power of Visual Modeling. *Rational Process Workbench is a process-authoring tool built as a Rational Rose add-in. It brings the power of visual modeling and UML to process authoring, and allows you to build RUP Plug-ins.*

Distribution and Marketing Channel for Process Components: The RUP Exchange

The Rational Developer Network⁶ is an online community and knowledge resource for Rational customers worldwide. One of the Rational Developer Network's services is the RUP Exchange, which functions as a distribution channel for RUP Plug-ins, as well as an information source for building Plug-ins.

The RUP Exchange contains a hyperlinked list of currently available RUP Plug-ins. This list is continually updated as Plug-ins become available from Rational, our partners, and our customers. Clicking on a link to a Plug-in gives you its description and allows you to download it to your desktop.

Developers can find the latest guidelines for designing and building Plug-ins on the RUP Exchange. Once a Plug-in is built, it can be included on the RUP Exchange by using the RUP Plug-in upload page. Making your RUP Plug-ins available to other RUP users can be of significant value to the RUP community, and it can give you and your organization exposure as experts in a certain technology, tool, or domain.

The RUP Exchange also lists companies that will help you package your knowledge into a RUP Plug-in, for a fee. You can get help in a wide array of areas -- for example, using Rational Process Workbench for modeling your Plug-in, or transforming Word files into the Plug-in format.

You can find the RUP Exchange in the RUP Knowledge Center within the Rational Developer Network. There you will find RUP-related discussions,

various software engineering articles, and other material of interest to project members.

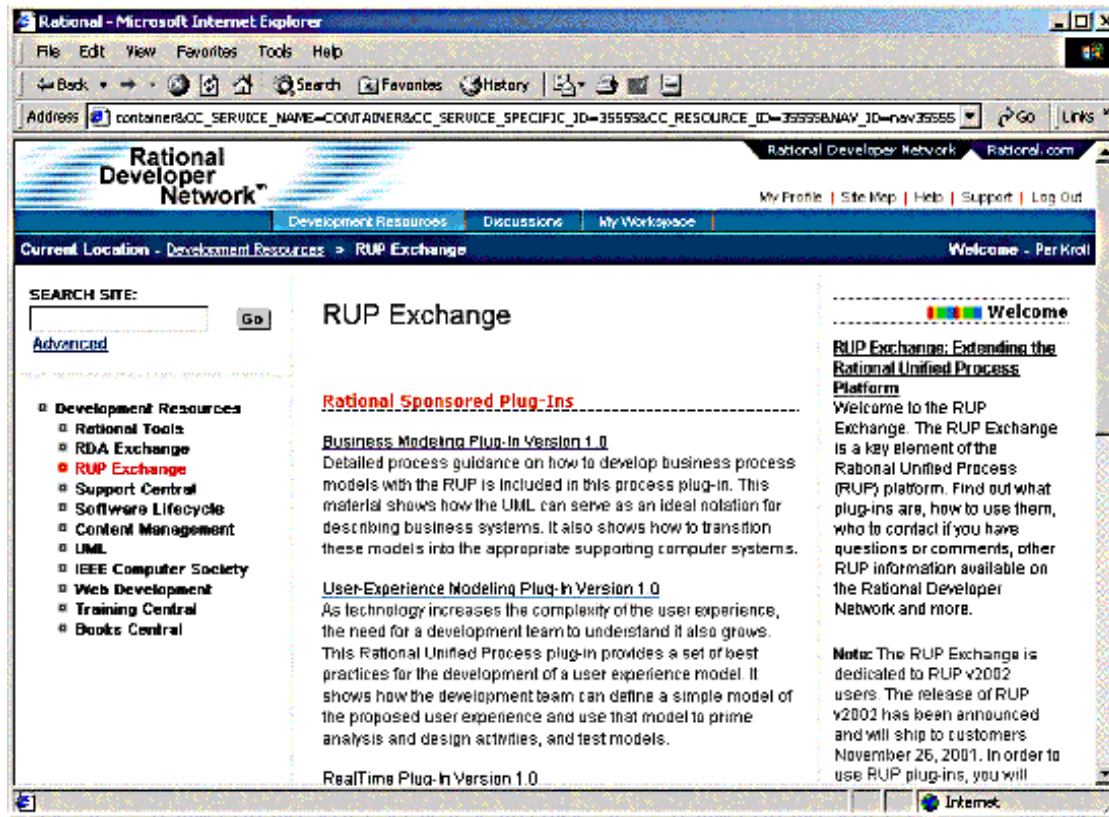


Figure 4. The RUP Exchange on the Rational Developer Network Provides The Latest Information about RUP Plug-ins. The RUP Exchange allows you to find new or updated Plug-ins from Rational, and from Rational partners and customers. You can also find guidelines on building Plug-ins, and you can upload Plug-ins you have built yourself.

Deployment Mechanisms for Process Components: RUP Builder

With RUP Builder, a new tool in the RUP product, you can select which Plug-ins to deploy for your project. First, let's introduce some new concepts:

- A **RUP Configuration** consists of a RUP Base, and a selected set of RUP Plug-ins. RUP Configurations are assembled using RUP Builder.
- A **RUP Web site** is a generated RUP Configuration.

The RUP ships with RUP Builder, the RUP Framework (which contains a RUP Base), and a number of Plug-ins. And as Rational, our partners, and our customers place more Plug-ins on the RUP Exchange, you can download these and use them in your RUP Builder.

RUP Builder organizes your Plug-ins into "configurations." It ships with a number of predefined configurations, and you can create additional configurations as you require. Once you have defined which Plug-ins belong to a configuration, RUP Builder validates that the selected pPlug-ins are compatible. Once you have selected a set of Plug-ins that do not conflict with each other, RUP Builder allows you to generate a RUP Web site from your configuration. This Web site has the look and feel of the RUP as you know it today. It has the tree control, navigation buttons, and search capabilities you

are used to, but the actual content is based on the particular Plug-ins you have selected for your configuration.

A key feature of RUP Builder is that it allows you to right-size your process. This means that you can choose a small, medium size, or large process, by selecting and de-selecting Plug-ins. Rational is committed to allowing your project to define the specific process you need, and we strive for continuous improvements in this area.

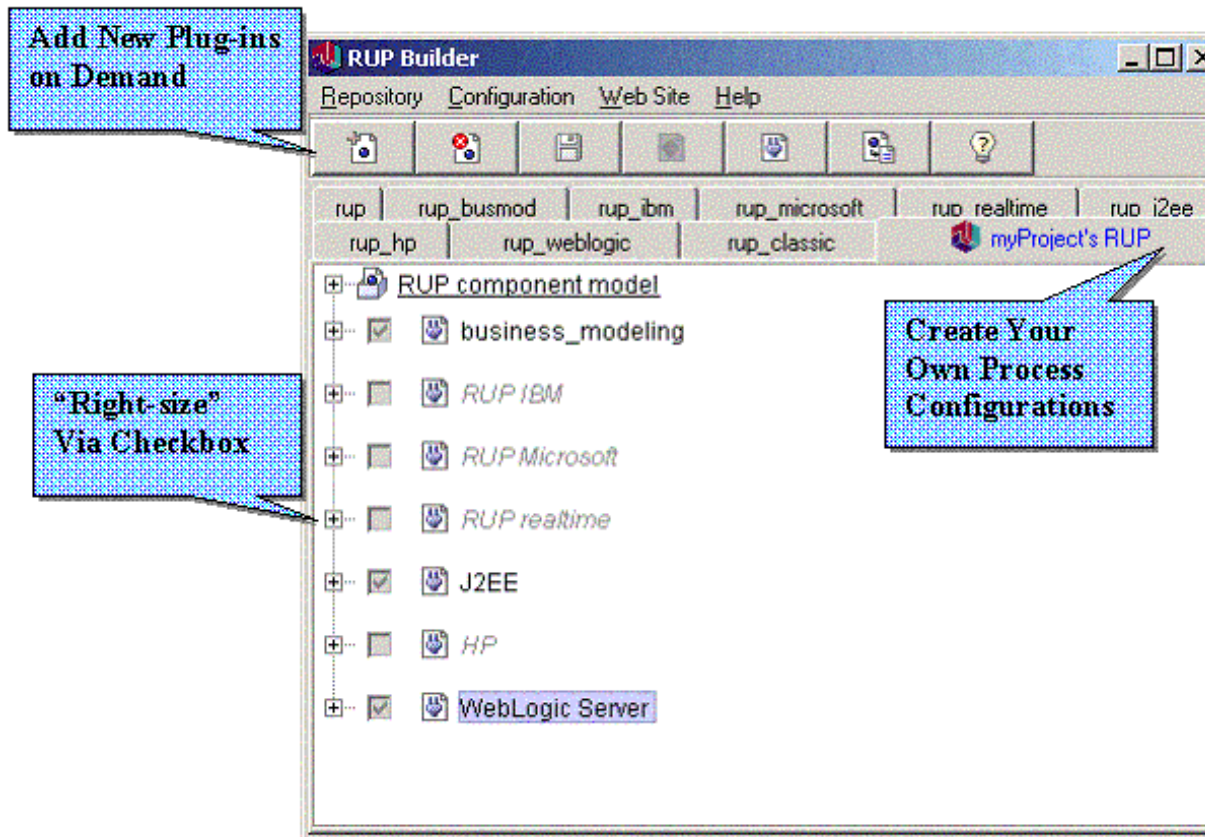


Figure 5. RUP Builder Allows "Right-sizing" to Customize Your Process. RUP Builder allows you to "right-size" the process you use in a project, by making your process smaller or larger. This is done by selecting which Plug-ins should be included in a RUP Configuration, and generating a RUP Web site from your Configuration.

The generated Web site, or instance of RUP, can be further customized to address specific project needs. This is typically done by producing a *Development Case*, which guides team members as to which parts of the RUP to use and how to use them. More guidelines for customizing the RUP are in the Process Engineering Toolkit included in the Environment discipline of the RUP.

Accessing Best Practices

All of the capabilities described above are only valuable if analysts, developers, testers, database administrators, configuration managers, project managers, deployment managers, and other team members can effectively use the know-how captured in the Plug-ins. Therefore, the last piece in our improved process framework is the new capabilities of the generated RUP Web site.

You can view the RUP Web site through your favorite Web browser. You will

find the following features that help you access pertinent information from the RUP knowledge base:

- A *Getting Started* tour to acquaint you with the RUP product.
- A *search engine and index* to make it easy to find information.
- A *role-based presentation of material* to allow you to rapidly access all relevant parts of the process for the responsibilities of that role.
- *Graphical navigation and extensive hyperlinking* that allow you to drill down into details in the areas of most interest to you.

RUP Integration With Tools: Tool Mentors and Extended Help

The bulk of the RUP is tool independent, but at the end of the day, practitioners need to understand how to implement the process with the tools at hand. *Tool mentors* provide step-by-step guidelines for implementing the various RUP activities using the tools on your desktop. Tool mentors describe which menus to select, what to enter in dialog boxes, and how to draw diagrams to accomplish the tasks specified in the RUP.

Tool mentors are available for Rational tools, as well as for partners' tools such as IBM WebSphere Application Server and BEA WebLogic. Customers and partners can write additional tool mentors, and tool mentors can be included in RUP plug-ins, as can any process element in RUP.

Extended Help provides context-sensitive process guidance within the various tools. For example, if you are trying to use the Class Diagram editor in Rational Rose and you do not know what to do next, you can open "Extended Help" from the Rose tools menu to get a list of the most relevant topics within the RUP, depending on the context -- in this case, a Class Diagram in Rose.

Extended Help is available not only from Rational tools, but it can also be integrated with any tools through a simple API, allowing partners to integrate their tools with the Rational Unified Process. This further promotes the notion of the Rational Unified Process platform assisting you in developing software using a variety of tools, on a variety of hardware and software platforms.

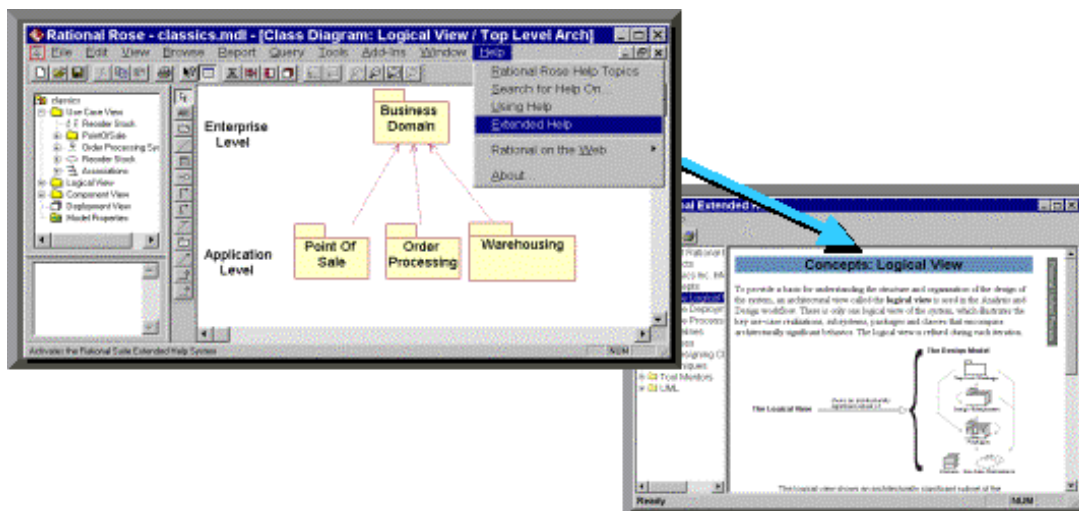


Figure 6. RUP Context-Sensitive Extended Help. *Extended Help provides context-sensitive help from the tools you are using. When launched, it presents a list of the most relevant topics in the RUP.*

The combination of Tool Mentors and Extended Help provides a powerful two-way integration between the RUP and the tools at your desktop. This integration helps practitioners make more effective use of their tools, allowing them to get more value out of their tool investment, and facilitating effective implementation of the process.

Conclusion

The Rational Unified Process is an industry-wide platform for software best practices. It allows Rational, its partners, and its customers a better way to package their know-how into process components, encapsulate them as RUP Plug-ins, and distribute them through the RUP Exchange on the Rational Developer Network. Plug-ins currently provide content from IBM, Microsoft, BEA, Sun, HP, and other companies.

RUP users can find out about, and download, RUP Plug-ins from the RUP Exchange on the Rational Developer Network.⁷ Using RUP Builder, they can "right-size" the process they use for their specific project by selecting the Plug-ins they want to use, and based on that selection they can generate a project-specific RUP Web site. The know-how captured in Plug-ins is now easily accessible for the practitioner through a Web browser. Extended Help and Tool Mentors provide a two-way integration with the tools enabling effective implementation of the process.

Notes

¹ For a review of the waterfall approach, see "Going Over the Waterfall with the RUP" by Philippe Kruchten in the September 2001 issue of *The Rational Edge*:
http://www.therationaledge.com/content/sep_01/t_waterfall_pk.html

² Note that the product was originally called the Rational Objectory Process, and was named the Rational Unified Process in 1998

³ The Objectory Process was developed 1987 by Objective Systems, founded by Ivar Jacobson.

⁴ See OMG Document ad/01-06-05 for more info: <http://cgi.omg.org/cgi-bin/doc?ad/01-06-05>

⁵ More information about this can be found at RUP Exchange within the Rational Developer Network. See below.

⁶ www.rational.net (registration required)

⁷ Ibid.



For more information on the products or services discussed in this

article, please click [here](#) and follow the instructions provided. Thank you!

Copyright [Rational Software 2001](#) | [Privacy/Legal Information](#)