

The Future of Rational Rose

Prior statement on Rose expected lifetime (February 20, 2007) [located here](#).

Executive Summary: End of Marketing of the Rational Rose products, excepting Rational Technical Developer, is on the one-year horizon. Rose will continue to be supported for years to come.

Market conditions and product requirements can always be counted on to change. And so as with all commercial software products, it was inevitable that eventually Rational Rose would either be retired and replaced by something else, or else modified in ways that would make its later versions unrecognizable to someone familiar only with its early versions. Consider how the market and technologies have changed over the course of the current decade...

- 1999 – MDD market begins to express desire to have modeling embedded in an IDE. Rational Rose customers express desires for capabilities such as multiple models that can directly reference each others' elements.
- 2000, 2001 – Microsoft introduces .NET. IBM introduces Eclipse. Visual Studio and Eclipse begin to emerge as likely IDE market leaders. Rational XDE is introduced in response to market drivers and Rose enhancement requests, and supports installation into Visual Studio 2003 and Eclipse-based IDEs
- 2002, 2003 – both IBM (with Eclipse) and Microsoft (with Visual Studio) begin adding substantial new modeling infrastructure to their IDEs. As the UML2 specification begins to gel, the *de facto* reference implementation of the UML metamodel is built on Eclipse EMF. IBM acquires Rational
- 2004 – IBM introduces Rational Software Architect and Rational Software Modeler as the initial offerings in a new product line that is natively based on Eclipse modeling infrastructure, including the UML 2 metamodel. Adoption of Eclipse soars and it begins to emerge as the *de facto* industry standard open source development tools platform. The Eclipse Foundation becomes a fully independent entity and its list of contributing members continues to grow.
- 2005 – a third-party licensing issue forces IBM to begin the end of life process for the XDE product. Microsoft introduces Visual Studio 2005, marking its first delivery of modeling capabilities. Microsoft ends support for Visual Basic (VB6)
- 2006 – IBM introduces Rational Systems Developer and Rational Data Architect as additions to the Eclipse-based product line. IBM also contributes substantial code to the Eclipse GMF project.
- 2007 – IBM introduces Rational Modeling Extension for Microsoft .NET as the latest addition to the Eclipse-based product line
- 2008 – IBM acquires Telelogic and its product portfolio

In addition to the points noted above, it also became clear along the way that traditional UML modeling and "Model Driven Development" were no longer perceived as compelling to the same extent as when they were "fresh and new" in the 90s. The fundamental powers of abstraction, automation, and re-use needed to be enhanced, broadened, and delivered in a more configurable, usable, and extensible form. Customers made it clear that MDD must support *Business Driven* Development. They expressed that they were experiencing pain due to the communication gap between development and IT operations, and the resulting frequent deployment failures. They expressed growing interest in abstraction (modeling) languages other than UML. Consequently, we have recognized that our products must span and integrate semantic domains such as BPMN and BPEL for business process modeling and execution, SQL for data modeling, 3GLs for code modeling, a specialized language for operational and technical architectures, and others in addition to UML.

Accordingly, the architecture of the Eclipse-based product line – the Rational Software Architect family of product -- provides for capabilities such as

- ability to configure the extent of visible UML tooling and semantic tree content, and the ability to quickly and easily build richly tooling custom UML profiles, which combined represent a means to effectively create customized domain specific modeling languages tailored to your particular problem and solution domains
- powerful development automation capabilities – "Patterns Based Engineering" – based on pattern and transformation frameworks and rich transformation authoring tools
- "mixed modeling" with diagrams that depict elements from multiple semantic domains such as UML, SQL, Java, and C#

Had we attempted to evolve Rose in response to all of these forces and requirements, it would look nothing like it does today. It would be Rose in name only. Evolving Rose to ...

- support multiple, cross-referenceable models
- install into an IDE
- support UML 2
- support modeling of other semantic domains

... and so on, would have meant wholesale changes to its architecture, to its user experience, and to its model formats.

Customers would have been required to undergo a number of model migration and retraining efforts along the way. So instead we chose to introduce Rose's eventual replacements under other names while leaving Rose available in the market, and to define paths and processes for migrating from Rose to the replacements. This has enabled our Rose customers to keep using Rose in their production projects without significant disruptions, as they evaluate the new products and plan ahead for their eventual adoption.

Now the time is drawing near when the Rose products (with the exception of Rose Technical Developer) will be retired. The retirement process will involve two significant milestones

1. End of Marketing. A formal End of Marketing Announcement is made, and then actual market withdrawal (when the product can no longer be purchased) follows in 90 days
2. End of Service. A formal End of Service Announcement is made, and then actual end of service (when support under standard maintenance can no longer be obtained) follows in 12 months

For the Rose family (Rose Technical Developer excepted) End of Marketing is likely to occur in the first half of 2009. End of Service is likely to be somewhere in the range from late 2011 to late 2014, with Extended Support Agreements available for an additional two years.

Normally, when an IBM software product is withdrawn from marketing, the part numbers for renewing maintenance on that product are retired. This generally means that maintenance can only remain in force for at most a year after the market withdrawal (exceptions include customers who write multiple years of maintenance into ELAs). But we are sensitive to the fact that many of our customers use Rose in large, long-running projects that cannot absorb adoption of new tools in mid-project. So we will deviate from standard practice in this case by making it possible for customers to continue to renew maintenance on Rose until 12 months prior to the End of Service date. In other words, the maintenance renewal part numbers will be retired at the time of the formal End of Service Announcement.

To summarize:

- you can count on being able to purchase additional Rose for a minimum of seven (7) months beyond the time of this writing
- you can count on being able to use Rose on a supported basis for several years to come
- your ownership of Rose entitles you to the replacement products
- the replacement products represent significant functional capabilities and potential business value over-and-above Rose, and they will continue to be the focus of significant product development efforts
- our investments in Rose which have already been at "close to maintenance levels" for some time, will become increasingly limited. Rose defects and enhancement requests that have meaningful counterparts in the replacement products are being addressed in those replacement products only, not in Rose
- you should begin planning now for eventual adoption of the replacement products