

The UML and the Rational Unified Process

From the Editor

As we receive your reactions, reflections, and, in some cases, corrections to the articles published, we may feature them here. This column is another way we hope to enhance our interactions with the software engineering community and our readers.

THE UML WAS developed to help maintain coherence among the many design and implementation views and artifacts in an iterative development lifecycle, keeping all stakeholders aligned as changes are made from iteration to iteration. Mr. Miller's characterization of the UML as designed to support Waterfall development processes is false.¹ Having risen in the context not of academia but rather of real-world, complex software-intensive systems, the UML was developed as a pragmatic tool that reflected best practices in software engineers, such as the concepts of multiple views (as now codified in ISO/IEC 42010), continuous integration, and continuous deployment (both of which preceded and became a part of modern agile methods).

Beginning in 1982, Rational employed iterative development to create its integrated software engineering tool set. The Rational Environment, first released in 1985, included direct support for iterative development, incorporating knowledge gained during its construction. To successfully deploy this product, Rational's field teams engaged with customers to help them transition from Waterfall to iterative

development. In developing the UML during the mid-1990s, Grady, Jim, and Ivar incorporated the experience gained by these field teams as they worked with customers developing complex, long-lived software using an iterative lifecycle, later formalized in the Rational Unified Process. This evolution took place not only in the context of real, complex systems but also in collaboration of the inventors of Waterfall (Win Royce); spiral methods (Barry Boehm); and the software engineering work of Parnas, Liskov, and others.

The UML came into being during a particularly important sea change in the nature of computing: from algorithmic languages to object-oriented ones; from mostly monolithic, single computer systems to highly distributed ones; and from systems that could be built by small teams to large teams of teams. Indeed, the UML and the Rational Unified Process did not reflect the ossified, high-ceremony practices of the earlier eras; they represented the beginning of our current era of software engineers and, through their concepts, have made possible—and are very much consistent with—modern agile approaches. 🌀

Dave Bernstein

Contact: dave.bernstein@comcast.net

Grady Booch

Contact: egrady@booch.com

Reference

1. J. Doolittle, "Jeremy Miller on Waterfall versus Agile," *IEEE Softw.*, vol. 37, no. 4, p. 107-C3, July/Aug. 2020. doi: 10.1109/MS.2020.2987493.

We welcome
your letters.

Send them to software@computer.org. Include your full name, title, affiliation, and email address.

Letters are edited for clarity and space.