

INDUSTRY SPOTLIGHT: MODELING

How UML makes a DevOps-driven digital transformation possible

Sparx Systems Enterprise Architect modeling platform aims to change development

BY ALEXANDRA WEBER MORALES

What's the best way to build complex, software-intensive systems? A powerful approach, according to the Australian software vendor Sparx Systems, places visual modeling tools at the hub of a DevOps-style operation. The company's flagship modeling platform, Enterprise Architect, was commercially released in 2000 and continues to rise to the challenge of faster software delivery.

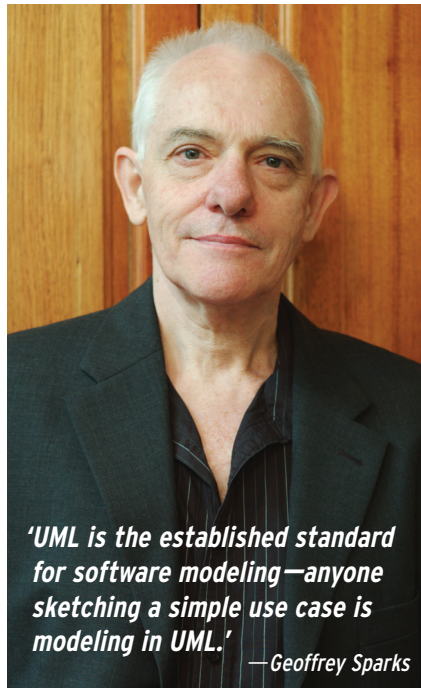
We spoke to Geoffrey Sparks, CEO and founder of Sparx Systems in Creswick, Australia, about how UML-based graphical modeling tools assist in enterprise digital transformation.

SD Times: What is the Sparx approach to DevOps? What problems do you hope to solve?

Sparks: Digital transformation at the enterprise level can only be achieved when it is underpinned by a solid platform to deliver and support new applications, services and technologies. Through transparent collaboration, Enterprise Architect encourages increased productivity between development (Dev) and IT operations (Ops) by eliminating functional silos.

Enterprise Architect has been built as a team-based visual modeling platform, an approach that continues for groups of people working on the same projects, sharing information, ideas and models. We like to say Enterprise Architect was enabling DevOps before the term was coined.

Collaboration lies at the heart of the Sparx Systems support for DevOps, which in turn, improves the software development life cycle through automation and best practices. Prior to



DevOps, project-delivery cycles lacked transparency and the progress visibility the managers had gathered from assumptions rather than from objective data. Within the integrated Enterprise Architect project workspace, artifacts can be viewed and updated with version control, code review, and Continuous Integration tools. This is the level of functionality that defines Enterprise Architect as a leading DevOps solution.

What are the advantages of a single repository?

Using a single platform helps to eliminate problems that arise from using a complicated DevOps tool chain. Everything you need is available at your fingertips, with the added advantage of traceability, integrated communication tools, model-driven design, and powerful visualization tools to improve understanding. The addition of cloud technologies and a

shared repository allows team members to contribute to the DevOps process from anywhere in the world. Enterprise Architect has been designed from the ground up to improve communication, enhance collaboration, and facilitate information exchange between all stakeholders and team members. Enterprise Architect also provides tools for resource allocation, task tracking, project management and team communication, to ensure everyone can contribute to and measure the success of a project.

How prevalent is UML in organizations that have a successful DevOps culture? How does UML assist in automating aspects of the delivery pipeline?

UML is the established standard for software modeling—anyone sketching a simple use case is modeling in UML. A substantial body of online research provides testimony to the value of UML on software quality. Within those organizations that are undergoing digital transformation to improve operational efficiencies (many of whom are Sparx Systems customers), there is a prevalence of UML tools deployed to manage application delivery, a process which relies implicitly on code quality assurance.

Within the digital transformation imperative, there is a natural tension that exists in the flux between legacy systems, which are “keeping the lights on,” and integration of new technologies.

Visualizing the software legacy, taking control of software development for evolving systems structure, navigating through complex systems, and selective exposure of detail can all be achieved using UML. The adoption of standards-based UML tools is a major step toward

continued on page 30 ►

INDUSTRY SPOTLIGHT: MODELING

UML drives DevOps digital transformations

◀ continued from page 29

technical debt reduction and associated quality improvement, as they provide the templates and frameworks to reduce risk and increase efficiency, while supporting collaboration between stakeholders across the enterprise.

How does Enterprise Architect assist with code generation, version control, code review and Continuous Integration (among other activities)?

Enterprise Architect includes a number of design tools to create models of software, automated code generation, reverse-engineering of source code, binaries and schemas, and tools to synchronize source code with the design models representing the software system. The programming code can be viewed and edited directly in the integrated code editors within Enterprise Architect, which provide intelligent code completion and other features to aid in coding.

Another compelling aspect of the environment is the ability to trace the implementation Classes back to design elements and architecture, and then back to the requirements and constraints and other specifications, and ultimately back to stakeholders and their goals and visions.

From an engineering and quality perspective, the most compelling advantage of this approach is that the UML models and therefore the architecture and design are synchronized with the programming code. An unbroken traceable path can be created from the goals, business drivers and the stakeholder's requirements, right through to methods in the programming code.

Enterprise Architect's Visual Execution Analysis tool set vastly improves the accuracy of Continuous Integration by allowing the developer to debug and profile the running code in comparison with the model. Automated reports and model generation, based on code sequences and code timing, afford a greater understanding how the code is operating in real time. Time Aware

modeling allows the Dev team to then plan the next steps for the project iteration without touching the current working model. Different design scenarios can be planned out, signed off by the stakeholders, and then promoted into the current working model.

How is requirements definition and management a key to effective (and speedy) DevOps in software-defined businesses?

Effective gathering of requirements and their accuracy can dictate the future success of any software development project. Enterprise Architect has the capability to source requirements in a variety of forms, native to the methods of each stakeholder's role.

Enterprise Architect hosts complete traceability, from strategic modeling to implementation, architectural definition, deployment and maintenance. Cloud-based repositories, discussions, impact-analysis tools, auditing, reporting and a host of other capabilities make Enterprise Architect the ideal platform for storing and working with requirements.

In addition, the Specification Manager is a document-like interface that provides an easy and familiar environment for creating and editing requirements without needing to use diagramming or visual tools. This is particularly beneficial for the provision of inputs gathered from non-technical yet essential elements of the enterprise.

What sort of software delivery pipeline productivity metrics can be derived from artifacts contained in Enterprise Architect? Can a "last mile" deployment problem be anticipated in UML design upfront?

As a team-based modeling tool, Enterprise Architect has a lot of features built in that facilitate correct software/process at the time of deployment:

Automatic Test Generation from the model provides a set of test scripts, test sets, test cuts and test steps for testers to use to test the system through the conceptual passes before being deployed, effectively allowing a user to

implement Test-Driven Design.

Because the project can be managed within Enterprise Architect's extensive project-management features (like Gantt charts, resource allocations, Kanban, workflow engine, etc), "last mile" problems, like dependent parts of the system not being ready on time, become evident early in development.

For process development, Enterprise Architect offers simulation, so users can explore and identify optimal "what if" scenarios prior to real-world implementation, saving them time and money.

Finally, Enterprise Architect is an end-to-end tool where everything from the initial mind-maps and strategic decisions can be traced to delivery, through requirements, use cases, process diagrams, down to the deliverables, code, business rules, etc. You can use the project itself to make sure you are delivering what has been asked for.

Tools like the Traceability Matrix and Gap Analysis tools make it very easy to spot holes in requirements and the downstream activities that will be deployed to realize these requirements. It also shows if extra functionality has been introduced to the project that was never asked for in the first place, with the goal of reducing scope creep.

Can iterative design and development at the project level end up transforming a company?

Definitely. From its inception, Enterprise Architect has been designed to used as a "master plan" for enterprise continuous improvement. Once the enterprise has been modeled based on the collection of requirements, further refinement may include simulation, testing, scenario analysis and much more. Furthermore, a subset of the enterprise, such as a single business process, can also be captured, modeled, analyzed and improved upon... possibly even automated into software. Once the repository is developed, this can be used to manage the generational evolution of any business unit. ■