Building smarter products
To support your business objectives

Rational. software

Model-driven systems development—delivering innovation with demonstrable business value.

Highlights

- ***Speeds delivery of innovative products with a program-wide development environment that enables clear understanding of requirements across systems and disciplines***
- ***Improves quality and helps lower costs and risk with continuous validation of system requirements throughout the product lifecycle***
- ***Capitalizes on distributed development with automated processes and effective collaboration among diverse teams***

Despite the ongoing focus on product lifecycle management (PLM), software and integrated product failures still plague companies. A US$1 billion prototype rocket self-destructed 40 seconds after takeoff due to a bug in the on-board guidance software. A software glitch forced recall of 75,000 cars that stalled at high speeds. Failure to properly calibrate all components in an automatic windshield wiper system resulted in cars sold with nonfunctioning wiper systems.

Such failures occur, not because of performance issues at the individual component level, but because of problems with interactions at a system level. Many of today's products are, in fact, systems of systems. Features are delivered through integration with other systems and processes. For example, sensors in an automobile's safety system can be programmed to assess the severity of a crash and integrated with a communications system to pass accident details to first responders.

Existing product lifecycle management tools and processes are pretty good at helping product teams verify that features and subsystems are developed correctly. However, they are not as good at helping companies determine that the right system is being developed and that it delivers the expected outcomes.

Shrinking development times only make the problem more challenging. Being first to the marketplace is of no use if products do not work or do not meet the needs of buyers. Hitting a window of opportunity in the current marketplace requires significant changes in the way value is delivered. The challenge today is to get the requirements right for the entire system.

To meet this challenge, the product development process is evolving from a focus on three-dimensional (3-D) computer-aided design (CAD) and mechanical bill-of-material (BOM) management to the design of holistic systems—a process that includes requirements management and traceability across engineering disciplines, including mechanical, electrical and software. When software is effectively fused with microelectronic, actuator, sensor and mechanical technologies, it creates unprecedented leaps in product value. In fact, software is becoming the key ingredient in product innovation and differentiation. It is also introducing a whole new layer of complexity, cost and risk.

The IBM solution for model-driven systems development is designed to help you deliver complex systems more quickly. It provides a structured and visual approach for collaborative systems design and development that can help you to:

- *Refine marketplace demands into robust systems requirements.*
- *Reduce the cost of producing high-quality systems by identifying system errors early, when they are less expensive to correct.*
- *Effectively collaborate across diverse hardware and software teams.*

Accelerate delivery of innovative products with a common view into systems requirements

The first challenge in developing a complex product is to transform amorphous marketplace needs into a set of requirements. The model-driven systems development solution from IBM helps stakeholders view and analyze the product in the context of a system—not just the physical product but also its connection with the broader environment. Understanding the system in context reduces confusion over requirements and clearly establishes what the system must do under what constraints.

By modeling the entire system early in the development process, you can simulate various product and architectural alternatives and make changes before engineers do a lot of detailed design that may have to be reworked later. Consider this example of an image satellite ground support system: system requirements included the ability to plan missions, control the satellites and process the collected data for analysis. Accordingly, the design included construction of three subsystems: mission planning, command and control, and data processing. Each subsystem was given to an independent team for development. During the project, each team independently discovered the need for a database to capture the satellite's orbital history. Each team built its own separate database, using separate formats. However, because the orbital information had to be consistent for the overall system to operate correctly, additional effort was required to ensure these three databases remained synchronized.

The IBM approach leverages models in the design to avoid this kind of costly mistake. Models enable effective stakeholder interaction. They provide a common visual language to promote common understanding and a framework for considering the integration of concerns across all of the engineering disciplines, as well as with customers, suppliers and other involved players such as contract managers. Often it is easier to point to a picture or diagram than it is to describe something in words. The very act of modeling or diagramming can force people to be more concrete and specific.

By providing a visual representation of product requirements to support the system, the IBM solution enables stakeholders to build architecture and design dependencies based on collaboration and shared responsibilities. From this model, they can predict system behaviors and perform trade-off analysis to determine which design choices make the most sense. For

example, when designing a vehicle, if the requirement for fuel economy increases, the top speed of a vehicle will decrease. Or when designing an aircraft, if the range requirement increases, tradeoffs will have to be made in size and payload. Early model execution and simulation helps to quickly identify and resolve issues and reduce costs.

Having determined a set of system behaviors, team members can more quickly and accurately create logical structures meant to support those behaviors and map product capabilities to specific parts of the system. The system model provides a synchronization point, allowing each discipline to create its own set of models without losing sight of the bigger picture.

Find errors early when they are inexpensive to correct

The more complex the product, the greater the risk of failure. At any time in the product development process or product lifecycle, you run the risk of adding or changing something that "breaks" something else. The earlier you find and address issues, the greater the probability of project success. It would be better still if you could anticipate problems and avoid them altogether.

Continual requirements validation

Managing risk aggressively is the essence of the model-driven systems development process framework included in the IBM solution. It provides rigorous process and tooling capabilities to help you decompose the system effectively, derive requirements and maintain traceability. Dependency relationships are created automatically, helping improve and accelerate impact analysis. You can analyze multiple viewpoints simultaneously to help ensure that both functional and nonfunctional requirements are met. With clear visibility into the status of requirements and traceability from each requirement to the system as a whole, you can determine that you are not only building to the right bill of material, but you're also building to the right set of requirements.

In addition, the IBM solution is designed to support iterative development. IBM experience has shown that iterating through the production of a set of reusable artifacts improves both the artifacts themselves and the system that is the end product. In the process of defining context, identifying dependencies and specifying the distribution of responsibilities, each progressive step can highlight ambiguities in previous steps and uncover any problems or issues in design.

Effective change management

As parts of the system are completed—or changes occur—modeling and simulation capabilities can help you understand, prior to testing, how different pieces and parts of the system work together and how changes could affect overall system performance.

This is more than just being able to do a digital mock up of a mechanical model. A change to a mechanical component can affect one or more electrical components, each of which, in turn, may be controlled by software. Just finding which elements are affected is complex enough, but the real key is to understand the significance of a change. For instance, the change may require using a part from a different supplier that operates under a different contracting structure, meaning that the purchasing organization could be involved in the change process.

With the IBM solution, you can assess the impact of a change across the entire system. By simulating system outcomes and conducting trade-off analyses, you can proactively:

- *Identify what modifications need to be made.*
- *Calculate the associated costs.*
- *Balance these costs against the benefits.*
- *Validate that the product still addresses customer, marketplace, performance and quality requirements after the change is made.*

Changes can be made more intelligently with knowledge of the end result available at the time of the change—before the product is in the hands of the customer.

Streamline process and promote collaboration across diverse teams

Although well-versed in electrical and mechanical engineering technologies, many manufacturing companies have not necessarily adjusted to the increase in software content within their products. In particular, they have not adopted effective software development methodologies. Many organizations lack formalized practices. Expertise is often in the heads of long-term employees and difficult to replace. Paper-based and manual processes hinder efficiency. Without tool-based automation, many processes can be time-consuming and expensive.

The IBM solution unifies software with other project disciplines through the use of models. And it automates best practice processes that help reduce software development time and improve impact analysis across the system.

Using models, systems and software engineers can more clearly understand and analyze requirements, define design specifications, test system concepts using simulation, and automatically generate code for direct deployment on the target hardware. Development teams can standardize processes and automate repetitive tasks to boost productivity and enhance regulatory compliance through self-documenting data and workflows. As a result, engineers and developers are able to deliver more complex and intelligent designs in much less time.

For more information

To learn more about how the IBM Rational® solution for model-driven systems development can enhance lifecycle management of complex products, contact your IBM representative or IBM Business Partner, or visit:

ibm.com/software/applications/plm/rational.html

IBM Corporation
Software Group
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
August 2009

RAS14035-USEN-00