



# TECH BRIEFS

ENGINEERING SOLUTIONS FOR DESIGN & MANUFACTURING

## Software is the Driving Innovation for Consumer Electronics



TomTom, a leading provider of in-car location and navigation devices, uses PTC Integrity, an application lifecycle management solution that manages requirements, models, code, and test.

**Y**ou may not realize it, but many of the bells and whistles in the products that you engage with on a daily basis are actually powered by software.

The vast array of features, including the advanced electronics in your vehicle — as well as many of the special capabilities in televisions, toys, and household appliances — are innovations that have been made possible by software. Software in products holds the key to innovations that improve quality, safety, and ease-of-use, as well as add new functions. Software simply makes products smarter. Think about how software has changed our experiences with products. Think about the smart products you really love to use. Can you imagine a day without using them?

In the past decade, a significant shift has occurred for discrete manufacturing organizations where software now delivers the majority of functionality and differentiation in numerous products we use in our everyday lives. In many cases, software is now the product, and hardware has been relegated to the role of delivery platform.

Discrete manufacturing companies that can harness the power of software for product innovation will become leaders in their respective industries.

According to industry research firm Aberdeen Group, the most successful discrete manufacturers are 30% more likely to use electronics and embedded software to bring innovation to products. [Source: Product Analytics to Engineering On Schedule and On Budget, 2010].

There are many reasons why these manufacturers have started to realize the power of software, and why they are making the shift in developing their products in a very different way, including:

- *Innovation and differentiation* – Software allows manufacturers to create a unique user experience with products. A CTO at one automotive company said that over 50% of the buyers of its vehicles do so in part because of the features driven by software.
- *More product variants* – Software allows manufacturers to build more customer-specific product variants in less time, with less cost. Nokia's move to the

Microsoft OS for smartphones means they now can deliver the same hardware to many countries around the world with no need for localized keyboards — the software provides the right keyboard for the right region and language.

- *Reduce product manufacturing costs* – If you manufacture 10 million units of a product and can save \$5 per device by moving functionality from custom hardware components into custom software that runs on a commoditized component, you save yourself \$50 million dollars in manufacturing costs. It's really that simple. There are examples from many industries that are transitioning to this way of developing products, including automotive, high-tech products, industrial, aerospace and defense, and medical device manufacturers.
- *Improve customer satisfaction, after product delivery* – There are many ways to improve customer satisfaction, but a well-designed product can actually improve over time through software updates. These products include the car that can automatically download

# Consumer Electronics

software updates for service recommendations, the smartphone that updates its GPS system, or the refrigerator that tells you when it's time to change the water filter.

While software is providing these organizations with competitive advantages and offers many cost benefits, it also significantly increases the risk and complexity of product development and delivery processes.

## The Challenges of Software Innovation

Software engineering is a new and different way of developing products and bringing them to market. Instead of a well-understood physical process, software development is empirical, requiring a highly iterative and collaborative approach. This makes it difficult to determine the state of release readiness because you can't physically inspect software for completeness or quality. Also, the flexibility of software to be copied and reused has created a proliferation of software-driven product variants that are hard to manage. As a result, software in products increases complexity and, therefore, quality may be compromised. For example, software bugs contained in automobiles and medical devices can pose huge safety risks. Dealing with software complexity can also cause schedule delays, impacting time-to-market and company profits.



PTC's Creo suite enables engineers to design, model, and simulate all aspects of a car, including the electronics.

Complexity is magnified when defects are being tracked. When a defect is found in one product, what about other related products that also include this defect? It's easy to fix by changing a line or two of code, right? But what if that line of code was reused (copied) to 10 or 100 or 1,000 other products? And suppose the development organization is siloed and little collaboration exists. Defects like this cost organizations millions of dollars each year to manually find and fix across product variants.

Several other areas including testing, software change, and compliance all create complexity in the management of software development:

- Fixing, tracking and testing software is typically done manually, resulting in delays and poor quality.
- Software changes at 10 to 100x the rate of hardware changes. Managing change as requirements that are changed late in the process becomes cumbersome, especially when the change affects multiple product variants.
- New compliance regulations (ISO 26262 in automotive, for example) also increase complexity, which require traceability in software development processes so that companies can easily demonstrate compliance.

Without formal integrated processes for requirements, change management, testing, and reporting, the potential for errors and delays can be massive and expensive.

## A New Approach is Needed

How can these challenges be met? And what do designers need to know when designing, simulating, and manufacturing these software-intensive products?

Design engineers need to understand that software development processes are part of the entire product development process, and they must recognize that the dynamics around the development of software are far different than those of electrical and mechanical design and development, and to take these differences into account when implementing the software process. It is also important to determine the interdependencies of the software development process to the overall system-wide process. Finally, the process needs to



Motorola creates designs of its cellphones and other mobile devices in PTC Creo.

extend across the entire design and delivery spectrum. These processes should allow for efficient and innovative design and development, while minimizing the likelihood of introducing defects into the product.

Another way to ensure that a product development solution is comprehensive and integrated is to adopt a single product to manage core engineering artifacts across the product development lifecycle. With an end-to-end automated software engineering solution, software development resources can be focused on core product development and product innovation, leading to more competitive product lines and accelerated time-to-market with new features.

Discrete manufacturing organizations that make these investments to ensure that software development is managed effectively and accelerates innovation in products can become the industry leaders and deliver the types of products their customers are demanding.

*This article was written by Jake Simpson, Division General Manager, Application Lifecycle Management, at PTC in Needham, MA. For more information, visit <http://info.hotims.com/45599-121>.*