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.
Introduction

You may not realize it, but many of the “bells and whistles” in the products that you engage with on a daily basis — from cell phones to automobiles — 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.

The most successful discrete manufacturers are 30% more likely to use electronics and embedded software to bring innovation to products.

Aberdeen Group, 2010

Product Analytics to Engineering On Schedule and On Budget

Companies in a range of industries are increasingly using software to deliver innovative, feature-rich products for some very good reasons:

• Software innovation costs much less when you factor in the near zero incremental manufacturing cost of software.
• Software, as the name indicates, is more flexible and changes can be made at a fraction of the time and cost.
• Software allows organizations to create more product variants at less cost to pursue new markets and new customers.

The trend toward software innovation is prevalent in many industries, particularly in the automotive, aerospace and defense, industrial, medical device, and electronics and high tech sectors.

Software Makes the Difference

In Automotive: Automatic, intelligent braking systems on many high-end vehicles react in a potential crash situation even if the driver doesn’t.

• Did you know? Cars will require 200-300 million lines of software code in the near future. (Source: Frost & Sullivan)

In Aerospace and Defense: Sensors in unmanned aircraft gather surveillance data for the US military.

• Did you know? The Airbus A380 uses almost 1000 software components supplied by more than 40 system and software companies located on 3 different continents (Source: SITA)

In Medical Devices: Software in implantable pacemakers monitors cardiac rhythms in heart patients.

• Did you know? In the last 7 years, 500 medical device recalls were related to software defects and malfunctions (Source: FDA Survey)

In Electronics and High Tech: From voice-activated texting to wireless internet access to privacy and security features — cell phones (and many other electronic devices) get their intelligence from software.

• Did you know? Embedded Software is responsible for Time-to-Market. Software development now determines the delivery date (Source: Panasonic Corporation)

Use of software to deliver customer-perceived product value continues to accelerate.

Gartner Group, November 2011

Chapter 1: The Challenges of Software Innovation

Software represents a huge opportunity for innovation and competitive differentiation. But for discrete manufacturers, a shift in focus from hardware to software also presents many challenges.

First of all, it is a completely 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.

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 1000 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.

Over 50% of new product launches fail to meet original business goals. Top factors include excess cost and inability to respond to new demand.

Several other areas including testing, software change, and compliance all create complexity in the management of software development. For instance, 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. Managing change as requirements 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, requiring 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!

How Embedded Software Increases Complexity

Example: A single change request could affect 20 products (or more). The change request then proliferates to the requirements documents which must be updated and authorized, and to the individual test plans which must be modified to test the new requirement. If there are 20 products that are affected throughout the organization, this process can quickly become extremely complex.

In Chapters 3 and 4, we’ll discuss how best to address these challenges and look at several examples of companies that have adopted best practices and engineering solutions that have delivered significant value to the business. But first, let’s look briefly at where we are now.
Chapter 2: Current Software Engineering Solutions

Many of the solutions available on the market today are not designed to address the complexity of embedded software development. In an effort to save money, some engineering organizations piece together solutions using Microsoft Office (e.g., spreadsheet, word documents, etc.) and Open Source solutions. These are grown organically as needs arise and result in a set of disconnected processes and products.

Engineering resources are assigned to build, maintain and integrate these home-grown solutions, costing the company money that could be spent on engineers devoted to enhancing and innovating the product line — the business’s core competency. So the “free” solution comes at quite a price!

The other alternative is an enterprise “solution” made up of software “point” products that each address only one aspect of the development cycle, such as requirements management, test management, or change management.

Siloes can lead to problems with traceability and change management.

Product Requirements → Product Design → Change Management → Testing

Requirements must be traceable bi-directionally – what is the requirement, what software addresses the requirement and when tested, does it meet the requirement?

Gartner Group, November 2011

Without integration between various elements, it’s difficult to assess the requirement change, and how that new requirement should be tested once the software change is made. These operational siloes hinder collaboration and traceability, leading to additional costs in rework, change management and compliance reporting. Many organizations assign development resources to produce software that will integrate various tools. What would seem to be a one-time effort becomes an on-going problem as different development tools are updated at different times, and code must be constantly updated to maintain integration. The problem is compounded if your integration expert decides to leave the company, taking along the knowledge and history of the custom code.
Chapter 3: A New Approach is Needed

How can companies overcome these challenges to create a development environment that supports innovation and collaboration and ensures quality and efficiency? One way to address this problem is by improving the software development process.

Organizations must integrate software development with hardware development, adopting industry specific best practices across the product development lifecycle — and extend those best practices to partners and external suppliers of product components. Once the process is understood, a solution should be implemented which allows for the automation and management of the full end-to-end process.

Aberdeen’s Report “System Design: Get it right the First Time” published in August 2011 compared “Best-in-Class” (high-performing in terms of meeting revenue projections and product release schedules) companies to all other companies. Figure 1 illustrates how system knowledge is managed.

Figure 1: How Best-in-Class Companies Manage System Knowledge

By employing integrated efficient processes throughout the engineering lifecycle to design, develop, and test software — costs are driven down, reliability is improved, defects are minimized and innovative features can be easily added and changed.

Best-in class companies are 36% more likely to integrate knowledge and information across disciplines. By linking requirements to specific test cases, these successful companies ensure that software not only “works”, it also meets the customer’s requirements. By tracing requirements throughout development stages, change and test management is facilitated and information is readily available for compliance reports.

In order to address the challenges of embedded software development, organizations must adopt an integral solution that will unify processes throughout the product lifecycle and across product lines, improving end-to-end integration and traceability. By doing this, the enterprise can:

- Relate together all engineering components
- Reuse requirements, test plans and software code
- Orchestrate the software development process within the core engineering and business processes
- Provide real-time visibility into release readiness
- Facilitate collaboration across engineering disciplines and multiple organizations
- Integrate with existing engineering tools and enterprise applications
- Scale ahead of enterprise needs
Chapter 4: Accelerating Innovation in Action

The best 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.

Rather than cobbling together several different technologies, the solution should manage all aspects of product engineering including requirements management, configuration management, test management, defect tracking, change management and reporting. This built-in integration between disciplines:

- Provides a “single source of truth” for software driven product engineering, allowing instant visibility into product readiness across the entire product portfolio
- Enables collaboration across globally distributed development organizations and sharing of requirements documents, test plans and change requests—facilitating workflows, approvals and impact analysis

Pre-defined views can filter data based on organizational roles and permissions, ensuring that users only see information relevant to job position (improving both security and ease-of-use). The solution should be flexible and customizable, allowing your business to tailor the solution to reflect your engineering processes. By mirroring your company’s own workflows and business practices, the tool will be more readily accepted and deployed throughout your engineering organization ensuring a rapid “time to value”. Packaged templates provide a starting point for companies still defining engineering processes and workflows.
With a single solution that is automated from end-to-end, software development challenges become opportunities. For example:

- Requirements, test plans, and software code shared among multiple products are easily reused, and efficiencies can be realized — accelerating development schedules
- Changes and bug fixes can be easily propagated to all models affected by a change, improving software quality
- Traceability of requirements through the development cycle eases compliance management and reporting
- A flexible, open architecture links engineering development to corporate business processes as well as customers, partners, and suppliers, thereby facilitating collaboration
- Errors that are difficult to trace in manual or point solutions are easily traced with integrated processes and an automated solution, thereby improving safety and quality

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. For companies, a range of industries, adopting this type of solution has made a huge impact. Let’s look at a few examples.

One of the world’s leading automotive companies adopted an automated end-to-end engineering solution to help manage the volume and velocity of engineering change driven from software. In this company, 90% of product changes are software-based. By deploying a single solution for the entire product development lifecycle, quality has improved, costly rework had been eliminated, regulatory reporting has been simplified, and requirements and change information is easily shared throughout the organization and with OEM partners.

A leading supplier of in-car location and navigation services used a single product engineering solution to create an early warning system that finds and corrects issues before schedules, quality, or costs are impacted. As a result, the first time right statistic improved from 80% to 97%. Release predictability was also improved, providing the ability to deliver as promised on-time products to customers.
A leading medical device company has reduced compliance reporting efforts by 99% — from up to 36 man — weeks per FDA regulatory submission to mere minutes, speeding time to market and improving safety in product engineering.

A pioneer in unmanned aircraft systems (UAS) for intelligence, surveillance, and reconnaissance was able to move from Capability Maturity Model Integration (CMMI, a software engineering process improvement approach designed to help organizations improve performance) Level 1 (i.e., unpredictable, reactive processes) to Level 3 (i.e., proactive processes customized for the organization) in 11 months by using a single automated solution to manage all core CMMI processes and artifacts. This company reached the 500,000 combat flight-hour milestone faster than any other UAS company.

Although these companies represent a range of industries, sizes, and growth stages, they all share common elements: they innovate through software and they all use a single solution to automate product engineering throughout the product development lifecycle. They are leaders in their market niche and they sustain this edge by getting products to market quickly. These industry leaders have streamlined the embedded software development process and automated best practices across the product development lifecycle. By improving process efficiencies, these companies are able to focus on their core business — accelerating innovation and speeding new products to market — which translates into larger market share, increased revenues, and higher profits.
PTC Integrity accelerates innovation of software-intensive products by connecting people, processes, and artifacts throughout the software and systems lifecycle.

PTC Integrity has an open architecture that integrates disparate tools into a streamlined engineering process, allowing orchestration of engineering change and collaboration across the technology supply chain. With PTC Integrity, engineering teams improve productivity and quality, streamline compliance, and gain complete product visibility, which ultimately drive more innovative products into the market.

To learn more about PTC Integrity, go to: PTC.com/products/integrity