Dealing with complex engineering projects requires a good plan from the start. Systems engineering and product lifecycle management (PLM) are the backbone for any design to be carried out successfully and as planned. IBM believes that systems engineering is crucial in meeting the demands of today’s complexity of products.

Others will agree; therefore, we have chosen a collection of articles focusing on the importance of systems engineering. The coverage includes a rundown of the major components of systems engineering: requirements engineering, systems modeling, integrated product change management, and quality management. Plus, the Aberdeen Group shares a report on how to implement successful system engineering practices that will lead to greater profitability, CPDA discusses requirements engineering tools and much more.

*This eBook is sponsored by IBM.*
**Featured Articles**

**Page 3**
*Confronting Product Complexity with Systems Engineering by John Myers, ConnectPress Editor*
An interview with Steve Shoaf, marketing manager for IBM’s system engineering division, on the benefits of implementing systems engineering.

**Page 6**
*Turning Product Development into Competitive Advantage by IBM*
The purpose of this paper is to explore smarter products and discover some best practices that businesses can employ to build smarter products and drive innovative technologies.

**Page 15**
*Systems Engineering: Top Four Design Tips to Increase Profit Margins for Mechatronics and Smart Products by Michelle Bouchon, Aberdeen Group*
This report offers guidance to implement successful system engineering practices that will lead to greater profitability and avoid the risks of excessive costs.

**Page 43**
*The Requirements Engineering Process: Panel Discussion on Requirements Management Tools by CPDA*
Hear from Cognition, Chrysler, IBM, PTC and Siemens PLM in this panel discussion from CPDA's PLM Road Map conference.

**Page 48**
*Innovation for smarter products: The IBM Approach to PLM*
Learn how the IBM PLM Approach can strategically transform business processes for saving costs and accelerating product innovation.

**Page 52**
*General Electric Conquers Collaboration Challenges with ENOVIA Software by John Myers*
Find out how General Electric’s Wind Energy division met the challenges of a global supply-chain with Dassault Systèmes ENOVIA software.
21st century engineering organizations face a new challenge as they strive to create the best products in the shortest amount of time possible. For many industries each new generation’s products is inevitably more complex than the previous one. Each new version integrates more systems from different engineering disciplines than the last. The automobile for example has spent the last hundred years evolving from a purely mechanical device into a combination of mechanical parts and electronics under the control of onboard computer systems.

This creates an increasingly difficult layer of complexity as more features are added and become tightly integrated. The result is a complex web of interrelated systems where a problem in one strand can devastate the functionality of the others.

To put this growing trend of product complexity into perspective Steve Shoaf, Go To Market Manager, Systems Engineering IBM Software, Rational, explained that as of 2008 over 162 million phones were sold that combine cell phone technology with computing functionality, such as Apple’s iPhone. Also, over 90 percent of innovation and research in the automotive arena is devoted to software and systems electronics.

These innovations take the form of numerous integrated features such as built-in global positioning systems and hybrid vehicle electric charging systems.

These products are by definition harder to work with because of their enhanced complexity which forces companies to integrate different teams with different skill sets to ensure the various engineering disciplines all work together. In addition this means that companies that previously worked exclusively with mechanical products are being forced to become software developers as well.

To help businesses cope with this challenge IBM is promoting their systems engineering approach to meeting product complexity. In fact IBM has over 50 years experience in systems engineering.

Shoaf explained that systems engineering allows businesses to plan their entire integrated product creation process and then simulate each step
of it as they proceed. “Systems engineering allows for a virtualization of entire systems, and significant benefits can accrue when it is integrated early enough that it can be used to flush out problems before they become too costly,” he added.

Systems engineering combines four concepts: requirements engineering, system modeling, integrated product change management, and quality management to create processes for planning and developing complex products thereby streamlining product development. It uses the concept of virtual product design in the product’s lifecycle to help organizations make better planning and feasibility decisions during the product’s creation. In addition by allowing everyone to work from the same set of requirements and collaboratively sharing system models users from multiple engineering disciplines—mechanical, electrical, software—can work as a single globally distributed team.

Requirements engineering focuses on the management of product requirements from the product’s inception to its final retirement. The goal is to capture customer and market needs and required features and ensure that the final product meets these needs. Requirements engineering helps development teams to focus effort, reducing development time while ensuring quality goals are met.

Systems modeling is a solution to help companies create a visual plan for how complex systems will work together, streamlining their development and integration throughout the design process. Modeling allows engineers to understand the system’s complex architecture by uniting mechanical, electrical and software engineering to create a system of traceability across teams and disciplines. This provides the ability to trace requirements from initial capture all the way to the feature implementation and delivery.

In addition, systems modeling also allows users to simulate the behavior of the entire system as it is developed to help businesses promote innovation while reducing cost and risk.

Integrated process change management synchronizes changes throughout the entire engineering process so that users can better understand the way changes in one discipline affect the entire process and view the impacts of changes across the entire product design. It also
helps to reduce the time it takes to propagate changes throughout the entire design team and to increase the visibility of schedules, including the impact of requirement and product changes.

Lastly, quality management is a system for tracking defects and ensuring that initial product requirements are met. In addition it can help promote the smooth flow of information between testers of products and the product’s developers, while also managing that complexity across multiple product configurations.

These core processes are then joined with the IBM Rational software development platform to ensure that the software providing the intelligence to the product is developed efficiently and smoothly integrated into the product design.

The systems engineering solution offered by IBM is made up of four core products, Rational DOORS for managing system requirements, Rational Rhapsody for systems modeling and architecture design and Rational Team Concert for collaboration across engineering discipline and teams managing change, and Rational quality management solutions for ensuring quality. In addition IBM’s system is platform agnostic and can integrate with most existing product lifecycle management (PLM) systems.

Effective use of systems engineering can lead to a number of significant business benefits. These include the ability to ensure the end product meets all of the necessary customer and market requirements. In addition users can also manage complexity while reducing times to market. These factors can then contribute to an overall higher level of product quality and higher degrees of product differentiation.

As engineering organizations are forced to confront the growing reality of product complexity they need to find a solution to that complexity and more importantly a way to manage it. For these reasons IBM is providing systems engineering as a way for businesses to organize their necessary information from a bird’s-eye perspective and then use that data to ensure processes are carried out in such way that each step of a complex product design is handled as efficiently as possible.
Turning Product Development into Competitive Advantage

A White paper by IBM

Best practices for developing smarter products

Executive summary
Smarter products. You hear them mentioned in trade journals, in boardrooms and in the press releases of your competitors. What are smarter products? How do you make them? And how do you make them better and faster than your competition?

The purpose of this paper is to explore smarter products and discover some best practices that businesses can employ to build smarter products and drive innovative technologies, such as:

- Employ a holistic design process that includes requirements management and traceability through all engineering disciplines
- Become value driven and continually assess how to optimize your products, services and projects
- Understand your product in the context of a system
- Provide all business disciplines across your enterprise with a single, shared view of all product requirements
- Perform extensive modeling early in the development process and use it to evaluate the impact of changes across the entire system

Whether you operate in the automotive, aerospace, telecommunications or electronics marketplace, business trends dictate that you need to deliver innovative products that incorporate advances in software, electronics and hardware to deliver a customizable experience to users. The key to creating this new class of offerings is to elevate software development to a strategic business process while leveraging advanced modeling capabilities to better integrate your different design disciplines.

What are smarter products?
As you exit your driveway, your car sends a signal to your home to arm the alarm system while your cell phone automatically synchronizes with your car’s voice command system. Having analyzed your recent driving patterns, the vehicle’s global positioning system (GPS) recommends a new route to cut fuel costs and avoid traffic delays. Finally, the car informs you that your antilock brakes need servicing and, after checking the schedule on your personal digital assistant (PDA), presents you with available appointment times.

This example is just the tip of the iceberg. Businesses around the world are
looking to revolutionize mundane tasks and daily activities with innovative technologies. They are beginning to see products not as solitary solutions, but rather as components of a greater system—one that can be adapted to fit the personality of each individual user or business. How? Manufacturers are tapping into innovations in software, microelectronics, actuators, sensors and mechanical technology and fusing these advances into a portfolio of “smarter products” supported by open standards in ways not previously possible. These “smarter products” are increasingly intelligent; realtime instrumented; and interconnected to other products, the Internet and relevant back-end IT systems.

Figure 1: Smarter products are extremely pervasive and demand a level of intelligence and integration previously unheard of.

**Why smarter products? Differentiation**
The six billion people on the planet all have unique needs, desires, hopes and approaches to getting things done and enjoying life. However, manufacturers are quickly realizing that individuality—the desired experience—can differentiate their offerings. Businesses and consumers are now craving personalization and integration of the products they rely on every day—products that understand the context of what they are doing and can adapt to help deliver results. This demand for smarter products is driving manufacturers and service providers to look for new, innovative ways to differentiate their products.

**The nature of smarter products**
Marketplace factors such as customer satisfaction are driving the trend toward smarter products, and to deliver, businesses are being forced to rethink the very nature of their products. Simply put, smarter products demand more from manufacturers that want to deliver a new experience to consumers. And these demands present themselves in a number of ways.
Systems of systems
As if designing and building complicated systems weren’t hard enough, many of today’s products, such as cars and planes are, in fact, systems of systems. Features are no longer isolated within individual products and are instead delivered through integration with back-office business processes. For example, in-vehicle security system vendors can now provide emergency services and can alert first responders with accident details gathered using vehicle sensors and passed through the vehicle’s security system to assess the severity of automobile crashes.

 Shrinking product lifecycles
Consumers seem more demanding today than ever. Not only do they want the smartest product—they also want it now. And before you know it, they’re looking for an upgrade. As a result, the service life for many products, especially electronics, is shrinking, forcing manufacturers to prioritize features over quality.

These same consumer demands are leading to shrinking development times, but fast to market may not necessarily mean right to market. Moreover, marketplace windows for specific capabilities are shrinking, making hitting a window of opportunity a significant challenge.

 Mechanical aspects are becoming commoditized as product value shifts to software
Today, the hardware that previously differentiated products has largely become a commodity, and it is becoming increasingly difficult to differentiate products based on electronics alone. Only a few years ago, an MP3 player was just that—a device that played MP3s. Now MP3 players must not only play music but also host music libraries, stream video, run applications, support messaging and offer games. And devices that cannot be easily updated with new functionality quickly become obsolete, destined for a local recycling center.

 Manufacturer or software company? The blurring line
The reality is that product manufacturers are now also becoming software companies, infusing the technological capabilities of electrical, mechanical and software components into a new generation of innovation. Unfortunately, many companies simply do not have the skill sets, resources or development platforms necessary to build and integrate the intelligent software that is needed.

Traditionally, software and hardware development teams have worked independently, but as new software-driven product functionalities take shape, the two disciplines need to evolve together to provide a high-quality working component. This is easier said than done because changes may affect components across multiple teams, leading to complex project management, testing and change control.
To develop smarter products, companies must also ensure that their products interconnect with the Internet and back-end IT systems. They must learn how to develop products that can talk together within their intended ecosystems using standardized communication protocols to break down the traditional silos between IT and systems, between competing companies, between vendors and between governments.

**Implications of an inefficient approach**

Failure to adapt to the new challenges for building and delivering smarter products can significantly affect a company’s bottom line and brand image. The speed of innovation in today’s world means that organizations must evolve now to remain competitive tomorrow. The risks of inaction are significant:

- Loss of marketshare leadership to more nimble organizations that can effectively innovate to meet marketplace demands and respond to competitive threats
- Loss of revenue and profit from the inability to develop complex products in a shorter cycle without compromising quality
- Increased development and customer service costs resulting from identifying problems toward the end of the design cycle

**Best practices for developing smarter products**

As we’ve seen, remaining competitive in the current marketplace requires significant changes in the way value is delivered in today’s products. And this value, in turn, is driving changes in the way these products are built—from a focus purely on cost to a focus on innovation with software as a major differentiator. Business drivers are shifting from departmental productivity improvement to globalized development. And the product development process itself is evolving from a focus on three-dimensional (3D) Computer-Aided Design (CAD) and mechanical bill-of-material (BOM) management to the

---

**Figure 2: The gaps that exist between vertical development silos can undermine project visibility, reaction to change and requirements mapping.**
design of a holistic system—one that includes requirements management and traceability through all engineering disciplines, including mechanical, electrical and software.

In fact, an Aberdeen Group study shows that those companies whose products include more software in the product mix do better than those whose products are more mechanical or electronic.¹ But to deliver smarter products, companies must focus on software development within an enterprise-level systems engineering approach. Moreover, because virtually no product these days exists in isolation, companies need to manage the systems of systems that compose the overall ecosystem within which the product will exist.

**Decide what, when and how**
The first questions to ask when building smarter products are what to build, when to release it to the marketplace and how best to deliver required functionality.

**Portfolio management**
The product portfolio decisions you make today will determine whether your company is still relevant tomorrow. With such a fine line between success and failure, companies must make product portfolio decisions based on fact—not guesswork, political agendas, intuition or the opinions of the loudest voice in the room.

Considering its importance, portfolio management is ironically one of the weakest areas of new-product development. Only 25 percent of businesses effectively rank and prioritize their projects, with only 21 percent of portfolios containing projects that are of high value to the organization. And fewer than 20 percent of companies have the right balance of projects in their development portfolios.²

Added to these facts, 76 percent of businesses have too many projects for the resources available, and only 21 percent have a portfolio management system in place to help them select their best projects.³

To increase the chances of marketing a successful product, organizations must become value driven. Benchmark analysis indicates that companies that are best in class at portfolio management are four times more likely to achieve margin advantages of 75 percent or higher for products on the market for less than two years.⁴ Becoming value driven requires a continual assessment of how value can be increased through the optimization of the products, services and projects that your organization delivers.

**Product management**
Proper management of the portfolio can have a dramatic effect on the value a company delivers to its customers. But management of the product itself is just as important. In fact, product managers are more frequently acting as the CEO of a product, bearing the responsibility for the strategic and tactical activities that help position a product in the marketplace and decide when to retire it.

Few things are more critical to the success of a company than its new-product launches, since many businesses notice higher margins on products that have been on the market for less than two years. However, product success is difficult to achieve as a large percentage if new-product launches fail.

A successful product delivers value to its customers and revenue to the business that creates it. But to ensure value and revenue, product managers must define
their target customer segment and understand its needs. To develop a product that delivers value in relation to the cost of development, product managers must also understand the value of different product features to different customers. By providing the right features to the right group of customers at the right time, a product can create value and revenue.

**Ensure that customer requirements are met**

To maximize the chances of success, companies need to ensure that smarter products deliver the specific capabilities defined not only by customers and the overall marketplace but also by how the product will operate as a system. Understanding the product in the context of a system will produce a set of requirements that will determine what the system does and how it does it.

Before the online collaboration capabilities that exist in today's product development environments, business processes operated in isolation, with the only communication being formal handoffs from one process to the next. Based on its research, the marketing department would outline product features and functions and give those requirements to the design department. Designers would consider those requirements when creating the master design, which was routinely documented in two-dimensional (2D) drawings or 3D computer models.

Problems with this approach were twofold. First, the design engineers had to interpret the marketplace requirements because their functions—or the “why” behind the requirement—were rarely specified. For example, a marketplace requirement called for a cup holder in a car but without any specifications. The result: cars on the road today with cup holders that can’t be easily reached, block access to the radio when in use, and sit above the dashboard computer, which is subjected to spilled beverages.

The second problem was, with the increase in software enabling smarter products, the means for satisfying requirements has dramatically changed—yet the means for communicating requirements has not. In the past, requirements for an MP3 player may have included volume control, implying the need for a volume button (hardware) and electronic volume controller (electronics). This requirement could be articulated in the product design by adding a document with the models for a volume button and underlying electronic components. However, today’s requirements may call for user-defined volume curves, with both preset and user-defined equalization capabilities, all of which are supported by software. Thus, the requirements must be understood not at a document level, but at a statement level. Moreover, these requirements must be associated with specific system capabilities, rather than simply with software modules.

To design against requirements correctly, marketing must be able to specify the intent behind product functions and capabilities. By doing so, marketers can guide development decisions in advance, rather than as checkpoints after the design is complete. To avoid confusion and miscommunication, all business disciplines across the enterprise—engineering, quality assurance and internal legal teams, as well as other peripheral stakeholders such as suppliers and customers—should have a single, shared view of all product requirements. And finally, the requirements specification should provide a means of quality testing, with product testing being conducted against initial marketing needs.

**Navigate complexity with modeling**

The infrastructures required to support smarter products are typically complex,
forcing businesses to pursue a modeling strategy that supports the entire system—not only the physical product but its connection with its environment: a systems of systems. By modeling the entire system early in the development process, businesses can simulate various product and architectural alternatives early in the development process when changes are much less expensive.

From these system models, businesses can perform trade studies to determine which design choices make the most sense and predict behaviors of the system and its structures. Having determined a set of behaviors, companies can create logical structures meant to support those behaviors and map product capabilities to specific parts of the system. This may require a variety of models to help with understanding of the system before the actual physical product is designed.

With an overall design of the system in place, a business can begin designing the component software, mechanical and electronic parts. The system model provides a synchronization point across these disciplines, allowing each to create its own set of models but then coordinating these through the system model.

The ability to model relationships between requirements statements and system-level capabilities (including, but not limited to, software capabilities) helps businesses:

- Ensure that requirements (along with their intent) are met.
- Devise and automate tests for software development.
- Generate software code automatically to implement specific functionality.

System modeling provides an abstraction mechanism that allows an understanding of the system as a whole, preventing situations in which the overwhelming system complexity can obscure the big picture. This allows architectural decisions to be made intentionally, not as an accident or afterthought, while providing a mechanism to ensure that the product addresses customer and marketplace requirements at all stages of development.

**Manage change**

Despite the best design efforts, changes to a product design inevitably occur—marketplace dynamics shift and customers may develop different needs. Understanding the effect of a change within the design of today’s smarter products can be challenging. In the past, finding which mechanical components were affected by a change to another mechanical component involved nothing more than simply traversing the product structure.

However, smarter products are much more complex, given that 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 understanding the significance of a change (i.e., identifying what modifications need to be made, determining the associated costs, balancing these costs against the benefit and ensuring that the product still addresses customer, marketplace, performance and quality requirements after the change is made).

Modeling provides a means to conduct trade-off analyses through simulation, assessing the effect of a change across the entire system within a huge network of complex interdependencies. System modeling also enables exploration of the effect a change could have throughout the distributed development team and the supply chain. For instance, a change may require using a part from a different supplier, which operates under a different contracting structure, meaning that the
purchasing organization could be involved in the change process.

The key to successful change management is integration across engineering disciplines, tools and applications as well as back-office processes that touch on the product development process. With the ability to collaborate and share information, 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.

Elevate software development to a strategic business initiative
Companies well versed in mechanical-engineering technologies have not necessarily adjusted to the increase in software content within their products. In particular, they have not adopted good software development methodologies.

Model-driven software development
Just like mechanical development, software development should be model driven. Using models, software engineers can more clearly analyze requirements, define design specifications, test systems concepts using simulation, and automatically generate code for direct deployment on the target hardware. Teams can improve productivity and reduce headcount, standardize processes and automate repetitive tasks to improve team efficiency while reducing time to value and enhancing regulatory compliance through self-documenting data and workflows.

Using model-driven development, telecommunications companies have seen productivity improvements, including 46 percent fewer designs behind schedule and 49 percent fewer designs cancelled.5 Similar to mechanical development, the results from model-based development are seen downstream from the development process in the form of fewer prototypes and physical mockups.

Collaboration enabled between development groups: increase visibility, remove silos
As software development has evolved from centrally located development facilities to globally distributed development, visibility of the activities of various teams has decreased. Walking down the hall to collaborate with a team member is no longer possible, given that a team member could be located across the country or across the world. Today’s development teams must leverage Internet connectivity, as well as a unified software development platform, to establish virtual colocation.

A unified platform helps remove development silos by enabling the integration of applications and the sharing of knowledge as well as the sharing of project status through management dashboards. Software quality, in turn, is improved through less rework, better project tracking and higher team satisfaction. Based on IBM experience, developers can reduce wasted rework by 25 to 50 percent by collaborating on work items, defects and build errors.

Governance and measurement
An often-overlooked aspect of software development is the ability to measure and govern how well the process is performing. In fact, governance, not compute power, is routinely the key to higher productivity, helping to improve overall productivity. And when trying to improve software development, it’s often helpful to use a four-step framework to measure capability improvement:

- Phase 1: Elicit and set business value objectives.
- Phase 2: Determine the solution components.
- Phase 3: Accelerate and monitor solution adoption.
• Phase 4: Review and communicate business results.

Using a measured capability improvement framework enables incremental, measured transformation of software delivery, accelerates adoption through out-of-the-box assets, provides flexible feedback on the business process and captures industry experiences in incremental adoption.

Toward a competitive advantage
In recent history, software development has become increasingly important in creating successful products and driving business results. And a holistic view of product development is critical to navigate the rapid changes in how value is delivered through smarter products. Businesses can no longer let products evolve or add capabilities as an afterthought. Instead they must be as rigorous in their software development efforts as they have been with any previous discipline, determining the effect of changes across departments, across companies and across IT ecosystems before making them. And in those efforts, IBM is leading the way in refining software development efforts for products to help you design and deliver the smarter products that your customers demand—before everyone else does.

For more information
To learn more about IBM Rational software and how it can help your company more efficiently deliver smarter products to the marketplace, contact your IBM sales representative or visit: ibm.com/software/applications/plm/rational.html

Endnotes


System Engineering: Top Four Design Tips to Increase Profit Margins for Mechatronics and Smart Products

Michelle Boucher, Aberdeen Group

Executive Summary

The current economy has driven many companies to look for ways to take cost out of their products and make their products even more appealing with more feature rich, intelligent products. This is driving manufacturers to improve their system engineering processes to enable them to develop products consisting of mechanical components, electronics, and embedded software. These products offer tremendous opportunities in terms of greater profitability when system engineering is done well, but when executed poorly, excessive cost is a significant risk. This report offers guidance to implement successful system engineering practices that will lead to greater profitability and avoid the risks of excessive costs.

Aberdeen used four key performance criteria to distinguish Best-in-Class companies. When compared to the Industry Average, the Best-in-Class are:

- Earning profit margins that are 2.3-times higher
- Taking three-times more cost out of products
- 20% more likely to meet product launch dates
- Experiencing development cycles that are 6.2-times shorter

Competitive Maturity Assessment

When compared to competitors, firms enjoying Best-in-Class performance share several common characteristics that support system engineering including:

- 40% more likely to evaluate design alternatives on multiple criteria
- 39% more likely to use system modeling to verify that design requirements have been met
- 36% more likely to link customer needs to requirements
Required Actions

In addition to the specific recommendations in Chapter Three of this report, to achieve Best-in-Class performance, companies must:

- Use multiple design criteria to define system architecture and add that criteria to the system requirements
- Requirements should be linked to higher level system functions as well as to the overall customer need it meets
- Leverage a model driven design approach to overcome communication barrier and verify requirements have been met

"Companies that desire to be Best-in-Class should consider better requirements definition early on. This includes all aspects of requirements management such as better process, tools, workflows, ownership, traceability, change management, and ability to use multiple formats. Requirements should also be linked to customer needs, which reduces drastic scope change later on. By achieving this desired state, we believe it will lead us to have more competitive solutions for our customers and more predictability in development costs and timelines (fewer overruns)."

~ Maryane Chapman, Director, Integrated Systems Engineering, Pitney Bowes

Chapter One: Benchmarking the Best-in-Class

One of the top pressures facing engineering executives today is increasing market demand to build "smarter" product as found in Aberdeen Group's June 2008 report, Engineering Executive's Strategic Agenda. Further supporting this growing need, Aberdeen's November 2008 report, Engineering Evolved: Getting Mechatronics Performance Right the First Time, found that 66% of the products developed last year contained embedded systems. Clearly developing smart products is an important trend manufacturers must pay attention to in order to be competitive in today's market. However, it is inherently difficult, and getting it right requires new approaches to developing products. One of these approaches requires making system engineering a core engineering discipline.

To understand successful system engineering approaches, Aberdeen studied the experiences of 150 companies from August to October 2009 to understand how which systems engineering practices result in bringing in more revenue while keeping costs down and staying on budget.

The Business Need for System Engineering

To understand the external factors affecting system engineering, respondents were asked to pick the top two pressures driving them to improve system
engineering processes. The top pressures all indicate a focus on customers (Figure 1).

**Figure 1: Top Business Pressures Driving System Engineering Improvement**

<table>
<thead>
<tr>
<th>Pressure</th>
<th>Percentage of Respondents, n=150</th>
</tr>
</thead>
<tbody>
<tr>
<td>Market demand for lower cost products</td>
<td>38%</td>
</tr>
<tr>
<td>Need to differentiate from competitors</td>
<td>28%</td>
</tr>
<tr>
<td>Market demand for more product features</td>
<td>27%</td>
</tr>
<tr>
<td>Need to launch products quickly (before</td>
<td>25%</td>
</tr>
<tr>
<td>competitors)</td>
<td></td>
</tr>
<tr>
<td>Reduced development budgets</td>
<td>21%</td>
</tr>
</tbody>
</table>

Source: Aberdeen Group, October 2009

The number one pressure on organizations is the need to take cost out of products. Clearly this is a reflection of the current economy that indicates the market has become more cost conscious. Companies are recognizing the cost saving opportunities of replacing mechanical components with software. However, they are realizing that to be successful, they need to be good at system engineering and this is driving improvement initiatives.

The second and third pressures show a need to align products with what customers want by differentiating them and providing more features. Companies are adding embedded software as a way to do this, but they are realizing this is also requiring improvements in system engineering to be successful. Launching before competitors is a time to market pressure which means companies are looking for better system engineering practices to improve the efficiency of their development process so that they can get their products to market first and capture market share.

**Fast Facts**

When compared to the Industry Average, successful system engineering enables Best-in-Class companies to:

- Earn profit margins that are **2.3-times** higher
- Take **three-times** more cost out of products
- Experience development cycles that are **6.2-times** shorter
Why is System Engineering Difficult?

Obviously the pressures driving improvements in system engineering demonstrate the effect system engineering has on the top and bottom line of a company. However, what makes system engineering so difficult in the first place? Figure 2 displays the top challenges of system engineering. Respondents were asked to pick their top three challenges.

Figure 2: Top Challenges of System Engineering

- Lack of cross functional knowledge: 37%
- Difficulty predicting system behavior: 37%
- Ensuring all design requirements are met: 34%
- Early identification of system level problems: 31%
- Cultural changes required to work across engineering disciplines: 31%

Source: Aberdeen Group, October 2009

The top challenges fall into three themes. The first is associated with bringing together a team of engineers from different engineering disciplines. With a team consisting of mechanical, electrical, and software engineers, there are natural silos of knowledge that must be overcome in order for them to work together to develop an integrated system. The second theme is predicting how the system will behave and identifying problems with system behavior as early as possible. Because the different components are all designed in tools developed for specific engineering disciplines, it is very difficult to get much insight into how the system will behave when it is put together. When problems are not found until the first physical prototype is built, it is often very late in the development cycle when it is far more expensive and time consuming to make changes and corrections. The final and third theme is making sure the product that was originally intended is actually what is designed and build. System level problems found late in the process often require sacrificing requirements just so that the system will work. In addition, simply not understanding the requirements means they will not be implemented correctly. This confusion is even more likely when requirements cross multiple engineering disciplines.
What is interesting is that all of these challenges point to things that are solved with good system engineering practices. System engineers have some expertise in each of the engineering disciplines involved so they are in a good position to bridge the lack of cross functional knowledge. They can use this expertise to make good decisions about the system architecture and plan the system requirements. By doing this work up front, the system is more likely be to designed and built as originally intended and it will be easier to validate that the requirements are met throughout the development process. With a better understanding of the requirements, it is also easier to catch problems with the design earlier in the process.

"Communication between engineering disciplines is key. Design / technical reviews across engineering teams in addition to those within teams help uncover issue earlier in the development phase."

~ Thomas Wright, Vice President of Engineering, ATSI Holdings, Inc

The Maturity Class Framework

To understand successful system engineering practices and the business impact it has upon companies, Aberdeen benchmarked the performance of study participants and categorized them as either Best-in-Class (top 20% of performers), Industry Average (mid 50%), or Laggard (bottom 30%).

To ensure that organizations are categorized according to the criterion that most accurately captures what organizations are trying to accomplish, Aberdeen first identified the top business objectives for improving systems engineering (Figure 3).

Figure 3: Top Business Objectives for System Engineering Improvements

Source: Aberdeen Group, October 2009
Based on these answers, four key performance measures that indicate success with achieving these objectives were used to distinguish the Best-in-Class from Industry Average and Laggard organizations. The performance of each of these tiers is displayed in Table 1.

Table 1: Top Performers Earn Best-in-Class Status

<table>
<thead>
<tr>
<th>Definition of Maturity Class</th>
<th>Mean Class Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Best-in-Class:</strong></td>
<td></td>
</tr>
<tr>
<td>Top 20% of aggregate</td>
<td>83% of products met</td>
</tr>
<tr>
<td>performance scorers</td>
<td>product launch</td>
</tr>
<tr>
<td></td>
<td>13% reduction in</td>
</tr>
<tr>
<td></td>
<td>development time</td>
</tr>
<tr>
<td></td>
<td>10% reduction in product cost</td>
</tr>
<tr>
<td></td>
<td>8% increase in product profit margins</td>
</tr>
<tr>
<td><strong>Industry Average:</strong></td>
<td></td>
</tr>
<tr>
<td>Middle 50% of aggregate</td>
<td>69% of products met</td>
</tr>
<tr>
<td>performance scorers</td>
<td>product launch</td>
</tr>
<tr>
<td></td>
<td>2% reduction in</td>
</tr>
<tr>
<td></td>
<td>development time</td>
</tr>
<tr>
<td></td>
<td>5% increase in product cost</td>
</tr>
<tr>
<td></td>
<td>3% increase in product profit margins</td>
</tr>
<tr>
<td><strong>Laggard:</strong></td>
<td></td>
</tr>
<tr>
<td>Bottom 30% of aggregate</td>
<td>36% of products met</td>
</tr>
<tr>
<td>performance scorers</td>
<td>product launch</td>
</tr>
<tr>
<td></td>
<td>10% increase in</td>
</tr>
<tr>
<td></td>
<td>development time</td>
</tr>
<tr>
<td></td>
<td>13% increase in product cost</td>
</tr>
<tr>
<td></td>
<td>3% decrease in product profit margins</td>
</tr>
</tbody>
</table>

Source: Aberdeen Group, October 2009

By meeting scheduled release dates, the Best-in-Class are able to address time to market pressures and can therefore start recognizing revenue sooner. This means they are better positioned to capture market share from competitors.

Also addressing the top objective of reducing development time, the Best-in-Class have been able to improve their efficiency and as a result have reduced development time over the last two years by 13%. Compared to the 10% increase in development time seen by Laggards, the Best-in-Class are clearly at a competitive advantage. They can start enjoying a return on their development investment much sooner than their competitors.

The Best-in-Class have also been extremely successful with taking cost out of their products. In fact, over the last two years, they have been able to remove three-times more cost than the Industry Average. By taking cost out of their product, they can address the top pressure driving system engineering improvement by meeting market demand for lower cost products and lower their prices. Alternatively, they can keep prices the same and enjoy higher profitably with larger profit margins.
Getting products to market on time and taking cost out of products enables the Best-in-Class to enjoy higher profit margins. However, even more important, this also indicates that they are doing a superior job of understanding what their customers want and delivering that. By releasing a product that aligns to what customers want, there is more demand for the product. Consequently, the Best-in-Class can charge a premium for their

Clearly there is a lot of opportunity for success and greater profitability with good system engineering practices. On the flip side, if it is not done well, it can be extremely expensive for a company and puts them at a competitive disadvantage. The question is, what practices have those Best-in-Class companies deployed that lead to their success?

"To facilitate collaboration and take full advantage of the expertise of the entire development team, have people in physical proximity to each other, or have frequent face to face working sessions plus very frequent networked meetings. Really reward cooperation and collaboration among team members, especially those from different disciplines. Celebrate successes and failures alike."

~ Engineering Manager, High Tech Company

### The Best-in-Class PACE Model

Using system engineering to achieve corporate goals requires a combination of strategic actions, organizational capabilities, and enabling technologies that are summarized in Table 2.

**Table 2: The Best-in-Class PACE Framework**

<table>
<thead>
<tr>
<th>Pressures</th>
<th>Actions</th>
<th>Capabilities</th>
<th>Enablers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Market demand for lower cost products</td>
<td>Increase understanding of changed requirements on design</td>
<td>System functions mapped to system requirements</td>
<td>Requirements management solution</td>
</tr>
<tr>
<td></td>
<td>Increase visibility into which subsystem fulfills each requirement</td>
<td>Trade-off studies investigate architecture alternatives</td>
<td>Simulation tools</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Requirements verified using models</td>
<td>Document Management</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Customer need linked to requirements</td>
<td>Integrated Product Data and Requirements Management</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Requirement verification status centrally managed</td>
<td>Product Data Management (PDM)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Owner defined for overseeing each requirement</td>
<td>Product Lifecycle Management (PLM)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>System engineer provides input into verification tests</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Design alternatives evaluated on</td>
<td></td>
</tr>
</tbody>
</table>
### Best-in-Class Strategies

Given the higher levels of profitability the Best-in-Class enjoy, they are clearly doing a better job of addressing the pressures driving improvements in system engineering and its associated challenges. They are doing this by improving system engineering and this is reflected in the strategies they are deploying to support their system engineering initiatives.

#### Figure 4: Best-in-Class Strategies to Improve System Engineering

<table>
<thead>
<tr>
<th>Action</th>
<th>Percentage of Respondents, n=150</th>
</tr>
</thead>
<tbody>
<tr>
<td>Make products more modular (interchangeable, reusable, etc.)</td>
<td>63%</td>
</tr>
<tr>
<td>Increase understanding of design impact due to changed requirements</td>
<td>46%</td>
</tr>
<tr>
<td>Increase visibility into which subsystem fulfills each requirement</td>
<td>38%</td>
</tr>
<tr>
<td>Implement a model driven design approach</td>
<td>38%</td>
</tr>
<tr>
<td>Increase ability to drive global changes across multiple disciplines</td>
<td>33%</td>
</tr>
</tbody>
</table>

Source: Aberdeen Group, October 2009

The Best-in-Class are far more likely than their competitors to deploy a number of strategies. All of these strategies support their ability to simplify the complexity of system design. This helps them manage the development process and requirements better so that they can be sure they deliver the products that customers want that will offer the revenue opportunities they are seeking. This also helps them adapt to changes as well so that they are implemented correctly and do not cause downstream problems later on when everything was not changed as required.

The most popular strategy for the Best-in-Class is to make products more modular. By making them more modular, it is easier to reuse components. Not
only does this save development time, but it means proven subsystems can be used again which reduces the risk of finding system level performance problems later on.

The Best-in-Class put a lot of focus on planning the system. They are 2.5 times more likely than their competitors to ensure they have visibility into which subsystem will fulfill each requirement. This ensures each requirement has been mapped to a subsystem, helping them address the challenge of ensuring design requirements are met in the final product. This also helps them understand how changes to the requirements affect the design. This allows them to better evaluate the impact of a change and ensure that the change is implemented correctly.

Another way the Best-in-Class simplify the complexity of the design is by implementing a model driven approach. This provides a visual reference for the system design that makes it easier for all engineers on the team to understand the system definition, which makes it easier for the different engineering disciplines to overcome the lack of cross functional knowledge, one of the top challenges of system engineering.

Changes are inevitable in the development process. To successfully implement a change requires understanding its full impact and changing everything that is affected. This is very difficult when making changes to a system consisting of interconnected components that cross multiple engineering disciplines. Often this results in changes that are only partially implemented which leads to problems found later on in the development process, another top challenge of system engineering. To address this, the Best-in-Class are three times more likely to increase their ability to drive global changes across multiple engineering disciplines. The other strategies they have implemented to identify how the requirements map to the subsystems also support their ability to execute this strategy.

"Modular design supports system design by enabling reuse which can reduce development costs and timelines."

~ Maryane Chapman, Director, Integrated Systems Engineering, Pitney Bowes
The role of a systems engineer has become increasingly important in the development of modern products as they evolve into integrated systems of mechanical components, electronics, and embedded software. However, it is not easy to find systems engineers. More and more colleges are offering Systems Engineering as a degree, but it will take time to address the increased demand for them. Even still, it takes time for them to become experienced enough to run a complete project. Survey responses show that systems engineers are more likely to be mechanical or electrical engineers or even project managers with the cross functional experience that has enabled them to become systems engineers.

Once a system engineer is hired, what is the best way for the engineering executive to leverage his or her expertise to justify the investment in the position? Overall, most companies use a systems engineer to define the system architecture (Figure 5).

**Figure 5: Responsibilities of a System Engineer**

![Figure 5: Responsibilities of a System Engineer](image)

Source: Aberdeen Group, October 2009

In the next chapter, we will see what the top performers are doing to achieve their performance advantages.

"Focus on the product requirements through-out the development and don’t let scope creep happen."

~ Engineering Manager, Industrial Equipment Manufacturer
Chapter Two: Benchmarking Requirements for Success

Chapter One demonstrated how the pressures and challenges driving improvements in system engineering as well as the business opportunities offered by implementing Best-in-Class system engineering strategies. Chapter Two explores the practices and enabling technologies the Best-in-Class deploy to execute those strategies.

Case Study — Danzco Inc.

Danzco Inc. manufactures unique products that solve design problems or offer improved performance over common products for logging, construction, hydraulics, machining, mechanical power transmission, and cold weather starting diesel engines. They specialize in small quantities of these specialized products. They pride themselves with the special attention they give their customers by focusing on their customer's success. It is this focus that they attribute to their own success.

It was this strong alignment to their customer's needs that drove them to incorporate new technologies into their products. Their customers needed products that cost less to operate, were easier to use, and had lower lifecycle costs. “In this economy, it is more important than ever to meet needs no one else is meeting,” says Ed Danzer, General Manager at Danzco. “A me-too product won't cut it right now. You need to think out of the box.” It was this realization that drove them to look to improve their ability to develop products that contain an integrated system of mechanical components, electronics, and software.

“One of the challenges was that we did not have all the design tools needed for the development of an integrated system or the skill sets,” observed Danzer. “We invested in simulation tools to help us manage the complexity of the design and have better visibility into product behavior.” As a result, they were able to take on new business and have identified an area for potential expansion – an impressive and exciting accomplishment in this economy.

“While the tools can get you a long way, they are not the silver bullet by themselves,” warns Danzer. “The software helps make better products if properly done, but garbage in, equals garbage out. There is still a learning curve for the tools and you have to know what you are doing.” The lesson is, when done well, system engineering processes offers new business opportunities, but there are challenges and knowledge barriers that must be considered to achieve success.

Competitive Assessment

Aberdeen Group analyzed the aggregated metrics of surveyed companies to determine whether their performance ranked as Best-in-Class, Industry Average, or Laggard. In addition to having common performance levels, each class also shared characteristics in five key categories: (1) process (the system
engineering process they use); (2) organization (defined ownership of processes); (3) knowledge management (how design data is managed and exposed); (4) technology (the tools that enhance and support system engineering); and (5) performance management (metrics used to assess product performance during development). These characteristics (identified in Table 3) serve as a guideline for best practices, and correlate directly with Best-in-Class performance across the key metrics.

Table 3: The Competitive Framework

<table>
<thead>
<tr>
<th>Process</th>
<th>System functions mapped to system requirements</th>
<th>Average</th>
<th>Laggards</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(63%)</td>
<td>(53%)</td>
<td>(38%)</td>
</tr>
<tr>
<td></td>
<td>Trade-off studies investigate architecture alternatives</td>
<td>(54%)</td>
<td>(42%)</td>
</tr>
<tr>
<td></td>
<td>Requirements verified using models (while design is under development)</td>
<td>(57%)</td>
<td>(45%)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Organization</th>
<th>System engineer provides input into definition of verification tests</th>
<th>Average</th>
<th>Laggards</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(74%)</td>
<td>(67%)</td>
<td>(53%)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Knowledge</th>
<th>Customer need linked to requirements</th>
<th>Average</th>
<th>Laggards</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(75%)</td>
<td>(59%)</td>
<td>(49%)</td>
</tr>
<tr>
<td></td>
<td>Requirement verification status centrally managed</td>
<td>(57%)</td>
<td>(51%)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Performance</th>
<th>Design alternatives evaluated on multiple criteria (i.e. performance, cost, and risk)</th>
<th>Average</th>
<th>Laggards</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(74%)</td>
<td>(56%)</td>
<td>(48%)</td>
</tr>
<tr>
<td></td>
<td>Product performance criteria defined prior to development work</td>
<td>(100%)</td>
<td>(81%)</td>
</tr>
<tr>
<td></td>
<td>Performance criteria defined within system requirements</td>
<td>(86%)</td>
<td>(75%)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Technology</th>
<th>System Engineering technologies currently in use:</th>
<th>Average</th>
<th>Laggards</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• 52% requirements management</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 67% simulation tools</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 74% document management</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 70% PDM</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 40% integrated product data/requirements management</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 41% PLM</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 42% requirements management</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 58% simulation tools</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 72% document management</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 55% PDM</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 29% integrated product data/requirements management</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 38% PLM</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 38% requirements management</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 50% simulation tools</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 57% document management</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 40% PDM</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 28% integrated product data/requirements management</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 18% PLM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Source: Aberdeen Group, October 2009
Capabilities and Enablers

Based on the pressures driving system engineering improvements, the challenges of system engineering, the strategies deployed, the findings of the Competitive Framework and interviews with end users, Aberdeen’s analysis of the Best-in-Class reveals that there are four key areas companies must focus on to optimize system engineering for more profitable products that are on schedule and on time:

- Capture and manage the needs of customers to optimize revenue potential
- Make sure the goals of all requirements are clear across all engineering disciplines to support collaboration and ensure requirements are met
- Analyze the system architecture to take cost out of products while still meeting customer needs
- Focus of performance validation throughout the development process to avoid finding expensive end of cycle system level problems that put schedules at risk

Processes, organizational responsibility, knowledge management, technology, and performance management all play a role in supporting these four things.

Process

The Best-in-Class processes that support system engineering simplify the complexity, catch problems earlier, and improve the ability to make decisions about system architecture (Figure 6).

Figure 6: System Engineering Processes

Source: Aberdeen Group, October 2009

The Best-in-Class are 66% more likely than Laggards to map system functions to system requirements. This clarifies how each requirement fits into the overall function of the system, making it easier for engineers to understand what is
expected from their work. Not only does this improve communication across engineering disciplines, but it also help ensure the final system will function as originally intended, a top challenge of system engineering.

"The introduction of MCAD software has enabled us to significantly reduce our engineering hours for product development by enabling designers to make smarter, more flexible part and product designs. This flexibility and speed allows us to quickly provide designs for new customer applications which also helps to capture new business. We also are able to quickly adapt to changes driven by CAE testing, validation testing, manufacturing and suppliers as well as changing customer requirements.

~ Mark Stirling, Manager, Continental Automotive

To make better decisions about system architecture, the Best-in-Class are 29% more likely than the Industry Average to conduct trade-off studies to investigate architecture alternatives. By investigating alternatives, they can have greater confidence that the final solution is cost effective, a top pressure driving improvements in system engineering, yet have the required performance to meet customer expectations, a top objective for improving system engineering. This is key to reducing product cost while maximizing revenue potential with products that meet the performance expectations of customers. System models help to support this process.

Another process the Best-in-Class are 27% more likely to implement than the Industry Average is to validate the requirements with system models and simulation models throughout the entire development process. This helps them catch system level problems as early as possible during the design process. As a result, they are more likely to meet their product launch dates and they avoid last minute changes that drive up product cost. Simulation provides insight into product behavior before a physical prototype exists. While it takes time to set up simulation models, the Best-in-Class have still been able to reduce the length of their development process and are meeting product launch dates better than their competitors, proving that the time invested up front is well worth the time it saves by avoiding significant problems found at the end of development.

**Organization**

When dealing with a cross functional team of engineers who are working on an integrated system, it is important that ownership is clearly defined. One of the tasks where responsibilities must be defined is the definition of verification tests. The Best-in-Class are 40% more likely to have a systems engineer provide input into the definition of verification tests. Since the system engineer is defining the system requirements and the system architecture, he or she is also in a good position to define how those requirements should be validated. In addition, by going through the process of defining how a requirement can be
validated, it improves the quality of the requirement definition because thought has been given to how the behavior would be measured when it performs correctly. This capability simultaneously addresses the challenges of making sure the requirements are met in the final product and avoiding problems late in the development cycle. With the requirements clearer, and the tests well defined and tied to requirements, it will be much easier to validate performance, verify requirements are met, and identify problems.

"Having a central data base and transparent workflows, provided by tools such as PLM, means less efforts for data/information retrieval. This also ensures that people work on the 'right' version and there are clear states of the project. This leads to easier collaboration and reduced development time which means better results (i.e. products). "~ Joachim Lentes, Head of Digital Engineering, Fraunhofer IAO

**Knowledge Management**

The information gathered by following Best-in-Class processes needs to be managed and exposed to support team collaboration and improve understanding (Figure 7).

Figure 7: Managing Knowledge for System Engineering

"To improve the success of the product, make sure you clearly understand the voice of the customer and meet their requirements."

~ Manager, A&D Company

One of the top objectives of improving system engineering processes is to improve the ability to meet customer expectations. This is not possible unless it is clear what customers want and those customer expectations are tied to product requirements. The Best-in-Class are 27% more likely than the Industry Average to link customer needs to requirements. By capturing and managing customer needs and typing that to a product requirement, it is clearer to the engineer what a particular component must do so that the intended requirements are met. It also makes it easier to validate the intended behavior because it is clearer how the customer would be using the product in a real use scenario. There is more demand for products that are aligned to the customer
and this higher demand means companies can enjoy higher profit margins on their products.

To support their ability to focus on performance validation, the Best-in-Class are 43% more likely than Laggards to centrally manage which requirements have been verified. This provides more visibility to the status of the project so that other engineers and management know what is working and what isn’t. This improves communication and helps the team collaborate better. By understanding what has been validated it is also easier to identify the root cause when a system level integration issue is found.

**Performance Management**

Defining system performance criteria is extremely important to clarify what is expected out of each requirement. This in turn makes it more likely that the system will perform as intended (Figure 8).

![Figure 8: Performance Management for System Engineering](image)

"With increased attention on the design specification and applying quality metrics, we have been able to reduce development time uncertainty and increase design reliability."

— Bob Stout, Scientist/Embedded Systems Architect, Microfirm

Source: Aberdeen Group, October 2009

By clearly defining the performance criteria, it makes it easier to validate that requirements have been met throughout the entire development cycle. To start things right, 100% of Best-in-Class companies define performance criteria before any development work is started. To make sure this performance criteria is constantly referenced during the development process, the Best-in-Class document it right in the system requirements. Everything starts with the requirements so, by putting in this effort up front with clearly defined requirements, the project has a much greater chance of success.
With performance criteria well defined, it is also easier for the system engineer to make better decisions about the system architecture. While taking cost out of the product is a top pressure, the Best-in-Class are 32% more likely than the Industry Average to look at multiple performance criteria, not just cost, when they define the system architecture. They look at the full picture thus ensuring they will deliver a product that truly meets customer needs in a cost effective way, allowing them to optimize product profit margins.

"What has made the biggest difference in our process for developing Mechatronic products is System Engineering and Requirement Management functionalities integrated on a Mechatronics PLM system."

~ Engineering Manager, Instrumentation Controls Manufacturer

**Technology**

Technology plays a very important role in supporting the Best-in-Class capabilities of system engineering. There are a wide variety of technologies in use to support system engineering. There are a few that are particularly differentiated and more likely to be used by Best-in-Class companies than their competitors, leading to superior performance (Figure 9).

**Figure 9: System Engineering Technology**

<table>
<thead>
<tr>
<th>Technology</th>
<th>Percentage of Respondents, n=150</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Data Management (PDM)</td>
<td>Best-in-Class: 70% All Others: 49%</td>
</tr>
<tr>
<td>Simulation tools</td>
<td>Best-in-Class: 67% All Others: 55%</td>
</tr>
<tr>
<td>Requirements management solution</td>
<td>Best-in-Class: 52% All Others: 41%</td>
</tr>
<tr>
<td>Product Lifecycle Management (PLM)</td>
<td>Best-in-Class: 41% All Others: 30%</td>
</tr>
<tr>
<td>Integrated Product Data/Requirements Management</td>
<td>Best-in-Class: 40% All Others: 29%</td>
</tr>
</tbody>
</table>

Source: Aberdeen Group, October 2009

One of the most important things for successful system engineering is making sure the requirements are clear and well understood by the entire development team. To support this, the Best-in-Class are 23% more likely than the Industry Average to use a requirements management solution. Requirement
management centralizes the requirements so that they are available to the entire development team. When there is a change to the requirements, the requirements management solution ensures that to updated requirement is made available to the entire team. In addition, requirement management solutions have features such as requirements traceability that allow the "life" of a requirement to be tracked. This means that the requirement can be traced from its origin, all the changes it went through, to how it was eventually implemented in the final product. This is especially important when there is a problem with meeting a requirement. In a product as complex as an integrated system, this can be very different to track down. Requirement management solutions also offer another important feature of managing a complex system - visibility to dependencies. This means when there is a change to a requirement, it is easy to know what else has been affected.

The Best-in-Class are more likely to centralize design data. Centralizing design data means everyone has access to the latest version, improving collaboration. Product Data Management (PDM) and Product Lifecycle Management (PLM) are used to support this. The solutions also provide version control so that data is not overwritten and it is clear what data is complete and which is still under development. PLM offers the benefit of PDM but also supports development processes such as change management, project management, configuration management, and collaboration.

The Best-in-Class are 40% more likely to use a solution that integrates requirements management with the storage of the design data. This allows the requirement to be linked to the design or code that met that requirement, improving the ability to confirm a requirement was implemented correctly, in addition to making it easier to manage the relationship between the requirements and design when changes are made, supporting two of the top Best-in-Class strategies for system engineering (Figure 4).

"MCAD software coupled with our PLM system permits us to manage product configurations very accurately. Configuration and change management controls implemented in the PDM system enforce best practices for control and release of designs and Product configurations. The PDM system also manages other non-CAD key specification documents for the products and component parts. All product data is related through product structures in the PDM system. Since the system is web based all data is available to users throughout the globe with access rights controlled by user roles."

~ Mark Stirling, Manager, Continental Automotive

The ability to manage documentation is also important. This includes details such as product specifications and test plans. Centralizing this type of information improves the ability for the team to work together and collaborate
because everyone has access to the same information and no one has to waste time trying to track down who has the latest version of a document.

Finally, simulation tools are very important to assess the performance of product behavior before a physical prototype is available. This makes it easier to catch problems as early as possible and also makes it possible to make more well information decisions. Simulation tools are available for each engineering discipline, but there are also some more advanced tools for integrated simulations that represent aspects of the design from multiple engineering disciplines. This provides even better insight into the integrated system behavior and makes it easier for engineers to understand the impact of their design decisions on other engineering disciplines.

"Evaluating different design alternatives using multiple criteria is very helpful. It starts everything right from the beginning. We can generate consensus about the right criteria & weights with a moderated process that advances the design substantially. Additionally the transparency for comparing different alternatives is a big benefit."

~ Joachim Lentes, Head of Digital Engineering, Fraunhofer IAO
One of the most important parts of developing systems is managing the requirements. The requirements can be considered the plan for the design. Any project without a good plan has little chance for success. On the other hand, a project with a good plan where it is clear to everyone what they need to accomplish, who will do what, and the interdependencies between the task each person is working on, is far more likely to succeed. This is exactly the effect well managed requirements have on the development team.

Many companies use spreadsheets to manage requirements. This makes sense because it is quick and easy to enter information. However, spreadsheets lack the capabilities that support the ability to execute the Best-in-Class strategies for system engineering. Understanding the impact of changed requirements on the design requires links from the requirements to the design. Without this understanding, the change will not be implemented correctly because system complexity makes it difficult to determine which portions were affected and which engineers need to know about it.

Spreadsheets can easily organize requirements, but are not well suited for managing interdependencies that would provide the required visibility to design relationships. Further, the ability to then drive the change across multiple engineering disciplines is even more complicated, especially without the ability to create role based and function based views of the requirements.

continued
With a lot of thought and careful organization a spreadsheet can be set up to increase visibility into which subsystem fulfills each requirement. However, one of the main reasons for doing this is to understand the impact of changes. To make a change to the spreadsheet, it becomes a very manual task of sifting through rows and rows of requirements to make sure each subsystem is updated correctly and even still, it is hard to know if everything that would be impacted was updated.

When dealing with something as complex as system design, solutions that were developed specifically with the challenges of managing requirements in mind should be strongly considered. Requirements management solutions have the capabilities that will support the ability to execute the Best-in-Class strategies for system engineering. These solutions provide traceability across engineering disciplines, enable different role based views of the requirements, and manage dependencies between requirements that are critical for change management. These features will make sure that that well developed plan continues to be valid throughout the life of the project because it will correctly reflect the changes made along the way. As a result, a requirements management solution is a powerful tool for guiding the project to success.

"Throughout my career in industrial embedded systems design, failures can cost lives, hard money (many millions of dollars in most cases), and catastrophic environmental damage. Specifications are dictated by industry, legal, and/or international standards. If those aren’t met, there simply is no product - period. This imposes a level of discipline in the design process that enforces software engineering rather than simply hacking code. The functional spec is simply a tool to capture the design, which facilitates the design review process. Some of the latest software development and management fads encourage less discipline, which is a mistake."

~ Bob Stout, Scientist/Embedded Systems Architect, Microfirm
Chapter Three: Required Actions

Fast Facts

- Use multiple design criteria to define system architecture and add that criteria to the system requirements
- Requirements should be linked to higher level system functions as well as to the overall customer need it meets
- Integrate code and design data with requirements management

Whether a company is trying to move its performance from Laggard to Industry Average, or Industry Average to Best-in-Class, or maintain its Best-in-Class advantage, the following actions will help spur the necessary performance improvements:

**Laggard Steps to Success**

- **Define performance criteria within system requirements.** Predicting system behavior is a top challenge for system design, but it is virtually impossible to get the behavior right if the performance criterion is not clear. Defining that criteria up front and putting it right in the system requirements makes it clearer to the designer what must be accomplished, thus increasing the chances of success. The Best-in-Class are 72% more likely to do this.

- **Map system functions to system requirements.** One of the reasons making sure the requirements are met in the final product is a top challenge is because the requirements are not understood. By mapping which function the requirement supports, the requirement becomes much clearer to the designer. The Best-in-Class are 66% more likely to do this.

- **Verify requirements with system models throughout the development process.** Identifying system level problems earlier in the development process is a top challenge of system engineering. By using system modeling tools to verify the requirements are met on an ongoing basis throughout the entire development process, many problems can be caught much earlier. The Best-in-Class are 63% more likely to do this.

**Industry Average Steps to Success**

- **Link customer needs to the requirements.** By aligning products to customer needs, there will be more demand for those products, thus allowing higher profit margins to be realized. The Best-in-Class are 27% more likely to do this.

- **Evaluate multiple performance criteria when defining product architecture.** Cost is a driving pressure for better system engineering, but it is important to also ensure the product meets customer performance expectations as well. By evaluating multiple criteria, more informed decisions can be made about the product architecture, optimizing its profitability. The Best-in-Class are 32% more likely to do this.

- **Integrate code and design data with requirements management.** Many Best-in-Class strategies involve the ability to improve change management. When code and design data are integrated with requirements, tracing the impact of changes is easier which is needed to
implement changes correctly. It is also easier to verify requirements were implemented correctly. The Best-in-Class are 38% more likely to have this level of integration.

Best-in-Class Steps to Success

- **Conduct a gap analysis to ensure all requirements are traceable to a component.** A top challenge is making sure all requirements are met. The Best-in-Class are already doing a variety of things to address this challenge. However, a gap analysis would identify any holes where the requirements will not be fulfilled. This also makes it easier to track down the root cause of why requirements were not met. System modeling is a way to do this. Currently, 21% of the Best-in-Class do this, showing it is on the cutting edge for successful companies.

- **Conduct a post mortem to analyze how many of the initial requirements were changed.** This process will identify areas of improvement for better initial requirements definition to increase the chance of getting them right the first time. Many of the strategies improve change management processes, but this step will help reduce the number of changes in the first place. Currently, 32% of the Best-in-Class are doing this, which is more than competitors, but it will help the Best-in-Class maintain a competitive advantage.

- **Integrate MCAD and ECAD design tools.** One of the top challenges is overcoming the lack of cross functional knowledge. One of the barriers to this is that each engineering discipline has its own design tools. Several solutions are now available that enable the integration of MCAD and ECAD, making it easier for mechanical engineers and electrical engineers to work together. Currently, 30% of the Best-in-Class are using these tools, but considering the impact the lack of collaboration between engineering disciplines has on system design, more should take advantage of this.
Aberdeen Insights — Summary

Good system engineering practices offer a lot of opportunity for greater profitability. However, poor system engineering practices are extremely costly. To be successful with system engineering, companies must focus on the needs to their customers and translate them into design requirements to maximize revenue potential. Requirements must be clear across all engineering disciplines. Understanding what each is responsible for will make collaboration easier and overcome the lack of cross functional knowledge across engineering disciplines. With a thorough trade-off analysis of the system architecture it can be optimized for the most cost effective, highest performing product. Finally, validating system performance throughout the development cycle is critical to catch problems as early as possible during the design process.
Appendix A: Research Methodology

Study Focus

Respondents completed an online survey that included questions designed to determine the following:

- What is driving companies to improve system engineering
- The challenges of system engineering
- The actions these companies are taking to improve system engineering
- The capabilities and technology enablers they have in place to support system engineering

The study identifies emerging best practices for system engineering and to provide a framework by which readers could assess their own capabilities.

Between July and August 2009, Aberdeen examined the use, the experiences, and the intentions of 170 enterprises in a diverse set of industries. Aberdeen supplemented this online survey effort with interviews with select survey respondents, gathering additional information on their strategies, experiences, and results.

Responding enterprises included the following:

- **Job title:** The research sample included respondents with the following job titles: Executive level manager (11%); VP/Director (20%); Manager (34%); Engineer (27%); and other (8%).
- **Industry:** The research sample included respondents from a wide cross section of industries. The sectors that saw the largest representation in the sample were aerospace and defense (14%), industrial equipment manufacturing (24%); industrial product manufacturing (14%); medical devices (12%), automotive (13%); high tech (16%), and other (7%).
- **Geography:** The majority of respondents (66%) were from North America. Remaining respondents were from Europe (24%), the Asia / Pacific region (8%), and from the rest of the world (2%).
- **Company size:** Twenty-five percent (25%) of respondents were from large enterprises (annual revenues above US $1 billion); 35% were from midsize enterprises (annual revenues between $50 million and $1 billion); and 40% of respondents were from small businesses (annual revenues of $50 million or less).
- **Headcount:** Thirty-seven percent (37%) of respondents were from small enterprises (headcount between 1 and 99 employees); 22% were from midsize enterprises (headcount between 100 and 999 employees); and 41% of respondents were from large businesses (headcount greater than 1,000 employees).
### Table 4: The PACE Framework Key

<table>
<thead>
<tr>
<th>Overview</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aberdeen applies a methodology to benchmark research that evaluates the business pressures, actions, capabilities, and enablers (PACE) that indicate corporate behavior in specific business processes. These terms are defined as follows:</td>
</tr>
<tr>
<td><strong>Pressures</strong> — external forces that impact an organization’s market position, competitiveness, or business operations (e.g., economic, political and regulatory, technology, changing customer preferences, competitive)</td>
</tr>
<tr>
<td><strong>Actions</strong> — the strategic approaches that an organization takes in response to industry pressures (e.g., align the corporate business model to leverage industry opportunities, such as product / service strategy, target markets, financial strategy, go-to-market, and sales strategy)</td>
</tr>
<tr>
<td><strong>Capabilities</strong> — the business process competencies required to execute corporate strategy (e.g., skilled people, brand, market positioning, viable products / services, ecosystem partners, financing)</td>
</tr>
<tr>
<td><strong>Enablers</strong> — the key functionality of technology solutions required to support the organization’s enabling business practices (e.g., development platform, applications, network connectivity, user interface, training and support, partner interfaces, data cleansing, and management)</td>
</tr>
</tbody>
</table>

Source: Aberdeen Group, October 2009

### Table 5: The Competitive Framework Key

<table>
<thead>
<tr>
<th>Overview</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Aberdeen Competitive Framework defines enterprises as falling into one of the following three levels of practices and performance:</td>
</tr>
<tr>
<td><strong>Best-in-Class (20%)</strong> — Practices that are the best currently being employed and are significantly superior to the Industry Average, and result in the top industry performance.</td>
</tr>
<tr>
<td><strong>Industry Average (50%)</strong> — Practices that represent the average or norm, and result in average industry performance.</td>
</tr>
<tr>
<td><strong>Laggards (30%)</strong> — Practices that are significantly behind the average of the industry, and result in below average performance.</td>
</tr>
</tbody>
</table>

In the following categories: |
| **Process** — What is the scope of process standardization? What is the efficiency and effectiveness of this process? |
| **Organization** — How is your company currently organized to manage and optimize this particular process? |
| **Knowledge** — What visibility do you have into key data and intelligence required to manage this process? |
| **Technology** — What level of automation have you used to support this process? How is this automation integrated and aligned? |
| **Performance** — What do you measure? How frequently? What’s your actual performance? |

Source: Aberdeen Group, October 2009
Aberdeen research indicates that companies that identify the most influential pressures and take the most transformational and effective actions are most likely to achieve superior performance. The level of competitive performance that a company achieves is strongly determined by the PACE choices that they make and how well they execute those decisions.

Source: Aberdeen Group, October 2009

<table>
<thead>
<tr>
<th>PACE and the Competitive Framework – How They Interact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aberdeen research indicates that companies that identify the most influential pressures and take the most transformational and effective actions are most likely to achieve superior performance. The level of competitive performance that a company achieves is strongly determined by the PACE choices that they make and how well they execute those decisions.</td>
</tr>
</tbody>
</table>

Source: Aberdeen Group, October 2009
Appendix B: Related Aberdeen Research

Related Aberdeen research that forms a companion or reference to this report includes:

- *Engineering Executive Agenda,* June 2008

Information on these and any other Aberdeen publications can be found at [www.aberdeen.com](http://www.aberdeen.com).

Since 1988, Aberdeen's research has been helping corporations worldwide become Best-in-Class. Having benchmarked the performance of more than 644,000 companies, Aberdeen is uniquely positioned to provide organizations with the facts that matter — the facts that enable companies to get ahead and drive results. That's why our research is relied on by more than 2.2 million readers in over 40 countries, 90% of the Fortune 1,000, and 93% of the Technology 500.

As a Harte-Hanks Company, Aberdeen plays a key role of putting content in context for the global direct and targeted marketing company. Aberdeen's analytical and independent view of the "customer optimization" process of Harte-Hanks (Information – Opportunity – Insight – Engagement – Interaction) extends the client value and accentuates the strategic role Harte-Hanks brings to the market. For additional information, visit Aberdeen [http://www.aberdeen.com](http://www.aberdeen.com) or call (817) 723-7890, or to learn more about Harte-Hanks, call (800) 456-9748 or go to [http://www.harte-hanks.com](http://www.harte-hanks.com).

This document is the result of primary research performed by Aberdeen Group. Aberdeen Group's methodologies provide for objective fact-based research and represent the best analysis available at the time of publication. Unless otherwise noted, the entire contents of this publication are copyrighted by Aberdeen Group, Inc. and may not be reproduced, distributed, archived, or transmitted in any form or by any means without prior written consent by Aberdeen Group, Inc. (071309b)
The Requirements Engineering Process: Panel Discussion on Requirements Management Tools

Moderated by Don Brown, Chairman, CPDA

At PLM Road Map™ 2009, our panel discussion on requirements management (RM) included individuals from leading software suppliers and a top end user.

**John Carrillo of IBM** consistently emphasized a broader perspective of “how we get better” that focused on the full requirements engineering process rather than just on requirements management. Without a requirements engineering process in place to ensure the linking of requirements through design and test, the system will not meet its intended objectives. There is a major need to standardize on common processes. Connecting the current processes represents another major step forward.

The challenge extends to issues concerning the centralization of requirements in a single-access repository as well. That is, best practices must be in place and supported broadly across the organization, with the parameters and patterns defined to support traceability; and to associate textual information as well as attributes to provide a context for the requirements. The approach then extends well beyond the issue of what should be done, by establishing a meaning and significance for the process. It becomes part of the culture; how a firm does business. This is the shift taking place for many, and what it means to have the software engineering become more integrated with mechatronics. The integration raises new demands back on the mechanical, electrical, and systems engineering domains.

**Mike Cronin, Chairman and CEO of Cognition**, positions requirements management as “a portal into the entire system model.” Requirements management should not be treated as one isolated stovepipe application, but rather as one view of the product model that also contains risk assessments, FMEA (failure mode and effects analysis), critical parameter management with transfer functions, and performance metrics. The critical information must be assembled into one unified model. Requirements cannot be met in isolation; they must be part of the unified system model.

In the “old days,” individual gurus ran requirements. The RM applications were not only hard to use, but the concept was difficult as well. The military
specification standards that everyone had to follow contributed to the complexity. Now these special cases no longer dominate the market. The vision of pervasive requirements management addresses the needs across the whole engineering organization. Anyone may have an initiative launched to improve the product development process. The effort must acknowledge system engineering without necessarily having to say, systems engineering, and it must embrace the new concept of a unified product model.

To broaden adoption to occasional users demanding ease of use, Cognition implemented live templates for nearly any operation within product development. With roughly one hundred templates provided out of the box, a template tailored to a company’s specific processes assists in working with Design of Experiments, or the House of Quality, or Cost Analysis. Simply fill in the template. If the projects are broken down into smaller tasks, and if people start with a template with only a little training, they accomplish useful work right away. The templates make it easy to dive in, and the support of a Web 2.0 interface lets new users get work done right away, even in complex applications.

The Cognition Cockpit also employs a broad definition of the “Voice of the Customer.” The voice is not only that of the customer, but also the voice of safety, of compliance, of the business; even of OSHA. A whole raft of issues impacts the early definition of customer needs. By grouping them together, the broad range of users eventually discovers they are all talking about the same thing.

Mike sees a wave of change affecting his customers. That wave is the recognition that requirements should be managed and accessed by all project engineers and tied to test and validation, to assure that all requirements are met, with the variation also reported.

In considering the voice of the customer, Lee Garf of PTC considers two different scenarios. The first relates to actually building a product for a market, addressing a broad range of requirements, and a broad range of customers, with inputs from multiple sources. A little bit of art and science coalesce in coming up with the key requirements facing the customer. The second scenario involves established customers. Many firms actually build for a specific customer. A spreadsheet or an exported list from a database may be all that is needed to start.

Very different flavors may be involved with the front end of the requirement process. In either case, the critical issues relate to getting those requirements into a system. Suppose at the basic level a spreadsheet serves as the initial tool. A lot of advantages flow from having that information in a database. One advantage, for example, is the ability to baseline the set of requirements following initial discussions with the customer. Then, developers later can go back in history to assess the actual agreement and commitments. The flowdown helps as the decomposition and allocation of requirements flows down to systems or functions, and then finally to the product. A database provides the
ability to manage those relationships. Whether a change is applied midstream or at the end of the process, a where used search lays out the impact. For any change in the product, in a feature for CAD geometry or in a line of code in software, the impact can be readily assessed. In the end, there is a need to verify and validate as well. Tools out there now support those efforts, which close a big gap in reconciling whether or not requirements are met. As both software and tools improve, it becomes much easier to trace the customer needs from the front end to requirements, and to trace the relationships across the full development process.

Across a broad spectrum of companies from Procter & Gamble to Intel to Boeing, Mark Sampson of Siemens PLM Software confronts a common set of symptoms. For example, late cycle integration problems can typically be traced back to a failure to keep up with a requirement. That is, a change in a requirement does not get communicated to a downstream user. Or on the downstream implementation side, an engineer will find he cannot complete a task to fulfill a requirement but the message does not reach the right places upstream. The missed communication subsequently shows up as an integration problem.

From a usability perspective, communication must be understandable by everyone. But their willingness to listen depends on their maturity – much like teenagers. Maturity-wise, organizations that have experienced near misses or actual project failures are interested in getting themselves back on track. They are willing to listen. They are motivated to get a handle on requirements to start defining what they actually have to do. They recognize they need to keep up with both the requirements and their impact.

Today, the requirements are too often managed outside the development process. The requirements from systems engineering must be fully integrated into the development process, rather than managed in separate tools, which is what the systems engineering experts are currently doing. Requirements communicate decisions made by the systems engineers, and knowing them in the context of the ongoing design means everyone can adjust their trajectory when they see a problem coming, versus trying to fix the damage after the fact. This idea of bringing requirements and the engineering behind them into the open is critical to providing a proactive means of seeing any problems coming.

A fully integrated framework can present the impact of a decision by a design engineer in terms of specific goals, whether that is framed in terms of the cost, the weight, or other targeted parameters. The approach focuses the means for managing the development effort on the right targets, and tracks the progress through the lifecycle against those targets. Risk assessment and project management are connected. The goal would be for developers to see the impact of any decision that causes a problem downstream.

The only way to get a handle on complexity rests with the right modeling and abstraction. The problem generally comes from not fully respecting system
models. Software developers treat their software models with respect; CAD developers treat their geometry models with respect. They both know individually that without a quality model, there is no product. Most do not, however, treat the system models with respect. Yet these system models become increasingly important to the survival of a product development organization. Everyone needs to get a handle on those models and get them under control.

Considering the impact of changes, one study determined that 3% of a set of product requirements changes every month. That in turn mandates two programs. First, shorten the lifecycle to get that product to market to hit the window of opportunity. Second, consider from the very beginning what may be needed architecturally to facilitate changes later in the lifecycle to make those changes easier to implement.

Because of the limited time-value of the input, the system needs to capture any requirement change immediately, not later. Capture the rationale as well. Get that information out there into the workflow, or wherever it shows up, to insure a thorough analysis of any impact. The ability to capture that impact can then influence any decisions made now versus later.

The only way to possibly keep up as the development cycle continues to accelerate is to integrate requirements with the lifecycle. Processing changes become part of the daily life in getting through the development process. For example, does the electronics business validate TVs? The first time a TV powers up is when the customer plugs it in. Electronics companies do not test TVs coming off the line because they rate test and validation as a waste. They pull it off because they calibrate each step in their process. They monitor the process that produces the televisions. They know when something is going to go wrong that will impact the quality of the television coming off the line. They validate the process of producing the TV, not the product itself. This represents a different way of thinking about the process. There is no question that most companies have a long way to go to get there. But everyone can start by capturing changes now, rather than later.

In terms of supporting a full requirements engineering process, Ed Griffor of Chrysler Group LLC reports that his engineers have all been trained with a basic introduction to CMMI (capability maturity model integration). As a discipline, CMMI appears incomprehensible at first. But with practice, the engineers begin to understand the process areas in terms of the data produced, and they acquire a concrete understanding of it. As they improve their understanding of the interdependencies in their own product development flow, they gain better insight into the relationships of who owns what data. Suddenly that CMMI environment becomes real.

Clearly, there are challenges in getting CMMI and process metrics accepted by mechanical engineers and designers. The light goes on as soon as metrics are
introduced. For example, Tata Consulting Services (TCS) built a development environment with certification at level-four maturity. They then deployed it inside of their organization, so people working at TCS interacted with the framework in every step of their daily work. They never mentioned the term *process enhancement*, however; nothing at all was said about it. That is the way to start. Do not use the word process; just let people work it.

Ed also considers respect to be critical, and respect must have an operational meaning as well. With the prodding of an Italian group, the position of “Product Data Engineer” was announced for every organization. Similarly, a cost engineering position was created in every organization, in every department. Ed considered those titles carefully, and came to the conclusion that Product Data Engineer represented one of the best titles he had ever heard. In a sense it created the respect that was needed.

http://cpd-associates.com/
Innovation for smarter products

The IBM Approach to PLM

A Whitepaper by IBM

Smarter products: The building blocks for a smarter planet
We live in a world in which a huge amount of data is being captured. Intelligence is being infused into systems and processes of every kind — and nowhere is this transformation more evident than in the creation of a new generation of smarter products.

Smarter products are the building blocks for a smarter planet. Embedded with increasingly sophisticated software and instrumentation, able to connect and communicate with other devices and fully capable of responding intelligently to user needs, these products are transforming the way our world works.

Organizations in every industry can create incremental value from global interconnection throughout products, systems, applications and the Internet. Indeed, the development of smarter products frees manufacturers to participate in the larger systems that define this new world.

Product lifecycle management: Plenty of strategic challenges
The complexity inherent in today’s environment profoundly affects product development and the product lifecycle. In response, product lifecycle management (PLM) has risen from the ranks of engineering and now requires a more strategic and end-to-end approach throughout the enterprise.

Companies must demonstrate the same level of proficiency in complex systems integration and software development as they have demonstrated in conventional product design and manufacturing. They must seek effective ways to manage and integrate software throughout a complex, ever-expanding supply chain in which components are sourced from different locations, arrive with software already embedded and are assembled in various combinations.

For most product companies, significant obstacles stand in the way of achieving this level of integration. The persistence of siloed development organizations runs counter to accelerating in product innovation. Evolving business models and the interdependency of supply chains add to the
challenge. And regulatory concerns increase the demands on product data traceability throughout the product lifecycle.

**Release the full power of innovation at your organization**

Some product companies have found a way to achieve success in the challenging world of product lifecycle management. Working with IBM®, these organizations use systems engineering to spark innovation, collaboration and efficiency throughout engineering domains. They take a new approach to product lifecycle management, one that delivers an integrated enterprise-wide set of capabilities and supports tighter collaboration within increasingly complex value chains.

**The IBM approach to product lifecycle management creates value quickly. Your company can:**

- Strategically transform business processes to build new capabilities, save costs, accelerate product introduction and create new markets.

- Make the most of the design chain by automating business processes that use existing investments in industry-leading applications and data.

- Adopt an advanced systems-engineering approach to manage all product interrelationships throughout engineering disciplines.

- Enable product lifecycle collaboration throughout your ecosystem of partners and suppliers.

- Build a strong competency in software development and delivery.

- Establish and maintain a secure, reliable, cost-effective and flexible PLM infrastructure.
IBM offers PLM capabilities that are unique in the industry
When your company incorporates PLM solutions with strategic business processes, it gains the power and process flexibility to meet the demands of today’s complex, interconnected world — and to prepare for a challenging future. With consultants and technical staff in 160 countries, IBM delivers systems, solutions and services to over 20,000 customers throughout all industries — from automotive makers to medical equipment manufacturers. Organizations use IBM experience, tools and capabilities to conceptualize, design and build innovative products. IBM offers a proven track record of delivering advanced solutions in a variety of markets. IBM supports its strong PLM focus through IBM Centers of Excellence, a network of IBM PLM ISV Business Partners, a world-renowned IBM research group and global delivery capability. When PLM is effectively deployed as an enterprise solution, the results can be powerful. IBM clients know to expect:

**Innovation.** Make the most of strong PLM capabilities throughout your company’s broader innovation agenda. IBM solutions provide measurable and real-world business value.

**Integration.** Combine technology and business insight through integration capabilities from IBM — the quickest way for your organization to create competitive advantage.

**Collaboration.** Capitalize on the IBM collaborative network of PLM global resources and IBM PLM ISV Business Partners.

**Experience.** IBM has been delivering PLM solutions for more than 25 years. IBM itself successfully deployed an integrated product development transformation and offers broad experience as a globally integrated enterprise.
IBM product lifecycle management
Discover an extensive portfolio of assets and services that delivers powerful PLM capabilities.

IBM strategy and transformation solutions make possible the global restructuring of product development at your company. With IBM strategy and business consulting, product development transformation solutions and operational process design services, you can unlock the full value of PLM. Discover the power of IBM PLM Business Value Accelerators, proven hardware-to-solutions strategy expertise and a forward-thinking approach to global integrated enterprise technologies.

IBM business solutions put predefined assets and services to best use, helping you create business value and differentiation. Built with an extensive software portfolio including IBM Rational®, IBM WebSphere®, IBM Lotus®, IBM Tivoli® and IBM DB2®, IBM provides solutions for product, portfolio and requirements management, model-driven software and systems development, design chain collaboration and optimization, asset management, master data management and analytics.

IBM infrastructure and operation solutions deliver lifecycle support of PLM implementations. With flexible and cost-effective application management services and Cloud computing from IBM, success comes quickly. IBM offers high-performance computing systems and storage solutions that, when coupled with IBM infrastructure managed services, make it possible for your company to implement a new approach to product lifecycle management. Change is possible. The tools exist today.
To learn more, please contact your IBM representative or IBM Business Partner or visit: ibm.com/solutions/plm
General Electric Conquers Collaboration Challenges with ENOVIA Software

By John Myers, ConnectPress Editor

General Electric’s (GE) Wind Energy division knows firsthand the frustrations and challenges of building a global supply chain.

At this year’s Dassault Systèmes Customer Conference Owen Mckenley, Global Configuration Manager, GE Energy, explained how product-lifecycle management could be used to meet these challenges.

Mckenley explained that because GE is made up of five different brands focusing on areas from technology infrastructure to consumer products to commercial entertainment there is a constant need for effective communication and collaboration, therefore creating a number of issues for GE Energy.

Difficulties include the need to create and leverage cross-business synergies within the GE infrastructure and working within GE’s internal IT and engineering infrastructure, an infrastructure that supports over 30,000 engineers and is integrated with a world-wide supply chain.

Working within this context, the GE Energy Wind division, designs, builds, and services wind turbines, wind farm equipment and related accessories with operations taking place in North America, Europe and Asia.

Wind products created by the organization include turbines in the 1.5 and 2.5 megawatt ranges, design for wind farms and lifecycle management and servicing.

Because of the global reach of the Wind division the organization was forced to work with many different offices each with its own bill of materials specifications. A problem that can cause delays or errors to occur as the organization struggled to communicate complex orders without the benefit of a unified system.

Similarly, the operation struggled with the fact that many parts in their database have duplicate part numbers and because the administration for these systems is spread throughout the world it is difficult to make the necessary corrections.

In addition, because the Wind division’s disparate systems did not share a single localized data management system none of the project administrators could update or change the configuration rules for data and part information.
These problems combined with the issue of a lack of localized change processes forced system administrators and project supervisors to work within the constraints of local systems and the rules set that had been laid out for them.

Also, Mckenley explained that throughout the system there was no standardized set of product definitions, which made it difficult to catch errors and ensure files were transferred and handled properly.

Further compounding these issues was a lack of project synchronization, leading to problems caused by differing time zones and periods of work availability and the need to manage specialized engineering resources across widely dispersed locations.

For this reason GE Energy adopted Dassault Systèmes’ ENOVIA product-life-cycle management (PLM) and collaboration system. ENOVIA is a tool-set for designers, engineers and managers for tracking and managing data throughout a global supply chain.

In the past GE energy has implemented software solutions in a piece by piece manner. A different software tool was used for every task; planning, design, status updates and tracking each had an independent program assigned to it. ENOVIA allowed all of those problems to be solved with a single solution.

ENOVIA was adopted by the Wind Energy department in 2007 and the company plans to have it implemented throughout GE Energy’s infrastructure soon. By 2010 the company plans to be using ENOVIA not just as a way to track documents and ensure file security but as a system to ensure that engineers and designers can work together to create ideas and plan how best to implement them.

Some of the uses being implemented today include maintaining integrated product design, tracking compliance documents, tracking bill of material exchanges and ensuring the correct protocols are followed during change management and tracking requirements management.

The result is a solution that allowed the teams at GE Energy to combine global engineering resources with global manufacturing resources. The engineering side of the equation was able to take advantage of digital product definition management, a feature that allows them to collaborate across a global network using a single system of record. They also benefitted from the fact that ENOVIA allows them to manage project definitions from any of their global locations.
Simultaneously, the manufacturing side of the operation was able to manage their data through a site-specific framework and manufacture products directly from project definition data.

The result is a system that allows the GE Energy team to take their designs, define them by customer demands, configure release schedule engineering, and plan to optimize the procurement and assembly of the necessary parts.

After the product release ENOVIA helped the team track the manufactured parts from the shipping phase to the process of implementing field changes, all the way to the design’s final end of life.

Summarizing his presentation Mckenley said that the ENOVIA software toolset helped take GE Energy’s Wind division through the processes of large-scale growth, developing lifecycle-focused part design, implementing a deeper supply chain and maintaining and tracking in-the-field servicing.

---

**About ConnectPress Ltd.**