Tech-Clarity

Issue in Focus:

Mastering the Development of Smart Products

Raising Process and Systems Maturity to Improve Quality and Time-to-Market

© Tech-Clarity, Inc. 2014

Table of Contents

© Tech-Clarity, Inc. 2014

Introducing the Issue

Products are getting smarter. Today's "smart" products leverage an intelligent combination of mechanical, electrical, and software to deliver capabilities that simply weren't achievable with traditional approaches. Products are more responsive, adaptable, interconnected, and portable than ever. The trend is significant. As Tech-Clarity's Developing Software-Intensive Products indicates, manufacturers have not only increased the *amount* of software in products over the last five years, but "*the importance of software in products and the level of product innovation driven by software have also increased significantly*."

Smart products are compelling to consumers. It's hard to compete with a car that drives and parks itself or a clothes washer that senses and adjusts to water conditions, fabric types, and dirt levels. Likewise, a medical device that monitors its performance and schedules service when it needs repair offers significantly higher customer value. Benefits will increase even further as smart products become more aware and communicative, such as mobile phones that interact with neighboring devices, appliances, automobiles, and more.

It's clear why consumers like smart products, but there are even more reasons for manufacturers to love them.

It's clear why consumers like smart products, but there are even more reasons for manufacturers to love them. As Tech-Clarity's Systems & Software Driven Innovation concludes, "*Leading companies will look to take advantage of (mechatronics) to improve their ability to tailor products, increase reuse, make agile updates to products in the field, reduce product cost, and lower product development cost*." Smart products are the new reality. What impact do they have on engineering and product development?

Improve Mechatronic Product Development Maturity

The advantages of smart products come at a cost. Developing complex, mechatronic products brings inherent complexity. It requires involvement from more design disciplines and demands synchronization between different design lifecycles. Tech-Clarity research identifies a number of practical challenges this creates, but more importantly shows that they lead to significant, negative business impacts (Figure 1). Specifically the study finds over one-half of companies suffer from time-to-market and quality issues due to mechatronics. This is a significant business issue given that time-to-market is such a significant driver of profitability and the market punishes high-tech companies for missing release dates.

Developing complex, mechatronic products brings inherent complexity.

© Tech-Clarity, Inc. 2014

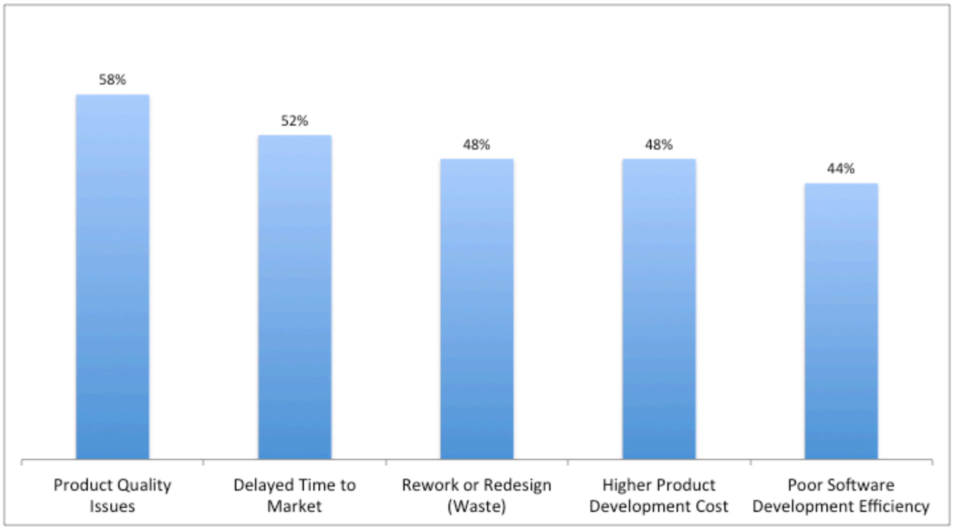

Figure 1: Negative Impacts Resulting from Mechatronic Product Development

Complexity is the new reality. Smart products will continue to grow and become more complex. The time has come for companies to raise the maturity of mechatronic systems development to manage complexity and take greater advantage of opportunities. Fortunately, our research shows that better processes and enabling technology improve results by providing a platform for integration, visibility, and traceability for smart product development. This paper outlines a number of tangible steps to improve mechatronic product development maturity with proven best practices and technology.

The time has come for companies to raise the maturity of mechatronic systems development to manage complexity and take greater advantage of opportunities.

Define the Target

The first step in effective smart product development is formally documenting product targets to ensure that everyone is working to the same end goal. This simple step is frequently given too little time and attention, particularly when "missed customer expectations" commonly ranks as a top reason for product failure. Why bother developing a product without the right features, or one that suffers from feature bloat and costs too much? Effective requirements management, however, is challenging because different domains have different formats and systems. Fragmented information makes it hard to find the single source of truth, requirements are not kept up to date, and design inertia takes over. The result is that important requirements can be overlooked or products risk being overdesigned.

Fragmented information makes it hard to find the single source of truth, requirements are not kept up to date, and design inertia takes over.

Product managers must consolidate and prioritize needs and demands from multiple sources including customers, market analysis, technical specifications, and regulatory mandates. Consolidated requirements must be prioritized and decomposed to a level of detail that is actionable. Proven best practices for requirements management include version controlling requirements and ensuring they live through the entire product lifecycle, serving as guidelines to design and validate the product. In addition, forward-thinking companies save time and improve quality by intelligently reusing requirements across products and product generations, although blind reuse can result in over-specifying products.

Systems must support centralized, cross-discipline requirements and make them visible across the product development team.

Mature companies have found that documenting requirements centrally increases productivity and improves results. Systems must support centralized, cross-discipline requirements and make them visible across the product development team. They need to be clearly documented and easily maintainable, not tucked away in a forgotten document, so they are kept up to date and changes are communicated. Requirements also need to be integrated with downstream designs, deliverables, and tests through the product lifecycle to allow traceability from product concepts through final validation.

Plan the Product

Once requirements are determined it’s time to design the overall system. This is a critical point in the process because getting the conceptual model right prevents costly rework and reduces late changes. It’s important to get the architecture right up front in order to properly allocate requirements to mechanical, electrical, or software disciplines. Investing the right time and effort here provides huge returns later in product design and development. It’s also critical to validate requirements are addressed from the very first systems architecture.

Getting the conceptual model right prevents costly rework and reduces late changes.

Despite the proven value of getting product architecture right the first time, many companies find it hard to invest time in the early design when deadlines are tight. Others don’t know how to confidently determine the way that mechanical, electrical, and software components will interact, and don’t feel they can realistically predict system

© Tech-Clarity, Inc. 2014

behavior. But nothing is more frustrating than finding a problem too late after it is locked in to the architecture.

Solutions that support systems modeling and simulate systems behavior are critical to help architects optimize systems.

Best practices in system design include systems modeling, conducting tradeoff studies early in the product lifecycle, and simulating system behavior to validate and optimize architecture. These processes allow designers to confidently develop their portion of the design knowing that they will result in a cohesive product. Another best practice is to carefully define and manage interfaces to allow different disciplines to work in parallel with the knowledge that designs will converge into a working system. Solutions that support systems modeling and simulate systems behavior are critical to help architects optimize systems conceptually prior to allocating functions to domains.

Validate Early and Often

Getting products right the first time saves costly and time-consuming rework. It also prevents locking in suboptimal design decisions. Many choices are easy to change early in design because there is more flexibility and windows of opportunity are still open, but that flexibility rapidly diminishes later in the product lifecycle (Figure 2). Simulation helps validate design decisions but has its challenges. Complex mechatronic products have numerous factors to analyze, including vibration, noise, heat, power, performance, EMF, interference, and more. It’s hard enough to interpret individual analysis results and determine how to change designs, but even harder to determine what needs to change holistically.

Complex mechatronic products have numerous factors to analyze, including vibration, noise, heat, power, performance, EMF, interference, and more.

Simulation now allows analysis and validation of a variety of design decisions and their impact on weight, energy consumption, vibration, noise, heat dissipation, and more. Simulation capabilities have matured to allow designers to experience and validate how customers will experience products in their own environment. Mathematical modeling can even simulate real world behavior, for example conducting a virtual drop test before any physical prototypes are built.

Best practices include early simulation at the product level to allow designers to validate system behavior.

© Tech-Clarity, Inc. 2014

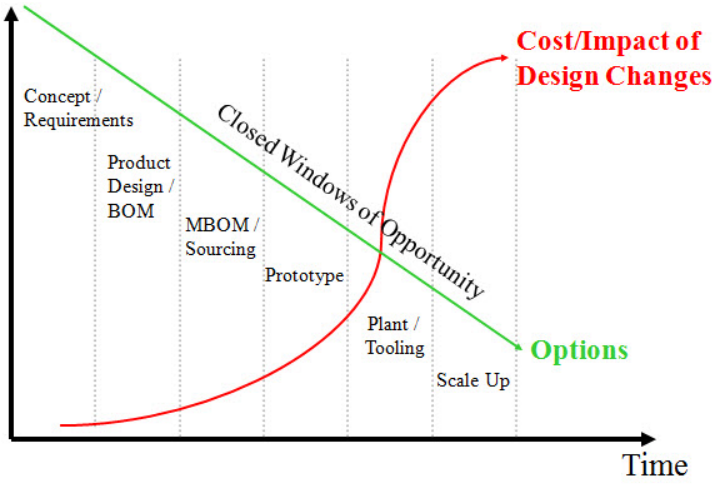

Figure 2: Closing Windows of Opportunity by Time

Best practices include early simulation at the product level to allow designers to validate system behavior. As designs progress, virtual prototypes help ensure fast design convergence. Simulation should be applied at the systems level but also within individual disciplines, for example using electronics simulation and software emulation to ensure requirement are met effectively. Leading companies also leverage analysis to address commercial concerns such as cost and supply availability in addition to technical performance.

Best in class technology offers visualization techniques to help designers and analysts interpret and communicate results.

Enabling technology must provide specialty tools to analyze multiple aspects of designs. Further, they need to determine the interaction between different factors, requiring multiphysics capabilities. Best in class technology offers visualization techniques to help designers and analysts interpret and communicate results. These techniques create their own complexity due to large, complex files, leading forward-thinking companies to use simulation lifecycle management (SLM) tools to control simulation models and analysis results and keep them in context with underlying designs.

Enabling technology should support informal information sharing and interaction in addition to formal processes.

Although simulation is a technical discipline, it's important to structure collaborative design reviews and socially-oriented information sharing to keep people involved in the process. For example, mature companies collaborate to get buy-in and ensure that targets

© Tech-Clarity, Inc. 2014

and plans are understood and achievable. Enabling technology should support informal information sharing and interaction in addition to formal processes.

Design by Discipline and Validate to Interfaces

Once requirements are allocated by discipline and interfaces are defined, different components of the product can be designed in parallel. Mechanical, electrical, and software design have their own methodologies and lifecycles. Maintaining parallel processes frees up software engineers to iterate more rapidly and change much later in the lifecycle while others may follow a more waterfall-oriented approach.

Managing multi-discipline design efforts is challenging because designers in different disciplines use very different toolsets and follow different design lifecycles.

But managing multi-discipline design efforts is challenging because designers in different disciplines use very different toolsets and follow different design lifecycles. For example, software development iterates more rapidly and code remains dynamic far later in the product development cycle than physical parts. This can make engineers feel like there are “too many moving parts” and makes it very hard to manage change across disciplines.

During parallel design efforts, designers from each discipline should focus and communicate on interfaces to ensure future convergence. But parallel does not mean disconnected. Although processes are separate, it’s best practice to share progress and collaborate along the way and continue to validate at the product level throughout the lifecycle. As decisions are made by each discipline, facts can replace assumptions in the virtual prototype to validate the product experience.

Software technology must help by providing visibility and enabling collaboration at both the domain level and across disciplines.

Software technology must help by providing visibility and enabling collaboration at both the domain level and across disciplines. It must also provide centralized project and deliverable management to coordinate design efforts. More mature companies have found that integration between product lifecycle software and design tools helps manage deliverables and enables an effective change management process. The keys to success are centralization and integration. Centralized management also helps designers find and reuse prior designs at the system and domain levels to improve productivity and quality.

© Tech-Clarity, Inc. 2014

Aggressively Manage Change

Although companies should try to avoid late changes, design changes are a fact of life. Change requests can come from different sources as requirements change or as new information is discovered. Companies need to capture issues and bugs, enhancement requests, component changes, and other issues and manage them through a structured process. Effective change management is essential to quality and time-to-market. Change management within a single discipline is difficult. Determining the impact of change across disciplines is much harder, particularly considering the high frequency and late occurrence of changes common in software development.

Companies need to maintain an integrated view to understand the impacts and effectively manage change across disciplines.

Best practices for change management include a thorough impact analysis when a change is proposed or introduced. As A Ten Point Guide for Streamlining Product Development with Systems Engineering says, "*Make sure that change, and more importantly the impact of change, is managed across all levels of the design*." Changes in one discipline frequently impact designs in other disciplines, such as changing a chip might change board layout and heat dissipation, or adding a new software feature might change expected power consumption. Companies need to maintain an integrated view to understand the impacts and effectively manage change across disciplines. Best practices also expand impact analysis beyond the current design to find and proactively correct issues in related products or configurations.

Best practices also expand impact analysis beyond the current design to find and proactively correct issues in related products or configurations.

In order to manage change, companies must execute a highly visible, formal change management process. Enabling technologies help manage the change process and provide broad visibility. They offer centralized access to information and integration with information and tools across domains. Mature companies also use enabling technology to support impact analysis, enable root cause analysis, and provide where-used information to propagate change to the right products and variants.

Mature companies use enabling technology to support impact analysis, enable root cause analysis, and provide where-used information to propagate change to the right products and variants.

© Tech-Clarity, Inc. 2014

Enable an Integrated Lifecycle View

Clearly today's smart products demand smart approaches to enabling technology. Developing Software-Intensive Products also offers best practices related to enabling technology for mechatronic system development. The research finds that leading companies, among other things, *"… have more integrated systems, are more likely to employ systems modeling, leverage integrated design solutions, and take a lifecycle approach to supporting systems-driven product development*." Leading companies are more likely to use formal data / lifecycle management solutions, while "*companies with disconnected point solutions are more likely to experience … negative impacts than those with integrated solutions or a single system*."

Leading companies are more likely to use formal data / lifecycle management solutions.

Most companies, however, do not have solutions that span disciplines and domains. They typically employ point solutions that help with specific tasks but fail to manage the complex network of data and their relationships, including requirements, designs, manufacturing plans, sustainability, simulation results, test plans, test results, and more. These tools are typically internally focused and don't manage the interaction and collaboration required with design partners and the supply chain. Too many companies are trying to support today's integrated products with yesterday's disconnected point solutions.

Too many companies are trying to support today's integrated products with yesterday's disconnected point solutions.

Best practice, then, is to leverage lifecycle management systems for mechatronic product development. While companies will certainly have different design tools by discipline, they need an integrated view of the product development process and an integrated way to manage deliverables. Centralized information and a common data model help promote visibility, traceability, and collaboration across design disciplines. This integrated approach helps designers manage the complex relationships between various elements of design and different variants and configurations. Smarter products demand more integration, transparency, and traceability in order to ensure quality, minimize rework, and get products to market quickly. Today's development platform must support this with an integrated view from requirements to test management, and provide a collaborative environment to get the best out of each design discipline and the supply chain.

Smarter products demand more integration, transparency, and traceability.

© Tech-Clarity, Inc. 2014

Conclusion

Smart products hold significant benefits for customers and for the companies that produce them. The value is compelling and has created a significant growth trend. It has also brought a significant, new level of complexity to product design and development. The complexity of smart products will only increase as they become more connected and the "internet of things" becomes a reality. As the earlier research concludes, "*Developing (mechatronic) products presents a conundrum for manufacturers, enabling strategic business benefits while simultaneously increasing product and product development complexity. The benefits are too important to ignore, so the impacts of complexity must be mitigated*."

High-tech and related companies must raise the maturity of the processes and tools they use to develop smart products.

High-tech and related companies must raise the maturity of the processes and tools they use to develop smart products. This report suggests best practices for requirements management, systems design, simulation, interface management, and change management. There are others as well, including optimizing designs for supply up front, closing the loop on issues from the field back to design, and leveraging designs for downstream product communication. These processes are proven and can provide significant value today.

Companies should implement integrated, lifecycle-oriented solutions to enable a cohesive, transparent product development process across disciplines.

In addition to best practices, companies should implement integrated, lifecycle-oriented solutions to enable a cohesive, transparent product development process across disciplines. These solutions allow companies to implement best practices to take advantage of the compelling benefits of smart products without suffering from costly, negative impacts to quality and time-to-market.

Recommendations

Based on industry experience and research for this report, Tech-Clarity offers the following recommendations:

- Follow proven best practices for developing mechatronic products
- Implement the right tools to enable design, including simulation and systems design tools
- Leverage lifecycle-oriented systems to integrate the various processes, people, and deliverables required to efficiently develop high quality, smart products.

© Tech-Clarity, Inc. 2014

About the Author

Jim Brown is the President of Tech-Clarity, an independent research and consulting firm that specializes in analyzing the business value of software technology and services. Jim has over 20 years of experience in software for the manufacturing industries. He has a broad background including roles in industry, management consulting, the software industry, and research. His experience spans enterprise applications including PLM, ERP, quality management, service lifecycle management, manufacturing, supply chain management, and more. Jim is passionate about improving product innovation, product development, and engineering performance through the use of software technology.

Jim is an experienced researcher, author, and public speaker and enjoys the opportunity to speak at conferences or anywhere he can engage with people with a passion to improve business performance through software technology.

Jim can be reached at jim.brown@tech-clarity.com. You can read additional research, watch Tech-Clarity TV, or join Jim's *Clarity on PLM* blog at www.tech-clarity.com. You can also follow Jim on Twitter at @jim_techclarity, watch him as a "dueling analyst" in the Tech4PD web show, or find Tech-Clarity on Facebook as TechClarity.inc.

© Tech-Clarity, Inc. 2014