

**Tech-Clarity**

*making the value of technology clear*

**Issue in Focus:  
Systems and Software Driven  
Innovation**

***Complexity and Opportunity  
in the Mechatronic Era***



## Table of Contents

Introducing the Issue .....	3
The Mechatronics Imperative .....	3
The Complexity of Mechatronic Products .....	4
Managing Change in Mechatronic Products .....	5
Working Together – The Systems View .....	5
The Opportunity to Innovate with Systems and Software .....	6
Competing through Software Driven Innovation .....	7
Enabling Software Driven Innovation .....	8
PLM and Software Driven Innovation .....	9
The Current Reality of PLM for Systems .....	10
Conclusion .....	11
Recommendations .....	12
About the Author .....	12

## Introducing the Issue

Modern products increasingly incorporate a combination of mechanical, electrical, and software components that allow innovative designers to take advantage of sensors and logic to solve problems and provide capabilities in new ways. This impacts the way companies innovate. As manufacturers develop their next generation of products, they are more likely to turn to electronics and software to make a “smarter” mousetrap as opposed to innovating solely in their mechanical design.

---

***Over the last decade, there has been a steady, fundamental shift towards increased software and electronics in traditionally mechanical products.***

---

Over the last decade, there has been a steady, fundamental shift towards increased software and electronics in traditionally mechanical products. Engineers have added more monitoring and more sophisticated controls, and have increased the amount of product functionality and value delivered through software as compared to mechanics. Examples of this shift range from automobiles, to mobile devices, to “simple” household appliances.

---

***The shift has driven higher levels of product development and engineering complexity.***

---

This changing paradigm requires multiple design disciplines to work together to develop a working system. Systems that rely more heavily on the combination of mechanics, electronics, and software demand more integrated engineering and validation processes. The shift has driven higher levels of product development and engineering complexity starting in the early requirements phase of a product, continuing through design, making a significant impact on product testing and validation, and continuing as a configuration management issue throughout the product lifecycle.

## The Mechatronics Imperative

The mechatronics trend is not likely to reverse itself because smart, mechatronic systems can simply provide better functionality than mechanics alone. Dealing with this complexity is mandatory to innovate and deliver high quality products. Beyond managing the complexity, the trend also creates opportunities. The increased influence of software on product capabilities and performance opens up new frontiers for innovation and competition because software is much more adaptable to change than physical parts, leading to product development, manufacturing, and maintenance cost savings and improved flexibility to rapidly address new market opportunities.

Because of the need to manage complexity and the promise of new innovation benefits, software development is moving from an afterthought to a core component of product innovation. Manufacturers that excel at this new lever of competition will drive higher levels of innovation and reduce cost, without suffering from quality and reliability issues that characterized the early adoption of mechatronic design.

---

***Software development is moving from an afterthought to a core component of product innovation.***

---

## **The Complexity of Mechatronic Products**

The increased reliance on electronics and software is fundamentally changing product development. The most noticeable shift is the increased complexity required to integrate people and designs across multiple disciplines. *Tech-Clarity Insight: Five Dimensions of Product Complexity* lists Mechatronics as the second of five dimensions of product complexity. The report indicates that, “*Modern automobiles may have 50 or more microprocessors managing everything from braking, to engine timing, to entertainment. This combination of mechanical, electrical, and software components, known as ‘mechatronics,’ has made products exponentially more complex.*”

---

***This combination of mechanical, electrical, and software components, known as ‘mechatronics,’ has made products exponentially more complex.***

---

The shift is significant. Many manufacturers of complex products now have as many software engineers as they do mechanical or electrical engineers. This is true even in traditionally mechanically driven industries such as automotive suppliers. Today, an average automobile may rely on 100 million lines of software code. It’s not just in the entertainment system, either. Software is well established in the powertrain and throughout the vehicle as well.

---

***If this complexity isn’t dealt with early in design, it can leave a big integration problem at the end of development.***

---

The complexity will continue to grow as products get smarter and products start to interact with each other as a “system of systems.” If this complexity isn’t dealt with early in design, it can leave a big integration problem at the end of development because the performance of the product is the result of the entire system working in harmony. The complexity can lead to rework and frequently compromises product capability and quality as suboptimal tradeoffs need to be made during late design rework. It can also result in quality issues, time to market challenges, and configuration problems.

## Managing Change in Mechatronic Products

Initial mechatronic design is complex, but change management brings on even greater challenges. Engineering change management now needs to be coordinated across disciplines. For example changing the hardware without adjusting the control software can result in unpredictable behavior. With platform-based designs, change management is even more complicated. Change may need to be applied selectively based on the product variant or even the stage of the product in its lifecycle. This leads to divergent paths for requirements that must be understood and managed over time.

Another complicating factor in managing change is the disparity in the change cycle between hardware and software components. Mechanical designs are highly flexible in the early parts of the design process, but as designs mature during product development physical elements get locked down due to constraints from production equipment, tooling, and component purchases. Because of this, the cost of change increases dramatically later in the design cycle. Software, on the other hand, remains more flexible and can be changed more easily and less expensively.

---

***This disparity in the pace and volume of change between software and hardware ... breaks inefficient change management processes developed to handle primarily mechanical products.***

---

This is true even after the release of the product. Today's software development approaches such as "agile" processes contribute to even more rapid change in software. This disparity in the pace and volume of change between software and hardware results in opportunity, as we will discuss later, but also stretches or breaks inefficient change management processes developed to handle primarily mechanical products.

## Working Together – The Systems View

The core of the mechatronics challenge is that different design departments typically work independently. In many companies, they don't share designs or interact on a regular basis. This is a big challenge as companies try to view the product as a whole across design disciplines with different perspectives and priorities.

---

***Product developers must understand and manage all elements of the systems, from requirements to tests.***

---

Companies need to take a systems view up front. Systems modeling techniques can provide a framework to understand system behavior and the underlying requirements. Beyond these models, product developers must understand and manage all elements of the systems, from requirements to tests. This is the only way to maintain quality and

prevent inefficiency in design and manufacturing. It will also help reduce the significant, manual compliance effort resulting from different disciplines acting in silos with no traceability at the systems level.

As companies adopt more modular and platform-based designs, the systems view will become even more important. As product variants that leverage the same hardware become more prevalent, designing and testing these different configurations will be much more challenging. For example, with multiple variants, sometimes only a portion of them should be updated with the new software and not all requirements are applicable to each configuration.

## **The Opportunity to Innovate with Systems and Software**

So why do companies put up with this complexity? Part of the reason is that they simply can't compete without the enhanced products that electronics and software can deliver. For example, simple mechanical products will not be able to compete with a braking system that automatically adapts to road conditions or a wing that automatically adapts to flight conditions. For that reason, mechatronics will continue to advance and the resulting complexity must be managed. Beyond the obvious improvements in product capability, taking a systems approach holds other strategic advantages, including:

- Improved ability to tailor products to customers or specific market needs
- Increased reuse of hardware and mechanics
- More agile corrections and improvements to products in the field
- Reduced product cost
- Lower product development cost

---

***Product performance and behavior can be much more adaptable when software plays a bigger role in determining product functionality.***

---

Product performance and behavior can be much more adaptable when software plays a bigger role in determining product functionality. This opens up significant benefits. For example, a company can develop a single physical product that can be adapted for a particular customer, market, or usage. A range of mobile devices from basic to advanced may include the same chipset, but have advanced features enabled via software only for the more expensive models. While at first blush it may seem too expensive to include hardware that will not be used in the lower end model, the ability to manufacture, test, and inventory a single hardware platform provides significant cost savings. Developing and reusing fewer physical platforms and differentiating them into different products via software drives up reuse and leads to economies of scale by increasing manufacturing and sourcing volumes.

The ability to adapt and configure products via software supports platform-based design where the same base product platform can be leveraged for multiple products, multiple geographies, and more. An example is a mobile device that works for different carriers, networks, and provides different levels of functionality from basic to advanced. The basic design can be developed and leveraged across the different use cases, driving lower product development costs and leading to cost savings from parts reuse.

### **Competing through Software Driven Innovation**

Software based innovation also enables the ability to add new features and capabilities to products that were not in the original design. The capability might be a hot fix for noise cancellation to address an unanticipated complaint, product issue, or new “must have” feature to invigorate sales or improve customer value. Not only can the capability be added for new products, it can change the functionality of a product that is already in the customers’ hands. This “hot fix” could prevent a recall that would traditionally be required to replace a mechanical part. The flexibility of software-driven capabilities also allows features that couldn’t be predicted to be included late in the design cycle or in production without disrupting manufacturing processes, supply chain plans, or product quality.

---

***The flexibility of software-driven capabilities also allows features that couldn’t be predicted to be included late in the design cycle or in production.***

---

While today this kind of product innovation may be expected for a smart device such as a phone, the same opportunity exists for other mechatronic products. Want to upgrade the handling of your agricultural equipment? Or the volume available from your amplifier? Why not download a new piece of software that transforms the product into something better or more tailored to your needs? In the near future, products will become more like today’s smartphones. By adding a new “app” to these devices, it can take on a wide variety of new functions, in effect being enhanced virtually from the inside with updated software.

---

***The opportunity will help companies compete in new ways – from saving cost through platform reuse to enhancing innovation and customer experience.***

---

Of course not every product can or should be modified in the field by users. For example, aftermarket sources will update an automobile’s software to enhance performance, also known as “chipping” a car. Auto manufacturers are not yet ready to support this type of aftermarket customization, and will therefore void the warranty. But the consumer demand is evident. Even for these products, software-driven innovation can allow changes later in the product design process with less disruption because there is no need to modify actual parts– not to mention tooling, molds, etc. On the other hand, software

can be changed more readily. The resulting opportunities are significant. While many companies will act based on the need to manage complexity, the opportunity will help companies compete in new ways – from saving cost through platform reuse to enhancing innovation and customer experience.

## Enabling Software Driven Innovation

Whether a company's strategy is to simply manage the complexity of mechatronic products or to take advantage of the opportunities they offer, today's engineering processes and systems must evolve to support a systems view. One of the key issues is the ability to view the product in a holistic way. One of the best tools systems engineers can employ is the "V model." The concept is to start and end with requirements that span different design disciplines and view the product system as a whole. These requirements are mapped to corresponding tests and validation procedures to ensure the system design will function as planned. As described in Wikipedia (Figure 1), "*The left side of the "V" represents the decomposition of requirements, and creation of system specifications. The right side of the VEE represents integration of parts and their verification.*" Others have proposed standards that expand this concept into a "triple V" to specifically address systems, software, and hardware.

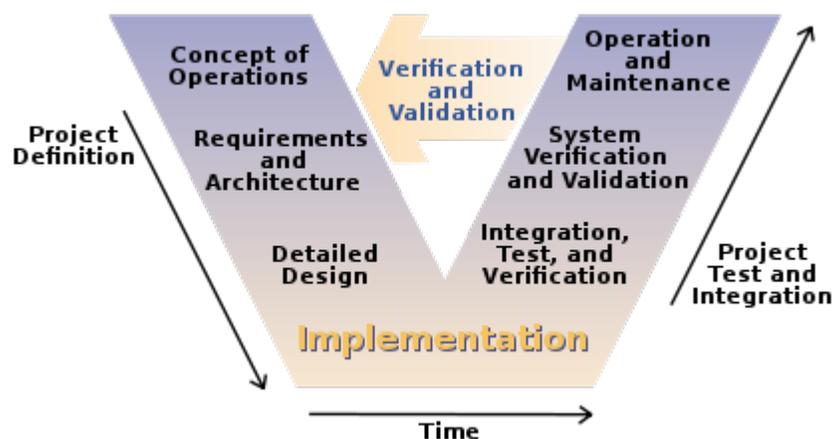


Figure 1: V Model from Wikipedia

---

***There is currently no single tool today that allows companies to see the big picture and the details together.***

---

There have been a number of systems design tools developed to support a systems view, but there is currently no single tool today that allows companies to see the big picture and the details together. The tools either provide a holistic view at the high level or dive into

details for individual design disciplines. The tools developed for mechanical, electrical, and software designs have all been developed to manage their respective functions independently, and are very good at what they do. Each of these disciplines has their own tools and their own data management:

- **Mechanical** - Product data management (PDM) systems are built to store CAD files and other design data. They understand complex configurations and relationships between CAD files.
- **Electrical** – Electronic design automation (EDA) systems help design electronics. They frequently don't have their own data management so electrical designs end up in another PDM system that isn't intimate with the design details.
- **Software** – Application Lifecycle Management (ALM) solutions help software engineers manage their development artifacts from requirements and design specs to code and test artifacts. These tools manage the relationships between these deliverables and manage change throughout the software development cycle.

---

*The ideal solution would provide transparency, traceability, and understand the complex relationships between all of the deliverables.*

---

These data management solutions do more than manage files. They understand the underlying design. Although each of the disciplines has good tools, there is no system that manages all of the design disciplines equally. The ideal solution would provide transparency, traceability, and understand the complex relationships between all of the deliverables. But creating a common PDM that covers all of these elements is far from trivial, so independent PDM / design management systems are today's best bet.

## PLM and Software Driven Innovation

Although individual data management solutions are the most feasible choice, companies need the big picture. They must be able to see how requirements are met and verified whether they are addressed by hardware, electronics, software, or a combination. They also need to be able to trace those requirements and how they are fulfilled throughout the product lifecycle. Using disparate PDM systems brings up the following questions:

- What system manages the overall product lifecycle?
- Where is the product development project managed?
- What drives product development and engineering processes?
- In what system do engineers collaborate?
- Where is the system design?
- How are requirements shared?
- How are test and validation traced back to the requirements?

---

***Overarching information including requirements, systems models, and associated test cases are not suited to a fragmented environment.***

---

This is where a comprehensive Product Lifecycle Management (PLM) system can add value. Ideally this is a unified system that handles these systems-level functions in an integrated way. While some elements can stand alone in separate tools by discipline, overarching information including requirements, systems models, and associated test cases are not suited to a fragmented environment. These elements should be managed in a cohesive PLM environment. Some elements, such as change orders, will need to exist at the systems level to manage changes to requirements and other system level data, but also at the PDM layer to provide an understanding of the underlying changes to product configurations and related data.

### **The Current Reality of PLM for Systems**

Today, there is no solution that provides the detailed design management required by each of the disciplines and an overarching PLM layer. Current PLM and ALM solutions do not provide the full picture. Neither of these systems holds a central view of all of the requirements and can trace them through the designs that achieve them. Therefore, there is no way to implement the V-model in today's PLM systems and cover the validation and verification across design disciplines. Short of developing a systems-centric PLM solution, companies will have to build tight integration between ALM and PLM or develop a "layer" on top of these solutions.

---

***In the long term, manufacturers that want to manage the complexity of mechatronic products should look to PLM vendors with the vision to pull a systems-based innovation solution together with a systems level data model that ties into each of the individual design disciplines.***

---

Development of a ubiquitous PLM layer to pull all of the disciplines together requires work on the part of the manufacturer. These companies can leverage existing solutions by discipline and integrate them in a composite application approach (Figure 2). Service oriented architectures (SOA) and application program interfaces (APIs) can help the integration processes. Alternatively, they could integrate ALM and PLM. The combined solution will require choosing one of the solutions to serve as the systems PLM layer and manage the full process. In the long term, manufacturers that want to manage the complexity of mechatronic products should look to PLM vendors with the vision to pull a systems-based innovation solution together with a systems level data model that ties into each of the individual design disciplines.

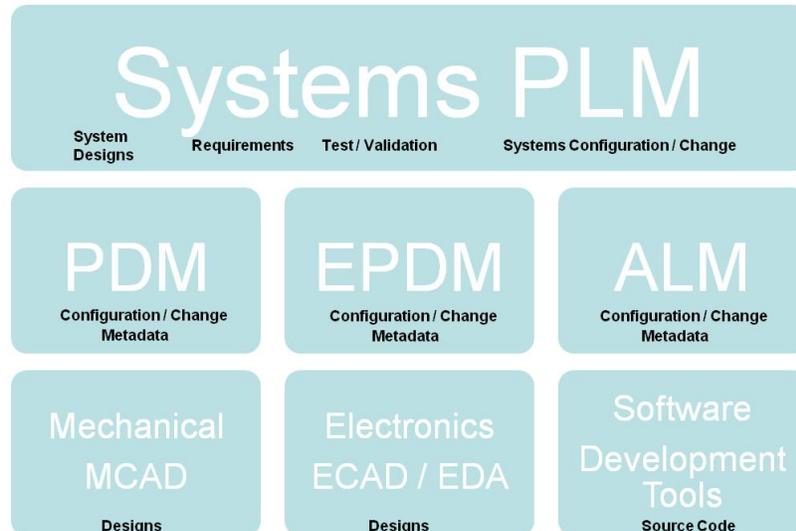


Figure 2: Tech-Clarity Systems PLM Framework

## Conclusion

The trend towards mechatronics will continue because the benefits are simply too compelling to ignore. Software will play an increasing role in determining product behavior and the value it delivers to customers. Manufacturers across industries must find a way to manage product design and change management complexity or suffer from issues including delayed new product introduction, poor quality, and costly recalls.

---

***Software driven innovation is the new frontier of competition, helping companies drive more rapid innovation and creating opportunity due to the increased flexibility and agility of software-based products.***

---

Leading companies will look to take advantage of software driven innovation 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. They will go beyond managing complexity to look for new opportunities. Software driven innovation is the new frontier of competition, helping companies drive more rapid innovation and creating opportunity due to the increased flexibility and agility of software-based products.

Systems-level PLM solutions promise to unify the development of the disparate design disciplines in mechatronic products. Today, companies have to build this PLM layer on their own or enhance and modify a combination of PLM and ALM solutions to provide the full systems view from requirements through validation. In time, systems-level PLM

solutions will undoubtedly be available from software vendors, but today the integrated solution requires work on the part of the manufacturer.

## Recommendations

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

- Manage mechatronic complexity to improve quality and reduce design cycle times
- Leverage software to more rapidly and cost-effectively innovate and add capabilities to products and product platforms, even late in the design cycle
- Take advantage of the new competitive differentiators enabled by software based innovation
- Enable designers from each discipline within the entire system to leverage the appropriate design and data management tools
- Provide a PLM layer to manage systems-level processes such as requirements, test cases, and change management in an integrated way, or integrate ALM and PLM and choose one as the systems-level PLM solution
- Over time, look for the PLM layer and integrated data management and design tools that support the new paradigm of systems and software based innovation

## About the Author

Jim Brown is the President of Tech-Clarity, an independent research and consulting firm that specializes in analyzing the true business value of software technology and services. Jim has over 20 years of experience in software for the manufacturing industries, with a broad background including roles in industry, management consulting, the software industry, and research. He has worked as a manufacturing engineer as well as a software developer. His experience spans enterprise applications including PLM, PDM, ERP, quality management, service, manufacturing, and others. Jim is passionate about improving product innovation, product development, and engineering performance through the use of software technology and social computing techniques.

Jim is an experienced researcher, author, and public speaker and enjoys the opportunity to speak at conferences or anywhere that he can engage with people that are passionate about improving business performance through software technology.

Jim can be reached at [jim.brown@tech-clarity.com](mailto:jim.brown@tech-clarity.com), you can find him on Twitter at [@jim\\_techclarity](https://twitter.com/jim_techclarity), or you can read his blog at [www.tech-clarity.com/clarityonplm](http://www.tech-clarity.com/clarityonplm).