SIEMENS

Ingenuity for life

Siemens Digital Industries Software

Getting started with systems engineering

Key capabilities and benefits of systems engineering

Systems engineering is an interdisciplinary field of engineering focused on providing manufacturers with a system-level view of the product that enables them to account for the related lifecycle processes from business, engineering, manufacturing, service and end-of-life perspectives. This system-level understanding helps reduce costs, mitigate risk and improve product quality by allowing organizations to synthesize, analyze and optimize the interdependent product relationships. This white paper identifies the capabilities and best practices that a systems engineering solution and process approach should provide to facilitate those business benefits.

Contents

Executive summary

To address today's product complexity, many industries are adopting a systems driven approach to product development. A cornerstone of this approach is the intelligent integration of systems engineering and requirements management.

Systems engineering and requirements management will help you address today's challenges by understanding a product in its entirety as well as the processes used to plan, develop, manufacture and sustain it. By defining, monitoring and measuring the relationships between these processes, you can systematically:

- Avoid costs and ensure compliance
- Accelerate development and mitigate risk
- Reduce rework and optimize system performance
- Improve quality and eliminate integration problems

All of which increase the chance of achieving marketplace success.

There are multiple systems engineering methodologies and solution implementations. However, to deliver these advantages, you need an intelligently integrated approach to systems engineering that supports your business-specific needs. It must address cross-domain dependencies (mechanical, electronic, software and wire harness) as well as manufacturing processes.

At its foundation, a system engineering approach must be able to provide best practice solutions for:

- Requirements driven product development
- Model-based systems definition
- System/subsystem-level analysis and verification
- A standardize enterprise-wide data/interface library

In addition, the data and processes must be tied to core PLM capabilities such as schedule management, product configuration management, cost management and issue and change management.

Choosing the right systems engineering solution will enable you to remain competitive and profitable while creating the best products possible.

Business challenges

Each new generation of products is more sophisticated and complex than the previous, so you face substantial challenges as you strive to remain competitive and profitable while creating the best products in increasingly shorter timeframes.

To meet the demands of today's global marketplace, design and manufacturing centers are scattered across the globe with exponential growth in the number and location of external suppliers. Next, the growth in software-driven electronics creates the need for more efficient and effective sub-system integration and better communication between all engineering and cross-functional disciplines. And finally, manufacturers must manage compliance to customer needs or contractual commitments, as well as a myriad of governmental mandates, environmental standards, industry regulations and corporate rules and initiatives.

Building complex products while addressing these challenges has raised the stakes for achieving product success. It is safe to say that the more complex a product, the higher the risk associated with planning, developing, manufacturing, marketing, selling and sustaining that product in a highly competitive global marketplace.

- If a product fails to meet customer expectations, its marketplace acceptance will be dubious at best
- If a product fails to conform to established regulatory requirements, product makers face the potential of heavy non-compliance penalties
- If concept design is not coordinated with the downstream implementation then start-overs, redesign and rework can result in a runaway project
- If system/sub-system interfaces are not documented and managed, then design reuse is severely limited and the discovery of late stage integration errors will occur
- If implementation or engineering changes are not properly aligned with requirements, a program's strategic objectives can be lost
- If bad design decisions are not caught upfront or if their impact is not understood, rampant redesign can result in unnecessary development or warranty costs

Systems engineering capabilities enable you to assess the effectiveness of your product initiatives from multiple business and product perspectives. Seamless access and traceability to intelligently integrated product information – requirements, system definition, modeling tools and interface specifications – enables everyone across the value chain to make better, more informed decisions.

Best-practice solutions

The goal of systems engineering is to enable every stakeholder in a product or program to recognize that their decisions can impact everyone else who participates in the product lifecycle. Equally important, systems engineering solutions facilitate impact analysis and trade-off decisions. The result is optimized products that balance the effect of multiple business and product considerations, including market appeal, product cost, performance, manufacturability, safety, disposability, regulatory compliance, usability, maintainability and total quality.

A systems engineering approach to product development enables companies to address a variety of fundamental business needs.

Business needs addressed by a systems engineering solution

Fundamental business need	How Systems Engineering addresses those needs
Avoid costs and ensure compliance	Avoiding costs and ensuring compliance liberates requirements from standalone applications by capturing, managing and communicating them throughout the enterprise. Requirements are allocated to functional requirements that are segmented by design domains (software, electronic, electrical and mechanical design), or cross-functional areas (project management, finance, manufacturing, service, etc.) and distributed across the entire value chain and product lifecycle. A critical process in any program, product or project is the final verification that requirements have been met. By linking requirements to budgets, schedule, product configurations, physical design, test cases and manufacturing processes you can: • Perform verification checks to validate and identify requirements compliance as well as any non-compliant or requirement “exception” status • Generate traceability reports required for proof of compliance to project plans, contracts or potential standards investigations ATK develops and manufactures armament, mission and space systems, with the goal to ensure that their customers accomplish their objectives – whether they involve a military operation, a satellite launch or a technological breakthrough. Systems engineering and requirements management help ATK meet its goals. “We can go all the way from a customer’s set of requirements, their statement of work, to the tasks in a schedule and we can trace all the way back to that statement of work and know why we’re doing that task. We’re doing that task to comply with this portion of the ATK’s PLM strategy which starts at the onset of a project.” *David Riemer, VP Science and Engineering*

Fundamental business need	How Systems Engineering addresses those needs
Accelerate development and mitigate risk	You can accelerate development and mitigate risk by allowing development teams to define and model the system, product or program from the "top down" and integrate its subsystems and components "upfront" so that every discipline can fully understand the impact of their product decisions. Early establishment of this system-level view provides everyone involved with a common understanding of how all the subsystems and processes fit together to achieve the final objectives. Developers can segment the product/system into different product configurations with different assemblies and different versions that represent all of the variations that arise, including product variants, options and add-ons that can be developed to extend the profitability of the program. This definition provides total program visibility so that systems engineers can define the problem statement. Decision makers can leverage systems-level metrics to make optimized cross-domain tradeoff decisions and informed impact assessments when changes are proposed. At Strukton Civiel the aim of the systems engineering project was to improve risk analysis and to provide the organization with more and better information, so that project complexity can be managed more effectively. The systems engineers turn the divergent branches of the project into 1) what should be built (object tree), 2) the requirements that have been established (requirement tree), 3) the activities the requirements will require (activity tree), 4) the schedule and finally 5) the risk tree. "What we have done with the systems engineering project is rise to a higher level so that we get a genuine picture of all the systems, tasks, responsibilities, schedules and risks." *Frank Hoekemeijer, Manager Strategy and Development Department*
Reduce rework and optimize system performance	Reducing work and optimizing system performance enables development teams to accurately investigate a wider range of design alternatives, assess trade-off options and validate solutions against customer needs, contractual requirements and safety regulations, such as ISO 26262. Rather than relying on physical prototypes, which are costly and in many cases do not help you identify "variability" or configuration issues, developers can validate system/sub-system interactions across domains using modeling, simulation and analysis tools to analyze and optimize the system or product across all possible configurations earlier in the process. Because the system is defined and modeled development teams can better assess the impact a requirement, design or manufacturing change will have on the system.

Fundamental business need	How Systems Engineering addresses those needs
	Budgets for weight, cost, power, time are defined, managed, monitored to provide an easily understood view of the value and sensitivity of the factors involved in any trade-offs that may be necessary.
	At Cofely Energy & Infra, the scope and cost of engineering projects requires a clear and detailed understanding of how and how well a technical solution functions to satisfy a specification. This level of understanding requires a systems engineering process, in which requirements are registered and translated into functions and systems that must be validated – from quotation to project delivery, maintenance and ongoing operation. "We use the systems engineering capabilities predominantly for requirements management, systems definition, analysis and design management. The extra effort in the engineering and analysis phase saves time later in the process. Impact analyses show the largest benefit. When the customer changes requirements, we now have the ability to say where to expect the impact of these changes. This removes an important part of the risk for both our clients and us." *Rico Verhage, Systems Engineer and Consultant*
Improve quality and eliminate integration problems	A managed, standardized interface and message library enables development teams to capture and share a common definition of all the interface, signals and messages. Once these interface items are associated to subsystem interfaces, design teams and suppliers can accurately and efficiently determine what message and format their sub-systems will generate (publish) and their sub-systems will receive (subscribe). Validating all the publishers and subscribers for all the interfaces and messages ensures proper sub-systems communication, function and performance.
	At Boeing models were developed that allowed all air vehicle systems and interfaces to be modeled to the card and software application level. As a result, all airplane systems are represented in the 787 FI model. Interface activities have identified hundreds of thousands of discrepancies, which are getting resolved prior to lab and flight testing. "In terms of an Interface Control Document (ICD) database tool, we have collectively accomplished something that other companies have attempted and not achieved; we have modeled the logical interfaces of virtually the entire airplane and created a database that software design tools can assimilate with minimal human intervention. This required a lot of hard work by the entire team, but it will pay off by enabling efficient data management throughout the life of the 787 Program." *Mike Sinnett, Boeing, Chief Systems Engineer, 787 Program*

The goal is to enable every stakeholder in any system, product or program to recognize that their decisions impact everyone else participating in the product lifecycle. Essentially, a systems engineering solution must facilitate impact analysis and trade-off decisions.

It must also enable product makers to assess the effectiveness of their product initiatives from multiple engineering and business perspectives, allowing companies to understand all of the related lifecycle processes.

The end result is optimized products that balance the effect of business and engineering considerations, including market appeal, product cost, performance, manufacturability, safety, disposability, regulatory compliance, usability, maintainability and total quality.

To accomplish all of this, a systems engineering solutions must be able to support best-practice solutions for:

Requirements driven product development, which enables product teams to define, capture, manage and leverage product requirements that originate in multiple sources and make them available for enterprise access/usage through a single managed source of PLM information.

Model-based systems definition, which enables product teams to use graphical building blocks to create hierarchical structures that represent a product and its processes from multiple system-level perspectives. Linking these interrelate views together provides a whole product perspective that project management, development, manufacturing and service teams can leverage for cross-domain and cross-discipline optimization.

System/subsystem-level analysis and verification, which enables product teams to analyze and validate sub-system interaction using a wide variety of commercially available or internally developed modeling, simulation and analysis tools earlier in the process. The models must be managed and linked to specific versions or configurations so designers can find and re-use the correct versions of these models.

Budgets constraints, which include quantifiable targets like cost, weight, power, time and other critical measurements, must be tracked and managed so you can establish goals and track progress towards meeting those goals. They also enable product teams to understand the interdependent relationships that exist between different aspects of the product and how these relationships impact one another.

Standardize enterprise-wide interface library, which enables product teams to define, manage and validate the interfaces, messages and signals being shared and communicated between sub-systems. Since the number of interfaces and messages can range from several dozen to several thousand, the ability to identify and validate all the publishers and subscribers is crucial. If not done accurately, this can lead to over design, additional cost, increased weight, reduced performance or worse...system failure.

A systems engineering solution's capabilities

While the general requirements discussed in the preceding section are helpful in conceptualizing the best-practice solutions needed by a PDM system, the following table describes detailed capabilities required by these solutions.

Best practice solution	Required capabilities
Requirements driven product development	Requirement driven development helps you capture requirements using familiar tools such as Microsoft Office® applications – the most commonly used office automation and requirement capture tools – and the ones most companies are already familiar with. They must be maintained in a single, secure source so the requirements can be easily shared across all stakeholders. Design teams can import them from existing documents and export them in pre-defined formats to Word or Excel. Requirements can be defined as in a variety of formats including text, equations, images, models, videos and more. Requirements must be treated as design objects that can participate in project plans, structured workflows, product configurations, and issue and change management processes. As they evolve, or are re-used across multiple systems, products or programs they must be versioned and maintained so they can be traceable back to their sources. You need to link and trace requirements to the various products, configurations, subsystems and design domains (mechanical, electrical, electronics and software) as well as across the other functional areas of the company including test, manufacturing, service and maintenance. When requirements are linked to physical implementation and test, it easy to determine whether there is test coverage for all items. Requirements effectivity track requirements as they are allocated, re-use or evolve based on: • Release: Requirement that apply to a release • Date: Requirements that apply as of a specific date, within a date range or after a specified date • Serial Number: Requirements applied to a serial number, end item (with revision rules) or serial number range (from-to) • Version/range of versions: Requirements that apply to revision "B" of a requirement (or set of requirements), a set of requirements associated with revision "B" of parts • Requirement variant: Requirements that only apply to certain variant conditions

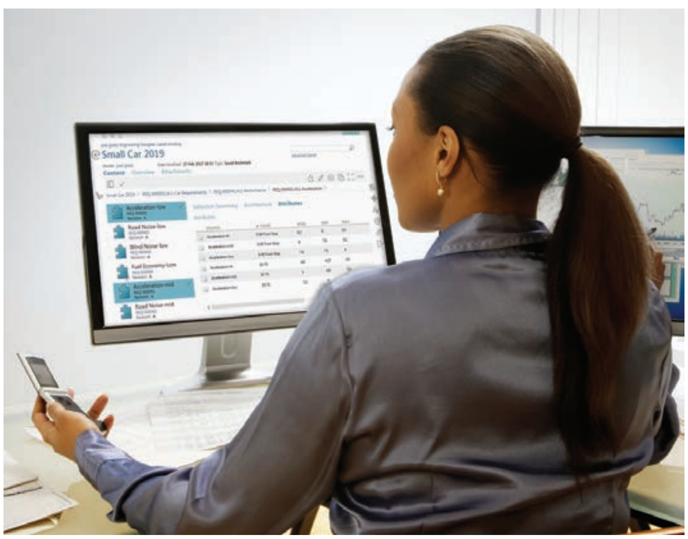
Best practice solution	Required capabilities
	To help downstream consumers understand the thought process for a requirement or assess the impact of requirements change, a decision support "authoring" application is needed. This application can capture requirements background information such as why is a requirement included, what tradeoffs were considered, were changes made to the requirement, and more. Link and note indicators should identify which requirements are linked or have additional information or issues associated with them. You need to verify requirements implementation and generate requirements compliance reports. As the number of trace links increases, a graphical relationship browser, trace matrix or other features are needed to see and verify all the relationships that go between the different structures.
Model-based systems definition	Using best-in-class integrated architecture modeling tools, design teams create functional block diagrams, system block diagrams, UML diagrams, IDEF diagrams, network diagrams, flowcharts and more. The diagramming process should automatically populate the design object's definition in a single PLM database. To ensure compliance, these objects must be linked to the associated requirement as well as the downstream implementation. Since bad things typically happen at the interfaces, as each of the subsystems are being defined the solution must capture detailed information about the interfaces and messages being shared through the connections and ports (detailed definition of each function's inputs and outputs). The information will provide a common understanding across the supply chain of what messages each subsystem will communicate, their format, description and more. Decision support notes to capture issues, actions, questions/answers or trade-off rationale should be attached to design objects. These notes help capture and communicate information required by downstream stakeholders that is often lost. This information will be especially useful during initial implementation or when assessing the impact of a change.
System/subsystem-level analysis and verification	To accurately validate a broad range of design alternatives, a systems engineering solution must support integration with your home-grown and commercially available modeling and simulation tools. One of the most popular commercial tools for modeling and multi-domain simulation is MathWorks®' MATLAB® and Simulink®. In addition, support for other modeling tool associated with cost, behavioral, functional, logical, SysML/UML, FMEA/reliability, co-simulation, vibration, manufacturing, human and more are required. As models evolve, or are referenced in multiple products or configurations, they must be managed in a single secure source along with their dependencies and relationships. This helps designers find and maximize the reuse of proven designs and models. This capability is especially critical when design teams need to validate the design or assess the impact a change will have on the design or the models it uses.

Best practice solution	Required capabilities
	Budget management capabilities enable organizations to establish quantifiable goals or performance targets such as cost, weight, power, time and other critical measurements. The budget values are associated with a product structure or configuration. The values are rolled up periodically to monitor how well they are tracking to the goal. Managing budget information provides everyone an easily understood view of the value and the sensitivity of factors involved in trade-off decisions, as well as helping assess the potential impact of a change.
Standardize enterprise-wide interface library	Information captured as part of the product structure interfaces and messages should be organized and managed in a secure central library so that it can be re-used on a single project or leveraged across multiple programs and platforms. This provides everyone in the supply chain with the ability to find, reference, validate and re-use proven interface information. Here again, managing the data in a standardized enterprise-wide library will help in assessing the potential impact of a change. You can generate a matrix of publishers and subscribers. Once interface items are associated to subsystem interfaces, design teams and suppliers can accurately and efficiently determine what message and format their subsystems will generate (publish) and their sub-systems will receive (subscribe). Design teams use the information during implementation to better determine the size, location, routing pattern and terminal points for a wire harness or communication networks, or whether additional gateways are required. You can validate all the publishers and subscribers for all the interfaces and messages. This ensures proper sub-systems communication and performance. Without this visibility, people rely on prototypes and testing to find problems versus a design and validation approach. Lastly, interfaces, signals and messages should be subject to the same set of configuration rules, workflow and change management process as other objects.

Conclusion

Siemens Digital Industries Software has extensive experience helping companies like yours achieve their systems engineering goals. With more than 5 million users worldwide, Siemens Digital Industries Software's Teamcenter® software suite delivers a proven systems engineering solution that enables you to start leveraging and controlling your product knowledge and processes through a single secure source. From there, Teamcenter can be expanded with the following additional Systems Driven Product Development solutions to meet your complex product development needs and strategic initiatives.

Whether your focus is starting a Systems Engineering approach or implementing a systems driven product development initiative, contact us today – we can help you achieve your business goals.



Extend systems engineering to systems driven product development

Systems engineering	Model-based systems engineering
Requirements driven product development	Requirements driven product development
Model-based systems definition	Model-based systems definition
System/subsystem-level analysis and verification	System/subsystem-level analysis and verification
Standardize enterprise wide interface library	Standardize enterprise-wide interface library
	Project/schedule management
	Engineering process/ workflow
	Mechatronics
	Visualization
	BOM management
	Manufacturing

Siemens Digital Industries Software

Headquarters
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 972 987 3000

Americas
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 314 264 8499

Europe
Stephenson House
Sir William Siemens Square
Frimley, Camberley
Surrey, GU16 8QD
+44 (0) 1276 413200

Asia-Pacific
Unit 901-902, 9/F
Tower B, Manulife Financial Centre
223-231 Wai Yip Street, Kwun Tong
Kowloon, Hong Kong
+852 2230 3333

About Siemens Digital Industries Software

Siemens Digital Industries Software is driving transformation to enable a digital enterprise where engineering, manufacturing and electronics design meet tomorrow. Our solutions help companies of all sizes create and leverage digital twins that provide organizations with new insights, opportunities and levels of automation to drive innovation. For more information on Siemens Digital Industries Software products and services, visit siemens.com/software or follow us on LinkedIn, Twitter, Facebook and Instagram. Siemens Digital Industries Software – Where today meets tomorrow.

siemens.com/software

32661-C8 2/20 H