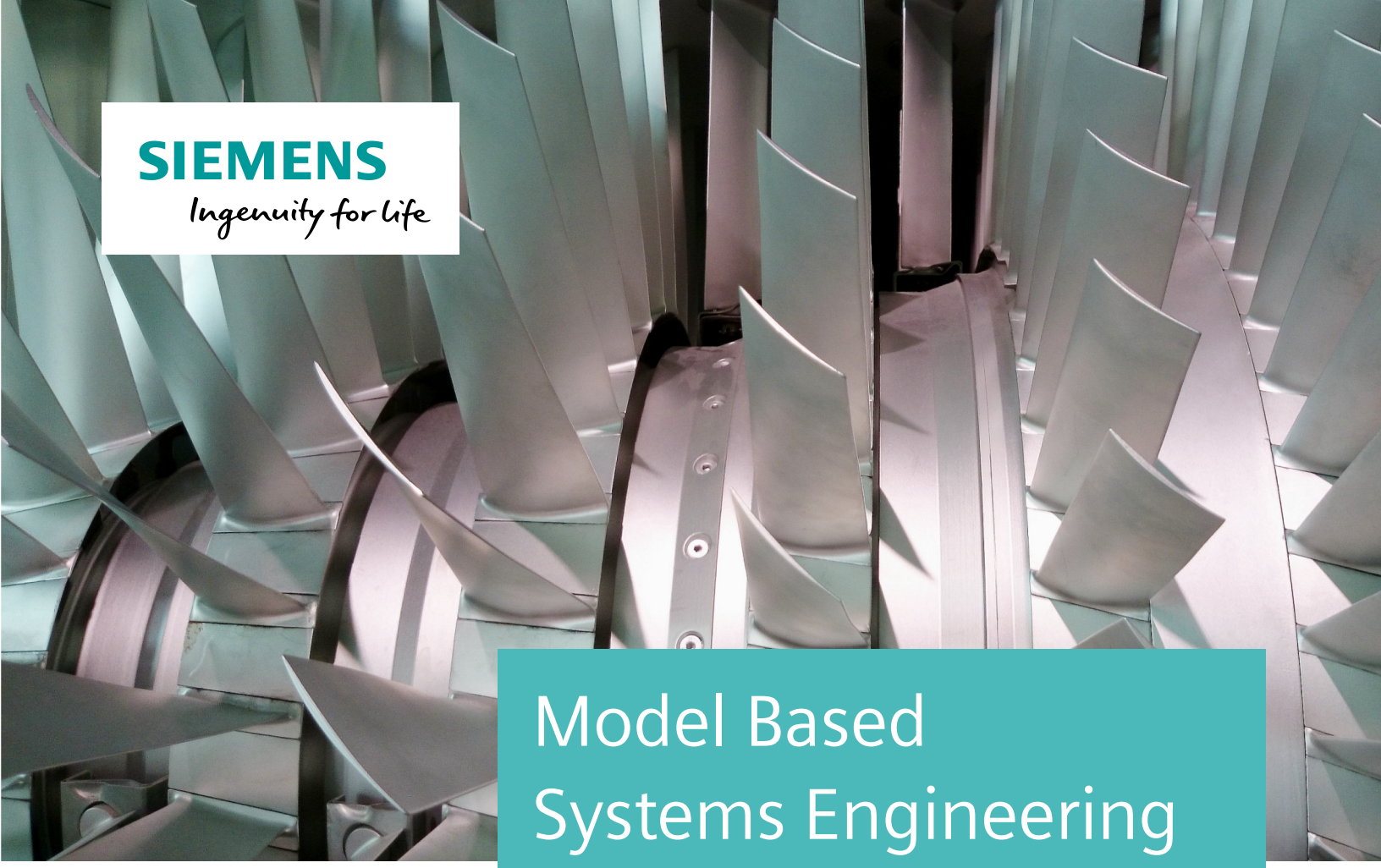


SIEMENS
Ingenuity for life

Model Based Systems Engineering

Orchestrate the technical content and scope of your program throughout the entire lifecycle

Like an aircraft, the aerospace systems engineering process is extraordinarily complex and only becomes more so with every new innovation and innovation comes frequently in this industry. However, organizations face a paradox; the demands that innovation must come faster and cheaper. Product integration issues are often the glaring culprit in slowing down the process and driving up costs, sending companies into a seemingly never-ending loop of testing and fixing.

Aerospace and defense products are becoming increasingly complex and highly integrated

More innovation means more complexity

As technological innovation evolves and the aerospace industry becomes more complex, companies have realized that aircraft requirements must grow in both total number and rigor if they are to deliver the best and safest possible product. Designing, validating, certifying and delivering aircraft to market inevitably becomes more difficult. In view of this challenge, leaders at aerospace companies are placing more importance on the systems engineering function to identify and manage the requirements, interactions and interfaces that accompany this increased level of complexity.

Addressing the complex requirements of today

Leaders have built strong teams of systems engineers, created processes, and given their teams modeling technology and tools to translate customer needs into product requirements and design product architectures. By checking these boxes, they believe that their teams are equipped for success.

The systems engineers charged with developing these product architectures have very limited control over systems and their requirements. Their goal is to learn how these toolsets work and how to use them to quickly deliver what they are responsible for –architecture design and requirements management – to the best of their ability. Program leadership can see process challenges occurring throughout the development lifecycle, but the systems engineers below them typically only articulate issues with their own toolsets and their own functional areas, rather than the impact these issues have on the bigger picture of development.

Though the process of identifying requirements and design architectures is increasingly fraught with unpredictability due to product complexity, companies often feel they cannot afford to wait for the concept design to be perfect, lest they miss deadlines and lose their opportunity in a competitive market often populated with small and nimble players. Their only option is to begin later design phases in parallel. As a result of this pressure to keep moving forward, an engineer might knowingly hand off a suboptimal concept design on schedule with the belief that the design process is still in its early phase. He or she may think that if there is a design flaw, it will be identified and fixed in testing.

Further downstream, there is a growing frustration regarding product integration issues. At this point in development, thousands of internal individuals and third-party suppliers may be working on a program and living these integration challenges as they appear. Leaders do not have time, or insight, to figure out where the integration issues originated, so they commonly attribute them to existing process challenges such as a bad interface drawing or design document iteration. Thus they solve the problem by adding more people or working more hours. Organizations today are pressured to fix integration challenges with greater speed and dexterity in the face of conflicting industry trends toward lower cost of innovation and increasing regulatory pressure, so they fix it as quickly as possible to deliver a validated finished product.

Integration issues are allowed to thrive

On paper, the process of developing an aircraft should be as simple as designing, building, and certifying projects sequentially. In practice, however, the constant clash between the need to iterate to meet performance goals and arrive at the best possible solution, and the need to keep moving forward toward certification, culminates in a reality where design and execution happen concurrently, with no reliable goal post. Integration issues tend to flourish as a result, appearing with virtually no predictability. The lack of common understanding between leadership and systems engineers traces back to the concept design phase, and is perpetuated by the problem with legacy tools and processes.

This lack of understanding is due to the fact that implementation of modeling tools is limited to the conceptual design phase. There is not a channel to effectively communicate the iterative work done in this phase of development downstream as changes happen. Concept design is meant to be the map for development, but even with the most advanced modeling tools, it effectively happens in a vacuum and is thus negated. This leads to cost increases, schedule delays, and a poor perception of scope creep as numerous revisions run rampant.

As the process continues to move forward without a fully validated design to predicate it, companies hope that whatever problems they face can be fixed later. However, the cost of design changes goes up exponentially the later in the process they take place. In one assessment of lifecycle cost impacts, NASA found that the cost to change a design direction could go from a factor of 3-6 in the early stages of development to a factor of 500-1,000 in late stages of product testing.

If individual functions continue to focus on executing their own tasks as quickly as possible without addressing the needs of the full project development lifecycle, then the overarching programs suffer. Adopting integrated toolsets alone will not solve this problem. A shift in mindset and processes, enabled by new tools, will move the focus from system modeling to enabling the full optimization of the product lifecycle. Leadership must usher in this shift with a vision that prioritizes solutions that address the complete ecosystem to improve their program execution.

The better path forward for systems engineering

If the process of aircraft development is a body, systems engineering is its brain, informing every action of the other areas. Leaders have the right idea in putting a greater emphasis on systems engineering in view of mounting complexity challenges, but to truly manage complexity at scale requires a step further. Leaders must not allow systems engineering to operate as a niche function completely cut off from the rest of the development process. It should be leveraged as the overarching program driver, with complete traceability of design changes to identify and address problems proactively and early on.

A total shift in mindset as well as toolset is needed to address the integration challenges that plague nearly all aerospace companies. The future of systems engineering requires an iterative, connected and informed process in conceptual design, which is optimized across multiple domains to select the right product architecture and solutions and allow for anticipation and implementation of integrations and changes in real-time. By making this switch, companies guarantee that the finished aircraft they are working toward matches what was originally defined in concept even as the design changes, the program becomes more complex, and the stakes get higher.

This shift will directly correlate to better regulatory and requirement adherence, fewer integration issues, and delivery on schedule and on budget because previously disparate parties will be working toward a single common truth and outcome.

By connecting the work done in concept design, companies can fix and debug problems early in the design process and focus on verification, rather than continuing to look for problems in testing and building. The result is greater confidence that the products that go to market meet performance requirements and are, without question, safe.

Implementation of a real-time digital twin of design, manufacturing, verification and certification throughout development with complete traceability of design history will be crucial to executing this orchestrated approach to model-based systems engineering processes. This digital twin will facilitate a real-time view of the complete process that allows companies to predict and address integration issues in a virtual environment early on and prevent them entirely as a result. This technology supports goals including orchestration, traceability, and target performance optimization.

- Accelerate your new product development and become more agile
- Get your design right the first time through up-front systems engineering
- Manage complex product and supply chain integration
- Full lifecycle solution to deliver safe, reliable products faster – and within budget

Orchestration of technical programs

A product lifecycle management architecture that is open, interoperable and easily integrated with other systems can be used to drive development of interfaces and subsequently manage product integrations through the detail design and testing phases of development. Companies can orchestrate the content and scope of programs across functional areas by empowering disparate vendors and functional areas to work toward a single common goal outlined in the concept design and to continuously track requirements and implementations. This will encourage streamlined collaboration and consistent information, which will ultimately accelerate development time and reduce the risk of errors that delay product certification and customer gratification.

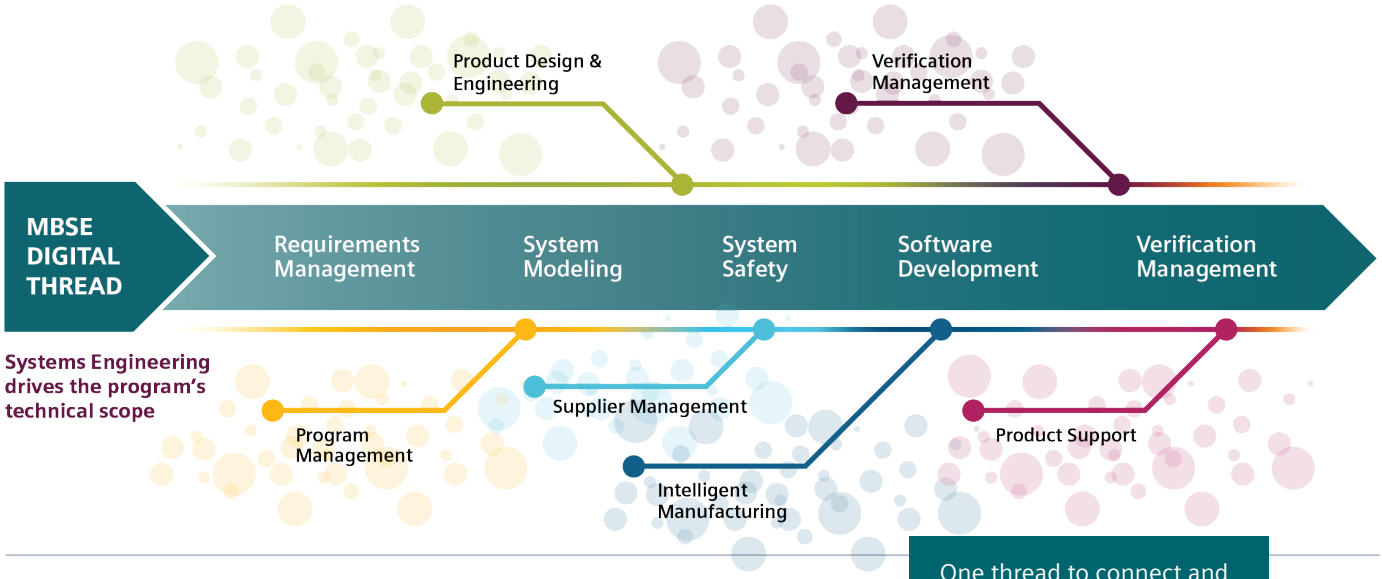
Traceability of design decision implementations in context

A digital thread of traceability allows companies to know when a decision was made and why, throughout the development process. This helps retain and transfer that knowledge in context so product definition does not shift even as the people working on the program do. Traceability will also facilitate better preparedness in the event of audits during the certification process.

Success in meeting target performance on the first try

A digital twin provides continuous visibility and context into the design changes taking place during model-based systems engineering, giving companies the control they need to design and develop products that meet target performance before they're ever built. This switches companies from a mindset of needing to work out kinks in testing to truly knowing a system will work once the pieces come together. The resulting confidence level in validation and certification will enable the bulk of development time to be spent on design and innovation, thus leading to a competitive edge in the industry.

By treating model-based systems engineering as a crucial element of overall program execution, organizations can support the creation of reliable, compliant, and innovative products that map back to the original concept regardless of the new complexity that innovation brings. Implementing the right tools to support program wide orchestration will be the underpinning that enables companies to approach model-based systems engineering through this new lens and enable transformational change in their programs.



One thread to connect and guide all domains.