SysEne
Whitepaper

Systems Engineering: Increasing importance in today's connected world

Products and systems in every industry are getting increasingly connected and complex as part of larger systems-of-systems. Our expectations of the performance, reliability, safety, security, and cost-effectiveness of these systems continue to increase. Systems engineering practices are adapting to meet these challenges, and organizations that embrace these best practices will be best prepared to deliver their planned products and systems.

The trend of connected systems

Every major industry is trending towards smarter, connected systems. In the energy industry, we are transitioning to a "smart grid". In automotive, we are transitioning to mobility-as-a-service and autonomous driving. In aviation, we see the rise of unmanned aerial systems, or drones and collaborative manned/unmanned systems. In maritime, we find a rise of remotely operated and autonomous underwater vehicles. In health care, we are transitioning to a unified ecosystem that is patient-centric, provider-enabling, and delivers a payer-savvy interoperable information exchange. Software is becoming smarter, more connected, and offering new functionality across all those complex systems. The overall drivers of efficiency, safety, and convenience will continue to push all industries towards further smarts and connectivity.

The future is now, and unanticipated new products are emerging

The potential of smart, connected systems was once the realm of science fiction, from geosynchronous communications satellites from Arthur Clarke, to our cellular phones from Star Trek, to autonomous cars from Isaac Asimov. While we see many of these systems in various forms today, we are also developing new valuable products and services that weren't anticipated even one year ago.

For example, our smartphones are going from a connected communicator (Star Trek) to a platform for continuous delivery of customer value in the form of applications, services, and information. Smartphones are even becoming the "brains" of many systems, such as low cost medical devices for the developing world.

Self-driving cars are well on their way to becoming intelligent, efficient, traffic-optimizing, and safer transportation systems with vehicle-to-vehicle and vehicle-to-infrastructure connectivity. Additionally, cars are becoming "infotainment systems on wheels"

Our energy grid is transforming into a smart grid that can manage demand, store energy, improve efficiency, allow for intermittent renewables, and improve resiliency. One example of additional upside is significantly improving wind power by taking data from a number of wind farms, weather stations, and weather satellites, and then with high-powered computing, creating highly accurate wind power forecasts, which can then optimize operations for increased power.

In manufacturing, companies are adding new, recurring revenue streams based on services wrapped around the core product. This enables manufacturers to amortize much of the development cost over the product's service life. It also challenges the end of the idea of "planned obsolescence", since the physical product is now a platform for the delivery of higher-value services.

The challenge of smart-connected systems today, and tomorrow

While the smart, connected systems are providing more value for us, they are increasing the complexity of the products, systems, and product development. Our expectations of acceptable system performance have also increased dramatically. Systems must function, be reliable, and be safe in a wider range of scenarios, interactions, and environments than ever before. Systems have to be more sustainable, across all segments of the value chain, from cradle-to-grave. Increased competition requires cost effectiveness both at system acquisition and for a total cost of ownership through the life-cycle.

All of these challenges are making it increasingly difficult for organizations, owners, and developers to deliver success. The degree of challenge has increased exponentially, and it is difficult for organizations to adapt.

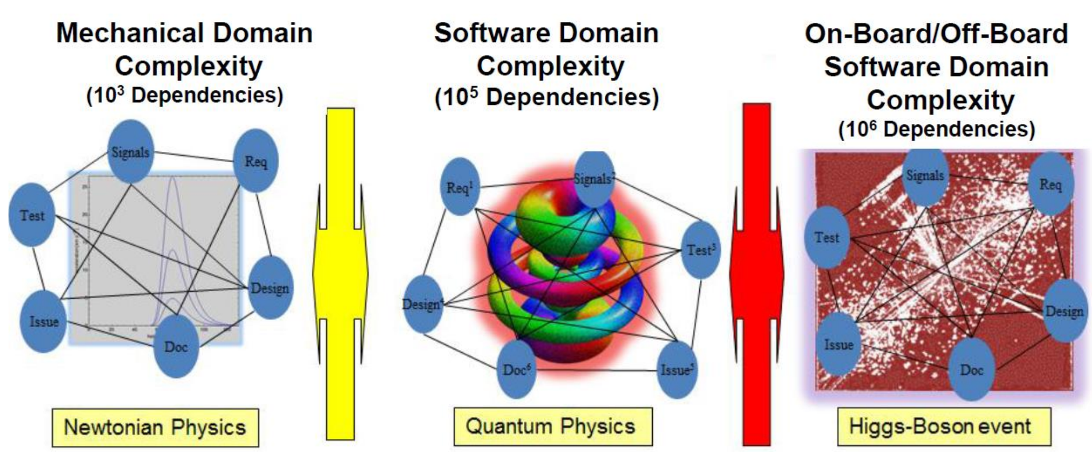
More complex systems require a deeper understanding of needs, risks, interactions, requirements, technologies, and potential impacts than ever before.

The shift towards increasing amounts of software within each product, and towards connecting all the products together, creates additional challenges. Software is growing increasingly large, with exponential interactions and nodes, which makes defects harder to find.

Software is much quicker to iterate and change than hardware systems, but often software is integrated with hardware systems. The more conventional methods of product development that were more suited for hardware systems (e.g. waterfall) have to be adapted to incorporate software-intensive systems methods (e.g. agile).

Additionally, the integration of hardware systems and software systems is very challenging, as the way of thinking of software systems is very different than the way of thinking of hardware systems, and combining them together requires integration specialists that can think in both worlds.

Going forward, the challenges of complexity will only get tougher.



The impact of complexity on product, system, and project delivery

Unfortunately, in all industries, we see more quality problems, higher development costs, and longer project timelines than ever before. Development effort scales disproportionately with complexity. Each new feature, behavior, or non-functional requirement (e.g. the "-ilities" such as reliability or security) brings its own additional complexity and commensurate development effort.

Additionally, risks are increasing. There are even more opportunities for project failure with more interactions, escalating regulatory requirements, and greater risk of project failure and challenges throughout the supply chain. The consequences of failure are becoming even more costly, with higher warranty costs, increased brand damage, and higher penalties. This requires organizations to put more effort into design for reliability, quality assurance, and verification activities.

Development ecosystem changes

Our world is more connected, with more product and system development being done globally. This results in more competition, more regional customization, global development teams, and global supply chains.

New technologies, materials, processes, and innovations are becoming available at an increasing rate, and many of them game-changing. The challenge for any developer and systems engineer is to adapt to each new technological change, to be able to take advantage of it.

Product development is also getting more complex. For example, development is getting more collaborative, enabling customers and competitors to agree on standard interfaces, which helps drive down costs and increases market share and interoperability. At the same time, development is also getting more strategic, taking advantage of differentiation where it makes sense to take advantage of core technology, competencies, and customer values.

The increasing value of good systems engineering

With the increase in complexity, systems engineering is adding more value than ever before. Shortfalls in systems engineering lead to either cancellation of already expensive systems or even greater expense in terms of total cost of ownership or human life. An analysis of the effects of insufficient systems engineering for software-intensive systems shows, out of 161 projects, a significant increase in rework costs and time as a function of source lines of code (SLOC); averages of 18% rework for 10,000 SLOC, and 91% rework for 10 million SLOC. Conversely, adequate investment in systems engineering results in highly cost-effective systems, such as in the cases of Global Positioning Systems, or the Next Generation Medical Infusion Pump Case Study.

Systems engineering situation in most industries

Many organizations do not have a formal systems engineering function, and they still have most of the engineering organized by discipline. These organizations tend to rely on discipline leads (e.g. a control systems manager), or a project manager to take on the systems engineering functions such as requirements development, systems analysis, or systems integration. Too often, each team delivers their subsystem or code and then problems are not detected until late stage system tests, or worse, out in the field. This creates expensive late-stage rework, production line retrofits, or expensive warranty replacements and damage to reputation.

Late stage problems consume resources, which would be better spent on future iterations that are lower in cost and higher in quality, or on new product lines.

Even if an organization has a mature systems engineering organization, the explosion in complexity, software, and changes to the supply chain continues to require improvement in systems engineering practices.

Systems engineering practice enhancement

Systems engineering practices have to adapt to all the new demands, complexity, and changes to the ecosystem, and provide the increasingly important value in product, system, or project delivery. Conventional systems engineering approaches more from the prior mechatronic era are insufficient for today's additional connectivity and software-intensiveness.

Voice of the Customer and requirements elicitation must be aligned with the product and marketing strategy, so that the customer and stakeholder engagement can be tailored accordingly. Some parts of the product can be standardized in industry, to drive down costs, improve interoperability, and improve the overall value proposition for the customer. Other parts of the product should be differentiated according to its value to the customer or the developer's strengths.

Requirements definition and requirements management have to be able to handle both the hardware requirements and the software requirements according to their unique development natures. Requirements traceability is becoming more critical for regulatory and safety compliance.

Model-based engineering has to be consistent or interoperable across all domains – mechanical, electrical, software, and systems – so that there is one "truth" in the data and design specifications.

Model-based systems engineering (MBSE) is an emerging set of tools and practices, which promises to bring visual automated tools to the systems engineering domains of requirements, architectures, and behaviours, just like computer-aided design (CAD) did for mechanical engineering. While MBSE is still developing, there are many successful applications, including at companies like Boeing or Ford.

Systems engineering has to allow for increased collaboration across the global supply chain, global development centers, and with customers. Global development is best practiced by taking advantage of the high bandwidth of technical communication of a collocated multidisciplinary development team, and mitigating the challenges of lower bandwidth of technical communication between global development centers.

Systems engineering has to account for the whole product lifecycle, from cradle-to-grave, including material compatibility, disposal requirements, and changes to its operating environment throughout the product life-cycle.

Systems engineering needs to continue to adapt to the increasing amounts of software in future systems and enable efficient development of complete systems.

SysEne recommends that today's developments require systems engineering practices that are adapted to higher complexity, a more global environment, higher expectations, and increasing amounts of software. To learn more about our systems engineering practice enhancement services, please visit:

http://www.sysene.com/capabilities/systemsengineeringdesign