
Model-Based Systems Engineering (MBSE) Workshop

Executive Summary

Today, we must view products as integrated systems, reflecting the exciting evolution of their capabilities. Unlike earlier products with limited functions, modern innovations are designed to interact with humans and their environments seamlessly. They communicate and collaborate, as seen in the remarkable swarm capabilities of drones.

This expansion of functionality inspires a new era of product development. In the past, a handful of specialists could create simple, domain-specific products, integrating software, electronics, and mechanics. They focused on their expertise while products executed processes deeply rooted in their software. Now, decision-making is enriched by complexity, and the physical capabilities of products are broadening, demanding excellence from their components.

For instance, consider the evolution from a passive spring-damped chassis to an active chassis that uses cameras to monitor the road and adapt its damping to real-time conditions.

In response to these advancements, Model-Based Systems Engineering (MBSE) has emerged as a transformative discipline, fostering a system-domain-agnostic approach. The emphasis is on the desired functions of products rather than traditional component designs.

Participants embark on a journey to discover how modern product systems, such as drones, come to life and how we can manage various development domains to usher in practical, powerful, and market-driven innovations.

Workshop Snapshot

Item	Description
Duration	1-day workshop
Workload	≈ 8 hours total
Format	On-site or video
Language	English (materials/tools in English; B2+ recommended)
Target Learners	Software and hardware engineers, soldiers who want to prepare for a civilian career in Ukraine or Europe
Outcome	Modeled demonstrator with model-based artifacts and templates for the complete life cycle of a mechatronic system
Integration	Can be used as the basis for company-owned projects

Workshop Structure

Block 1: Systems Theory
Introduction to the term System
The Systems Thinking approach
The Problem-Solving Process
Block 2: Systems Engineering
The key elements of Systems Engineering
The Systems Architecture Approach
The Systems Engineering life cycle model
Block 3: Model-Based Systems Engineering
Introduction to the Modeling Approach
Modeling a project
Analysis of the Model Effectiveness

Block 1: Systems Theory

By the end of this block, students can:

- Explain what a “system” is (boundary, elements, environment) with everyday examples
- Sketch a system map showing inputs, outputs, and feedback loops
- Describe the system's attributes for creating effective systems and communication
- Differentiate complicated vs. complex systems and discuss emergence and trade-offs
- Apply a simple problem-solving cycle (problem → objectives → solution → validation)
- Choose the right level of abstraction and clearly state assumptions

Block 2: Systems Engineering

By the end of this block, students can:

- Describe the purpose and value of Systems Engineering in product/project success
- Name and use the key SE elements: stakeholder needs, requirements, architecture, interfaces, risk, V&V, and configuration/change control
- Write precise, testable requirements with acceptance criteria and rationale
- Allocate and trace requirements to subsystems and interfaces
- Select and tailor a life-cycle model (e.g., V-Model vs. iterative) to a given project and justify the choice
- Plan verification and validation (fundamental V&V matrix with methods, responsibilities, and entry/exit criteria)
- Identify and treat risks using a lightweight hazard/risk register

Block 3: Model-Based Systems Engineering (MBSE)

By the end of this block, students can:

- Explain what a model is and when MBSE adds value vs. documents/spreadsheets
- Use core modeling constructs (e.g., in SysML: requirements, use cases, blocks, activities, states) to represent needs, structure, and behavior
- Build a small project model that connects stakeholders, requirements, architecture, and interfaces
- Maintain traceability in the model (needs $\rightleftharpoons$ requirements $\rightleftharpoons$ design $\rightleftharpoons$ test)
- Create simple model checks/analyses (requirement coverage, interface consistency, impact of change)
- Assess model effectiveness with clear criteria (e.g., fewer late changes, faster reviews) and propose improvements
- Generate views/documents from the model to communicate with non-modeling stakeholders