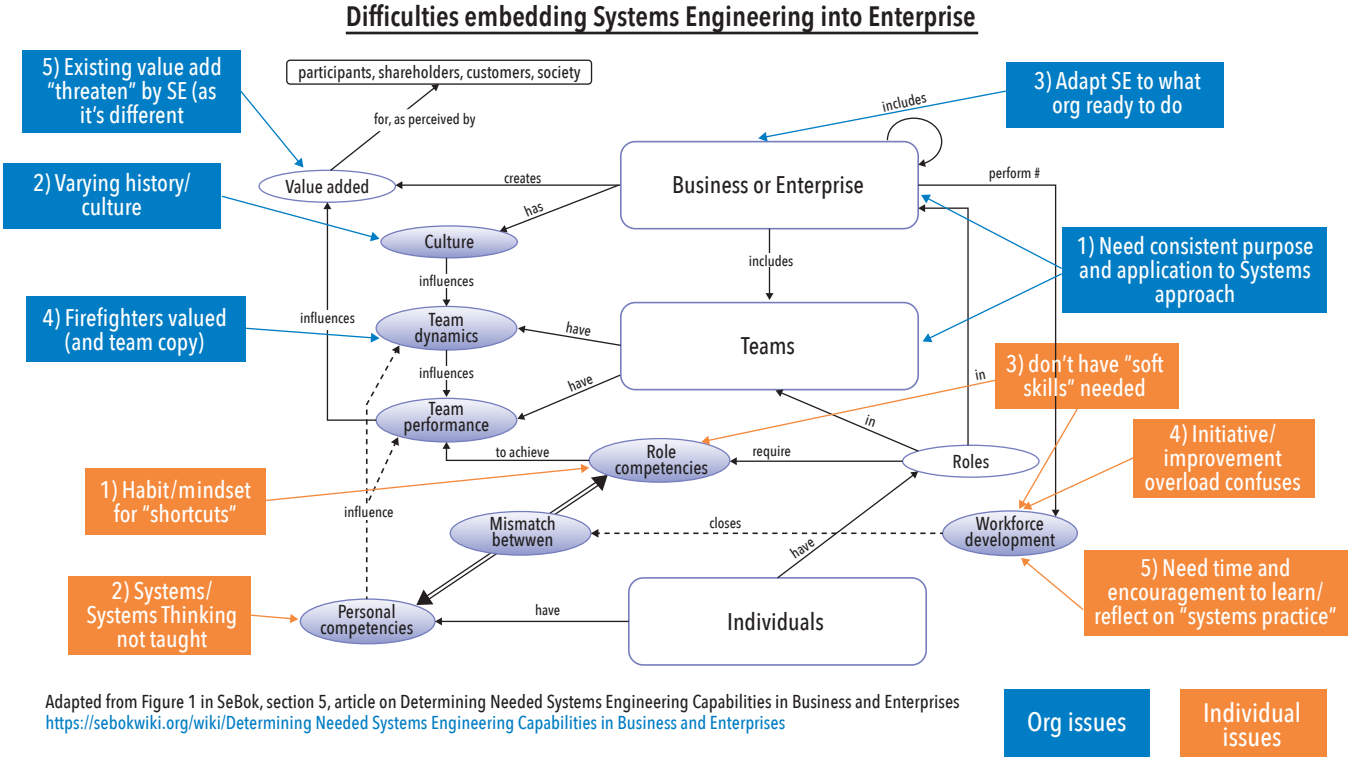
INSIGHT

This Issue's Feature:

Systems Engineering Contributions from Europe, Middle East, and Africa



Adapted from Figure 1 in SeBok, section 5, article on Determining Needed Systems Engineering Capabilities in Business and Enterprises
https://sebokwiki.org/wiki/Determining Needed Systems Engineering Capabilities in Business and Enterprises

Illustration credit: from the article
What Is the Role of a Systems Engineer In an Engineering Organization?
by Richard Beasley (page 29)

FEBRUARY 2026
VOLUME 29 / ISSUE 1

A PUBLICATION OF THE INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING

INCOSE

ANNUAL CONFERENCE ON SYSTEMS ENGINEERING RESEARCH 2026

6-9 APRIL 2026 | ARLINGTON, VA

CSER 2026
conference on systems engineering research

The CSER 2026 theme, *Intelligent Digital Twin-enabled Systems Engineering for 21st Century Sociotechnical Systems*, reflects the growing importance of integrating digital twin technologies with systems engineering practices. This theme emphasizes the need for innovative approaches to design, analysis, and decision-making in sociotechnical environments—where complex systems principles converge with human, organizational and technological factors.

REGISTER NOW

SYSTEMS ENGINEERING RESEARCH CENTER

INCOSE

GEORGE MASON UNIVERSITY

INSIGHT

A PUBLICATION OF THE INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING

FEBRUARY 2026 VOLUME 29/ ISSUE 1

Inside this issue

About This Publication

INFORMATION ABOUT INCOSE

INCOSE's membership extends to over 27,000 members and CAB associates and more than 200 corporations, government entities, and academic institutions. Its mission is to share, promote, and advance the best of systems engineering from across the globe for the benefit of humanity and the planet. INCOSE charters chapters worldwide, includes a corporate advisory board, and is led by elected officers and directors.

For more information, click here:
The International Council on Systems Engineering
(www.incose.org)
INSIGHT is the magazine of the International Council on Systems Engineering. It is published six times per year and

OVERVIEW

features informative articles dedicated to advancing the state of practice in systems engineering and to close the gap with the state of the art. ***INSIGHT*** delivers practical information on current hot topics, implementations, and best practices, written in applications-driven style. There is an emphasis on practical applications, tutorials, guides, and case studies that result in successful outcomes. Explicitly identified opinion pieces, book reviews, and technology roadmapping complement articles to stimulate advancing the state of practice. ***INSIGHT*** is dedicated to advancing the INCOSE objectives of impactful products and accelerating the transformation of systems engineering to a model-based discipline.

Topics to be covered include resilient systems, model-based systems engineering, commercial-driven transformational systems engineering, digital engineering, artificial intelligence, natural systems, agile security, systems of systems, and cyber-physical systems across disciplines and domains of interest to the constituent groups in the systems engineering community: industry, government, and academia. Advances in practice often come from lateral connections of information dissemination across disciplines and domains. ***INSIGHT*** will track advances in the state of the art with follow-up, practically written articles to more rapidly disseminate knowledge to stimulate practice throughout the community.

Editor-In-Chief William Miller
insight@incose.net +1 908-759-7110

Layout and Design Chuck Eng
chuck.eng@comcast.net

Member Services INCOSE Administrative Office
info@incose.net +1 858 541-1725

Officers
President: Michael D. Watson, *Leidos*
President-Elect: Stephen Cook, *Shoal*
Secretary: Stueti Gupta, *BlueKei Solutions*
Treasurer: Alice Squires, *ESEP, University of Arkansas*

Directors
Director for Americas Sector: Renee Steinwand, *ESEP, Booz Allen Hamilton*
Director for EMEA Sector: Sven-Olaf Schulze, *CSEP, Huennemeyer Consulting GmbH*
Director for Asia-Oceania Sector: Quoc Do, *ESEP, Frazer-Nash Consultancy*
Technical Operations Director: Tami Katz, *Ball Aerospace*
Services Director: Chris Browne, *CSEP, The Australian National University*
Director at Large: Jeff Anderson, *Boeing*
Director at Large: Annabel Fraga, *Universidad Carlos III de Madrid*
Director at Large: Annika Meijer-Henriksson, *Saab Aeronautics*
Director at Large: Robert Wirthlin, *Ford Motor Company*
Executive Director:** Steve Records, *INCOSE*

*** Non voting*

PERMISSIONS

*** PLEASE NOTE: If the links highlighted here do not take you to those web sites, please copy and paste address in your browser.**

Permission to reproduce Wiley journal Content:
Requests to reproduce material from John Wiley & Sons publications are being handled through the RightsLink® automated permissions service.

Simply follow the steps below to obtain permission via the Rightslink® system:

- Locate the article you wish to reproduce on Wiley Online Library (http://onlinelibrary.wiley.com)
- Click on the 'Request Permissions' link, under the ‹ARTICLE TOOLS› menu on the abstract page (also available from Table of Contents or Search Results)
- Follow the online instructions and select your requirements from the drop down options and click on 'quick price' to get a quote
- Create a RightsLink® account to complete your transaction (and pay, where applicable)
- Read and accept our Terms and Conditions and download your license
- For any technical queries please contact customercare@copyright.com
- For further information and to view a Rightslink® demo please visit www.wiley.com and select Rights and Permissions.

AUTHORS – If you wish to reuse your own article (or an amended version of it) in a new publication of which you are the author, editor or co-editor, prior permission is not required (with the usual acknowledgements). However, a formal grant of license can be downloaded free of charge from RightsLink if required.

Photocopying
Teaching institutions with a current paid subscription to the journal may make multiple copies for teaching purposes without charge, provided such copies are not resold or copied. In all other cases, permission should be obtained from a reproduction rights organisation (see below) or directly from RightsLink®.

Copyright Licensing Agency (CLA)
Institutions based in the UK with a valid photocopying and/or digital license with the Copyright Licensing Agency may copy excerpts from Wiley books and journals under the terms of their license. For further information go to CLA.

Copyright Clearance Center (CCC)
Institutions based in the US with a valid photocopying and/or digital license with the Copyright Clearance Center may copy excerpts from Wiley books and journals under the terms of their license, please go to CCC.

Other Territories: Please contact your local reproduction rights organisation. For further information please visit www.wiley.com and select Rights and Permissions.
If you have any questions about the permitted uses of a specific article, please contact us.

Permissions Department – UK
John Wiley & Sons Ltd.
The Atrium,
Southern Gate,
Chichester
West Sussex, PO19 8SQ
UK
Email: Permissions@wiley.com
Fax: 44 (0) 1243 770620
or

Permissions Department – US
John Wiley & Sons Inc.
111 River Street MS 4-02
Hoboken, NJ 07030-5774
USA
Email: Permissions@wiley.com
Fax: (201) 748-6008

ARTICLE SUBMISSION insight@incose.net

Publication Schedule. ***INSIGHT*** is published six times per year. Issue and article submission deadlines are as follows:

- April 2026 issue – 2 January 2026
- June 2026 issue – 1 March 2026
- August 2026 issue – 1 May 2026
- October 2026 issue – 1 July 2026
- December 2026 issue – 1 September 2026
- February 2027 issue – 1 November 2026

For further information on submissions and issue themes, visit the INCOSE website: www.incose.org

© 2026 Copyright Notice.

Unless otherwise noted, the entire contents are copyrighted by INCOSE and may not be reproduced in whole or in part without written permission by INCOSE. Permission is given for use of up to three paragraphs as long as full credit is provided. The opinions expressed in ***INSIGHT*** are those of the authors and advertisers and do not necessarily reflect the positions of the editorial staff or the International Council on Systems Engineering. ISSN 2156-485X; (print) ISSN 2156-4868 (online)

ADVERTISE

Readership
INSIGHT reaches over 27,000 members and CAB associates and uncounted employees and students of more than 130 CAB organizations worldwide. Readership includes engineers, manufacturers/purchasers, scientists, research and development professionals, presidents and chief executive officers, students, and other professionals in systems engineering.

Issuance	Circulation
2026, Vol 29, 6 Issues	100% Paid

Contact us for Advertising and Corporate Sales Services
We have a complete range of advertising and publishing solutions professionally managed within our global team. From traditional print-based solutions to cutting-edge online technology the Wiley-Blackwell corporate sales service is your connection to minds that matter. For an overview of all our services please browse our site which is located under the Resources section. Contact our corporate sales team today to discuss the range of services available:

- Print advertising for non-US journals
- Email Table of Contents Sponsorship
- Reprints
- Supplement and sponsorship opportunities
- Books
- Custom Projects
- Online advertising

Click on the option below to email your enquiry to your nearest office:

- Asia and Australia corporatesalesaustralia@wiley.com
- Europe, Middle East and Africa (EMEA) corporatesaleseurope@wiley.com
- Japan corporatesalesjapan@wiley.com
- Korea corporatesaleskorea@wiley.com

USA (also Canada, and South/Central America):

- Healthcare Advertising corporatesalesusa@wiley.com
- Science Advertising Ads_sciences@wiley.com
- Reprints Commercialreprints@wiley.com
- Supplements, Sponsorship, Books and Custom Projects busdev@wiley.com

Or please contact: Marcom@incose.net

CONTACT

Questions or comments concerning:

Submissions, Editorial Policy, or Publication Management
Please contact: William Miller, Editor-in-Chief
insight@incose.net

Advertising—*please contact:*
Marcom@incose.net

Member Services – *please contact:* info@incose.org

ADVERTISER INDEX

February Volume 29-1

CORPORATE ADVISORY BOARD — MEMBER COMPANIES

3DSE Management Consultants
Advanced Systems Engineering, LLC
Aerospace Corporation, The
Airbus
Albers Aerospace
AM General LLC
Analog Devices, Inc.
ANSYS, Inc
Arcfield
Auburn University
Australian National University
Aviage Systems
Aviation Industry Corporation of China, LTD
BAE Systems
Bechtel
Becton Dickinson
Belcan Engineering Group LLC
BlueHalo Labs, An AV Company
BMT Canada
Boeing Company, The
Booz Allen Hamilton Inc.
Boston Scientific Corporation
BTS Software Solutions
California State University Dominguez Hills
Caltech
Capgemini Engineering
Carnegie Mellon Univ. Software Engineering Institute
Change Vision, Inc.
Colorado State University Systems Engineering Programs
Commercial Aircraft Corporation of China, Ltd (COMAC)
Cornell University
Cranfield University
C.S. Draper Laboratory, Inc.
Cubic Corporation
Cummins, Inc.
Dassault SystËmes
Defense Acquisition University
Deloitte Consulting, LLC
Denso Create Inc
Dentsu Soken Inc
DigiFlight , Inc.
Drexel University
Eaton
EMBRAER
FAMU-FSU College of Engineering
Federal Aviation Administration (U.S.)
Florida Institute of Technology
Ford Motor Company
GE Aerospace
General Dynamics
General Motors
George Mason University
Georgia Institute of Technology
Hitachi Energy
Honeywell Aerospace Technologies
Huawei Technologies Co. Ltd
Idaho National Laboratory
IQNOX, LLC
ISAE – Supaero
ISDEFE
IVECO Group
Jama Software
Jet Propulsion Laboratory
John Deere & Company
Johns Hopkins University
KBR, Inc.
KEIO University
L3Harris Technologies
Lawrence Livermore National Laboratory
Leidos
LEONARDO
Lockheed Martin Corporation
Los Alamos National Laboratory
Loyola Marymount University
Magna
ManTech International Corporation
Marquette University
Massachusetts Institute of Technology
MBDA (UK) Ltd
Medtronic
MetaTech Consulting Inc.
Missouri University of Science & Technology
MITRE Corporation, The
Mitsubishi Electric Corporation
Mitsubishi Heavy Industries, Ltd
Modern Technology Solutions Inc
National Aeronautics and Space Administration (NASA)
National Reconnaissance Office (NRO)
Naval Postgraduate School
Nissan Motor Co, Ltd
Northrop Grumman Corporation
Pacific Northwest National Laboratory
Parametric Technology GMBH PTC
Pennsylvania State University
Petronas International Corporation Limited
Prime Solutions Group, Inc
Project Performance International (PPI)
Purdue University
RealmOne
Redwire Space
Rolls-Royce
RTX
Saab AB
SAFRAN
SAIC
Sandia National Laboratories
Saudi Railway Company
Shanghai Formal-Tech Information Technology Co., Ltd
Shell
Siemens
Sierra Nevada Corporation
Singapore Institute of Technology
Southern Methodist University
Space Dynamics Laboratory
SPEC Innovations
Stevens Institute of Technology
Strategic Technical Services LLC
Studio SE, Ltd.
Swedish Defence Materiel Administration (FMV)
Systems Planning and Analysis
System Strategy, Inc (SSI)
Taiwan Space Agency
Tata Consultancy Services
Terumo BCT
Thales
The George Washington University
The University of Arizona
The University of Texas at Arlington
The University of Utah
Torch Technologies
TOSHIBA Corporation
Trane Technologies
Tsinghua University
UK MoD
UNCOMN
Universidade Federal De Minas Gerais
University of Alabama in Huntsville
University of Arkansas
University of California San Diego
University of Connecticut
University of Maryland
University of Maryland, Baltimore County
University of Maryland Global Campus
University of Michigan, Ann Arbor
University of New South Wales, The, Canberra
University of South Alabama
University of South-Eastern Norway (USN)
University of Texas at Austin
University of Texas at El Paso (UTEP)
US Department of Defense
Vector Informatik GmbH
Veoneer US Safety Systems, LLC
Virginia Tech
Volvo Cars Corporation
Volvo Construction Equipment
Wabtec Corporation
Wayne State University
Weber State University
Wichita State University College of Engineering
Woodward Inc
Worcester Polytechnic Institute (WPI)
Woven by Toyota, Inc.
Yulista Services , Inc.
Zuken, Inc

FEBRUARY 2026
VOLUME 29/ ISSUE 1

FROM THE EDITOR-IN-CHIEF

William Miller, insight@incose.net

We are pleased to publish the February 2026 issue of *INSIGHT* published in cooperation with John Wiley & Sons as a magazine for systems engineering practitioners. The *INSIGHT* mission is to provide informative articles for advancing the state of the practice of systems engineering. The intent is to accelerate the dissemination of knowledge to close the gap between the state of practice and the state of the art as captured in *Systems Engineering*, the Journal of INCOSE, also published by Wiley.

The February issue of *INSIGHT* celebrates industry contributions across Europe, the Middle East, and Africa (EMEA) advancing the practice of systems engineering. We have selected a sample of EMEA-sourced articles addressing the following themes:

- Functional-Outcomes Driven Tailoring (FODT)
- Loss-Driven Systems Engineering (LDSE)
- Security in the Future of Systems Engineering (FuSE)
- Unique Abilities of the Systems Engineer
- Archimedes Initiative
- Agility in the Future of Systems Engineering (FuSE).

These themes span the five enumerated categories in the *Systems Engineering Vision 2035* (SEV2035): applications, practices, tools & environment, research, and competencies. The selected EMEA authors and their topics are as follows:

1. Barend Botha – Functional-Outcomes Driven Tailoring in Modern Complex Engineered System Development
2. David Endler – Integrating Loss-Driven Systems Engineering Activities
3. Roar Georgsen and Geir Køien – Very Small Entities (VSEs): Outsourcing Risk to the Supply Chain Is Placing Systems Security Engineering on a Clay Foundation, but Playing Games May Help
4. Juan José López García and Daniel Patrick Pereira – Analyzing System Security Architecture in Concept Phase Using UAF Domains
5. Richard Beasley – What Is the Role of a Systems Engineer in an Engineering Organization?
6. Alan Harding – Systems Skills … From Here to Diversity
7. Wouter Leibbrandt, Jacco Wesselius, and Frans Beenker – TNO-ESI – Systems Engineering Methodologies for Managing Complexity in the High-Tech Equipment Industry: Our Roadmap
8. Domenik Helms, Patrick Uven, and Kim Grüttner – Modular Over-the-air Software Updates for Safety-critical Real-time Systems
9. Laurent Alt and Mikaël Le Mouëlli – How Large Scale Agile Can Operate Systems Engineering in the Future
10. Sophie Plazanet and Juan Navas – Model-Based Systems Engineering as an Enabler of Agility.

We thank the contributing authors. We hope you find *INSIGHT*, the practitioners' magazine for systems engineers, informative and relevant. ■

FEBRUARY 2026
VOLUME 29/ ISSUE 1

Functional-Outcomes Driven Tailoring in Modern Complex Engineered System Development

Barend Botha, Chief Engineer, Halcon, bwbotha@tadhole.com
Copyright ©2025 by Halcon. Permission granted to INCOSE to publish and use.

■ **ABSTRACT**
Modern engineered systems demand lifecycle strategies that are responsive, risk-aware, and aligned with evolving needs. Historically, systems engineering has been methodology-driven, with formal frameworks such as the V-Model or waterfall used to guide process rigor and governance. These methodologies provided structure and control, especially for large, safety-critical engineered programs. However, as systems have become more complex and adaptive, the limitations of rigid methodological adherence have become increasingly apparent. This article retraces the reasoning behind moving from methodology-driven to function-driven, to outcomes-focused, and finally to functionally enabled outcomes. It introduces the concept of functional-outcomes driven tailoring (FODT) as a unifying framework that drives lifecycle performance through functional alignment to mission outcomes.

BACKGROUND AND MOTIVATION

The design, development, and deployment of modern engineered systems take place within an environment of increasing complexity, uncertainty, and operational urgency. These systems must integrate a wide variety of disciplines—mechanical, electrical, software, and human factors—while satisfying stringent safety, performance, and reliability requirements [1][2]. At the same time, the evolving pace of threats, the rapid shift in enabling technologies, and pressure to reduce cost and time-to-field have challenged traditional systems engineering approaches [3][4].

Historically, large engineered programs as commonly found in the aerospace and defense or power generation industries have relied on methodology-centric frameworks such as the waterfall model or the V-model to manage complexity and enforce rigor [1][2]. These models emphasize control and sequential validation, which, although effective in risk-averse environments, often inhibit adaptability and timely feedback. The resulting rigidity can limit innovation, delay decision-making, and create misalignment between engineering execution and stakeholder expectations [4][8].

This growing disconnect has led practitioners to explore alternative approaches, including agile, model-based systems engineering (MBSE), spiral, and hybrid models [2][5][9]. While each offers certain advantages, their application is often fragmented or misaligned due to inconsistent tailoring and lack of functional coherence. What remains missing is a unifying strategy that selects methodologies not based on institutional habit, but based on their ability to fulfill the functional needs of each development phase while supporting desired system outcomes [7][8].

This paper builds the case for functional-outcome-driven tailoring (FODT) as that unifying approach. It does so by examining the evolution from methodology-driven thinking to function-driven and outcome-based alignment—and ultimately to a practical framework that integrates both.

HISTORICAL CONTEXT AND NEED FOR CHANGE

Before diving into function-driven engineering, it's helpful to assess the core functional outcomes traditionally targeted by major systems engineering methodologies. While most of these methods were designed around structure and process, they do embed assumptions about what constitutes functional success [2][8]. Table 1 shows the various traditional methodologies and their outcome orientation.

Table 1. Methodologies and their outcome orientation

Methodology	Primary Functional Focus	Typical Outcome Orientation
Waterfall	Sequential task completion and documentation compliance	Process completion over mission adaptation
V-Model	Rigorous verification and traceability through structured decomposition	Functional completeness and testability
MBSE	Architectural integrity and traceable system behavior	Early defect reduction and integration clarity
Agile	Rapid feedback and incremental capability delivery	Stakeholder usability and responsiveness
Spiral	Iterative risk reduction through prototyping and stakeholder feedback	Learning-focused development
Concurrent Engineering	Cross-functional coordination to compress timelines	Time-to-market with acceptable integration risk
Lean	Waste elimination, efficiency, and value flow	Cost and schedule optimization
Six Sigma	Defect reduction through statistical control	Quality and process reliability
Digital Twin/PLM	Real-time system performance alignment with evolving use conditions	Continuous supportability and system insight

Table 2. Functional objectives and suitable methodologies

Phase	Functional Objective	Suitable Methodologies
Needs Analysis	Understand stakeholder need and system purpose	MBSE, Simulation, Spiral
Concept Exploration	Explore technical feasibility and trade-offs	Agile, Spiral, Prototyping
Requirements Definition	Translate needs into clear, testable requirements	V-Model, MBSE
Design & Development	Build structured, integrated, verifiable solutions	Agile, MBSE, MAHD
Verification & Validation	Demonstrate requirements have been met	Waterfall, V-Model
Production	Produce systems at quality and scale	Lean, Six Sigma
Sustainment & Disposal	Monitor, support, upgrade, decommission	PLM, Digital Twin

These methodologies each serve valuable functions, but they tend to excel in specific lifecycle contexts. None fully span the functional and outcome needs of a complex engineered system across all phases [3][5][8]. This highlights the need to shift from strict methodology adherence to functionally-aligned, outcome-focused tailoring.

Traditional systems engineering frameworks such as the waterfall model, the V-Model, and US Department of Defense (DoD) 5000 lifecycle management processes were developed to impose structure and control on complex defense and aerospace programs [1][3]. These methodology-driven approaches focused on phase-based documentation, gated reviews, and sequential design-validation cycles. While valuable for their rigor, these methods often decouple engineering activity from the functional intent and mission outcomes they were meant to support [2][4].

In practice, projects followed the process — even when the process was not fully aligned with system complexity, urgency, or stakeholder context [7]. As a result, engineering teams optimized compliance over effectiveness, leading to:

- Delays due to rigid phase boundaries
- Inability to adapt to emerging insights
- Misalignment between system performance and mission need.

This prompted the evolution toward a more function-driven model of development.

FUNCTION-DRIVEN ENGINEERING

Function-driven engineering emphasizes that each phase of the lifecycle has a purpose that must be functionally fulfilled [2][5]. Rather than beginning with a fixed methodology, this approach asks: *What must be achieved at this stage?* and then applies the most suitable methods to support that function [1][8]. Table 2 gives the functional objectives and suitable methodologies.

Advantages:

- Enables lifecycle-phase-specific precision
- Promotes architecture alignment and functional integrity
- Encourages early technical risk identification.

Limitations:

- May optimize internal performance but neglect stakeholder value
- Risks over-engineering without clear external validation criteria.

This led to a growing emphasis on not just achieving internal functions, but also aligning those functions to externally validated outcomes [4][7].

OUTCOMES-DRIVEN ENGINEERING

Outcomes-driven engineering flips

Table 3. Lifecycle phase functional-outcome tailoring

Lifecycle Phase	Functional Objective	Example Methodologies
Needs Analysis	Clarify outcomes and required capabilities	MBSE, Simulation, Spiral [5][8]
Concept Development	Investigate technical feasibility	Agile, Spiral, Prototyping [2][9]
Concept Definition	Define architecture, requirements, validation plan	MBSE, V-Model, MAHD [1][2]
Detailed Design	Engineer traceable, modular, compliant solutions	MBSE, Agile, MAHD [9]
Integration & Test	Validate alignment to stakeholder outcomes	V-Model, Waterfall [1][2]
Qualification & Acceptance	Certify readiness and system assurance	Waterfall, Six Sigma [2][8]
Production & Deployment	Ensure scalable delivery with traceable quality	Lean, Six Sigma [2][8]
Operations & Support	Monitor performance and ensure supportability	Digital Twin, PLM [3][10]
Disposal	Execute end-of-life strategies	PLM, V-Model [3][6]

the focus to the end result: *what must the system deliver to the mission, the warfighter, or the customer?* This approach prioritizes:

- Operational effectiveness
- Stakeholder satisfaction
- Measurable success criteria [3][7].

It drives decisions based on mission utility rather than engineering completeness.

Advantages:

- Maintains alignment with evolving mission goals [3][4]
- Enhances value delivery and stakeholder relevance [7][8]
- Supports rapid reprioritization in dynamic contexts [4].

Limitations:

- Can sacrifice traceability and technical depth if not controlled [2]
- Risks superficial designs if functions aren't rigorously linked to outcomes [8].

Based on the advantages and limitations of the two approaches, modern engineering must integrate both views [4][8].

FUNCTIONAL-OUTCOME-DRIVEN TAILORING (FODT)

FODT represents the synthesis of the two previously discussed perspectives: the architectural discipline of function-driven engineering and the mission-oriented priorities of outcomes-driven thinking. Rather than choosing one or the other, FODT recognizes that systems engineering must tailor methods based on both the functional requirements of each lifecycle phase and the operational outcomes they must support [1][2][4]. It is the logical culmination of the evolution from rigid, methodology-driven approaches toward a more adaptive, purpose-built product development framework [3][7].

FODT is inherently disruptive because it shifts the focus from rigid process adherence to functional and mission-driven adaptability. It breaks away from legacy lifecycle orthodoxy and encourages dynamic, real-time decision-making. While this may initially appear to challenge conventional program structures, it actually enables greater alignment between engineering actions and strategic needs [4][6]. In this context, FODT provides the framework to transition from fixed lifecycle models to structured adaptability, which is increasingly essential in modern, fast-moving environments developing complex engineered solutions [3][5].

FODT combines the rigor of function-driven approaches with the adaptability and relevance of outcomes-driven thinking. It recognizes that:

"Functions are the means. Outcomes are the goal. Tailor the method to align both."

Definition: FODT is a strategic, adaptive systems engineering strategy that uses the functional purpose of each lifecycle phase to select and sequence methodologies in order to achieve validated mission outcomes [2][8]. Table 3 shows the various lifecycle phases aligned with ISO 15288 and the functional objective to be achieved and suitable methodologies.

FODT does not lock projects into a lifecycle model, but rather offers an adaptive toolkit governed by the functional demands of each phase and the desired outcomes.

Instead of arguing which method should be applied, the question now becomes:

- "How do we ensure early risk identification and reduction?"
- "How do we maintain traceability with rapid iteration?"
- "How do we validate evolving user needs during development?"

WHY THIS IS THE RIGHT APPROACH:

Organizations developing complex systems are no longer struggling due to a lack of engineering expertise — they're struggling because their processes are misaligned with the real needs of their lifecycle phases. Too often, teams argue over *which* methodology should be used, instead of focusing on *why* and *how* a method addresses their specific challenges.

Functional outcome-driven tailoring (FODT) offers a constructive resolution. It keeps traditional methodologies in the toolbox, but changes the conversation: from enforcing a single lifecycle model, to selecting methods based on functional purpose, risk, and outcome alignment. This enables teams to use the right method at the right time for the right reason [1][2].

What makes this approach effective is not its complexity, but its practicality as it:

- **Decouples Principles from Rigid Models:**
  You avoid dogmatic enforcement of methods and instead focus on outcomes such as traceability, adaptability, risk reduction, and stakeholder alignment [2][8].
- **Enables Tailoring Without Losing Control:**
  Business units or projects are free to choose or combine methodologies, as long as they satisfy the core functional expectations [1][3].

- **Future-Proofs the Organization:** Methodologies evolve (Agile → MAHD, MBSE toolsets change), but the required functions remain stable – ensuring continuity in expectations [4][9].
- **Supports Cross-Project Governance:** Auditing and quality gates can focus on capability fulfilment (e.g., verified requirements traceability, change control), not just on process compliance [7][10].

This approach is not only logical—it is also **aligned with international systems engineering standards and best practices**:

- **ISO/IEC/IEEE 15288** advocates tailoring lifecycle processes to the nature of the system and project [1].
- **INCOSE Systems Engineering Handbook** encourages hybrid and integrated engineering approaches [2].
- **NATO STANAG 4728** mandates structured yet adaptable lifecycle management tailored to program complexity [6].
- The US **DoD Digital Engineering Strategy** promotes flexible, model-integrated methods that support outcomes across the system lifecycle [4].

CORE FUNCTIONAL CAPABILITIES FOR LIFECYCLE SUCCESS

While FODT focuses on tailoring methodologies to suit lifecycle phase functions and outcomes, its success also depends on sustaining a core set of system engineering capabilities. These capabilities provide the structural integrity needed to ensure that tailoring does not compromise system traceability, quality, or operational readiness [1][2][8]. Table 4 shows the core functional capabilities required for lifecycle success.

The effective implementation of functional outcome-driven tailoring (FODT) depends not only on methodology selection but also on the presence of foundational capabilities that ensure consistency, quality, and traceability throughout the lifecycle. These capabilities act as functional anchors that must be maintained—regardless of the specific methods applied—to ensure systems are robust, supportable, and aligned with stakeholder expectations [7][8].

FUNCTIONAL EVALUATION AND METHODOLOGY RANKING STRATEGY

To enable tailored method selection, each development methodology was systematically evaluated for its ability to support the functional objectives of each lifecycle phase. Each development methodology was assessed for its effectiveness in mitigating the core functional challenges at each lifecycle phase. These challenges include traceability, adaptability, integration, testing rigor, stakeholder alignment, and lifecycle support [1][2][8].

The scoring approach involved assigning relative effectiveness (1=highest, 10=lowest) for each methodology per lifecycle phase, considering:

- **Phase-Specific Purpose Fit** – how well the method supports the intended function of the phase
- **Breadth of Functional Coverage** – how broadly the method supports multiple functions in that phase
- **Risk Mitigation Capability** – ability to surface or retire technical and programmatic risk
- **Outcome Alignment** – strength in delivering validated, mission-relevant results.

Table 5 shows the ranking of the traditional methodologies most suitable for each lifecycle phase.

This structured assessment informed a lifecycle-phase-to-methodology relevance mapping, which is visualized in the next section as a heatmap [4][8].

VISUALIZING METHOD RELEVANCE ACROSS LIFECYCLE PHASES

Following the ranking, a heatmap was generated to visualize the relevance of common systems engineering methodologies at each lifecycle phase based on their ability to fulfill phase functions and enable outcomes.

This heatmap helps identify diminishing importance across the lifecycle. For instance, agile and MBSE are highly relevant in early phases—such as needs analysis, concept exploration, and design—but gradually taper in significance during

Table 4. Core functional capabilities required for lifecycle success

Capability	Purpose
Requirements Traceability	Link stakeholder needs to verified functionality
Risk-Driven Decision-Making	Align effort with uncertainty and technical risk
Early Validation of Assumptions	Catch flaws before they compound downstream
Iterative Learning and Feedback	Integrate findings without resetting plans
Configuration and Baseline Control	Govern evolving design and verification artifacts
Design Integration	Ensure coherence across subsystems and disciplines
Milestone-Based Quality Assurance	Provide checks for design and integration health
Lifecycle Sustainment Planning	Address support and obsolescence early
Stakeholder Collaboration	Maintain engagement, alignment, and feedback
Tailoring Governance	Formalize lifecycle method tailoring and deviations (e.g., SEMP)

Table 5. Methodology ranking by lifecycle phase

Lifecycle Phase	Top-Ranked Methodologies (1 = Best)
Needs Analysis	MBSE, Simulation, Spiral [5][8]
Concept Exploration	Spiral, Agile, Concurrent Engineering [2][9]
Requirements Definition	V-Model, MBSE, Six Sigma [1][2]
Design & Development	Agile, MBSE, V-Model [2][9]
Integration & Test	V-Model, Prototyping, MBSE [1][2]
Qualification & Acceptance	V-Model, Waterfall, Six Sigma [1][8]
Production & Deployment	Lean, Six Sigma, Digital Twin / PLM [2][3][10]
Sustainment & Disposal	Digital Twin / PLM, Lean, MBSE [3][10]

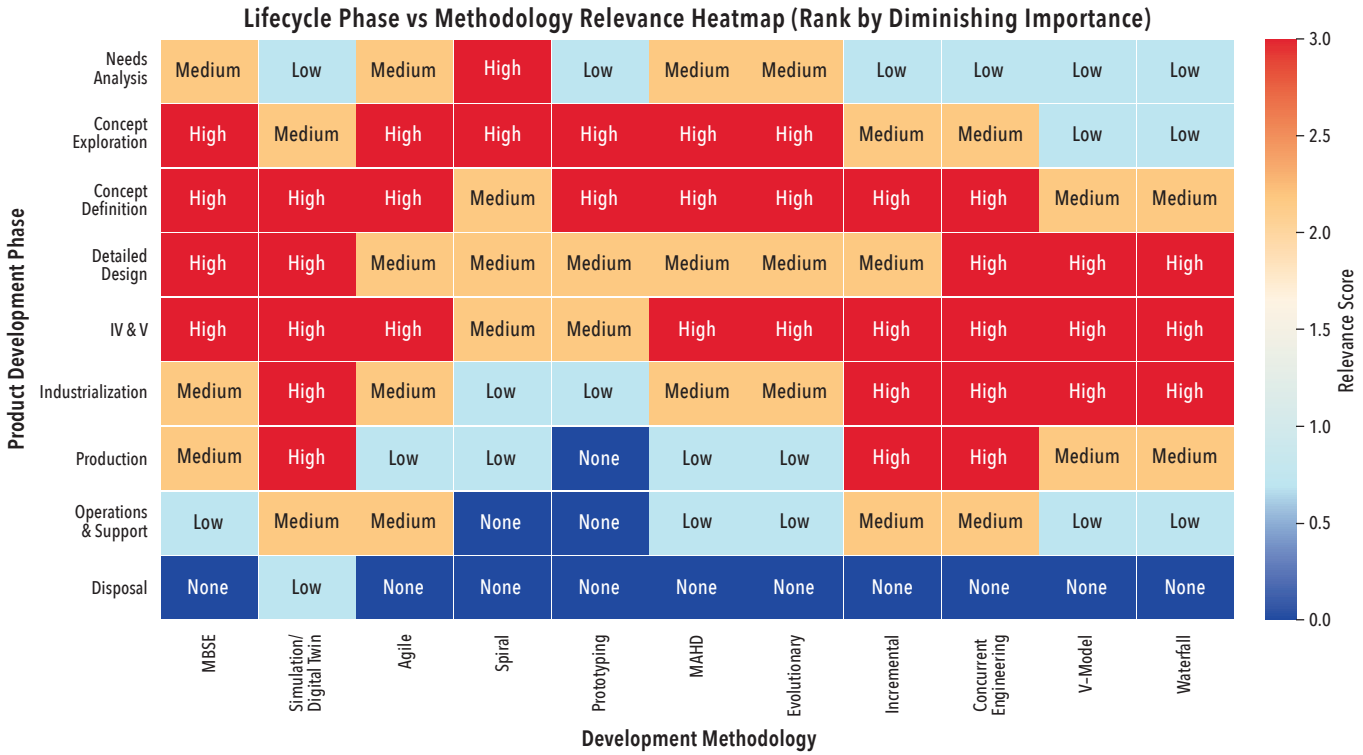


Figure 1. Methodology relevance across lifecycle phases

production and sustainment. In contrast, lean, PLM, and digital twin approaches become more relevant in the latter lifecycle stages, where efficiency, supportability, and real-time system insights become dominant [3][10] (See Figure 1).

A point of discussion is the placement of simulation and digital twin relative to MBSE. Some practitioners argue that simulation or digital twin should precede MBSE in the sequence. However, in the FODT logic, MBSE is positioned first because its initial role is to define system architecture, requirements, and functional behavior. Simulation then follows as a validation and prototyping tool that tests architectural soundness and performance assumptions [5][8].

These insights form the basis for outcome-function alignment that informs lifecycle tailoring decisions. By recognizing how methodology relevance changes over time and aligning it with system maturity and mission context, FODT enables teams to make decisions that reflect the full spectrum of systems complexity, risk posture, and outcome relevance [2][4][8]

A DISRUPTIVE SHIFT

The functional outcome-driven tailoring (FODT) framework represents a profound departure from traditional systems engineering practice. While its rationale is grounded in practicality and international standards, its implementation marks a cultural and operational disruption for most organizations. The disruption arises not from recklessness, but from the deliberate restructuring of how methods, control, and accountability are understood in complex system development [1][2][4].

- **It breaks the method-centric mindset**
  Traditional engineering organizations often promote specific methodologies as if they are fixed identities. FODT challenges this deeply embedded mindset. Instead of treating methodologies like immutable doctrines, it positions them as flexible instruments selected in service of specific functional objectives. In this way, FODT liberates organizations from dogmatic method adherence and reorients the focus on purpose-driven engineering [2][8].

- **It decouples life cycle phases from fixed models**
  FODT rejects the notion that a single methodology should govern the entire product lifecycle. Rather than mapping one method end-to-end, FODT emphasizes selecting the best-fit methodology for each lifecycle phase. This is done based on the functional objective, the phase-specific risk profile, and the method's demonstrated issue mitigation potential [4][7]. This radically differs from conventional process planning, which too often applies a rigid framework regardless of system complexity or maturity.

- **It demands systems literacy over procedural compliance**
  FODT raises the bar for both engineers and project managers. Success under this model depends not on filling out templates, but on understanding how each method serves the goals of a particular lifecycle stage. This requires systems thinking, lifecycle insight, and informed judgement — not checklist execution [2][8]. The shift redefines what competence and ownership mean in an engineering context.
- **It shifts control from process to performance**
  FODT redefines what "control" looks like in systems development. It is no longer sufficient to demonstrate adherence to a prescribed process. Instead, teams must show evidence of managing complexity, addressing risk, and progressing system maturity. This transition from process compliance to outcome assurance significantly alters program expectations and leadership responsibility [1][4].

In summary, FODT introduces a new contract with engineering discipline: one that emphasizes functional alignment, mission relevance, and adaptive control over procedural orthodoxy. While disruptive, it is a disruption in the service of agility, integrity, and strategic coherence. It also represents a positive strategic shift aligned with internationally recognized best prac-

Table 6. Stakeholder concerns and tailoring messages

Stakeholder	Key Concerns	Your Positioning Message
QA/Compliance	Loss of standardization or auditability	"We're not removing structure — we're clarifying it. Every project will still produce the same quality gates, artifacts, and traceability. But we're allowing method tailoring to better match the risk and maturity of each phase."
PMO/Program Managers	Schedule, scope creep, planning complexity	"This approach makes planning easier, not harder — by aligning effort with real needs at each stage. It avoids over-investment early and late-stage surprises. Method selection is risk-based and documented in the SEMP."
Executives	Cost, reputation, accountability	"Our goal is to reduce rework and failure risk by choosing smarter development tactics. This increases our delivery confidence and agility without compromising oversight. We're building in flexibility without losing control."
Engineering Managers	Loss of methodological identity or clarity	"We're not abandoning V-model, agile, or MBSE — we're turning them into tools to solve specific problems, instead of rigid processes. Teams still plan and justify their lifecycle approach — but now they do so with precision."
Customers/ Regulators	Consistency, documentation, transparency	"We remain fully compliant with ISO 15288 and defence acquisition standards. Our tailoring is governed, documented, and reviewed at lifecycle reviews. We still hit every gate — just with smarter, risk-aligned methods."

tices, including ISO/IEC/IEEE 15288 [1], the INCOSE *Systems Engineering Vision 2035* [2], NATO lifecycle principles [6], and the US DoD Digital Engineering Strategy [4]. These frameworks all emphasize tailoring, integration, and outcome-oriented engineering as essential to future capability development.

It will challenge the status quo — but in doing so, it solves the very problem most large engineering organizations face: **rigid processes failing to deliver under modern complexity and time pressure** [3][8].

IMPLEMENTATION CONCERNS AND ORGANIZATIONAL RISKS

While FODT introduces much-needed flexibility and outcome alignment, its successful adoption depends on the ability of organizations to manage several critical challenges:

- **Cultural Resistance to Change**
  Shifting away from entrenched, process-centric thinking may provoke discomfort or scepticism, particularly among experienced practitioners accustomed to legacy frameworks [2][7].

- **Inconsistent Tailoring Practices**
  Without clear guidelines, tailoring may become ad hoc or superficial, resulting in fragmented execution and lack of traceability across programs [1][6].

- **Competency Gaps in Systems Literacy**
  While the individual methodologies used in FODT are familiar, the challenge lies in knowing when and why to apply each method in alignment with lifecycle functions and mission outcomes. Tailoring requires engineers and managers to engage in cross-phase reasoning, something not typically emphasized in traditional training programs. Without developing this systems literacy, tailoring may devolve into guesswork or unjustified decisions [2][4][8].

- **Governance Misalignment**
  Existing review boards, assurance checkpoints, and quality systems may need redefinition to support performance-based rather than process-based evaluation [3][7].

- **Tooling and Infrastructure Readiness**
  Effective tailoring and traceability demand mature toolchains and data integration that not all organizations currently possess [4][10].

FODT's potential lies in its adaptability—but only if supported by structured governance, capability development, and strong leadership alignment. The transition is not trivial, but with a clear strategy and committed sponsorship, the benefits far outweigh the risks [1][3][8].

Importantly, adopting FODT does not mean starting from scratch — but it does require a meaningful culture shift. Many current debates around which methodology should be followed stem from stakeholders operating under different assumptions and priorities. FODT acknowledges these differences and provides a structured way to reconcile them by selecting methodologies based on functional and outcome-driven needs. In this way, FODT builds on what organizations already know, while resolving the friction caused by one-size-fits-all lifecycle models [2][4].

Potential concerns vary by stakeholder group, and it is important to proactively address them through tailored communication. Table 6 summarizes some key concerns of stakeholders and appropriate positioning messages:

CLOSING PERSPECTIVE: FROM PROCESS COMPLIANCE TO PURPOSEFUL DELIVERY

FODT reframes how we measure success in engineering. Instead of defaulting to the question: "*Are we following the process?*" — it challenges us to ask something more purposeful:

"Are we solving the right problems with the right tools at the right time?"

This shift is not simply rhetorical. It defines a new mindset — one that centers engineering discipline on strategic relevance, technical judgment, and delivery integrity. By prioritizing function and outcome over method allegiance, FODT enables organizations to engineer with clarity, agility, and purpose [2][4][8]. ■

REFERENCES

- [1] ISO/IEC/IEEE 15288:2023 *Systems and Software Engineering — System Life Cycle Processes*. International Organization for Standardization (ISO) and IEEE, Geneva, CH. https://www.iso.org/standard/81283.html.
- [2] INCOSE. 2023. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, version 5.0. Hoboken, NJ, USA: John Wiley and Sons, Inc, ISBN: 978-1-119-81429-0. https://www.incose.org/products-and-publications/se-handbook.
- [3] US Department of Defense. 2022. Adaptive Acquisition Framework (AAF) and DoD Systems Engineering Guidebook, U.S. Department of Defense, Office of the Under Secretary of Defense for Research and Engineering (OUSD(R&E)). https://aaf.dau.edu.
- [4] U.S. Department of Defense. 2023 (September). Digital Engineering Strategy, Update
Office of the Deputy Assistant Secretary of Defense for Engineering. https://www.acq.osd.mil/se/docs/DoD-Digital-Engineering-Strategy-Update-2023.pdf.
- [5] NASA. 2023 (March). Systems Engineering Handbook, SP-2023001762, Rev. 3, National Aeronautics and Space Administration (NASA). https://ntrs.nasa.gov/citations/2023001762.
- [6] NATO. 2021. STANAG 4728 Ed. 3, *Allied Systems Engineering Management*. North Atlantic Treaty Organization, NATO Standardization Office, Brussels, BE. https://nso.nato.int (restricted access; NATO login required).
- [7] Defense Acquisition University (DAU). 2022. Tailoring Guidelines and Lifecycle Planning Tools. Defense Acquisition University. Accessible via: https://aaf.dau.edu/tools.
- [8] SEBoK Editorial Board. 2025. *The Guide to the Systems Engineering Body of Knowledge (SEBoK)*, v. 2.13, N. Hutchison (Editor in Chief). Hoboken, US-NJ: The Trustees of the Stevens Institute of Technology. BKCASE is managed and maintained by the Stevens Institute of Technology Systems Engineering Research Center, the International Council on Systems Engineering, and the Institute of Electrical and Electronics Engineers Systems Council. https://www.sebokwiki.org.
- [9] Airbus Defence and Space. 2023. Digital Design and Engineering Framework. Technical Briefing, Airbus Group (internal distribution; cited with permission).
- [10] Lockheed Martin. 2022. Model-Based Engineering and Digital Thread Integration. Internal Strategy Document, Lockheed Martin Corporation (referenced under authorized NDA).
- [11] BAE Systems . 2023. Engineering Policy Directive 4000: Lifecycle Tailoring and Governance. BAE Systems Engineering Governance Group (internal publication).
- [12] Programme SCORPION (France). 2022. Capacité collaborative interarmes French Ministry of the Armed Forces (DGA). https://www.defense.gouv.fr/scorpion.

Integrating Loss-Driven Systems Engineering Activities

David Endler, de@davidendler.de
Copyright ©2020 by David Endler. Published and used by INCOSE with permission.

■ **ABSTRACT**
Loss-driven systems engineering activities are key to realizing successful systems. At the same time, loss-driven systems engineering assessments are, in most cases, complex. In real life projects, integrating loss-driven systems engineering activities in the system development activities might be difficult. In some cases, there is a lack of understanding the activities' importance and sometimes there are organizational barriers. To overcome those barriers, we propose an approach based on widely accepted standards. The difficulty is most existing systems engineering standards poorly describe loss-driven systems engineering activities and how they integrate with traditional engineering activities. This paper provides an approach to successfully accomplish this integration. It is extremely important to involve loss-driven systems engineers in every life cycle phase. At the same time, achieving a common integrated approach understanding is necessary.

■ **KEYWORDS:** Loss-Drive Systems Engineering; Integration; Development Process; Reliability; System Safety; Availability; Maintainability; Security; Resilience

INTRODUCTION

Developing complex systems involves various stakeholders with conflicting interests. Typically, a project manager, who delegates technical aspect responsibility to a lead systems engineer, leads these projects. Consequently, the lead systems engineer carries the responsibility to establish trade studies balancing conflicting technical needs and requirements to realize a successful system. Establishing integrated teams, comprising specialists from many different domains, achieves this. The INCOSE Fellows "systems engineering" definition reflects this very well, stating "systems engineering is a transdisciplinary and integrative approach (INCOSE 2020)."

While integrated team members may include many more experts (purchasing, marketing), this paper addresses the relationship between traditional

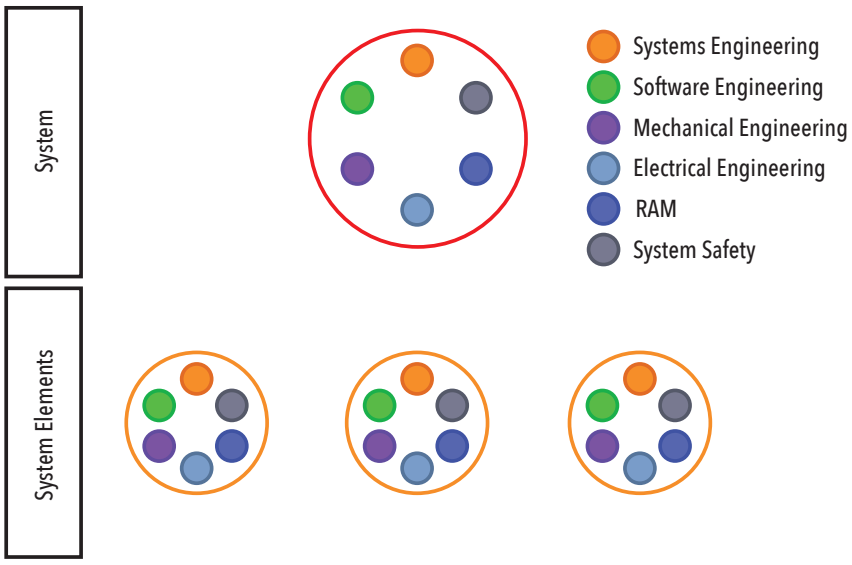


Figure 1. Example Integrated Team

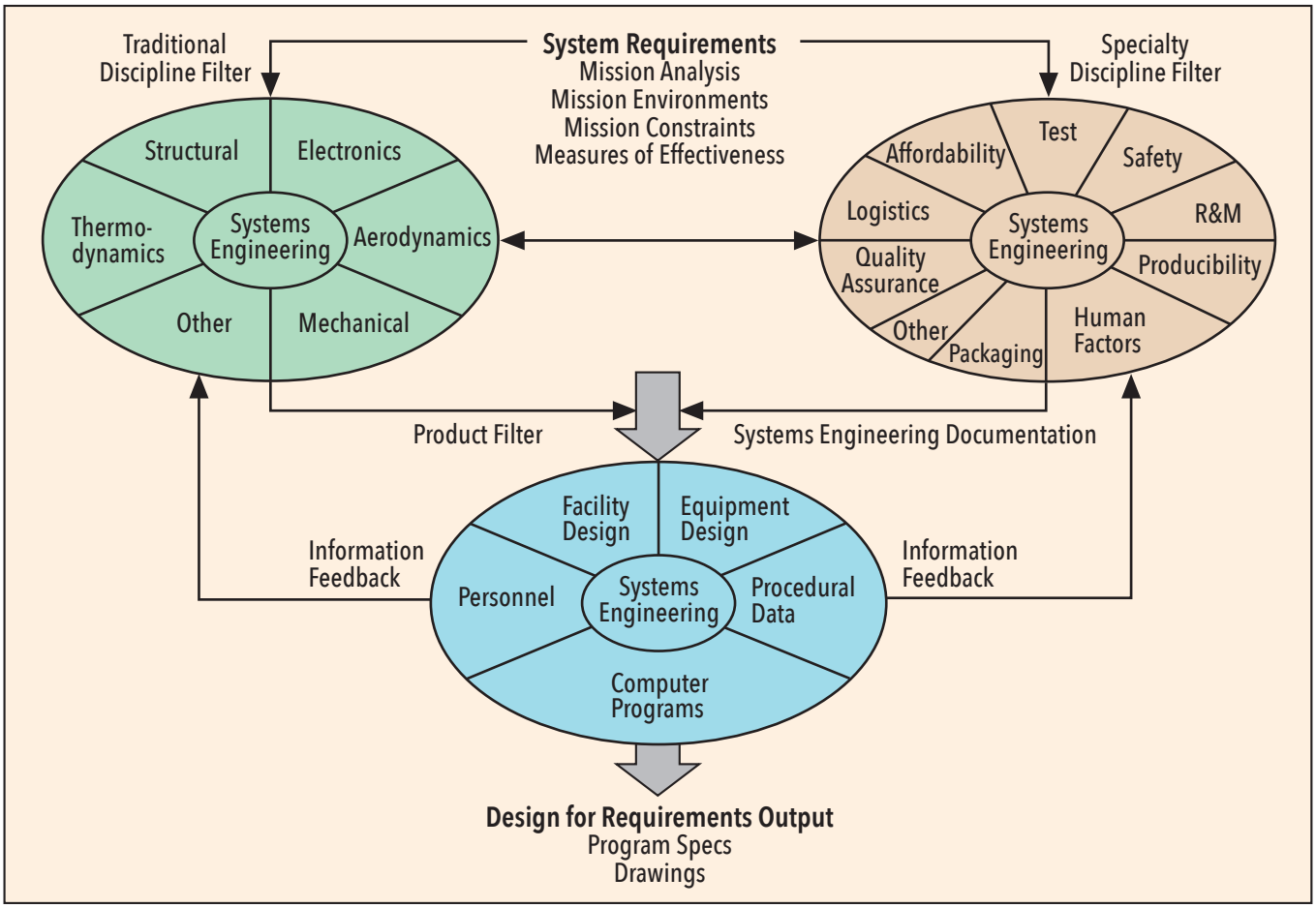


Figure 2. Integration Process for Specialty Engineering, see Part 6 Knowledge Area: Systems Engineering and Industrial Engineering, Figure 1, p. 873 (SEBoK Editorial Board 2019)

engineering activities (mechanical engineering, electrical engineering) and loss-driven systems engineering (LDSE) activities (reliability, availability, maintainability, safety (RAMS), and security, resilience, and recovery). Figure 1 shows an integrated team example with members from different engineering disciplines.

In an integrated team, the lead systems engineer is responsible for identifying agreeing interfaces, defining and allocating system requirements to the corresponding system elements, resolving conflicting requirement issues, and many more activities. In particular, they also must ensure respective specialists perform all analysis tasks required to develop functional and physical system architectures.

Many projects observe subliminal conflicts between traditional engineering and LDSE disciplines due to the very different LDSE discipline natures: where traditional engineering focuses on required capability delivery, LDSE addresses potential system of interest associated losses. Typical examples include, on one hand, obviously contradicting cost and functionality requirements and, on the other hand, safety and reliability requirements. Also observed, traditional field engineers directly allocating to one project. In many cases, loss-driven systems engineers allocated to several (sub-)systems or even to several different projects within the same organization. Consequently, integrated team affiliation was much stronger for traditional engineers compared to loss-driven systems engineers.

There are numerous examples of systems having poor RAMS properties such as Australia's Collins Class submarines (Defense Industry Daily 2015).

This paper proposes an approach to integrating all engineering disciplines to develop a system optimized for all disciplines involved (cost, functionality, reliability, and safety).

PROCESS DESCRIPTIONS

The first step to resolve conflicts between the parties involves analyzing process descriptions, identifying how LDSE aspects integrate into the system development.

Systems Engineering Process Descriptions

Typically, engineers from the traditional domain can easily apply systems engineering process descriptions such as ISO 15288:2015. The ISO 15288:2015 technical process activity description identifies LDSE aspects (clause 6.4.2.3 d)2) on page 53 or clause 6.4.3.3 b)3) on page 55). References include other ISO standards such as IEC 61508 (Functional safety) or ISO TR 18529 (Ergonomics), while not covering aspects like reliability, availability, or maintainability.

Interviews with engineers involved in traditional engineering show understanding process descriptions help their domain. So they focus on system functions and performance requirements, easily overlooking LDSE aspects. ISO 15288 appendix E.4 describing specialty engineering views amplifies this.

The same is true for other standard works such as BKCASE Systems Engineering Body of Knowledge (SEBoK) or INCOSE Systems Engineering Handbook Version 4. SEBoK Part 3 "Systems Engineering and Management" addresses requirements and logical architectures. Part 6 "Related Disciplines" covers LDSE aspects. Looking closer into INCOSE

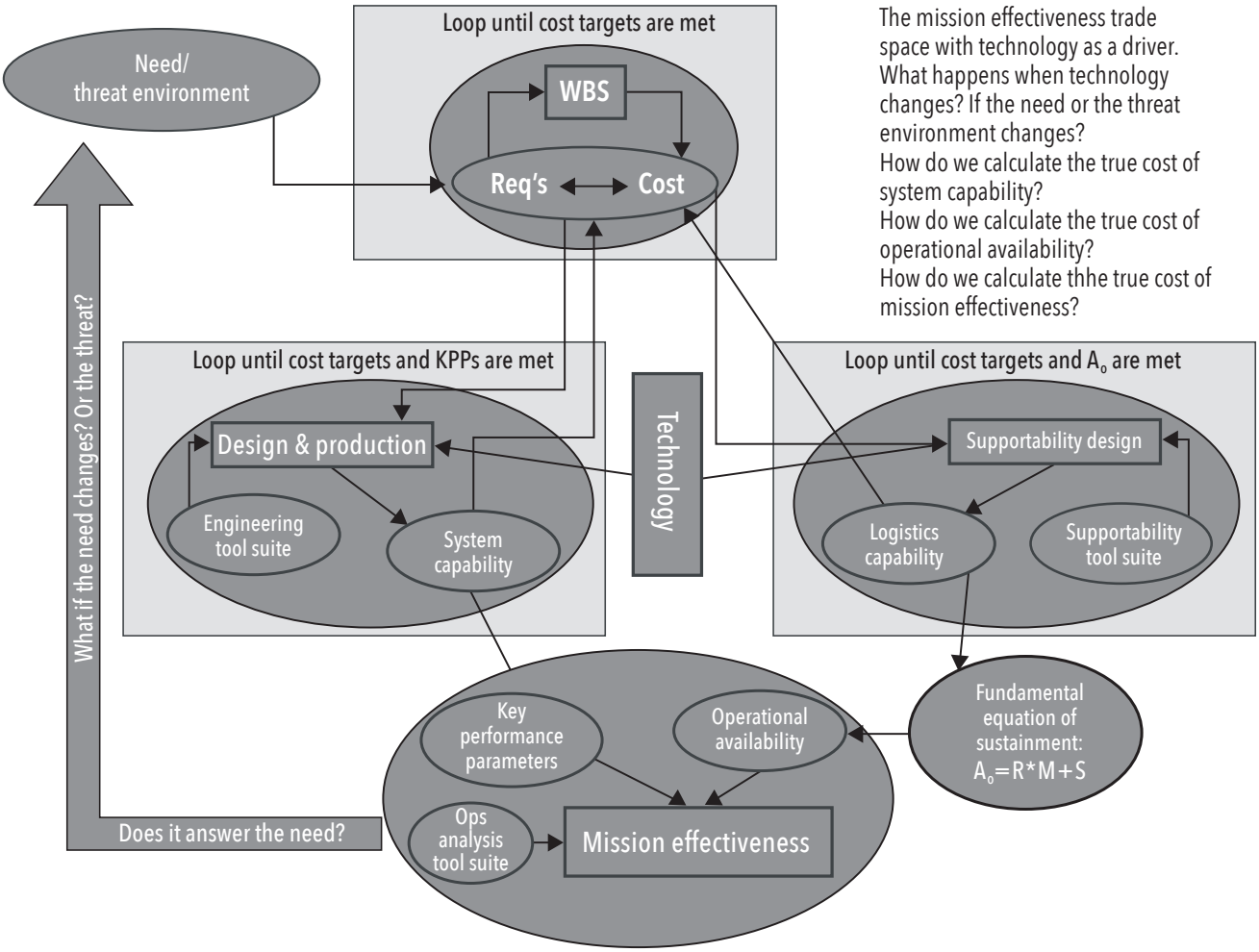


Figure 3. Affordability Cost Analysis Framework, see INCOSE (2015) Figure 10.4

Systems Engineering Handbook Version 4 the picture is the same: details about the technical ISO15288 processes found in chapter 4 whereas chapter 10 describes LDSE activities (and others).

People become even more confused when trying to understand graphical LDSE activity representations in SEBoK and INCOSE Systems Engineering Handbook Version 4. Figure 2 shows the integration process for specialty engineering activities from SEBoK v2.1.

The SEBoK figure shows many details and for some arrows the reader must figure out what they represent. Figure 3, taken from INCOSE Systems Engineering Handbook Version 4, carries even more details, more arrows, and raises many questions. The reader starts at the figure's upper right corner, then reads the questions on left upper corner, and finally gets lost in the arrow wilderness. Without studying the reference given it is impossible to understand the so-called "fundamental equation of sustainment" in INCOSE's work (2015).

In summary, current practice treats LDSE activities separately from traditional engineering activities and the LDSE activity presentation does not contribute to an integrated approach. LDSE activity importance receives very poor awareness. Consequently, this may lead to situations considering LDSE aspects too late in the development.

Loss-Driven Systems Engineering Process Description

The pictures changes when assessing LDSE activity standards. Standards from this domain do not only emphasize integrating LDSE activities is important in product development processes, it also describes how to accomplish this. As an example, Figure 4 shows a safety assessment process model for aircraft system safety assessments taken from SAE ARP 4754A. (see Figure 4.)

Just like in process descriptions presented above, standards related to other LDSE activities provide a systems engineering activity overview at the very beginning (MIL-HDBK-338B Electronic Reliability Design Handbook chapter 4.2). It strongly emphasizes LDSE activities heavily rely on the results from the traditional domains. At the same time, it explains LDSE activities make an essential contribution to successfully realizing the system, providing answers to questions like "How do we know when the design is adequate?" or "How is the effectiveness of a system measured?" (DoD 1998) when applying the methods defined.

Unfortunately, the approach presented limits validity to the LDSE activity area under consideration. Therefore, the approach cannot include other areas as well.

INTEGRATED APPROACH

The approach presented bases itself on experience gathered in several different industry projects. Even though the difficulties in integrating LDSE activities varied in severity, the fundamental root cause was always lack of understanding. Achieving a common LDSE importance and integration understanding requires an approach considering both domains.

Mutual Appreciation

Practice proves a process description close to the traditional process descriptions like ISO 15288 works to create mutual appreciation on both sides, traditional and LDSE.

The approach used bases on a systems engineering process description described in ISO 26702:2007 Systems engineering—systems engineering process application and management. Typically, both domains accept this standard even though it has not updated for quite some time. This standard starts with describing an integrated approach (ISO 26702:2007 chapter 1.1) and maintains this approach throughout the document (chapter 4.7.4, Table-1, or chapter 6.1.1). However, a document with more than 100 pages is not well-suited to gain engineer interest. Therefore, the aim to have something simple in hand helping achieve common understanding continues.

Finally, Figure 5 (next page) provides a simple approach. This figure, taken from ISO 26702:2007, has proven its worth in practice.

Modifying the boxes on the figure's right-hand side created common understanding. Those boxes adapt depending on the subject under consideration. The assessments performed do not only cover traditional aspects but LDSE activities as well.

Figure 5's big advantage is it comes from a widely accepted standard in the traditional domain. Also, it is very easy to show this is exactly the way to use the figure, referring to ISO 26702:2007 chapter 6.7.6. This chapter explicitly states each trade study life cycle must consider cost (chapter 6.7.6.1) and system safety aspects (chapter 6.7.6.3).

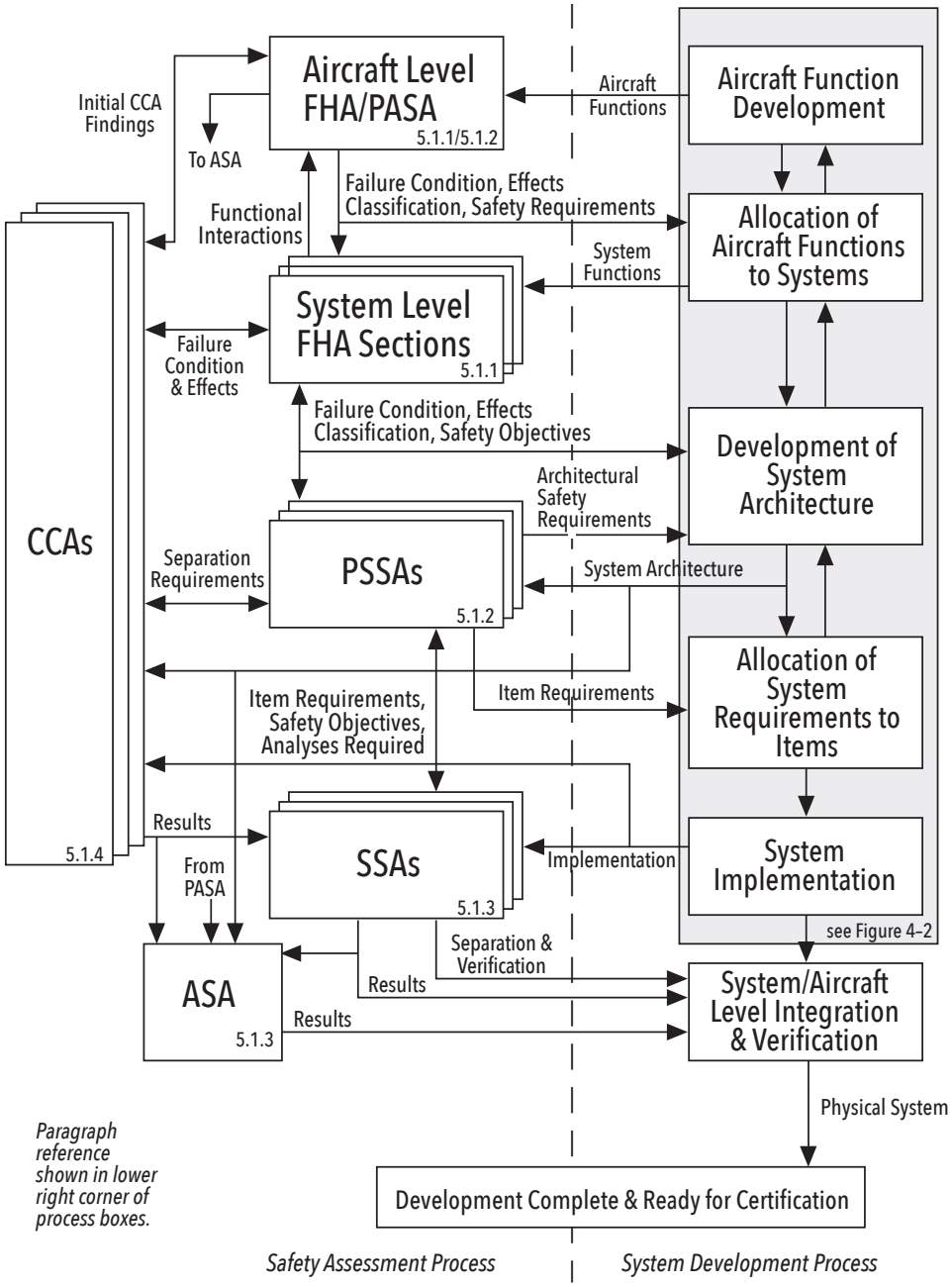


Figure 4. Safety Assessment Process Model, see SAE (2010) Figure 7

Embedding

The approach shown so far will create a common understanding. However, this might not be enough to anchor this common understanding in a sustainable way. To achieve this, we propose a workshop led by the lead systems engineer to establish a Figure 5 tailored version reflecting the current project's particular situation.

In this workshop, respective domains agree on the created results and place the arrows going from left to right and back. This reveals required rigor levels for assessments from the traditional domain feeding into the assessments performed by loss-driven systems engineers. It will also show the ways the assessments performed by loss-driven systems engineers influence the system design. The lead systems engineer will guarantee the agreements made align with any other project constraints.

This process interface definition—a Figure 5 tailored version—must document under configuration control. It has been very effective to print this tailored version as a large-size poster to put in the corresponding offices. ■

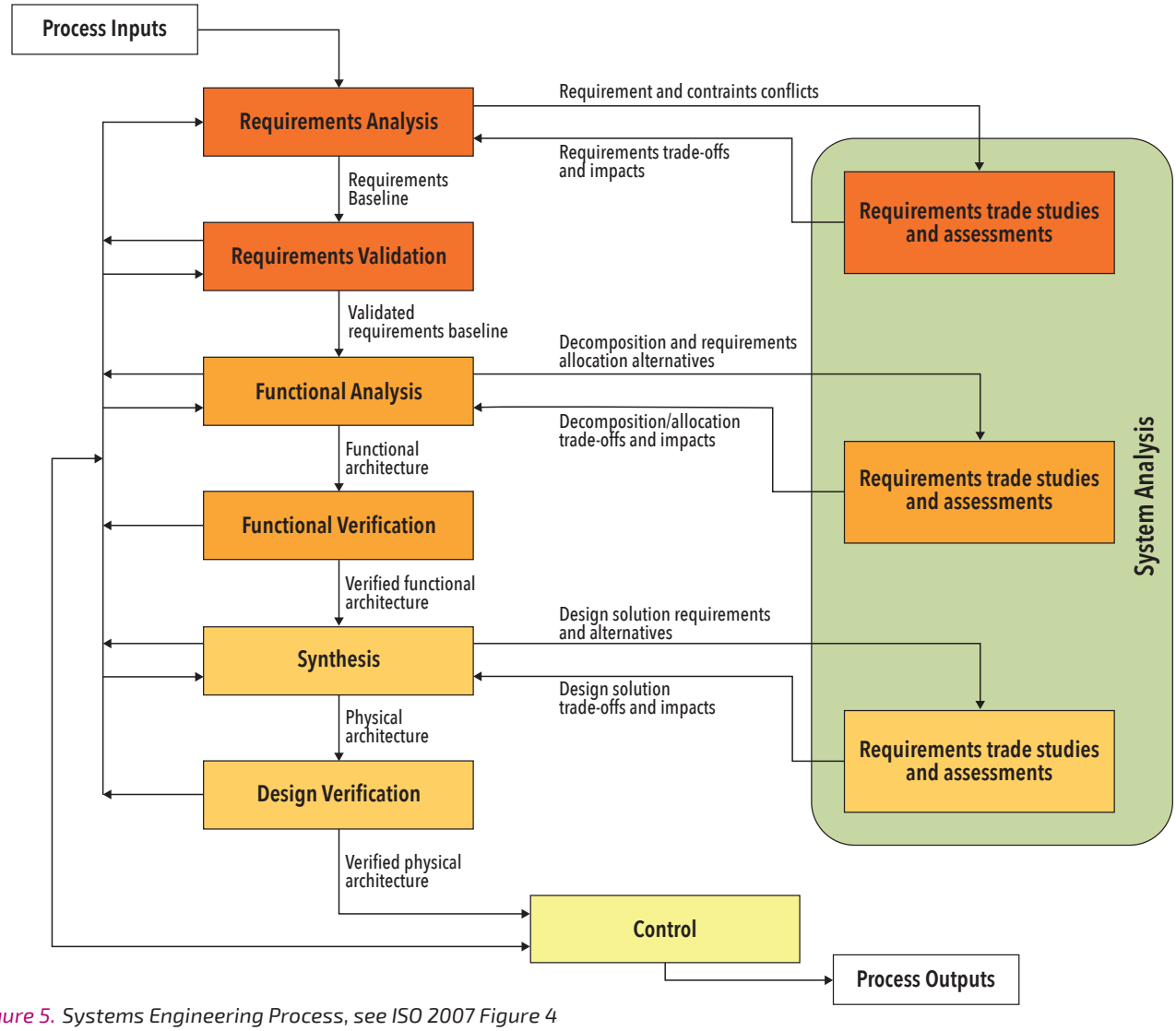


Figure 5. Systems Engineering Process, see ISO 2007 Figure 4

REFERENCES

- BKCASE. "Body of Knowledge and Curriculum to Advance Systems Engineering Project." www.bkcase.org.
- Defense Industry Daily. 2015. "Australia's Submarine Program in the Dock." https://www.defenseindustrydaily.com/australias-submarine-program-in-the-dock-06127/.
- INCOSE. 2015. INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, 4th Edition. Edited by D. D. Walden, G. J. Roedler, K. J. Forsberg, R. D. Hamelin, and T. M. Shortell. Hoboken, US-NJ: Wiley.
- ———. 2020. "Systems Engineering." https://www.incose.org/about-systems-engineering/system-and-se-definition/systems-engineering-definition.
- ISO (International Organization for Standardization). 2007. ISO 26702:2007. Systems Engineering—Application and Management of the Systems Engineering Process. Geneva, CH: ISO.
- ———. 2015. ISO/IEC/IEEE 15288:2015. Systems and Software Engineering—System Life Cycle Processes. Geneva, CH: ISO.
- SAE (Society of Automotive Engineers International). 2010. SAE ARP4754A:2010. Guidelines for Development of Civil Aircraft and Systems. Warrendale, US-PA: SAE International.
- Systems Engineering Body of Knowledge. 2019. "The Guide to the Systems Engineering Body of Knowledge (SEBoK), v.2.1." www.sebokwiki.org.
- United States Department of Defense. 1988. "Chapter 4.2." In Military Handbook-Electronic Reliability Design Handbook. Department of Defense.

ABOUT THE AUTHOR

Dr. David Endler works as a systems engineering consultant and training provider. He has participated in many large-scale projects such as systems engineering process definition for defense and renewable energies companies, lead systems engineer for major aircraft systems, safety and certification responsible for air traffic management systems. He has experience from many industries such as aerospace, automotive, renewable energies, and marine systems. Dr. Endler holds a PhD in physics from the University of Hamburg. He is the current INCOSE Technical Director (2019 to 2021), holds the SE-ZERT® Level A and INCOSE CSEP certificates and is an accredited training provider for SE-ZERT® trainings.

[Editor: Author biography was current when the paper was initially published in 2020.]

FEBRUARY 2026
VOLUME 29/ ISSUE 1

Very Small Entities (VSEs): Outsourcing Risk to the Supply Chain Is Placing Systems Security Engineering on a Clay Foundation, but Playing Games May Help

Roar E. Georgsen, roar.e.georgsen@usn.no; and **Geir M. Køien,** geir.koien@usn.no
Copyright ©2022 by Roar E. Georgsen and Geir M. Køien. Published by INCOSE with permission.

■ **ABSTRACT**
This article addresses the inherent risk in a supply chain that comprises primarily Very Small Entities (VSE) with little to no security proficiency and limited resources and incentive to prioritize system security. In a globalized economy based on outsourcing and risk-sharing, most engineering activities occur in the smallest companies, even for large and complex projects. The Future of Systems Engineering initiative (FuSE) appropriately has agility at the core of its Systems Security Engineering (SSE) foundation concepts, and VSEs are by their very nature agile. However, the line between agility and chaos may be thin, and engineers at VSEs must often accept a level of restraint and rigidity beyond their comfort level to achieve functional agility. The primary challenge in VSEs is adding structure without the necessary resources to enforce compliance manually. We propose that VSE focus their initial efforts on FuSE SSE Foundation Concepts that play into their nature and strengths as dynamic human social activity systems. Improvements in security proficiency and stakeholder alignment do not necessarily require much formal structure, and digital tools combined with social strategies can add structure to a resource-constrained environment. Games can be excellent low-cost tools to provide structure while minimizing resistance, and Agile Model-Based Systems Engineering (AMBSE) using digital models can support automated enforcement. Here we use the card game *Elevation of Privilege* (EoP) as an example. Within the context of a SysML Threat Model integrated into a larger System Model, players naturally treat security requirements as traceable functional requirements. Automated model validation, re-usable components and patterns enforce a Zero-Trust architecture, a sufficiently formal trust model to provide evidence-based assurance, yet achievable for small companies with limited resources.

NO PLAN SURVIVES CONTACT WITH THE ENEMY

"Remember that remote controlled wireless valve de-icer we briefly discussed six months ago? Well, the customer expects delivery on Monday."

Statements like this one are something we hear often. As researchers embedded in small supplier companies, we have first row seats to observe the difference between the theory and practice of systems engineering. In a typical scenario, the customer-facing side of a company, sales, or service, encounters an opportunity in the form of an unmet customer need. They then casually approach the engineer and ask whether implementing the requested feature is feasible. If the engineer answers anything other than a hard "no," the salesperson tells the customer "Yes." When the contracted delivery date approaches, and the engineer discovers that the feature in question has yet to be implemented, the engineer is forced by management to very quickly build a minimum viable product (MVP). "Minimum" in this context means the absolutely smallest possible set of observable system properties necessary to meet a strictly letter-by-letter interpretation of the contract required to avoid any immediate penalties. Needless to say, abstract and unobservable properties such as security, reliability and availability rarely make the cut.

MOST COMPANIES ARE SMALL. VERY SMALL.

Very Small Entities (VSEs) consist of organizations of five (5) to twenty-five (25) people and comprise the vast majority of all companies in the world. More than 92% of European enterprises are micro-enterprises, meaning they have fewer than nine (9) employees.

It is vital to identify and develop tools that support System Security Engineering (SSE) with this network of VSEs in mind. As practitioners of a discipline with roots

in some of the world's largest and most complex organizations, systems engineers must not lose sight of the fact that most engineering activities take place in a very different context from that which gave rise to systems engineering as a discipline. The design of secure systems needs to start from the premise that modern systems depend on an increasingly complex global network of small suppliers, most of which have no systems engineering capabilities, little to no security proficiency, and minimal incentive to prioritize system security.

VSEs typically have no experience working with standardized processes and lack experience in intentionally performing activities such as architecture, design, verification, test definition, and execution (Muñoz et al. 2021). They are aware of growing customer and legal demands to prioritize security, but they do not perceive this as adding value to their work. They experience the practices employed by larger companies as rigid and inappropriate to their context (Sanchez-Gordon, O'Connor and Colomo-Palacios 2015). VSEs are rarely required to document compliance with specific standards and prefer to produce only the minimum documentation required (Sanchez-Gordon, O'Connor and Colomo-Palacios 2015). When documented compliance is required, this does not necessarily correlate with actual engineering practices (Tran 2014). The result is that the actual risk embedded in the supply chain is hidden. As larger organizations increasingly systematize their security engineering efforts, their VSE suppliers struggle to keep pace. Consequently, measures such as risk-sharing partnerships (Figueiredo, Gutenberg and Sbragia 2007) intended to improve supply chain security can actually make the system less secure as responsibility and accountability move down the supply chain.

Any systems engineering tool or philosophy that cannot survive the frantic and sometimes chaotic world of the VSE runs the risk of becoming bureaucratic sugar coating concealing a house of cards built on a clay foundation.

CONTROLLING THE UN-CONTROLLABLE

The first instinct of an engineer when facing complexity is to measure, control and standardize. INCOSE contributed to the ISO/IEC 29110 family of standards in an attempt to address the needs of VSEs, and these standards provide a structural interface compatible with standards used by larger companies. It is possible to adapt the ISO/IEC 29110 standard to an agile workflow (Laporte and Miranda 2020; Muñoz, Mejia and Lagunas 2018; Muñoz, Mejia and Laporte 2019). Model-based systems engineering (MBSE), in this context Threat Modelling, has been suggested as a suitable process for dealing with the complexity of the modern threat environment. However, trying to enumerate all interfaces between systems and subsystems, identifying and enumerating all dataflows, quickly grows beyond the capacity of a VSE engineering team with limited time and resources. Despite the well-known benefits, threat modelling is often performed late in the engineering lifecycle and often not at all unless mandated by customers or regulatory authorities (Shostack 2014a).

The goal should not be to control complexity. Such a goal is not simply challenging; it is impossible in a very literal sense. Once the delusion of control has been cast aside, it is possible to seek out more appropriate mental models, ones that recognize that engineers are human beings with flaws and limitations, but that also have useful intuitive skills that can be leveraged.

A VSE ROADMAP OF SSE FOUNDATION CONCEPTS

The philosopher Andy Clark uses the image of a crew on a sailboat navigating rough seas as a model for how the human mind manages to navigate an infinitely complex world with limited, finite resources. A human being, according to Clark, is essentially a pattern matching prediction machine continuously and actively seeking and making new predictions. Rather than simply reacting to input, the human mind efficiently selects the next input, resulting in fast and frugal problem-solving routines. Our actions structure the physical, social, and technological worlds around us. The ship's crew harbors no expectation of fully controlling the chaos of the sea but manages to navigate the waves through communication, experience, and tools that embed that experience externally. Nevertheless, our ship's crew should never cling to the illusion of control. On the contrary, striving for more control than is possible may directly lead to losing what little control one may have. This is a very systems-oriented perspective, which is why it is so perplexing that so many systems engineering methodologies turn out so rigid in practice. We want to create more capable prediction machines making efficient judgements with higher accuracy. In biological systems capable of learning, this is done by generating and testing problem-solving routines with exposure to novel experiences. This creates new behavior that is largely automatic and thus inexpensive to maintain. Fortunately for us, engineers happen to be biological systems capable of learning.

When it comes to VSEs, increasing security proficiency in the systems engineering team must, by necessity, be primarily learning-based. In VSEs, "The systems engineering team" will be synonymous with "The engineering team." Even trained systems engineers are expected to perform domain-specific engineering tasks. On the one hand, this can be beneficial because SSE is always embedded in the overall engineering workflow, but on the other hand, this makes in-depth security expertise rare. As pointed out in (Dove et al. 2021), "proficiency is unlikely to be found in systems engineers that haven't spent considerable career time developing breadth and depth in security." However, training the permanent engineering team in basic security practice is a low-hanging fruit that is easy to maintain and will provide immediate benefits. Other strategies will have to be employed to capture the expertise of external security specialists in a way that can inform and guide the work of the permanent team. Complementary to individual learning is the SSE foundation concept of Stakeholder Alignment. This alignment can be viewed as a process by which individual agents interact to synchronize their mental models to match other agents. The same way all biological agents adapt their pattern matching routines based on feedback from their physical environment, human beings adapt to feedback from their social interactions. There exists a mode of such socializing that has been shown to be particularly well suited to interpersonal alignment, and that is broadly practiced: It is called "playing."

FUN AND GAMES

"Serious Games" are those games that "have an explicit and carefully thought-out educational purpose and are not intended to be played primarily for amusement." (Abt 1987) However, this passes over the point that amusement is an essential feature of games, which makes games different from mere simulation. Agile methodologies and team structures lend themselves particularly well to games, demonstrated by techniques such as Planning Poker (Grenning 2002) for task estimation. Protection Poker (Williams, Gegick and Meneely 2009) is a similar game used for software security analysis. Another Serious Game used for security analysis is the VOME project game Privacy (Hyperion 2011).

Part of why games work is their ability to induce a state of undistracted concentration often referred to as "Flow" (Csikszentmihalyi 2008). Games are intrinsically rewarding activities with clear goals and immediate feedback, not unlike crewing a sailboat. In a transdisciplinary game with players from different backgrounds, the game's structure can provide a set of "boundary

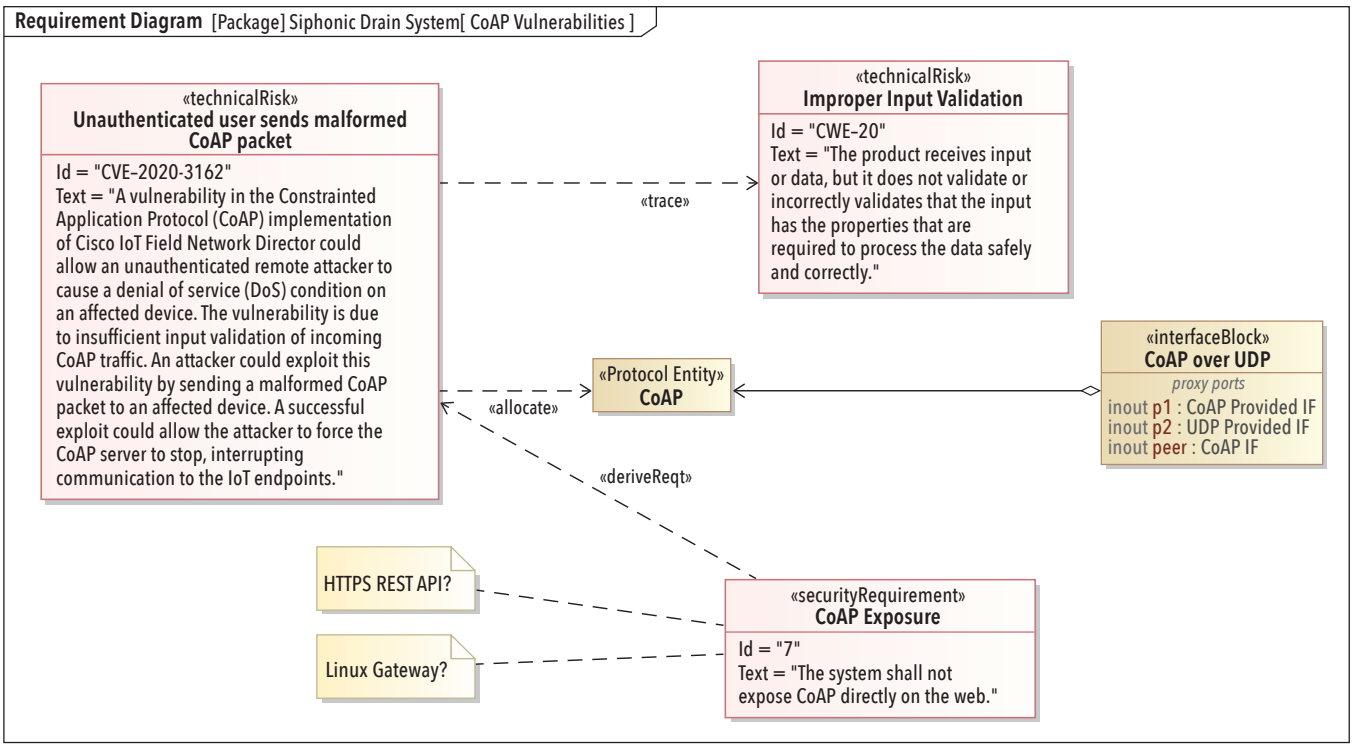


Figure 1. CoAP security requirements

objects" (Bowker and Star 1999). These objects exist in multiple domains but may play different roles in each area. As boundary objects align, their different roles clarify and one can communicate the roles as the game progresses. Stakeholder Alignment is an alignment of boundary objects, or put another way, a conversational construction of social interfaces, or a synchronization of mental models. In our work, we have attempted to apply these principles to threat modelling at VSEs.

THREAT MODELING

The goal of threat modelling is not to provide completeness or perfect consistency. Instead, it should focus on flexibility, efficiency, and being approachable for designers and implementers. Resistance to change as manifested in VSEs supports an approach where engineers perform threat modelling not as a separate exercise but as part of other engineering activities (Sanchez-Gordon, O'Connor and Colomo-Palacios 2015). If participants perceive the work as having immediate value to their own work, this can reduce resistance. Action Research (AR) is an interventionist approach to knowledge acquisition (Lewin 1951). AR proceeds from a praxis of participation, guided by practitioners' concerns for practicality, is inclusive of stakeholders' ways of knowing, and helps build capacity for ongoing change efforts (Bradbury 2015). Actively embracing uncertainty in this way can allow small teams to leverage natural human social and pattern matching skills. Still, we must balance the trust necessary to achieve this by a healthy level of systematic paranoia.

TRUST NO-ONE

An established concept is the "Zero Trust" (ZT) concept in the cybersecurity domain (NSA 2021), and recently there have been presidential executive orders issued in the US ordering the introduction of ZT principles and architectures in US federal infrastructure (*Executive Order on Improving the Nation's Cybersecurity* 2021). A central ZT tenet is that breaches are essentially unavoidable. Consequently, one must face up to a reality where unwarranted trust is dangerous.

One approach is to adopt a particularly paranoid form of ZT and recognize that an infinite number of actors constantly threatens a networked system. The word "actors" here is deliberate. While it is essential to focus on malicious actors such as attackers, it is easy to forget the threat caused by internal actors through incompetence, ignorance or simply a lack of time. When threat modelling, one should assume the model is incomplete and guaranteed to be incorrect in some way. This means accepting that most trust is unwarranted, that warranted trust has a cost, which in turn can inform what assets are worth paying that cost to protect.

In the context of threat modelling, ZT means paying particular attention to interfaces regarded as "trust boundaries." In a digital model, this is where one would attach security requirements. Figure 1 shows a known vulnerability and weakness in the CoAP protocol modelled as an extension to SysML requirements, along with a derived functional security requirement to mitigate the known risk. Because this technical risk is allocated to a specific interface model component, we highlight any use of CoAP in diagrams, as shown in Figure 2 (next page).

Implementing ZT with SysML is not as robust as using a formal model that supports trust based on proof. However, it is sufficiently machine-readable to allow a high degree of automated model checking, all within the resources of a VSE. Ideally, we want to leverage this potential for automation and combine it with the social strengths of VSEs that come to play during games.

PLAYING AT BEING PARANOID.

STRIDE is a threat model and modelling methodology developed at Microsoft (Howard and LeBlanc 2003; Shostack 2008), and the game *Elevation of Privilege* (Shostack 2014a, 2014b) (EoP) is a card game based on STRIDE. Each card in EoP represents a specific threat from the STRIDE model, and in our study, we modelled this as a SysML requirement. A digital model visualized the system, and as each player played a card, they had to explain to the group how the threat on the card per-

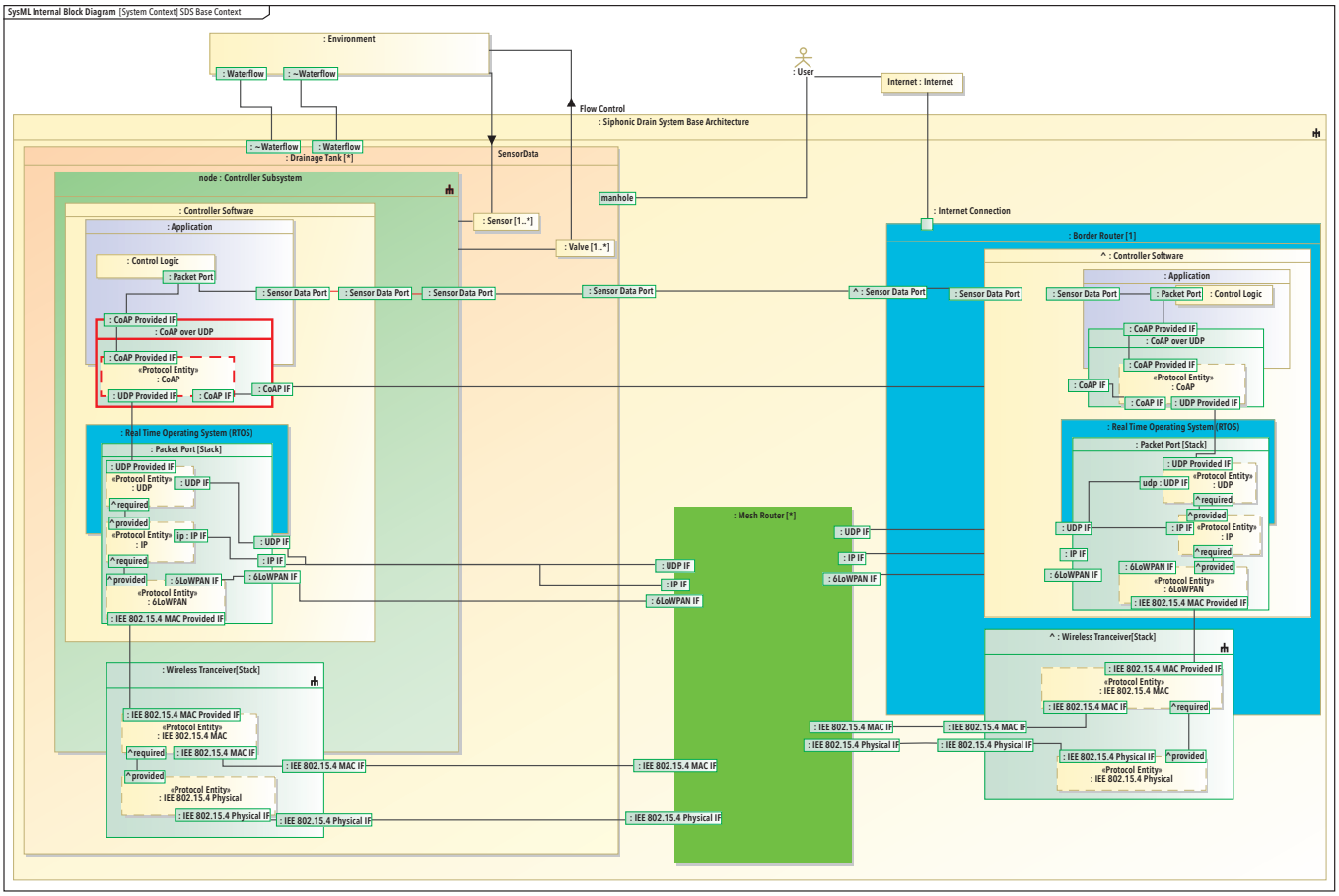


Figure 2. Automatic highlighting of interfaces with a known vulnerability

tained to the System-of-Interest (SoI) in the current context. The game continued until all players played all cards in the deck. Each card traces to more than one requirement. The game's purpose was to facilitate group discussion, linking cards to specific components, interfaces, or functions in the model. We added potential mitigations, notes, and requirements to the model as they came up during the game, as seen in Figure 1. Figure 3 shows a simple example of a STRIDE threat attached to a requirement.

At this point, we can return to the unlikely inclusion of a security expert on the VSEs permanent engineering team. We usually bring in outswide security experts to do the modelling. However, the internal team has the in-depth knowledge to assess the costs and tradeoffs involved in any potential mitigations. The security expert can provide highly specific knowledge, such as the CoAP example in Figure 1, and they can capture this in the model as shown. However, equally important is the role of the expert as educator and facilitator of the game.

Many engineers initially expressed skepticism when asked to participate in threat modelling. They did not perceive this as part of their job and were hesitant to commit their time to it. However, as the exercise progressed, players reported new insights due to the modelling process that directly related to their primary responsibilities. This created engagement and contributed to a broader range of perspectives integrated into the threat model. Because the game forced players to justify why a particular STRIDE threat was or was not relevant, they had to communicate their understanding of that part of the system in much greater detail than they would typically do with other stakeholders. Players generally preferred a

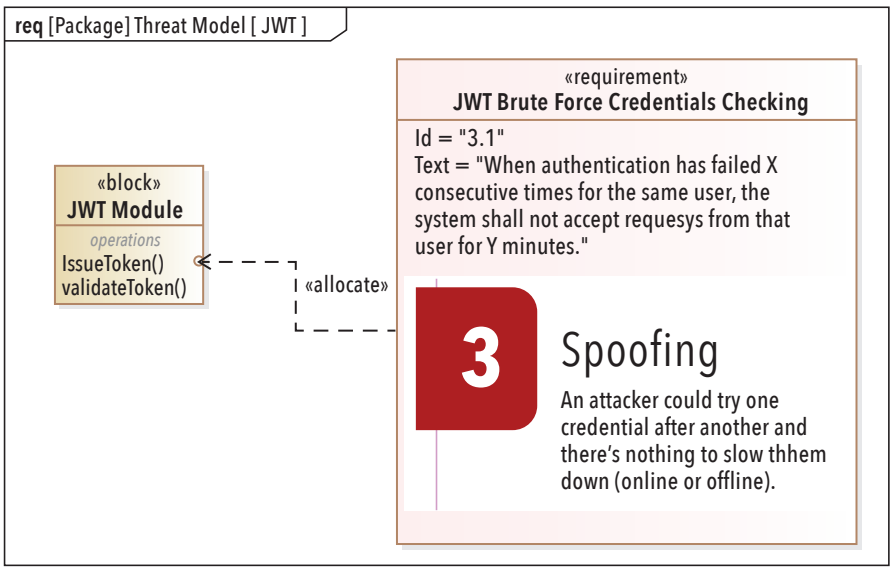


Figure 3. STRIDE SysML requirement

high level of consensus before modifying the model, so any inconsistencies the players negotiated, and players reported a better understanding of what other stakeholders needed from the system and why, especially when those stakeholders participated in the game. The game rules enforced a strict time limit on the modelling process. This time-boxing worked because the game started with the premise that completeness was not the goal.

VSES AND THE FUTURE OF SYSTEMS SECURITY ENGINEERING

The original motivation behind this work with VSEs was to help small historically non-ICT companies meet new requirements and build resilient systems to withstand the Denial-of-Service (DDoS) and ransomware attacks that were becoming a growing threat to their operations. However, it soon became apparent that VSEs collectively represent a substantial global risk factor. On the other hand, VSEs can provide an excellent opportunity to stress-test tools methodologies. Small companies are naturally agile and social organizations, and thus we should not assume they will exhibit the same dynamics as larger companies. Mature VSEs also differ from recent tech startups. Therefore, we should consider if different enabling factors might be necessary to help them build secure systems and prevent them from becoming a threat to others. The FuSE SSE Foundation Concepts are meant to be implementable without independencies, but the particulars of VSEs support prioritizing some before others. VSEs are resistant to change, so starting with improvements in individual security proficiency and application of that knowledge in a natural social setting such as games will be a low hanging fruit. Also, digital tools have improved immensely in recent years, and leveraging the potential of automation that comes with those tools can make other Foundation Concepts easier to implement, such as Modeling Trust or Security as Functional Requirements. ■

REFERENCES

- Abt, C.C. 1987. *Serious games*, University press of America.
- Bowker G.C., Star S.L.1999. *Sorting Things out: Classification and Its Consequences*. Cambridge, US: MA: MIT Press
- Bradbury, H. 2015. *The SAGE handbook of action research*, 3rd ed., London, UK: Sage.
- Csikszentmihalyi, M. 2008, *Flow: the psychology of optimal experience*. New York, US-NY: Harper Perennial.
- Dove, R., Willett, K., McDermott, T., Dunlap, H., MacNamara, D.P. and Ocker, C. 2021. Security in the Future of Systems Engineering (FuSE), a Roadmap of Foundation Concepts. *INCOSE International Symposium*, vol. 31, no. 1, pp. 175–194.
- Executive Order on Improving the Nation's Cybersecurity. 2021. *The White House*, viewed 6 December 2021, <https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/>.
- Figueiredo, P., Gutenberg, S. and Sbragia, R. 2007. Risk-sharing partnerships with suppliers: the case of Embraer. *Challenges in the Management of new Technologies*, World Scientific, pp. 241–262.
- Grenning, J. 2002. *Planning Poker or How to avoid analysis paralysis while release planning*, p. 3.
- Howard, M. and LeBlanc, D. 2003. Sage, *Writing secure code*, Pearson Education.
- Hyperion, C. 2011. It's all fun and games, until… no, wait, it is all fun and games. *Consult Hyperion*, viewed 6 December 2021, <https://chyp.com/2011/06/04/its-all-fun-and-games-until-no-wait-it-is-all-fun-and-games/>.
- Laporte, C.Y. and Miranda, J.M. 2020. Delivering Software- and Systems-Engineering Standards for Small Teams. *Computer* 53(8): 79–83.
- Lewin, K. 1951. *Field Theory in Social Science: Selected Theoretical Papers*, D. Cartwright (ed.), Harpers.
- Muñoz, M., Mejia, J., and Lagunas, A. 2018. Implementation of the ISO/IEC 29110 Standard in Agile Environments: A Systematic Literature Review. *13th Iberian Conference on Information Systems and Technologies (CISTI)*, pp. 1–6.
- Muñoz, M., Mejia, J. and Laporte, C.Y. 2019. Reinforcing Very Small Entities Using Agile Methodologies with the ISO/IEC 29110. in J. Mejia, M. Muñoz, Á. Rocha, A. Peña and M. fPérez-Cisneros (eds), *Trends and Applications in Software Engineering*, Springer International Publishing, Cham, pp. 88–98.
- Muñoz, M., Mejía, J., Peña, A., Laporte, C.Y. and Gasca-Hurtado, G.P. 2021. Beyond factors that motivate the adoption of the ISO/IEC 29110 in Mexico: An exploratory study of the implementation pace of this standard and the benefits observed', *IET Software*.
- NSA. 2021. *Embracing a Zero Trust Security Model*. National Security Agency.
- Sanchez-Gordon, M-L, O'Connor, RV and Colomo-Palacios, R 2015, 'Evaluating VSEs Viewpoint and Sentiment Towards the ISO/IEC 29110 Standard: A Two Country Grounded Theory Study', in T Rout, RV O'Connor and A Dorling (eds), *Software Process Improvement and Capability Determination*, Springer International Publishing, Cham, pp. 114–127.
- Shostack, A. 2008. *Experiences Threat Modeling at Microsoft*, p. 11.
- Shostack, A. 2014a. *Elevation of Privilege: Drawing Developers into Threat Modeling*, p. 12.
- Shostack, A. 2014b., *Threat modeling: designing for security*, Wiley, Indianapolis, IN.
- Tran, X-L. 2014., *Systems Engineering Tool Selection Framework for Australian Defence Small and Medium Enterprises*, PhD Thesis, University of South Australia.
- Williams, L., Gegick, M. and Meneely, A. 2009. Protection Poker: Structuring Software Security Risk Assessment and Knowledge Transfer., in F. Massacci, S.T. Redwine and N. Zannone (eds), *Engineering Secure Software and Systems*. Berlin, DE: Springer, Heidelberg, pp. 122–134.

ABOUT THE AUTHORS

Roar Elias Georgsen received a B.Eng. in Computer Engineering and an M.Sc. in Systems Engineering from the University of South-Eastern Norway (USN). Currently, he is the head of product development at Aiwell and Aiwell Water and an Industrial PhD Research Fellow with USN in Horten, Norway. His research interests include model-based systems engineering, digital transformation in small engineering teams, and integrated safety, security, and reliability design.

Geir M. Køien received his PhD from Aalborg University, on access security for mobile systems. He has also worked for many years in industry, including LM Ericsson Norway and Telenor R and D. During these years he worked extensively with mobile systems and with security and privacy. He has also worked with the Norwegian Defence Research Establishment and with Norwegian Communications Authority on various security and communications related projects. Currently, he is a professor with the University of South-Eastern Norway (USN).

[Editor: Author biographies were current when the paper was initially published in 2022.]

FEBRUARY 2026
VOLUME 29/ ISSUE 1

Analyzing System Security Architecture in Concept Phase Using UAF Domains

Juan José López García, juan-jose.j.lopez-garcia@airbus.com; and **Daniel Patrick Pereira,** daniel.pereira@airbus.com
Copyright ©2022 by Juan José López García and Daniel Patrick Pereira. Published by INCOSE with permission.

■ **ABSTRACT**
This paper presents combining MBSE (Model-Based System Engineering) and STPA (Systems-Theoretic Process Analysis) to mitigate security risks at an early stage of system development and to increase agility when developing or modifying architectures. The MBSE approach states that the systems development process should have a system model or a set of models as the unique source of truth. From the system model or a set of models, systems engineers of different specialties should be able to extract the information needed to perform their job. However, some specialties usually create their artefact apart from the model to perform the analysis, breaking the premises of MBSE to have a unique source of truth leading to out-of-date artefacts. This article proposes extending the Unified Architecture Framework (UAF) Profile (UAFP) to enable safety and security systems engineers to perform their analysis from the early stage of a system development process.

1. INTRODUCTION

This article addresses the FuSE concepts presented in Section 4 (refer to "Table 3. Concepts" Being Addressed in this Article" for further information). The systems that we are building today cross a wide variety of domains. Stakeholders demand higher reliability, and shorter product life cycles. Besides, global connectivity gives rise to system vulnerabilities. The systems engineering discipline, through an interactive top-down process, allows the systems engineer to understand the whole system. Several model and simulation practices are part of the formal systems engineering process which is the foundation of Model-Based System Engineering (MBSE). The formalized models support system requirements, design, analysis, verification, and validation activities.

The International Council on Systems Engineering (INCOSE) emphasizes how important MBSE (INCOSE 2014) is to manage design complexity including architecture, requirements, interfaces, behavior, and test vectors. However, some enterprise architecture frameworks miss addressing the inherent security aspects.

The Unified Architecture Framework (UAF) published by the Object Management Group (OMG) defines a complete set of stakeholder domains. The seven domains are the basis for creating several architecture views of an enterprise, as well as the systems that make up the enterprise. The domains allow for a logical and systematic flow of architecting activities.

The Unified Architecture Framework (UAF) is based on the Unified Profile for the United States Department of Defense Architecture Framework (DoDAF) and the United Kingdom's Ministry of Defence Architecture Framework (MODAF) (UPDM). UAF defines ways of representing an enterprise architecture that enables stakeholders to focus on specific areas of interest in the enterprise while retaining sight of the big picture. UAF intends to provide a standard representation for describing enterprise architectures using an MBSE approach. The UAF::Security profile illustrates the security assets, security constraints, security controls, families, and measures required to address specific security concerns. We observe a lack of elements that allow the systems engineer to conduct the safety and security analyses using this profile.

One of the main goals of the systems engineering processes is to deliver systems that are trustworthy. Security is an emergent property of a system and it shares the same challenges in its realization as other emergent properties like safety. To cope with systems engineering's goal, engineers must translate stakeholders' needs to provide adequate system security requirements related to the consequences associated with the loss of assets throughout the system life cycle. Employing system theory in early stages of system development enables engineers to leverage adequate functional security requirements, which would help to tackle the FuSE Concept 8 – Security as a functional requirement (for further information refer to Table 3. "Concepts Addressed in this Article").

System-Theoretic Process Analysis (STPA) (Levenson 2011; Levenson and Thomas 2018) is a hazard analysis meth-

od based on systems theory for analyzing undesired system behavior. Unlike the traditional hazard analysis techniques, STPA can apply at the early stage of system development to assist in identifying safety and security constraints. STPA derives an analysis in a control loop in terms of control actions, feedback, and communication. The control loop elements are known as a controller and a controlled process.

This article proposes extending the Unified Architecture Framework (UAF) Profile to enable safety and security systems engineers to perform their analyses at the early stage of a system development process. This work extends the UAF Profile to support the STPA elements. The remaining sections of this papers are as fol lows. Firstly, Section 2 surveys related works. The next section, Section 3 proposes the STPA UAF Profile. Section 4 discusses the proposed approach and outlines future work.

2. RELATED WORK

The following related work is a basis for the work presented in this article. With regards to MBSE and security risk analysis, and in line with FuSE Concept 8 "Security as a Functional Requirement, (Mažeika and Butleris 2020a; Mažeika and Butleris 2020b)" explores how MBSE can leverage the development and definition of secure systems. For this purpose, we tailor a specific MBSE profile which contains security and put it into practice with a specific example. The so-called MBSEsec Method (Model-Based Systems Engineering Method for Creating Secure Systems) has 4 steps: (1) identify security requirements, (2) capture and allocate assets, (3) model threats and risks, and (4) decide objectives and controls. All four steps use MBSE. This work is a clear example of how customization of MBSE and specifically a security-oriented MBSE method can enable security-by-design during system development, and how this could integrate with processes such as ARP 4754 (SAE Guidelines for Development of Civil Aircraft and Systems - https://www.sae.org/standards/content/arp4754a), with the outcome of defining security functional requirements.

Concerning the related work already performed for the FuSE Concept 5 "Architectural Agility" and FuSE Concept 1 "Security Proficiency in the Systems Engineering Team (Papke 2017)," exposes how organizations can reuse secure systems in the IOT by using MBSE. Specifically, how to design in an agile manner a secure system in dynamic environments in which threats are constantly evolving. Challenges such as tackling security during the complete "V" lifecycle and identifying and defining the threats together with reusable components are presented in Papke (2017).

3. PROPOSED APPROACH

The basis of the proposed approach starts after analyzing the UAF Domain Metamodel (DMM). This Metamodel provides the definition of concepts, relationships, and viewpoints for the framework. The UAF DMM is the basis for any implementation of UAF including non-UML or SysML implementations. UAF enables the modelling of strategic capabilities, operational scenarios, services, resources, personnel, security, projects, standards, measures, and requirements; which supports best practices through, separation of concerns and abstractions.

The UAFP is a UML/SysML implementation of the UAF DMM. The purpose of the approach presented in this section and basis of future work is to extend the UAFP to provide resources to evaluate the safety and security aspects based on the STPA analysis tool.

Using the already existing elements of UAF, the STPA analysis can be performed. However, we found it beneficial to extend the UAFP with new elements, required to perform the STPA analysis. The work described here reused as much as possible the UAFP elements, and only created those new elements required, with their purpose described. The authors added the new elements to the strategic and operational domains.

UAF's strategic domain describes the capability taxonomy, composition, dependencies, and evolution. The UAF's operational domain illustrates the Logical Architecture of the enterprise. It describes the requirements, operational behavior, structure, and exchanges required to support (exhibit) capabilities. The UAF's security domain defines the hierarchy of security assets and asset owners, security constraints, and details of where they are located. UAF's strategic domain ties to the UAF's operational domain. UAF's operational domain helps to define the problem in the UAF's security domain.

3.1. Proposed UAF and STPA Combination

The authors present a summary of the main steps of both UAF and STPA Analysis in Figures 1 and 2. The arrangement of the nine steps of the UAF illustrated in Figure 1 are in alignment with the UAF stakeholder viewpoints to produce the architectural views. Although the authors numbered the steps, one need not follow the sequence. In fact, some of the steps can occur simultaneously. The STPA integrates into the UAF Step 2 and UAF Step 3.

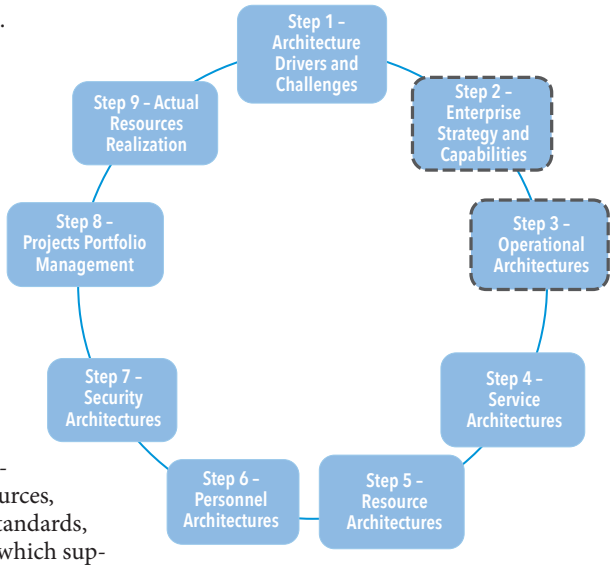


Figure 1. UAF main steps

Engineers conduct the four steps of STPA, illustrated in Figure 2, to identify safety and security constraints considering stakeholders concerns analyzing the operational system behavior. The authors present the main outputs of each step. An engineer performs the steps sequentially as follows:

- **STPA Step 1**: Consists of identifying its mission, key stakeholders, and system purpose and goal. In addition, the authors identify the following:
  - Losses;
  - Hazards;
  - System-level constraints.
- **STPA Step 2**: To model a system hierarchical control structure composed of control action, feedback and communication.
- **STPA Step 3**: Identification of an HCA (Hazardous Control Action). An HCA is a control action issued in a particular context and the worst-case scenario will lead to a hazard [H].
- **STPA Step 4**: Describes the causal factors that can lead to the Hazardous Control Action(s) [HCA(s)] and to the Hazard [H].

The purpose of UAF Step 2 is to describe the capability taxonomy, composition of capabilities, dependencies between capabilities, and evolution of the capabilities. The purpose of this step relates with the STPA Step 1. The UAF Step 3 purpose is to describe the requirements, operational behavior, structure, and exchanges required to support (exhibit) capabilities. As the engineer raises the system behavior here, the engineer conducts the remaining STPA Steps along with the UAF Step 3.

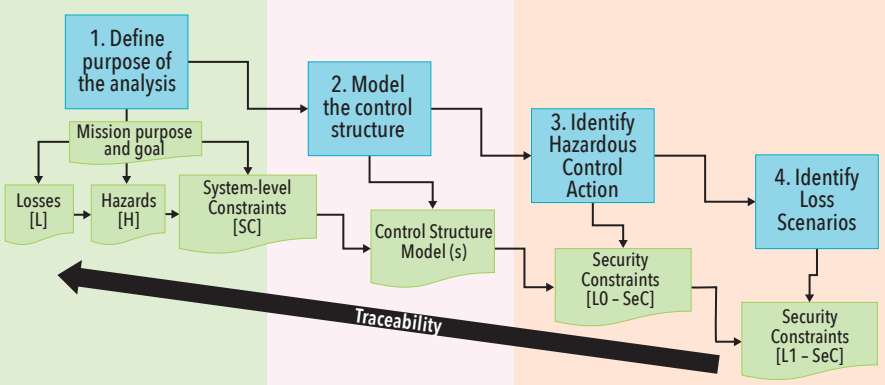


Figure 2. STPA main steps

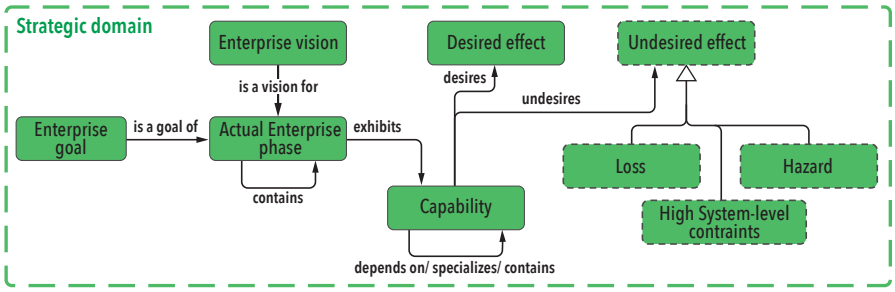


Figure 3. UAF strategic domain extension

In fact, the UAF Step 2 and UAF Step 3 should happen in parallel. If the STPA analysis raises the safety and security constraints, engineers leverage new capabilities to the system architecture. In this way, the UAF Step 7 generates the security capability as expected–the purpose is to illustrate security assets, security constraints, security controls, security control families, and the measures required to address specific security concerns.

In UAF Step 2 engineers define the capabilities. Capabilities are artefacts that can produce desired effects meeting the goals assigned to each deployment phase of the enterprise. Capabilities can achieve desired effects using ways (activities and behaviors) and means (physical and human resources) under certain conditions to perform enduring tasks. The STPA Step 1 defines the foundation of the analysis, covered here. We understand that we require additional elements (see dashed lines in Figure 3 below) to identify the STPA losses and STPA hazards.

Figure 3 illustrates the STPA elements in the Conceptual Schema for the UAF Enterprise Strategy and Capabilities domain. In the STPA context, the capability can produce undesired effects if not used properly. In this way, we bring the STPA concept of Loss, and Hazard to the UAF Strategic domain, therefore we can represent security as a loss that leads to an undesired effect. A Loss involves something that is valuable to stakeholders, and Hazard is a system state or set of conditions that, together with a particular set of worst-case environmental conditions, might lead to a Loss. The High System-level constraints generates from the Hazards.

The UAF Step 3 covers the remaining STPA Steps (2, 3, and 4). The UAF elements serve as model elements in the architecture views and the relationships. A functional control structure along with the Control Actions, Feedback, and Communication is on the left side of Figure 4, and in the middle is the containment window.

The Operational Performer is a logical agent that is capable to perform operational activities which produce, consume, and process Resources. The Operational exchanges asserts that a flow can exist between Operational Performers. The Operational Element flows between Operational Performers and the Operational Activities produces and consumes the Operational Element that the Operational Performers perform. These Operational Element flows have specific Operational Elements assigned to them. In the STPA context, the Operational Performers are the STPA Controller/Controlled process, and the Information Element represents the STPA control actions, feedback and communication.

The STPA Analysis phase requires adding new elements in the UAFP (see dashed-line in Figure 5) to cover the STPA elements of Step 3 (Hazardous Control Action) and Step 4 (Loss Scenarios). A Control Action (Information Element) issued inappropriately could lead to a Hazardous control action. A Hazardous Control Action is a Control Action provided that, in a particular context and worst-case environment conditions, can lead to a Hazard. The right side of Figure 2 illustrates a Control Action and two Hazardous Control Actions, the first one is not provided when required, and

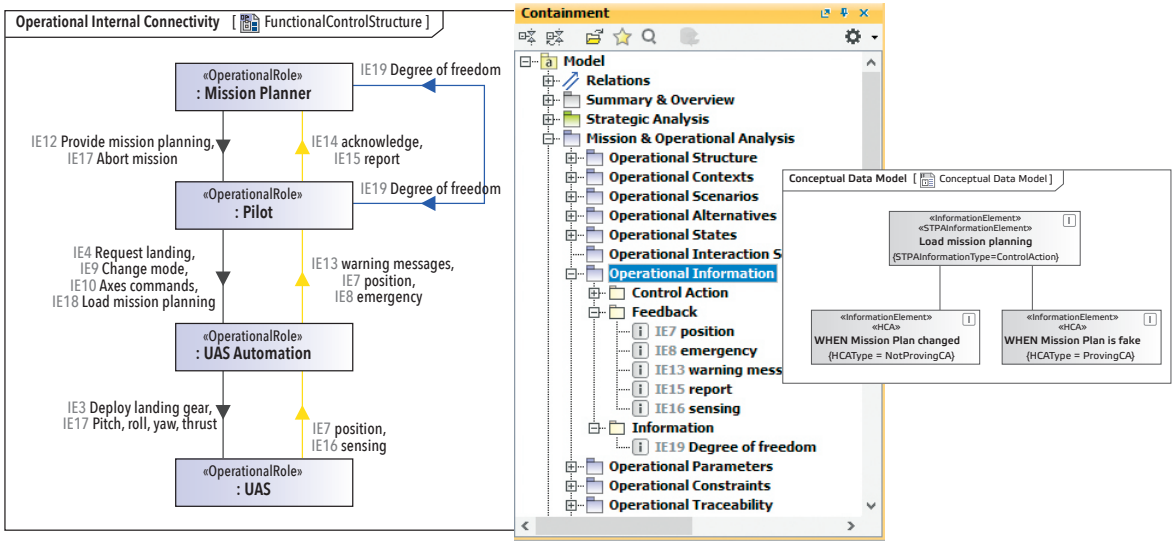


Figure 4. UAF operational connectivity (left) and operational information (right) which serves as control actions, feedbacks and communication in STPA

the second one relates to providing a lead to a Hazardous Control Action.

A Loss scenario leads to a Hazardous Control Action exploited by a causal factor. A Loss scenario has a severity, which is a qualitative indication of the magnitude of the adverse effect of a Loss scenario. An engineer evaluates the Severity in the same manner as a Functional Hazard Analysis, and with the following values: Catastrophic, Hazardous, Major, Minor, and No Effect.

Figure 6 illustrates the new elements in the UAFP that support the generation of Loss scenarios. The Operational Constraint is the only UAF element. The other elements guide the systems engineer to generate the safety and security scenarios. A loss scenario exploits a Causal factor. A causal factor is a STPA element that provides guide words to identify the loss scenarios.

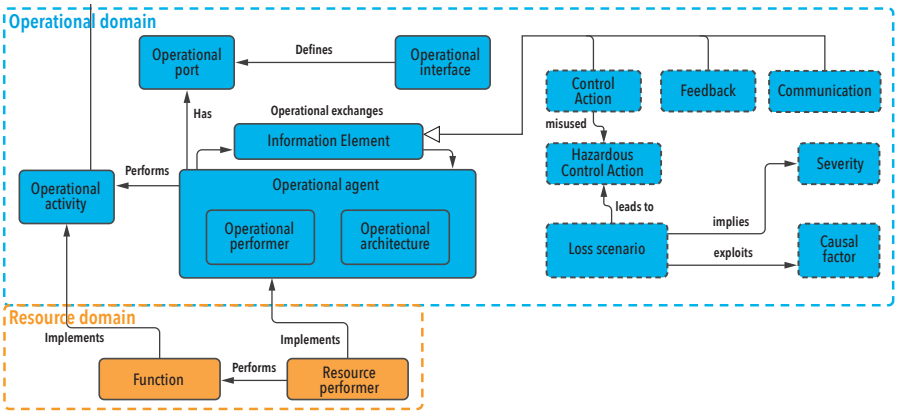


Figure 5. UAF strategic domain extension

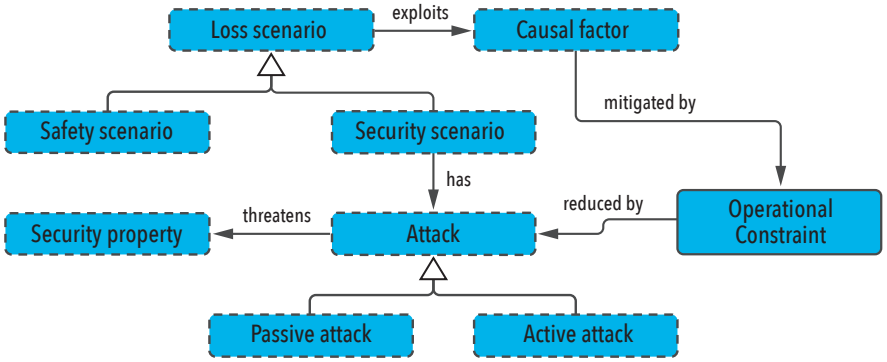


Figure 6. Loss scenario

The Safety scenario and Security scenario are subclasses of the Loss scenario. The Safety scenario covers the unintentional actions that describe how incorrect feedback, design errors, component failures, and other factors can lead to a Hazardous Control Action and Losses. Additionally, the Security scenario covers intentional actions, explaining how an adversary can introduce a control flaw. We added the elements of Security property, Attack, Passive attack, and Active attack based on (Pereira, Hirata, and Nadjim-Tehrani 2019) to support the identification of security scenarios.

In Table 1 we summarize the outcome of this first analysis presented in this article, showing the proposed modifications to integrate STPA analysis into the UAFP and its methodology. Upcoming work has the goal of extending the UAFP with a specific MBSE Profile which can bring the element into the context of the UAF analysis. For this, we used the CAMEO System Modeler and its Profile features (Table 2 next page).

4. SUMMARY

The presented work tackles the above mentioned concepts referenced at the beginning of this paper as Table 3 summarizes on the next page. ■

Table 1. STPA and UAF elements mapping

STPA Element	UAF Element	UAF description (from OMG 2015)
Purpose	Enterprise Goal	A Vision describes the future state of the enterprise, without regard to how it is to be achieved.
Goal	Enterprise Goal	A statement about a state or condition of the enterprise to be brought about or sustained through appropriate Means. An Enterprise Goal amplifies an Enterprise Vision that is, it indicates what must be satisfied on a continuing basis to effectively attain the Enterprise Vision.
N/A	Capability	An enterprise's ability to Achieve a Desired Effect realized through a combination of ways and means (e.g., Capability Configurations) along with specified measures. Capabilities are defined that can produce desired effects meeting the goals assigned to each deployment phase of the enterprise.
Controller / Controlled process	Operational Performers	A logical agent that is capable to perform operational activities which produce, consume, and process Resources.
Control Action / Feedback / Communication	Information Element	An item of information that flows between Operational Performers and is produced and consumed by the Operational Activities that the Operational Performers are capable to perform (see Is Capable To Perform)
Loss Scenario	Operational Interaction Scenario	A specification of the interactions between Operational Performers in an Operational Architecture

Table 2. STPA Elements to be included as part of UAF

STPA Element	UAF Realization	Proposal
<<HazardousControlAction>>	New Stereotype	Conceived as a misuse of Operational Information. The Operational Information which are identified as HCA will have this dedicated stereotype.
Capability::UndesiredEffect	New Attribute	Conceived as the counterpart of Capabilites::DesiredEffect. Capability::UndesiredEffect will be conformed of Hazards, Losses and Constraints. For this purpose, customization of the Class Capability is proposed to extend its attributes.
<<Hazard>>	New Stereotype	Hazard is conceived as a generalization of UndesiredEffect, and to be used to represent Capability::UndesiredEffects. A dedicated stereotype for it will be included in the profile
<<Loss>>	New Stereotype	Conceived as a generalization of UndesiredEffect, and to be used to represent Capability::UndesiredEffects. A dedicated stereotype for it will be included in the profile.
<<Constraints>>	New Stereotype	Conceived as a generalization of the UAF Element Security Constraints, and to be used to represent Capability::UndesiredEffects. A dedicated stereotype for it will be included in the profile.
HazardousControlAction ::LossScenario	New Attribute	The Operational Information with the stereotype <<HCA>> will have the attribute LossScenario. LossScenarios are conceived as a generalization of Operational Interaction Scenarios.
LossScenario::Severity	New Attribute	Attribute of Loss Scenario (Causal factor), which was a generalization of Operational Interaction Scenarios. Conceived as an Enumeration with the following literals: Catastrophic, Hazardous, Major, Minor, and No Effect.

Table 3. Concepts being addressed in this article

1-Security Proficiency in the SE Team	It has been noted that Security is typically seen as a decoupled competence when developing complex system, and that Security is typically left to the persons in charge of such transversal discipline. By bringing into methodologies (such as UAF) more awareness of how other analysis (as STPA) can benefit one from another, and how security analysis can be made available to SE Team bringing security concepts into play together with well-known methodologies, will most likely increase the security proficiency of system engineers and awareness of the importance of security by design and early security development.
5-Architectural Agility	Implementing early modelling and mitigating unwanted situations from the beginning do bring agility when defining the architecture of a system by anticipating to all possible scenarios. When using STPA a good overview of all scenarios and security constraints can be obtained, which will help being more agile when deciding which the functions are considered critical and the architectural components that may host these functions.
7-Capability-Based Security Engineering	Complex systems are made to meet different capabilities. This articles has shown how in UAF Step 2 these capabilities are brought into the design by using UAF Profile and MBSE tools. On top of that, an extension of these capabilities with crucial STPA elements (Losses, Hazards and Constraints) has been presented, therefore having the opportunity of modelling at early stages which are the Undesired Events (either security relevant or not-security relevant) our System of Interest may be affected by.
8-Security as a Functional Requirement	Typically UAF is used during conceptual and early stages of project. By using STPA together with this framework, in which Security concepts are already addressed, serves to analyze all hazardous and unwanted situation as well as the mitigations for them, which could be used to implement security safeguards during development and as a basis for eliciting security functional requirements when developing the systems that conforms the foundations addressed using a UAF analysis.

■ References and Author biographies: continued on page 32

What Is the Role of a Systems Engineer In an Engineering Organization?

Richard Beasley, richard.beasley@incose.net
Copyright ©2022 by Rolls-Royce plc. Published by INCOSE with permission.

ABSTRACT
This article discusses the role of a "specialist" Systems Engineer inside an engineering focused on ensures the engineering parts integrate to achieve the objectives of the whole – and so is embedding a systems approach throughout the organization – trying to "make Systems Engineering the way engineering is done". In this type of organization all engineers need systems engineering as a core skill (which is part of them becoming T-shaped". The specialist Systems Engineer needs to be more π-shaped, with specialism in the systems approach used to inform and guide all the other disciplines which need to be integrated together. Since systems engineering is an integrating discipline the group of systems engineers must not become "just another" technical silo.

Systems Engineering aims to ensure that the pieces of a system work together to achieve the objectives of the whole (INCOSE 2022). Studies show that Effective Systems Engineering increases the probability of successful development of a system (Elm and Goldensten 2012). This is true for both, engineered systems and the system that is used to engineer the systems the organization produces. To effectively engineer systems the pieces of the organization need to fit together into an effective "realisation system," see Figure 1 (Beasley and Pickard 2020). The enterprise level realisation system is a complex adaptive system, with at least as many (if not more) difficulties, challenges and complexity as the systems that are produced.

A common view of Systems Engineering roles is that they act as the glue or integration between other engineering roles (Sheard 1996). It is the author's contention that Systems Engineering is a discipline that an organization cannot add as another discipline or silo into a pre-existing engineering organization. Systems Engineering is an integrating discipline, pulling together the more "traditionally" specialist engineering roles. By itself, Systems Engineering does not produce successful systems – there must be the other engineering disciplines as well. Therefore, in the author's company, the long-term objective is "*to make Systems Engineering the way we do engineering*." This means that the "existing" engineers must be able to understand and participate in Systems Engineering activities. As a result, they provide introductory Systems Engineering and Systems Thinking training to all new graduate engineers entering the company and offer it to all existing engineers. Over 2,000 engineers have had Systems Engineering training in the company over the past 15 years.

This vision faces two important challenges/questions:

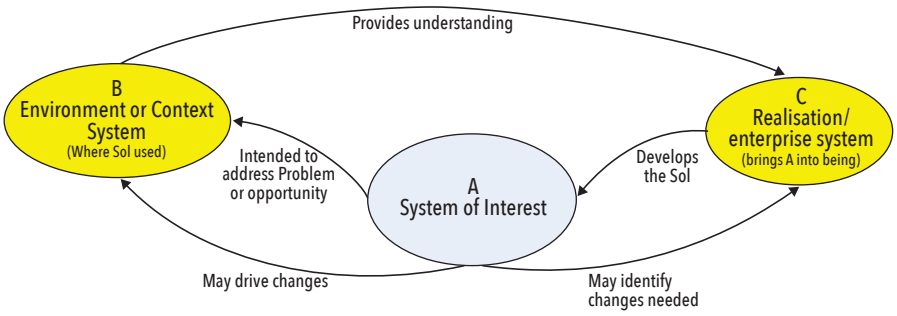


Figure 1. Three systems involved in developing and delivering product (Beasley et al, 2020)

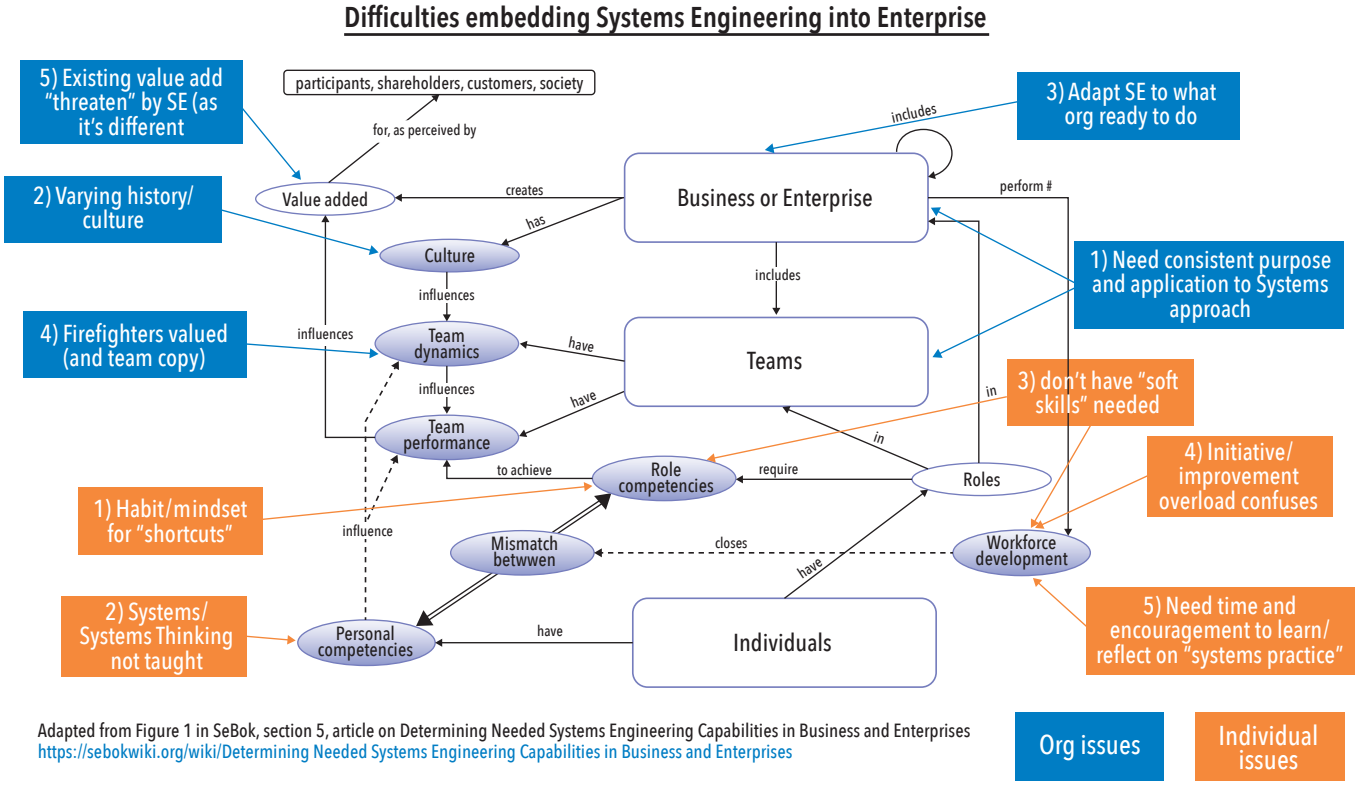


Figure 2. Summary of issues faced implementing systems engineering (adapted from Figure 1 in SEBoK, section 5, article on Determining Needed Systems Engineering Capabilities in Businesses and Enterprises (SEBoK Editorial Board, 2021), from Dunford et al, 2021

1. How do we improve the state of Systems Engineering practice across the whole organization?
2. If everyone is doing Systems Engineering, what is the role for a "specialist" Systems Engineer?

Section 5 of the SEBoK (SEBoK 2021) discusses what is needed in an organization to do Systems Engineering, and Figure 2 (Dunford, Beasley, and Palmer 2021) illustrates the range of problems mapped onto what is considered to be needed in an organization that does Systems Engineering. These problems divide into organization and individual issues. Several of the problems come from the nature of traditional engineering activity – which places strong emphasis and great reward on fighting fires rather than problem prevention. (Beasley, Pickard, and Nolan 2014).

Researchers have explored the problem of implementing Systems Engineering into Rolls-Royce (Dunford et al. 2013), which shows advances are needed in leadership, application, and expertise of Systems Engineering. Figure 3 represents the organization as a dynamic system, and improvement of Systems Engineering comes from leadership pull, meaningful application, and reflection on the application to develop and expand expertise.

This implies that, as stated in the vision "make Systems Engineering the way we do Engineering" all engineers in addition to their prime skill, need Systems Engineering. Rolls-Royce does through the organization processes embedding, Systems Engineering processes, and training all Engineers in Systems Engineering/ Systems Thinking. This helps make them "T-shaped" people. T-shaped is a metaphor to describe the "shape" of someone's skills—in this context the vertical line represents a depth of technical expertise, and the horizontal line the ability to collaborate across disciplines with experts in other areas. Some researchers argue that the "horizontal" collaboration includes the core systems engineering needed in "all engineering," providing skills (in addition to a deep engineering specialism) helping an approach to work together to ensure an optimised whole (rather than a part or single attribute). This leaves the question of the role of a more "specialist"

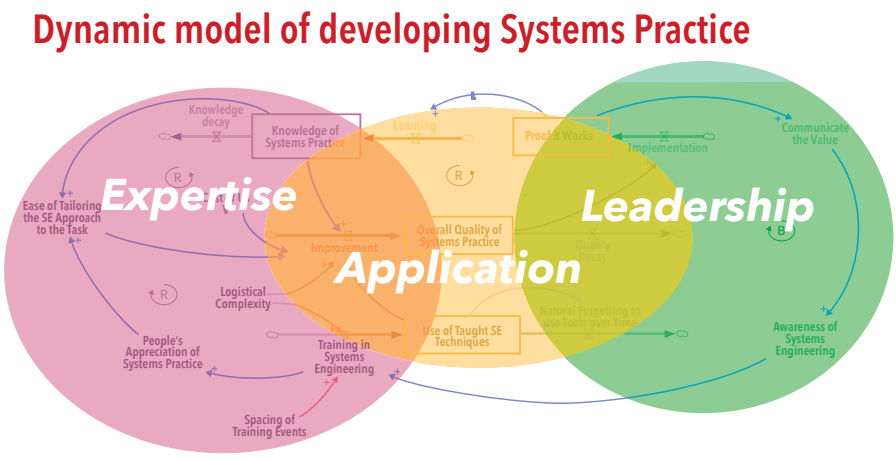


Figure 3. Spiral dynamics model (from Dunford et al, 2013)

systems engineer, which this article discusses below in terms of turning the T into a π-shaped engineer.

In Rolls-Royce the engineering teams generally look after either a specific system of interest (with the whole engine), one of the physical sub-systems (such as the combustor), the integrating "product system" (such as the engine secondary air cooling/sealing system, and oil system), specific components (a turbine blade), or an attribute (such as engine performance or weight) being the typical systems of interest). As well as the specific disciplines and technologies associated with their system of interest, it is important that they understand the insights that Systems Engineering gives in terms of context and understanding of the whole. This means it is considered important that the designers of a system of interest, those responsible for delivering it, are heavily involved in translating the system needs into requirements – they need to understand them and then deliver a solution that meets them. The specialist Systems Engineer provides a strong degree of specialist leadership in requirements elicitation and analysis. This emphasis on requirements should not result in the Systems Engineer being considered purely a "Requirements Engineer." Requirements are a part of, but not the complete scope, of Systems Engineering, and allowing the Systems Engineering to be seen as the sole "owners" and specialists in requirements would go against the "make Systems Engineering the way Rolls-Royce does Engineering" vision.

So: where does that leave the Systems Engineers?

There are two Systems Engineering roles in Rolls-Royce

1. Systems Engineering experts who have a deep specialism in the Systems Approach, Process, and Methods. People in this role provide expertise to support engineers in the project (via the Project Systems Engineers (described below). Additionally, they spend time developing Engineering/Systems Engineering capability (contributing to System C shown in Figure 1), and coaching/mentoring engineers to help develop the levels of expertise (the left-hand loop shown in Figure 3)
2. Project Systems Engineers (PSEs) embedded in project teams lead/facilitate the application of specific Systems Engineering practices in the project teams – both business and technical. This role involved being a full member of the project technical team and ensuring that effective Systems Engineering is both planned (working with the Program management) and executed. These aspects include:
   - facilitating Systems Thinking and Requirements analysis to understand the problem.
   - capturing/extending that understanding in Model-Based Systems Engineering
     - importantly, this modelling focuses on the enterprise level (using the Unified Architecture Framework (UAF), and working not only to understand end-user operation and the context for the Rolls-Royce product but to influence the end user solution with the Rolls-Royce power system capability
   - Taking the modelling into the definition of valid and complete requirements, solution architecture, and verification plans.
   - management/organization of the product data.
   - Ensuring that activities address the impacts of maturation of understanding.

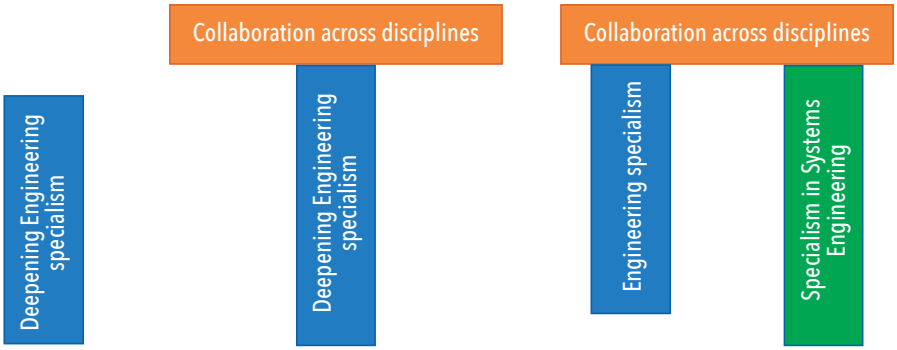


Figure 4. Narrow, T, and π-shaped engineers

These roles fulfil the Facilitator and Expert activities described in Annex C of the Systems Engineering Competency Framework (INCOSE 2018).

The Project Systems Engineer is the important interface between the product, technical engineering, and the program management/business layer. The important overlap between Systems Engineering is well recognized, (INCOSE UK 2020) is one of many examples, and the overlapping activities described in output from joint international working group with the Project Management Institute (PMI) (Rebentisch 2017). This does not make the Project Systems Engineer a variant of a Program Manager or diminish their technical engineering status – although it does open a pathway to a future career in Program Management or as a Project Executive. At its simplest, this responsibility is to make sure that the planning and organization of the System Development work recognises and includes the appropriate activities needed by the Systems Approach, and activities required by the uncertainties and risks identified by those activities. However, there is much more than that. A key aspect of a Systems Engineering approach is the recognition and integration of the needs of all the stakeholders for any given system of interest. For a technical product (such as what Rolls-Royce makes) it is too easy to focus only on the technical needs of the end users and the platform into which the Rolls-Royce power system goes. The Rolls-Royce business is a key stakeholder in the requirements for the product.

The Product Systems Engineer role is involved in helping to understand the business layer, and extracting the business stakeholder needs from that layer, to provide coordinated and consistent flow down into the product, service, and programme requirements. Once completed, there is still the work of recognising emergent technical risk as the product development matures understanding of the problems, including identification of assumptions and emerging concern.

In summary, this makes a Systems Engineer special, within the Engineering team. The Systems Engineering roles require "π-shaped" people as illustrated in Figure 4.

Being π-shaped implies that those in explicitly Systems Engineering roles have a technical engineering specialism and the connection between disciplines implicit in a "T-shaped" engineer, but they add a deeper Systems Engineering expertise and mindset (to enable the engineering teams to do Systems Engineering). Systems engineers become specialist generalists capable of working with a wide range of specialists and across a range of domains. The specialism becomes a focus on "how" engineering

is done, and focused on integrating the technical silos, and ensuring/helping them to work together.

It is important to note that the specialist systems engineers need to remain technical engineers. That technical engineering element of a systems engineer is vital for three reasons

a) Credibility with the other engineers
b) Ability to interpret/understand the technical details coming from the specialist engineers
c) Ability to understand the insights/issues identified in Systems Engineering and communicate these (as "the next question") back to the rest of the engineers, and the program management areas so that all gain and use the insight coming from the Systems Engineering prework that prevents problems.

Systems Engineering provides a vital input to ensure that product development is successful. Good Systems Engineering practice increases the probability of successful system development — allied to all the "traditional" disciplines of technical engineering. The specific challenge for Systems Engineers is that Systems Engineering is a transdisciplinary and integrative approach, and so different from the traditional, more "siloed" engineering disciplines. ■

REFERENCES

- Beasley, R., A. Pickard, and A. Nolan. 2014. "When "Yes" is the Wrong Answer." Paper presented at the 24th Annual International Symposium of INCOSE, Las Vegas, US-NV, 30 June–3 July.
- Beasley, R., and A. Pickard. 2020. "The Capability to Engineer Systems is a System Itself!" *INCOSE International Symposium* 30 (1): 1153-1168. https://doi.org/10.1002/j.2334-5837.2020.00778.x
- Dunford, C., R. Beasley, and E. Palmer. 2021. "Social Science Solutions for the Systems Engineer: What's Needed?" *INCOSE International Symposium* 31 (1): 699-712. https://doi.org/10.1002/j.2334-5837.2021.00863.x
- Dunford, C. N., M. Yearworth, D. M. York, and P. Godfrey. 2013. "A View of Systems Practice: Enabling Quality in Design." *Systems Engineering* 16 (2): 134-151. https://doi.org/10.1002/sys.21220
- Elm, J., and D. Goldenson. 2012. "The Business Case for Systems Engineering Study: Results of the Systems Engineering Effectiveness Survey" http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=34061.
- INCOSE. 2018. "Systems Engineering Competency Framework." https://www.incose.org/products-and-publications/competency-framework.
- INCOSE. 2021. "Systems Engineering Vision 2035." https://www.incose.org/about-systems-engineering/se-vision-2035.
- INCOSE UK. 2020. "Z11 Systems Engineering and Program Management." Z11_issue1.2_2020update_affinity_file (incoseuk.org).
- Rebentish, E. 2017. *Integrating Program Management and Systems Engineering: Methods, Tools, and Organizational Systems for Improving Performance*. Hoboken, US-NJ: Wiley.
- SEBoK. 2021. "Enabling Systems Engineering." https://www.sebokwiki.org/wiki/Enabling_Systems_Engineering#:~:text=Part%205%20of%20the%20Guide,described%20elsewhere%20in%20the%20SEBoK.
- Sheard, S. A. 1996. "Twelve Systems Engineering Roles." Paper presented at the Sixth Annual International Symposium of INCOSE, Boston, US-MA, 7-11 July.

ABOUT THE AUTHOR

Richard Beasley joined Rolls-Royce in 1986 with a Physics Degree from Bristol University, and an MSc in Gas Turbine Engineering from Cranfield University. After working on Integration Aerodynamics, Safety, Reliability and Life Cycle Engineering, he became Global Chief of Systems Engineering. In 2011 he became Rolls-Royce Associate Fellow in Systems Engineering. He was part of the BKCASE SEBoK author team, a leading author on the INCOSE SE competency framework, is a Past-President of the UK INCOSE Chapter, and is currently Services Director for INCOSE. He is a Chartered Engineer, Fellow of the Royal Aeronautical Society, INCOSE ESEP, and was a Visiting Fellow to the Systems Centre at Bristol University.

[Editor: Author biography was current when the paper was initially published in 2022.]

García and Pereira continued from page 27

REFERENCES

- INCOSE. 2014. A World in Motion —Systems Engineering Vision 2025.
- Leveson, N. 2011. "Engineering a Safer World: Systems Thinking Applied to Safety" Boston, US-MA: MIT Press.
- Leveson, N. and Thomas, J. 2018. "STPA Handbook," https://psas.scripts.mit.edu/home/, viewed on January 25, 2022.
- Mažeika, D. and Butleris, R. 2020a. "MBSEsec: Model-Based Systems Engineering Method for Creating Secure Systems." Appl. Sci.10, 2574.
- Mažeika, D. and Butleris, R. 2020b. "Integrating Security Requirements Engineering into MBSE: Profile and Guidelines" Hindawi, Security and Communication Networks Volume 2020, Article ID 5137625.
- Object Management Group 2015. *UAF Specification Business Motivation Model Version 1.3*, https://www.omg.org/spec/BMM/1.3/PDF, viewed on January 25, 2022.
- Papke B. L. 2017. "Enabling design of agile security in the IOT with MBSE," 12th System of Systems Engineering Conference (SoSE), pp. 1-6.
- Pereira, D.P & Hirata, C. and Nadjm-Tehrani, S. 2019. "A STAMP-based Ontology Approach to Support Safety and Security Analyses." *Journal of Information Security and Applications*, 47: 302-319, ISSN 2214-2126.

ABOUT THE AUTHORS

Juan José López García is an Aerospace Systems Engineer with over 5 years of experience working in military aircraft development projects and systems engineering tasks, focusing on requirements management, architecture design, and MBSE. He is currently working as a cybersecurity architect on behalf of Airbus Defence and Space.

Daniel Patrick Pereira obtained a Computer Engineering degree from Santa Cecilia University in 2003, a M.Sc. in Computer Engineering focus on embedded software in 2009 from Federal University of Amazonas, and a Doctoral in Electronic Engineering and Computing from the Aeronautics Institute of Technology in 2020. He has worked as systems security engineer and compliance expert for commercial aircraft companies in Brazil and Japan. Currently, he works as cybersecurity architect for Airbus Defence and Space.

[Editor: Author biographies were current when the paper was initially published in 2022.]

Systems Skills…From Here to Diversity

Alan Harding, alan.harding@incose.net
Copyright ©2022 by Alan Harding. Published by INCOSE with permission.

■ **ABSTRACT**
Competency, the ability to do things, is at the heart of how systems engineers realise successful systems. Over the years INCOSE and partners have codified what this means, initially in the INCOSE UK Competency Framework (INCOSE UK 2010) later adopted globally by INCOSE and used as the basis of the INCOSE Systems Engineering Competency Framework (INCOSE 2018) which, notably, included a new area of professional skills. This article considers competency from the perspective of Diversity, Equity, and Inclusion (referred to as DEI) using a variety of sources including the recently published INCOSE SE Vision 2035 (INCOSE 2022) and a variety of competency frameworks to offer a view of how the skills and competencies of systems engineers need to evolve in the future. Five new competencies are proposed, as are opportunities to improve the definitions of five more.

1. INTRODUCTION

It is now accepted that a well-led diverse team, where every member is included, respected, and valued, delivers improved results compared to an equally well-led team that is either less diverse or less inclusive. These results may be either in business terms like profit or in terms of an outcome or system that is more inclusive and hence better suited to its full target audience (Harding and Pickard 2019).

Systems engineers enable the development and sustainment of successful systems, working effectively across the disciplines and engaging widely with stakeholders. Arguably systems engineers are already diverse in outlook, but of course, improvement is always possible. Historically this diverse outlook has spanned the technical disciplines and application domains of systems engineering. In recent years the need to understand the social dimension of systems engineering has been recognised, especially since many systems, and arguably all systems of systems are socio-technical in nature.

Additionally, the context where systems engineering is practiced continues to evolve. Harding and Pickard further recommended that "systems engineering workforce and culture should be at the forefront of diversity and inclusion."

This article explores what is already said about how the abilities of systems engineers should contribute to diversity and inclusion, and it also reflects on what more needs to be said based on the current in social media and other forums. The article will discuss:

- Definitions
- Diversity as highlighted in the INCOSE Systems Engineering Vision 2035
- Diversity in the UNESCO Engineering Report 2021
- Insights from current competency frameworks
- Insight from wider sources on Diversity
- Summary
- Conclusions.

2. DEFINITIONS

INCOSE has adopted a set of definitions of Diversity, Equity, and Inclusion based on those established by ABET (ABET 2021), which are:

- Diversity is the range of human differences, encompassing the characteristics that make one individual or group different from another.
- Equity is the fair treatment, access, opportunity, and advancement for all people, achieved by intentionally focusing on their disparate needs, conditions, and abilities.
- Inclusion is the intentional, proactive, and continuing efforts and practices in which all members respect, support, and value others.

The successful application of these concepts throughout INCOSE and in the wider systems engineering perspective is intended to ensure that every person is comfortable being themselves and fully engages in leadership and all activities consistent with their individual talents, preferences, limitations, and ambitions.

Bringing this notion of "every person" to life, Figure 1 gives a view of the range of characteristics that can differ in each person. The author contends that even the apparently "pale, male and stale" cliché of a privileged and un-diverse white male (coined by NASA administrator Daniel Goldin in 1992) can mask individual characteristics such as caring responsibilities, neurodiversity, invisible health conditions or disabilities, or introversion which makes each of us unique. Hence, this discussion of skills for inclusion can be applied to any team or collective activity and for all situations where a system is being realised.

Reflection on the dimensions of diversity shown in Figure 1 highlights the need for competencies to address diversity and in-

Figure 1. Categorised dimensions of diversity (based on SEBoK original [Ref. 3], adapted from (Harding and Pickard 2019))

clusion as shown in Table 1. In this and the following tables, the mapping is to existing competencies in the INCOSE Systems Engineering Competency Framework (ISECF), together with:

- Proposed New competencies
- Opportunities to improve an existing competency

3. DIVERSITY IN THE SYSTEMS ENGINEERING VISION 2035

The INCOSE Systems Engineering Vision 2035 (INCOSE 2022) is the product of sustained collaborative activity with inputs from across the industry, academia, and government. One of the uses of the vision is to support collaborative efforts to advance the discipline and grow the skill base to meet current and future challenges related to systems development.

This article has reviewed how diversity, equity/equality, and inclusion are represented in the Vision (plus the closely related social aspects) and what insights into systems engineering competencies we can take from this. Figure 2 and Figure 3 shows what was identified by this analysis.

Consideration of Figure 2 and Figure 3 highlights the need for the competencies shown in Table 2 (next page), including four potential new competencies or areas of knowledge.

4. DIVERSITY IN THE UNESCO ENGINEERING REPORT 2021

The recent report "Engineering for Sustainable Development (UNESCO 2021) includes section 2.1 covering Diversity and Inclusion in Engineering.

This notes that "the inherent skills required from engineers are distinctly changing … the need for people with competencies that were previously described as 'soft skills' are increasingly being seen as the 'critical skills' of the future." The INCOSE Systems Engineering Competency Framework, as discussed earlier, recognises this with the inclusion of a set of professional competencies.

Observation from the Dimensions of Diversity	Related ISECF Competency
All systems engineers must have strong communication skills to enable them to engage empathetically with the widest possible range of stakeholders and colleagues.	Communications Emotional Intelligence
We must ensure that all systems engineers have a good understanding of the dimensions of diversity, how they affect peoples' preferences, and to inform how we interact with each other.	**New:** DEI Awareness
We must ensure that all systems engineers have a good understanding of sustainability with relevance to the social aspects which include environmental justice, human health and wellbeing, resource security, education, peace and political stability.	**New:** Sustainable Development Awareness (ecological, economic, social)
We must ensure that systems are conceived and realised that are inclusive for the widest possible stakeholder community.	**New:** Design for Inclusion (Inclusive Engineering)

Word search	Instances
• Diversity/diverse 16 mentions	• Diverse people/workforce/team (4)
• Inclusive/inclusion 1 mention	• Diverse stakeholders (4)
• Equity/equality 1 mention	• Diverse fields/domains (2)
	• Diverse spectrum of societal needs
	• Diverse set of solution alternatives
	• Diverse range of systems parameters (system health)
	• Demand for social equality

Figure 2. Mentions of diversity, equity, and inclusion in the Systems Engineering Vision

Word search	Instances
• Social equality – 23 mentions	• System/product social acceptability (5)
	• Changing global social environments (5)
	• STEM & Social sciences learning, curricula, skills (4)
	• Social needs (UN SDGs) (2)
	• Social responsibility and sustainability in system models
	• Non-engineered social and environmental systems
	• System social and ethical implications
	• Social equality
	• Social aspirations
	• Resolution of social problems
	• Increased reliance on social communication communities
	• PESTEL factors (incl. Social)

Figure 3. Mentions of social aspects in the Systems Engineering Vision

Table 2. Consideration of Systems Engineering Vision 2035

Competency	DEI	Social
Systems thinking (including PESTEL analysis)		Y
Requirement definition (understanding stakeholder needs)	Y	Y
Systems architecting (consideration of all viewpoints)	Y	Y
New: Design for Inclusion (Inclusive Engineering)	Y	Y
New: DEI Awareness	Y	Y
New: Sustainable Development Awareness (ecological, economic, social)	Y	Y
New: Social Science/Social Systems Engineering Awareness		Y

Table 3. Considerations from UNESCO report

Mentioned in UNESCO Report	Related ISECF Competency
Resilience (ability to adapt well in face of adversity or stress)	**Improvement**: Ethics and Professionalism
Agility (ability to react to change)	Planning
Ability to acquire new knowledge	Ethics and Professionalism
Team working	Team Dynamics
Communication	Communications

The subject report continues to say that "Competencies such as resilience, agility, the ability to acquire new knowledge, team working and communication will all become as important, if not more important, than the detailed technical knowledge that has previously been valued in engineering." Table 3 summarises how these aspects relate to ISECF competencies.

5. INSIGHTS FROM CURRENT COMPETENCY FRAMEWORKS

The competency frameworks used to underpin this discussion are as follows:

1. INCOSE Systems Engineering Competency Framework
2. INCOSE Technical Leadership Model
3. UKSPEC – The UK Standard for Professional Engineering Competence and Commitment
4. Atlas: The Theory of Effective Systems Engineers

5.1 INCOSE SYSTEMS ENGINEERING COMPETENCY FRAMEWORK

The INCOSE Systems Engineering Competency Framework (ISECF) (INCOSE 2018) is the most widely recognized description of the knowledge, skills, and experience needed to perform systems engineering. This framework, based on prior work in the UK, is summarized in Figure 4. It comprises competencies described in five themes:

CORE COMPETENCIES

Core competencies underpin engineering as well as systems engineering.

Systems Thinking	The application of the fundamental concepts of systems thinking to systems engineering;
Lifecycles	Selection of the appropiate lifecycles in the realization of a system;
Capability Engineering	An appreciation of the role the system of interest plays in the system of which it is part of;
General Engineering	Foundational concepts in mathematics, science and engineering and their application;
Critical Thinking	The objective analysis and evaluation of a topic in order to form a judgement;
Systems Modeling and Analysis	Provision of rigorous data and informaion including the use of modeling to support technical understanding and decision making.

PROFESSIONAL COMPETENCIES

Behavioral competencies well-established within the Human Resources (HR) domain. To facilitate alignment with existing HR frameworks, where practicable, competency definitions have been taken from well-established, internationally-recognized definitions rather than partial or complete re-invention by INCOSE.

Communications	The dynamic process of transmitting or exchanging information;
Ethics and Professionalism	The personal, organizational, and corporate standards of behavior expected of systems engineers;
Technical Leadership	The application of technical knowledge and experience in systems engineering together with appropriate professional competencies;
Negotiation	Dialogue between two or more parties intended to reach a beneficial outcome where difference exist between them;
Team Dynamics	The unconscious, psychological forces that influence the direction of a team's behavior and performance;
Facilitation	The act of helping others to deal with a process, solve a problem, or reach a goal without getting directly involved;
Emotional Intelligence	The ability to monitor one's own and others' feelings and use this information to guide thinking and action;
Coaching and Mentoring	Development approaches based on the use of one-to-one conversations to enhance an individual's skills, knowledge or work performance.

MANAGEMENT COMPETENCIES

The ability to perform tasks associated with controlling and managing Systems Engineering activities. This includes tasks associated with the Management Processes identified in the INCOSE SE Handbook.

Planning	Producing, coordinating and maintaining effective and workable plans across multiple disciplines;
Monitoring and Control	Assessment of an ongoing project to see if the current plans are aligned and feasible;
Decision Management	The structured, analytical framework for objectively identifying, characterizing and evaluating a set of alternatives;
Concurrent Engineering	A work methodology based on the parallelization of tasks;
Business and Enterprise Integration	The consideration of needs and requirements of other internal stakeholders as part of the system development;
Acquisition and Supply	Obtaining or providing a product or service in accordance with requirements;
Information Management	Addresses activities associated with all aspects of information, to provide designated stakeholders with appropriate levels of timeliness, accuracy and security;
Configuration Management	Ensuring the overall coherence of system functional, performance and physical characteristics throughout its lifecycle;
Risk and Opportunity Management	The identification and reduction in the probability of uncertain events, or maximizing the potential of opportunities provided by them.

TECHNICAL COMPETENCIES

The ability to perform tasks associated primarily with the suite of Technical Processes identified in the INCOSE SE Handbook.

Requirements Definition	To analyze the stakeholder needs and expections to establish the requirements for a system;
System Architecting	The definition of the system structure, interfaces and associated derived requirements to produce a solution that can be implemented;
Design for…	Ensuring that the requirements of all lifecycle stages are addressed at the correct point in the system design;
Integration	The logical process for assembling a set of system elements and aggregates into the realized system, product, or service;
Interfaces	The identification, definition and control of interactions across system or system element boundaries;
Verification	The formal process of obtaining objective evidence that a system fulfills its specified requirements and characteristics;
Validation	The formal process of obtaining objective evidence that a system achieves its intended use in its intended operational environment;
Transition	Integration of a verified system into its operational environment including the wider system of which it forms a part;
Operation and Support	When the system is used to deliver its capabilites, and is sustained over its lifetime.

INTEGRATING COMPETENCIES

This competency group recognizes Systems Engineering as an integrating discipline, joining activities and thinking from specialists in other disciplines to create a coherent whole.

Project Management	Identification, planning, and coordinating activities to deliver a satisfactory system, product, service of appropriate quality;
Finance	Estimating and tracking costs associated with the project;
Logistics	The support and sustainment of a product once it is transitioned to the end user;
Quality	Achieving customer satisfaction through the control of key product characteristics.

Figure 4. Complete listing of competencies in the INCOSE Systems Engineering Competency Framework

Table 4. Consideration of INCOSE Technical Leadership Model

Leadership attribute	Related ISECF Competency
Thinks strategically – includes considering whole range of stakeholders who benefit and are impacted by the full lifecycle of the product. Opportunity to improve definition by highlighting the diverse range of stakeholders who both benefit and are impacted by the system development.	**Improvement:** Technical Leadership
Fosters collaboration – collaboration across and within diverse groups.	Technical Leadership
Communicates effectively – ability to communicate and influence in language suited to diverse groups.	Communications Facilitation
Enables others to be successful – the ability to recognize potential and to enable all members of the team to be successful, including those traditionally overlooked or who do not assert themselves for cultural or personal reasons.	Coaching and mentoring
Demonstrates emotional intelligence – importance is increased when engaging with diverse groups of stakeholders, encountering different cultures, expectations, working practices, and more. Opportunity to improve definition by broadening definition to cover self-awareness, self-regulation, motivation, empathy, and social skills.	**Improvement:** Emotional Intelligence (Goleman 1996)

core, professional, management, technical, and integrating.

This article uses this framework as the basis of its findings/discussion and will map other sources to it. In the main, two questions will be addressed:

1. Which are the key competencies that enable systems engineers to foster and operate well in diverse situations (and to realise inclusive products)?
2. Are there other competencies or refinements to what we have that would help systems engineers achieve even more?

Within the detailed descriptions of these competencies, only one explicitly calls out diversity as intended in this article. This is the "Emotional Intelligence" competency whose description notes that "Systems Engineering involves interacting with many diverse stakeholders." Elsewhere a "diverse range of projects" is referred to, but this relates to a systems engineer having to understand and contribute to various projects.

5.2 INCOSE TECHNICAL LEADERSHIP MODEL

The INCOSE Technical Leadership Model (Godfrey 2016) provides a view of what is required to be a leader in systems engineering, in addition to the practitioner skills required. Six interrelated attributes are identified in the model shown in Figure 5.

The attributes highlighted in green have a direct bearing on diversity and inclusion and are described in Table 4.

5.3 THE UK STANDARD FOR PROFESSIONAL ENGINEERING COMPETENCE AND COMMITMENT

The UK Standard for Professional Engineering Competence and Commitment (Engineering Council 2020) commonly known as UKSPEC, defines the competence and commitment requirements for people to be Professionally Registered as Engineers in the UK, giving a wider view of engineering professional competencies. While this is a UK framework, it is widely recognised internationally and is also typical of professional engineering frameworks from other nations.

Figure 6 shows the five UKSPEC competencies. The competency areas highlighted in green have a direct bearing on diversity and inclusion, as shown in Table 5 and expanded below.

- B1 – "Take an active role in the identification and definition of project requirements, problems and opportunities" – engaging with and understanding the viewpoints and needs of all stakeholders, including those diverse stakeholders who have not traditionally been considered.
- C1 – "Communicate effectively with others, at all levels, in English" – ability to communicate and influence in language suited to diverse groups. This may be in whichever language(s) are required and includes the need to communicate with people who have differing needs to enable comfortable and effective engagement.
- C3 – "Lead teams or technical specialisms and assist others to meet changing technical and managerial needs" – the ability to recognize potential and to enable all members of the team to be successful, including those traditionally overlooked or who do not assert themselves for cultural or personal reasons.
- D3 – "Demonstrate personal and social skills and awareness of diversity and inclusion issues" – includes emotional intelligence; creating, maintaining, and enhancing productive working relationships, resolving conflicts; and understanding and supporting the needs and concerns of others.

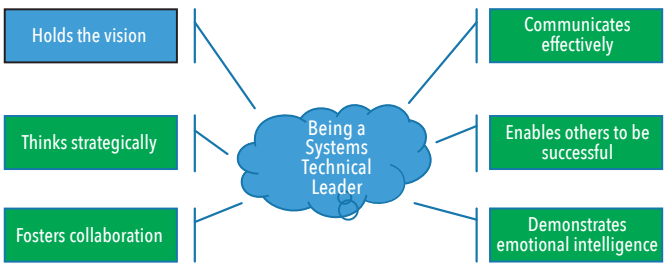


Figure 5. Attributes of a systems leader (INCOSE Technical Leadership Institute) redrawn for clarity

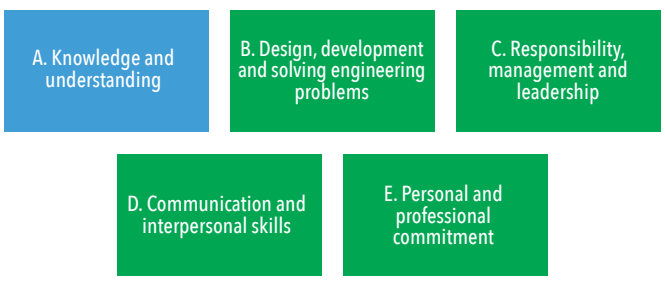


Figure 6. UKSPEC competencies A-E

Table 5. Consideration of UKSPEC competencies

UKSPEC competency area	Related ISECF Competency
B1 "Take an active role in the identification and definition of project requirements, problems and opportunities"	Systems Thinking Technical Leadership Requirements Definition
C1 "Communicate effectively with others, at all levels, in English"	Communications Emotional Intelligence
C3 "Lead teams or technical specialisms and assist others to meet changing technical and managerial needs"	Technical Leadership Team Dynamics Coaching and Mentoring
D3 "Demonstrate personal and social skills and awareness of diversity and inclusion issues"	Communications Emotional Intelligence **New:** DEI awareness
E5 "Understand the ethical issues that may arise in their role and carry out their responsibilities in an ethical manner". Opportunity to improve definition for instance includes applying awareness of UN Sustainable Development Goals, social justice, diversity equity and inclusion.	**Improvement:** Ethics and Professionalism

Table 6. Consideration of Atlas proficiency areas

Atlas Proficiency Area	Related ISECF Competency
1. Math/Science/General Engineering	No direct relevance to DEI
2. System's Domain & Operational Context	No direct relevance to DEI
3. Systems Engineering Discipline	No direct relevance to DEI
4. Systems Engineering Mindset • Big-picture thinking • Paradoxical mindset • Adaptability • Abstraction • Foresight and vision	Systems Thinking **New:** Adaptability
5. Interpersonal Skills • Communication • Listening and comprehension • Working in a team • Influence, persuasion, and negotiation • Building a social network Opportunity to improve definition of "Communications" by referring to diverse audiences, listening and comprehension, and building and maintaining professional/social networks. Opportunity to improve definition of "Negotiation" by also referring to influence and persuasion.	**Improvement:** Communications Team Dynamics **Improvement:** Negotiation
6. Technical Leadership • Building and orchestrating a diverse team • Balanced decision making and rational risk taking • Guiding diverse stakeholders • Conflict resolution and barrier breaking • Business and project management skills • Establishing technical strategies • Enabling broad portfolio-level outcomes Opportunity to improve definition of "Team Dynamics" by referring to diversity and inclusion within teams; and we should consider splitting out conflict resolution as it does not only exist within a team.	**Improvement:** Team Dynamics Systems Thinking Communications

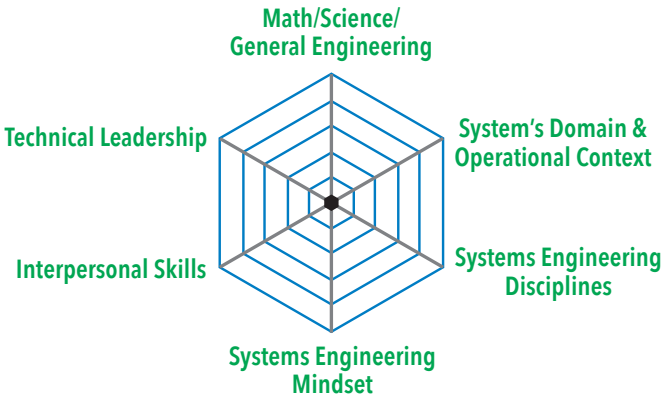


Figure 7 Atlas proficiency areas for systems engineers

- E5 – "Understand the ethical issues that may arise in their role and carry out their responsibilities in an ethical manner" – includes applying awareness of the United Nations Sustainable Development Goals, social justice, diversity equity, and inclusion. This social dimension is growing in importance to systems engineers because often systems are socio-technical and may even include a political aspect.

As well as the specific competencies, the requirement for professional commitment within UKSPEC also calls for registrants to "show that they have adopted a set of values and conduct that maintains and enhances the reputation of the profession" in areas including recognising inclusivity and diversity.

5.4 ATLAS: THE THEORY OF EFFECTIVE SYSTEMS ENGINEERS

The US SE Research Center Helix program created the Atlas model (Hutchison et al. 2018) of "general principles and ideas that relate to the subject of what makes systems engineers effective and why." It talks in terms of the proficiencies of systems engineers, where proficiency is defined as "the quality or state of knowledge, skills, abilities, behaviors, and cognition." Figure 7 shows the six Atlas proficiency areas.

Reflection on the Atlas proficiency areas shown in Figure 7 highlights the need for competencies to address diversity and inclusion as shown in Table 6. The Atlas Proficiency Area number 4 is a key aspect for systems engineering in a diverse and changing context and is a strong contender to pull through from Atlas into ISECF, noting the Atlas definition: "The overall ability to deal with ambiguity and uncertainty, this involves the abilities to be open-minded, understand multiple disciplines, deal with challenges, and the ability to take rational risks."

6. INSIGHTS FROM WIDER SOURCES ON DIVERSITY SKILLS

Research for this article has included wider reading including on social media, and discussion with colleagues familiar with the field of DEI. Although not comprehensive, this has given a good sense of what is being highlighted as valuable when building and operating teams that are inclusive and hence are environments for a diverse range of individuals to thrive and give of their best.

- Systems Thinking – given the range of differences that people may have from one wwwanother (see Figure 1) both systems thinking and perspective taking are vital tools to understand and make sense of situations. It is also important to recognise and test any assumptions made, as these could risk stereotyping people.
- Empathy – much stress is given to the importance of empathy (the ability to understand and share the feelings of another) and both having and showing genuine interest in others to learn something new from them and build open and trusting relationships.
- Culture – it is clearly important to recognise cultural differences and to understand how to work together appreciating and recognising these differences in groups as well as in individuals. These could be aspects such as recognising different national and religious holidays, traditions around alcohol and mixing between sexes, or aspects such as differences in decision-making, leading, and communicating (Meyer 2014). This must be done, of course, while avoiding stereotyping people.
- Communications – it is vital to communicate inclusively, for instance listening with full attention, giving space and time to those who need longer to express themselves (those speaking in languages other than their own, those with speech/hearing difficulties, those needing a translator or person to sign for them). Also choosing a style, tone, and language to be as helpful as possible to the audience such as preferring simple language to jargon, speaking slowly as a courtesy to those less familiar with the language, and offering simultaneous translation/sub-titles to online audiences. It is also important to understand cultural differences regarding communication (women not wanting to challenge authority, different cultures not habitually asking questions).
- Psychological Safety – the need to ensure psychological safety within the team, thus allowing everyone to "be able to show and employ oneself without fear of negative consequences of self-image, status or career" (Kahn 1990). This includes confronting bias, any form of exclusion or intimidation, and mitigating microaggressions which can require significant courage.
- Delegation, Coaching and Mentoring – various sources highlight the positive contribution to inclusion that effective delegation, as well as coaching and mentoring can provide. Taking the chance to empower a less obvious candidate, perhaps combined with coaching and support could make a significant change to both an individual and the team dynamics. All feedback should be honest and fair and needs to draw heavily on the cultural awareness mentioned earlier to be received constructively. Mentoring can also be two-way, for instance in reverse mentoring methods.
- Meetings – it is stressed how important it is to run inclusive meetings, ensuring a fair chance for all to express themselves and managing the agenda to allow appropriate amounts of discussion. This is a way of accommodating individuals with markedly different seniority and status, communications styles, and confidence levels. This can be made easier or harder when using online communications which have become so prevalent during the Covid pandemic.

7. SUMMARY

This article has reviewed a range of sources on diversity and inclusion and has identified many insights into the competencies that systems engineers will need now and in the future. This section draws those insights together, mapping them back to the INCOSE Systems Engineering Competency Framework as shown in Figure 8, where competencies are labelled as follows:

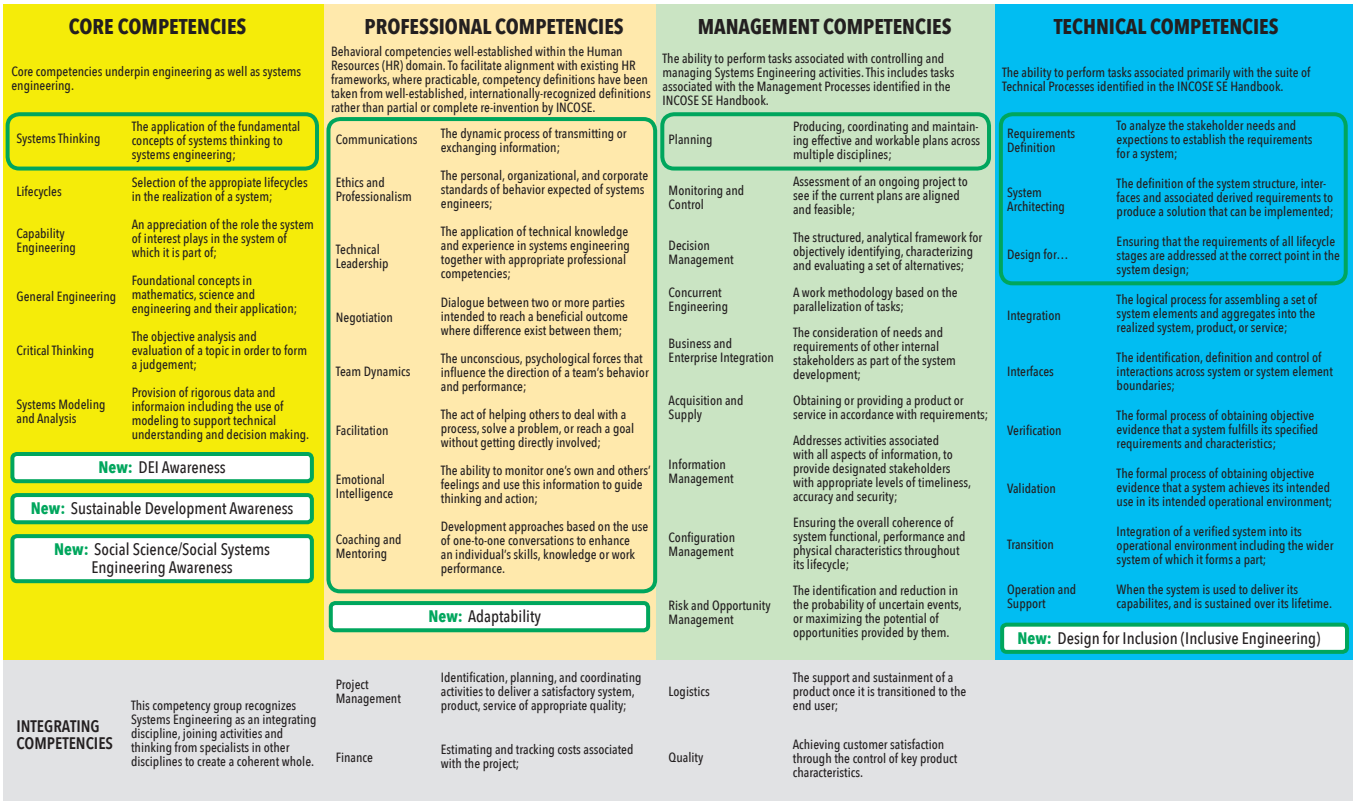


Figure 8 Modifications to the INCOSE Systems Engineering Competency Framework identified by this article

Table 7. New and improved competencies

New Competencies	Improved Competencies
DEI Awareness	Ethics and Professionalism
Sustainable Development Awareness	Technical Leadership
Social Science/Social Systems Engineering Awareness	Negotiation
Adaptability	Team Dynamics
Design for Inclusion (Inclusive Engineering)	Emotional Intelligence

- New (labelled as new)— additional competency suggested by the article.
- Improvement (labelled with an "I")— competency is relevant to diversity and inclusion and the article has identified an opportunity to improve its definition.
- Relevant (outlined in green)— competency is relevant to diversity and inclusion with no change.
- The remaining competencies have not been considered as particularly relevant to diversity and inclusion in this article.

The key conclusions of this article are as follows:

Core Competencies

- Systems Thinking is highly relevant to DEI because it imparts the ability to discern and reason about multiple systems, including socio-technical, cultural, political, and more.
- Three new core competencies have been identified, positioned as core because they should underpin both engineering and systems engineering in the future.

Professional Competencies

- All these competencies are relevant to DEI – this is not a surprise as these all relate to both the individual systems engineer and engaging/leading/serving other stakeholders.
- Insights identified in this article suggest the opportunity to improve definitions of five of these competencies, to better highlight what is needed to foster and operate in diverse and inclusive situations.

Technical Competencies

- This article has highlighted the relevance of the three technical competencies that sit "earlier" in a traditional system lifecycle
- Design for Inclusion (Inclusive Design) has been proposed as an addition to the "Design For …" competency to widen the mindset of systems engineers when helping realise systems for necessarily diverse stakeholder communities.
- While this article did not address it, the author recognises that validation, operation, and support are also likely to be affected by diversity and inclusion considerations.

Management Competencies

- The UNESCO report recommends that engineers will need to be more agile in reacting to change, which is mapped to the Planning competency.

Integrating Competencies

- No specific mention of these competencies in this analysis.

8. CONCLUSIONS

This article set out to offer a view of how the skills and competencies of systems engineers need to evolve in the future. The INCOSE Systems Engineering Competency Framework remains a good basis to describe what knowledge, skills and behaviours are needed by a systems engineer.

Based on the consideration of DEI factors in this article, Table 7 lists the new competencies that have been identified, and the opportunities to improve the definitions of five more. The competency area where most change is proposed is the "Professional" competencies, which seems reasonable given the interpersonal and social emphasis of DEI.

As a "sense-check," the most frequently mentioned competencies (existing and new) in this article are lister below in rank order. They provide a good summary of the critical areas where systems engineers, and those who lead and develop them, will need to focus on the future in order to fully complete the journey "from here to diversity:"

1. Communications
2. Requirements Definition
3. Technical Leadership, Emotional Intelligence, DEI Awareness (New) ■

REFERENCES

- ABET. 2021. "Diversity, Equity & Inclusion." https://www.abet.org/about-abet/diversity-equity-and-inclusion/.
- Engineering Council. 2020. *UK Standard for Professional Engineering Competence and Commitment (UK-SPEC) 4th edition*. London, UK: Engineering Council. https://www.engc.org.uk/EngCDocuments/Internet/Website/UK-SPEC%20fourth%20edition.pdf.
- Godfrey, P. 2016. "Building a Technical Leadership Model." *INCOSE International Symposium* 26 (1): 757-772. http://dx.doi.org/10.1002/j.2334-5837.2016.00191.x
- Goleman, D. 1996. *Emotional Intelligence: Why it Can Matter More Than IQ*. London, UK: Bloomsbury.
- Harding, A. D., and A. Pickard. 2019. "Towards a more Diverse INCOSE." *INSIGHT* 22 (3): 14-20. https://doi.org/10.1002/inst.12254
- Harding, A. D., and A. Squires. 2022. "Diversity, Equity, and Inclusion." SEBoK, 20 May. https://www.sebokwiki.org/wiki/Diversity,_Equity,_and_Inclusion.
- Hutchison, N., D. Verma, P. Burke, M. Clifford, R. Giffin, S. Luna, and M. Partacz. 2018. *Atlas 1.1: An Update to the Theory of Effective Systems Engineers*. Hoboken, NJ: Systems Engineering Research Center, Stevens Institute of Technology.
- INCOSE. 2018. "Competency Framework." https://www.incose.org/products-and-publications/competency-framework.
- ———. 2021. "DEI-100: Diversity, Equity, and Inclusion." https://www.incose.org/docs/default-source/policiesbylaws/policy_dei-100-_-diversity-equity-and-inclusion.pdf?sfvrsn=3ba98c6_6.
- ———. 2022. "Systems Engineering Vision 2035." https://www.incose.org/about-systems-engineering/se-vision-2035.
- INCOSE UK. 2010. "Systems Engineering Competencies Framework." https://incoseuk.org/Documents/Groups/UKChapter/SE_Competencies_Framework_Issue_3.pdf.
- Kahn, W. A., 1990. "Psychological Conditions of Personal Engagement and Disengagement at Work." Academy of Management Journal 33 (4): 692–724.
- Meyer, E. 2014. *The Culture Map: Breaking Through the Invisible Boundaries of Global Business*. New York, US-NY: Public Affairs.
- UNESCO. 2021. "Engineering for Sustainable Development." https://en.unesco.org/reports/engineering#:~:text=By%20presenting%20case%20studies%20and,energy%2C%20responding%20to%20natural%20hazards%2C.

ABOUT THE AUTHOR

Alan Harding is a past-president of INCOSE and INCOSE UK and is a systems engineer with over 37 years' experience. He is a Global Engineering Fellow with BAE Systems Digital Intelligence, a Chartered Engineer, a Fellow of the IET, and a Freeman of the Worshipful Company of Engineers. During 2020 Alan led the activity to establish INCOSE's first ever policy on Diversity, Equity and Inclusion, DEI-100.

[Editor: Author biography was current when the paper was initially published in 2022.]

TNO-ESI – Systems Engineering Methodologies for Managing Complexity in the High-Tech Equipment Industry: Our Roadmap

Wouter Leibbrandt, wouter.leibbrandt@tno.nl; **Jacco Wesselius,** jacco.wesselius@tno.nl; and
Frans Beenker, frans.beenker@tno.nl
Copyright ©2022 by Wouter Leibbrandt, Jacco Wesselius, and Frans Beenker. Published by INCOSE with permission.

■ **ABSTRACT**
The high-tech equipment industry brings complex industrial products to the market with high speed, enhanced functionality, a better cost-performance ratio, and greater integration into customer workflows. Driven by digitalization, the complexity of these systems continues to grow steeply. To manage this complexity, continuous innovation in systems engineering methodologies is needed. TNO-ESI targets to 1) create impactful and industrially applicable systems engineering methodologies and 2) provide innovation support to the industry to get these applied in an industrial context. The ESI research program is defined through a roadmapping process that follows two tracks: a roadmap that maps industry needs and related research and development requirements and a roadmap that describes the developments in the expertise areas necessary for addressing these industry needs. In this paper, we describe the ESI mission, our way of working and activities, and explain the roadmapping process and the roadmaps.

INTRODUCTION TO ESI – WHO ARE WE

TNO-ESI (Embedded Systems Innovation, ESI for short) (www.esi.nl) is an open innovation research center with strong partnerships with industry-leading high-tech equipment companies and strong associations with fundamental research of academia (both nationally and internationally). As ESI, we are part of the Netherlands Organisation for Applied Scientific Research, referred to as TNO or the TNO-ESI. By developing new systems design and engineering methodologies, we address the ever-increasing complexity the high-tech equipment industry faces in the systems it creates and maintains throughout the entire lifecycle. We are about managing complexity. Our research program aims to advance the high-tech equipment industry by improving the lead times and effectiveness of their product innovation processes and their products' functionality, quality, and societal impact. We contribute through a robust research program, dedicated innovation support, a focused competence development program, and various knowledge- and experience-sharing activities. We create impact by turning the latest insights in systems engineering methodologies into practice in the harsh reality of the industry.

THE DUTCH HIGH-TECH EQUIPMENT INDUSTRY

The Dutch high-tech equipment industry is developing world-class systems for diverse business markets. Along with other application domains, they focus on systems and equipment for the semiconductor industry, healthcare imaging, professional production printing, electron microscopes, warehouse automation, and combat management systems. These companies have much in common despite their apparent differences in markets and application domains. They all target the high end of their respective markets, serving an international customer base. They all make highly complex systems in relatively low numbers – typically hundreds per year and sometimes even fewer. And all systems operate in the field for a long time – twenty years is no exception. Finally, these companies also share a business driver: to digitalize their products and solutions.

This industry also commonly recognizes the advantage of joint innovation, particularly in research and development, to man-

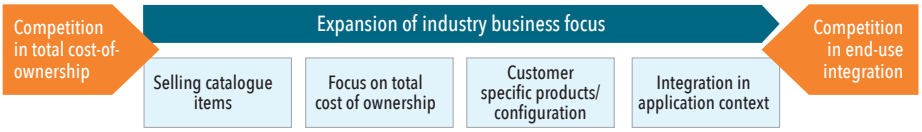


Figure 1. Expansion of industry business focus

age the ever-increasing complexity. Being non-competitors, they are very much open to working together. ESI's open-innovation model enables them to learn from each other and stay on top of market developments.

Over the years, the business focus of the industry has gradually expanded. Their initial focus was only on stand-alone (often high-performance) products (catalog items). This has become less and less a viable business proposition. As a basis, products must provide the required functionality, performance, and a competitive total-cost-of-ownership (TCO) with strict product quality and reliability requirements. However, they must increasingly be adaptable to individualized customer needs, provide flawless integration and cooperation with other products and end-customer processes and applications, and preferably be sufficiently future-proof to accommodate continuously changing operating requirements during their lifecycle; see Figure 1. Digitalization is the driver and enabler, bringing new complexity and system dynamics challenges.

DIGITALIZATION CHANGES BUSINESS

Product innovations are pre-dominantly realized as a complex multi-disciplinary interplay of software and physical components (– cyber-physical systems). Value, cost, and complexity have shifted increasingly into the software. Leveraging digital technologies has become a key engineering competence for the high-tech equipment industry (without reducing the need to be competent in managing complex physical technologies). Digitalization has brought new opportunities, challenges, and market expectations to the industry, including a demand for regular updates and upgrades.

CREATING IMPACT AND OUR WAY OF WORKING

The mission of ESI is to impact the Dutch high-tech equipment industry by embedding cutting-edge systems engineering methodologies to cope with their products' ever-increasing complexity. This mission defines our activities and our way of working. Methodologies here are meant as consisting of formalisms, techniques, methods and tools as explained in (Heemels 2007).With newly developed knowledge of methods, we target individual products or applications, foster synergies, and share and exchange methodologies and knowledge in an open-innovation setting. For successful innovation and value take-up by the industry, systematic attention must be given to all required elements of the knowledge chain. This consists of the following:

- agenda setting and programming (translating industrial challenges into a research program),
- applied research (executing the research program in cooperation with industry and academia),
- consolidation (knowledge base for general use, professionalized tooling),
- dissemination (presenting, sharing, discussing, demonstrating results, enabling service providers to apply our results), and competence development.

We have found that for our applied research to be impactful, it is vital to work on real-world, business-critical industrial challenges. To access the often company confidential information, on-site presence is required. Therefore, we conduct our work at the premises of our industry partners, a way of working called *industry-as-a- lab*. This requires trust that we have developed and nurtured over the years with our industry partners, with whom we execute applied research projects which may span several years.

POSITIONING ESI'S RESEARCH

The positioning of our research is depicted in Figure 2. We have carefully chosen the meta-2 level. This means that ESI does not create products (meta-0); this is the work of our industrial partners by applying, among others, systems engineering and system architecting methodologies (meta-1). Our focus is on delivering innovations in the methodologies required for industrial systems engineering practices in their product realization. This creates focus and opportunities to fully exploit the collaboration with our industry partner, their suppliers, and our academic partners. Now and then, with our peers, like the centers presenting themselves in this volume, we reflect on how to organize and conduct such research (meta-3).

Another axis is to look at technology-ready levels (TRL). Our emphasis is on TRL 4-7. Higher TRLs are addressed by industry, service providers, and tooling companies, while we leave fundamental research at TRL 1-3 to our academic partners.

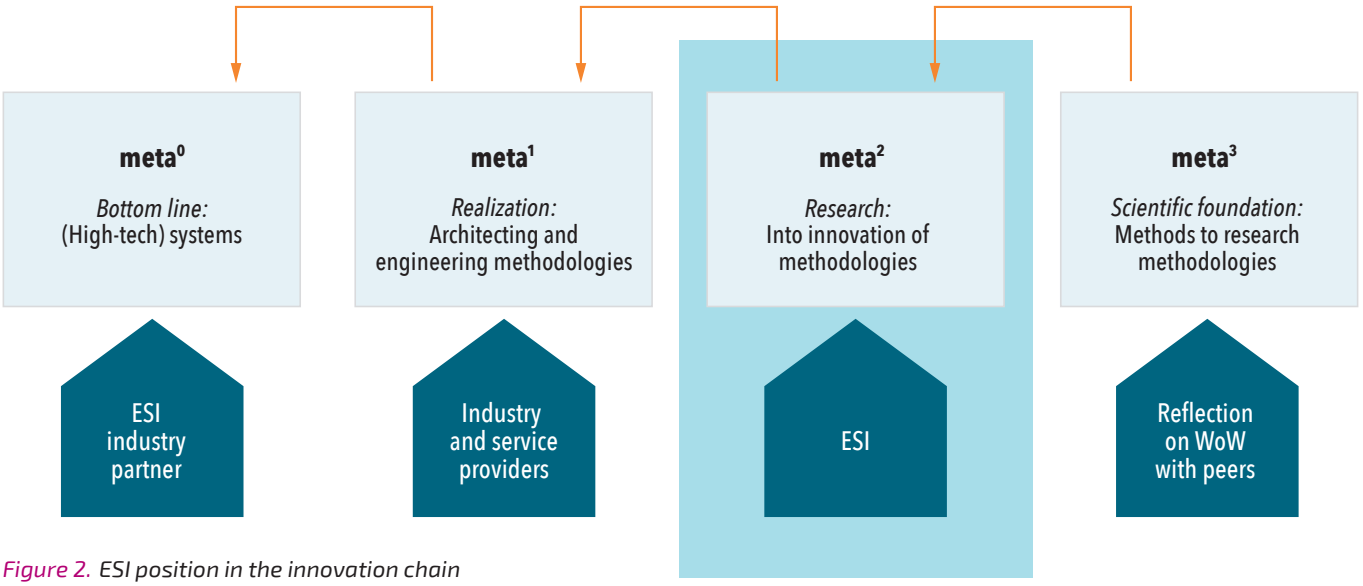


Figure 2. ESI position in the innovation chain

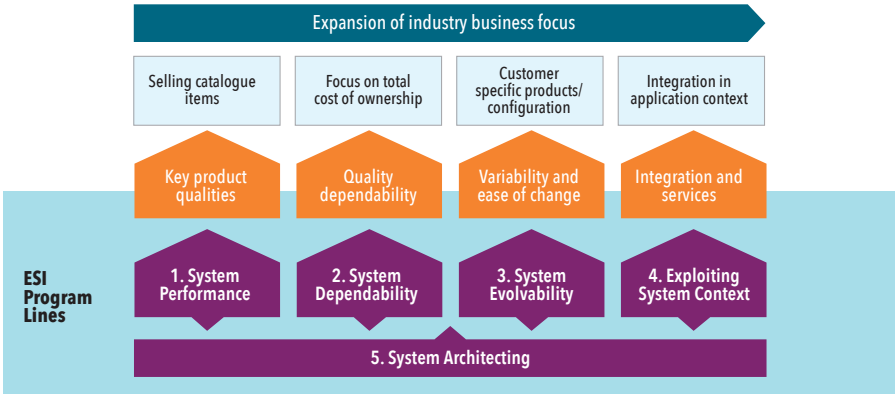


Figure 3. ESI program lines

PROGRAM LINES

During the 20-year existence of ESI, the focus of our research evolved, following the needs of the industry. This resulted in 5 research program lines, as depicted in Figure 3 (see https://esi.nl/research/program-lines for project results). Four program lines can be coupled to the life-cycle phase of products and businesses, and the fifth support these with methodologies for systems architecting:

- Program Line 1 – **System Performance**
  This program line focuses on the performance of systems in a broad sense: the system provides functions; how well does the system do this? The program line targets methodologies that enable engineers to get control of those system qualities that give value to their systems in the market. Since trade-offs are common in these cases (enhanced throughput might result in reduced output quality), methods are needed to relate the many system qualities and to provide methods to perform trade-off analysis to optimize design choices.

- Program Line 2 – **System Dependability**
  As systems are increasingly business/mission-critical, it is not enough to deliver the correct functionality and performance; it is critical that they also dependably deliver this: unforeseen system failure or performance degradation has to be avoided.

  ESI develops methodologies to address this challenge in the Systems Dependability program line. It also addresses the various lifecycle phases of systems, currently centering around two topics: (i) verification and validation and (ii) system diagnostics.

- Program Line 3 – **System Evolvability**
  Newly released high-tech equipment systems are usually evolutionary improvements of existing systems, in which components evolve at different speeds, software often having the highest evolution and obsolescence speed. There is a need for regular system updates during service in the field that can last multiple decades. That is a challenge and an opportunity: on the one hand, many versions need to be serviced, while the update business can be significant and profitable.

  This program line addresses that in several ways: methods for defining and managing component interfaces to improve system methods to deal with legacy software and systems, and methods for configuration management and architectural modeling of system diversity.

- Program Line 4 – **Exploiting Systems Context**
  Today's high-tech equipment is hardly ever performing its function in isolation. The systems will be integrated into the customer context regarding customer processes/workflows, data exchange, or physical integration. The equipment becomes part of (customer-specific) systems-of-systems.

  This program line focuses on: (i) integrating equipment into systems-of-systems; (ii) systems that are intelligent and that adapt to their operational environment, and (iii) optimizing the performance of equipment integrated into a system-of-systems and optimizing the performance and dependability of systems-of-systems as a whole.

- Program Line 5 – **System Architecting**
  Finally, the System Architecting program line supports the other program lines with research on system architecting methodologies. The program line delivers model-based system architecting methodologies with a strong focus on creating models that link customer and business value to architecting and engineering decisions.

  A special track in this program line is on the value and adoption of MBSE.

ROADMAPPING PROCESS

To ensure that the research of ESI is relevant for the Dutch high-tech equipment industry, we regularly conduct a process to take stock of the needs of the industry. We recently completed this biannual process in the summer of 2022. The approach we took this time was structured as sketched in Figure 4.

In a series of meetings, we discussed two topics with each industry partner of ESI, eight in total, individually:

- Strategic business directions, opportunities, and challenges;
- The key capabilities that are needed to achieve these.

After each of these discussions, we analyzed the outcome, and per partner, we mapped the business and capability needs to the area of (systems) engineering methodologies. A one-page summary was composed per partner containing an overview of (i) business needs; (ii) required capabilities, and (iii) ESI opportunities to

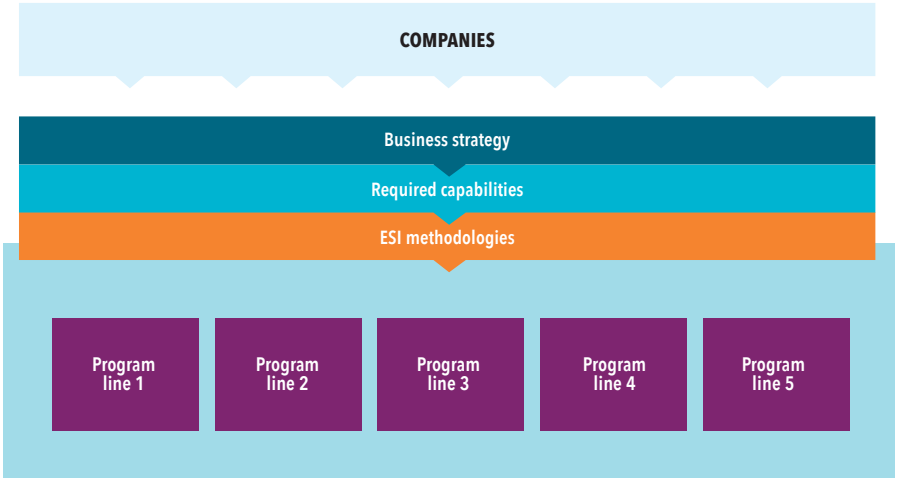


Figure 4. Roadmapping process

support the business with developing and embedding innovative methodologies.

The aggregated picture was created in a subsequent workshop with all industry and academic partners, and we identified common challenges, needs, and priorities.

Thus, we identified ample methodological opportunities for research in the next five years, as the basis for the demand-driven roadmap and as input for the 2023 research program.

ROADMAP

The process sketched above resulted in an updated industry-driven research roadmap of ESI. It combines the continuation of running program lines, shifts of focus in running program lines, and initiatives for new program lines. Having consulted a broad range of leading industries in the Dutch high-tech equipment domain, we are confident that it covers well the needs of the high-tech equipment industries.

Trends and Characteristics

Resulting from this process, we have identified several critical trends and characteristics and needs that follow from that:

i. **Enhanced criticality**: Systems are increasingly critical, and methodologies to assure dependability (also in systems of systems) are key.
ii. **Enhanced diversity:** Customer-specific systems diversity is growing, combined with deeper integration into customer processes and systems of systems, asking for improved methodologies for diversity management and efficient and effective verification and validation of diversified product families.
iii. **Continuous innovation and updating:** The market expects products that are kept up to date. Therefore, methods are needed to ensure easy, dependable updates, guaranteeing system functionality and performance for diversified installed bases. A more agile system engineering method is asked to support such fast innovation.
iv. **Climbing the value chain:** Equipment manufacturers climb up in the value chain, providing higher-level services to customers. By doing this, the scope of their architecting expands quickly beyond their equipment. They need to understand systems beyond their equipment, and they need broader domain knowledge.
v. **High demand for engineering experts:** A general observation is that experts are hard to find. New team members take a long time to maximize productivity and quality. This calls for "democratization."

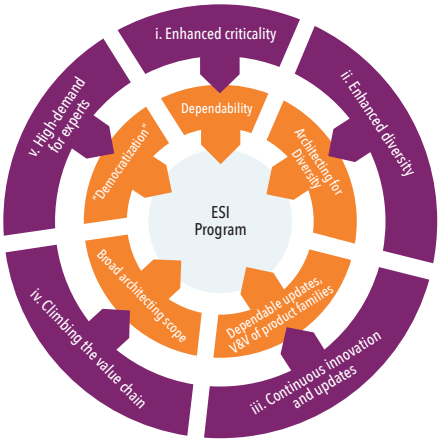


Figure 5. Trends and characteristics in the high-tech equipment industry and the resulting needs

- *Democratization of the systems*: the industry seeks ways to make the operation of its systems less complex by leveraging artificial intelligence, machine learning, and other smart algorithms. This raises a new, related challenge: how to optimally design their systems for AI integration combined with context-sensitive, adaptive system behavior?
- *Democratization of research and development R&D*: making it easier to develop the equipment. The industry seeks methodologies and tools that support the complex tasks of engineering the equipment, for example, bringing AI-based algorithms and other advanced algorithms to the engineers' fingertips.

Roadmap Priorities

Given these trends and characteristics, ESI discussed with its partners the priorities for their research, which led to the following high-priority topics:

- Architectural methods to create more modular systems; methods to design, describe, and enforce system and module interfaces, including inference of legacy interfaces.
- Methods to enhance the efficiency and effectiveness of verifying and validating highly diversified systems and systems-of-systems. This applies for V&V of new system releases but, maybe even more importantly, for system and software updates and upgrades.
- Methods for smart diagnostics and for architecting systems with optimal diagnosability.
- Methods to architect systems for optimal integration of AI/ML in critical high-tech equipment (engineering for AI).
- Methods applying AI to optimize the efficiency and effectiveness of R&D teams (hyper-automation for R&D, AI for Engineering).

Updates to the Program Lines?

Based on these priorities, we have reconsidered our existing program lines:

- The existing five program lines remain relevant for the high-tech equipment industry.
- In these program lines, we will address the opportunities and challenges of AI/ML.
- We expect to see more research in the evolving systems and systems in context programs, focusing on system diversity and systems-of-systems aspects, respectively.
- In the exploiting systems context program line, we expect to work on system adaptivity using AI and ML: how do we realize system adaptivity in high-tech equipment and combine this with the required dependability of critical equipment?
- In the systems architecting program line, we expect to combine these challenges in methods for systems architecting. A special focus is expected for MBSE, digital engineering, and agile (model-based) systems engineering.

Next to these shifts, we see new topics emerging. For these, we will start dedicated studies with our partners to define new research initiatives (in the existing program lines or potentially in a new program line): *system democratization* and *R&D democratization*.

EXPERTISE TEAMS AND ROADMAPS

The research at ESI draws on the knowledge and insights built up at ESI and our partners over the years and is clustered in seven expertise areas:

1. **System performance engineering:** methods and tools to address the performance challenges in system design, which are cross-cutting and often require a holistic view. The approach is primarily model-based, and the current practice is that models support design decisions. Based on a recent field survey conducted by ESI (Van der Sanden 2021), the vision is that models will act as authoritative sources of truth and form the basis for automated synthesis of implementation artifacts supporting optimal system performance over the lifecycle.
2. **Software legacy and rejuvenation:** methods and tools to support the developers of industrial software in understanding their codebases and in (semi-)automatically improving them

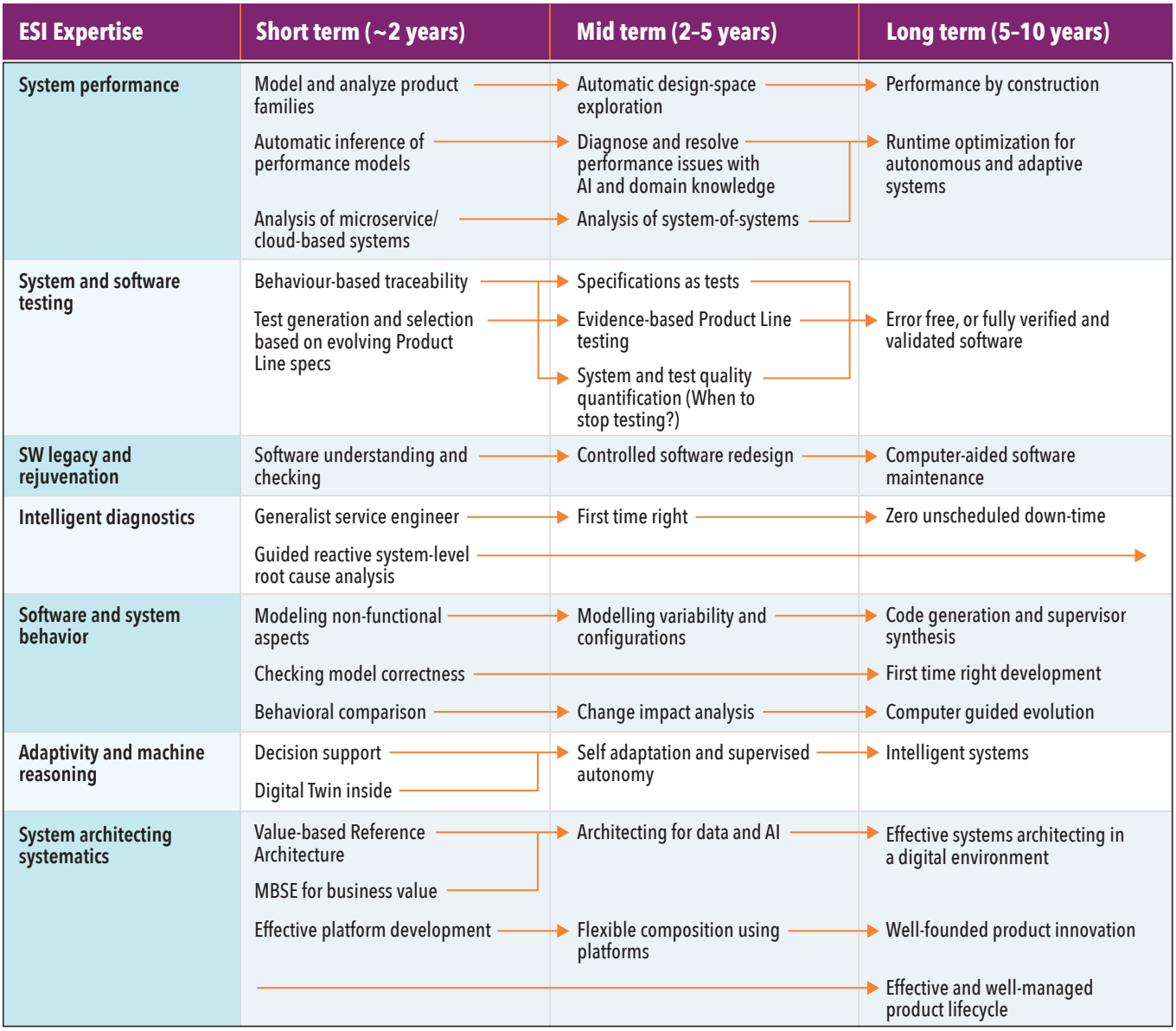


Figure 6. Summary view of the ESI expertise roadmaps

and reducing the accidental complexity. Legacy code can be scrutinized and rejuvenated using static analysis (Mooij 2020) and dynamic code analysis (Aslam 2020). The vision is to establish continuous computer-aided software maintenance, so code never gets old.

3. **System and software testing:** the objective is to guarantee system quality. This expertise focuses mainly on improving the effectiveness and efficiency of software testing at the system level (see Hendriks 2020). The strategy is to leverage the availability of models to automate testing as much as possible, thus speeding up the test process and enriching the test suites, for example, using model-based approaches (Tretmans 2019). The vision is to achieve error-free or fully verified and validated software and systems.
4. **Intelligent diagnostics:** identifying the root cause of system performance degradation and complete system failure. The aim is to speed up this process by automatically providing correct, timely information, thus reducing the knowledge required for troubleshooting (Barbini 2021). The vision is to achieve first-time-right diagnostics and, ultimately, zero unscheduled downtime.
5. **Software behavior:** the aim is to move beyond the traditional structural description of software and systems and to describe the behavior in an actionable way. This is done by capturing how the building blocks affect their surroundings, as described in (Schuts 2018). This includes static and dynamic behavior of a single interface, multiple interfaces, an entire component, multiple components, and the complete architecture. This must make it possible early in development to comprehensively analyze the behavior and detect issues. The vision is to establish an authoritative source of information with complete descriptions of the behavior and to achieve correctness by design.
6. **Adaptive systems and machine reasoning:** addresses the need for systems to autonomously change and adapt over their lifecycle, for instance, using AI techniques. An important topic is the perfection of knowledge-based and data-driven digital twins and their concurrent in-system use (Pil 2022). While it is still early for these developments, the vision is to

establish self-adaptation and supervised autonomy of systems and, ultimately, truly intelligent, possibly even self-aware systems.

7. **System architecting systematics:** approaches, methods, and tools to advance the art of architecting and help R&D departments and system engineers deal with the ever-increasing complexity of high-tech systems (Wesselius 2022). This complexity owes to the systems being increasingly software and data-intensive and integrated into systems-of-systems. At the same time, development faces trends like continuous value delivery and growing demand for customization. The vision is to develop an effective, scalable, and deployable practice in system architecting to meet the needs of highly digitalized systems rich in data and AI content.

ESI experts in each of the above areas assemble in teams that develop a view of future developments within the area in a global sense. This view is captured in a detailed roadmap per expertise area. Each roadmap identifies trends and objectives on short (<2 years), mid- (2-5 years), and long terms (5-10 years). These trends are underpinned by the expected or needed solutions and capabilities, with the foreseen innovations in formalisms, techniques, and methods supporting these objectives. Detailed descriptions of each roadmap are beyond the scope of this paper. Figure 6 summarizes all the roadmaps, offering a comprehensive view of all the ESI expertise areas.

BRINGING THE INSIDE-OUT AND OUTSIDE-IN ROADMAPS TOGETHER

At ESI, we have thus created two roadmaps of different natures. First, we have presented the industry, demand-driven roadmap, leading to the definition of five program lines, each addressing specific problems and challenges the industry faces. We call this the outside-in roadmap. Subsequently, we discussed the expertise roadmaps focused on expected and desired methodological developments in each domain. We call this the inside-out roadmap. Though not identical, these roadmaps are by no means independent and uncoupled:

- the outside-in roadmap translates industry needs into solution directions worth exploring and applying, drawing on relevant expertise;
- the inside-out roadmaps describe the projected developments within the discipline, where these developments are, among others, driven by requirements from the industry.

The two roadmaps come together in the definition and execution of the research projects that constitute the ESI research program, as depicted in Figure 7.

Research projects are initiated to address an industrial challenge, focusing on a specific industrial use case. In projects, one or more methodologies will be developed and validated typically against the specific industrial use case. This way, the industry challenges identified in the outside-in roadmaps shape the ESI research program.

The execution of a project draws on one or (usually) multiple ESI expertise areas. The insights gained in a project contribute to extending the expertise of ESI. During the project definition phase, the relevant expertise roadmaps are consulted to check which roadmap items are covered by the set of projects. The expertise teams will indicate which roadmap items have a high priority to be addressed by the research program. Project plans will be amended accordingly, ensuring the roadmaps are executed. Thus, the inside-out roadmaps help define the program.

Soon we intend to bring the two roadmaps together in a more holistic view. Here, we will acknowledge that although the roadmaps are correlated. There also is and should remain, some tension. On the one hand, there will be industry needs where the potential solutions remain beyond the current possibilities. On the other hand, some methodological developments logically follow from pursuing a trend that may not yet address a clear industry need.

Our roadmap also feeds directly, through active participation, into national and internal research agendas, such as the Dutch HTSM systems engineering roadmap and the EU electronic components and systems roadmap, ECS-SRIA.

COMPETENCE DEVELOPMENT PROGRAM (CDP)

Methodologies only have value when organizations can put them into practice. This requires stepping into education and developing professional capabilities for designing high-tech systems. ESI has created several programs and learning tracks to support companies in developing and exploiting such competencies, focusing on long-lasting results. We recognize that to be effective theoretical classroom training needs to be applied in industry practice. Our learning tracks typically center around an executive-sponsored industrial use case brought in by the company of the participants. Our competence development program, the ESI academy, aims to cover our knowledge base across all expertise areas.

CDP ROADMAP

Our CDP program covers expertise areas 2, 5, and 7, as listed above. Contents-wise, the goal for the coming years is to cover all expertise areas.

From an educational point of view, the CDP activities have changed over the years from dominantly technical stand-alone courses to tailor-made learning interventions coupled with business and personal development needs. The impact of this move has proven to be very significant. We also moved from executing such tailor-made programs within a single company to settings where multiple companies team together while maintaining the same high quality and personalization. This has resulted in a further increase in impact.

We aim to further develop our approach

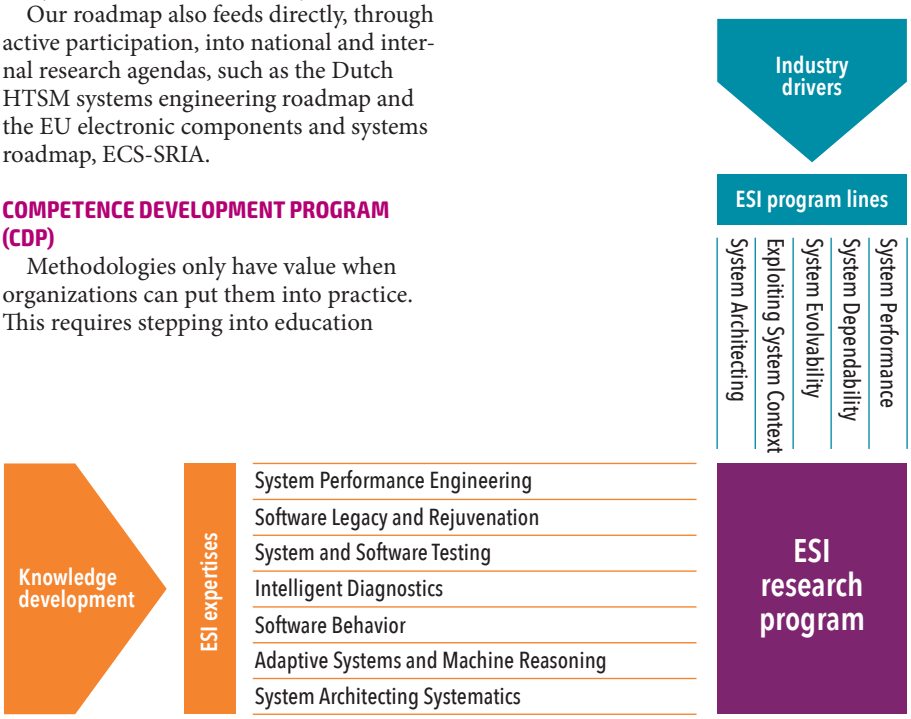


Figure 7. The ESI program lines, ESI expertise areas, and their respective roadmaps define and drive the ESI research program

by creating and intensifying multi-company thematic programs in which competence development and running research projects come together more directly.

In systems engineering, at least in The Netherlands, a lack of connection or alignment exists between mid-career education, as in the ESI career development program, and university and college education. ESI has co-founded an initiative to create a consistent program aligning concepts, mindsets, and methods across these different levels and stages of education. Going beyond the training of systems engineering as a process only, it aims to teach a systems engineering mindset in the context of business, domain, technical expertise, and leadership.

CONCLUSION

TNO-ESI leverages its strong and intimate partnerships with industrial companies to create and regularly update an overview of the high-tech equipment industrial needs and challenges in systems engineering. This leads to a demand-driven or outside-in roadmap organized in five program lines. Next, seven expertise roadmaps are created and regularly updated to lay out the expected and desired methodological developments. Together these define the ESI research program executed in collaboration with its university and industry partners. Similarly, these roadmaps steer the ESI competence development program to guarantee that knowledge transfer to industry reaches its full potential. ■

ACKNOWLEDGMENTS

Our thanks go to all involved in the ESI roadmapping process. This includes ESI research fellows and project managers, the members of the ESI Partner Board, and many co-workers at our partners' organizations. We also thank the ESI Partners and the Netherlands Organisation for Applied Scientific Research TNO for their financial support, management support, and collaborative efforts to conduct the research and reach the results described here. Further, we acknowledge the Netherlands Ministry of Economic Affairs and TKI-HTSM, the Netherlands Enterprise Agency (RVO), and the European ECSEL JU for financial support.

REFERENCES

- Heemels, W.P.M.H., E,H, van de Waal, G.J, Muller. 2007. "A design methodology for high-tech systems." In: *Boderc: Model-based design of high-tech systems*, edited by Maurice Heemels and Gerrit Muller: 11-26. Embedded Systems Institute, Eindhoven, The Netherlands.
- Van der Sanden, B., Y. Li, J. van den Aker, B. Akesson, T. Bijlsma, M. Hendriks, K. Triantafyllidis, J. Verriet, J. Voeten, T. Basten. 2021. "Model-Driven System-Performance Engineering for Cyber-Physical Systems." EMSOFT '21: *Proceedings of the 2021 International Conference on Embedded Software September 2021*: Pages 11–22. doi: 10.1145/3477244.3477985
- Mooij, A.J., J. Ketema, S. Klusener and M. Schuts. 2020. "Reducing Code Complexity through Code Refactoring and Model-Based Rejuvenation." *2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER)*: 617-621, doi: 10.1109/SANER48275.2020.9054823.
- Aslam, K., L. Cleophas, R. Schiffelers and M. van de Brand. 2020. "Interface protocol inference to aid understanding legacy software components." *Softw Syst Model 19*: 1519–1540. doi: 10.1007/s10270-020-00809-2.
- Hendriks, T., K. Triantafyllidis, R. Mathijssen, J. Wesselius, P. van de Laar. 2020. "A Virtual Test Platform for the Health Domain." In: *Validation and Verification of Automated Systems*, edited by A. Leitner, D. Watzenig, J. Ibanez-Guzman, 297-320. Springer, Cham, CH. doi: 10.1007/978-3-030-14628-3_21
- Tretmans, G.J., and P. van der Laar. 2019. "Model-Based Testing with TorXakis: The Mysteries of Dropbox Revisited." In: CECIIS: 30th Central European Conference on Information and Intelligent Systems, HR. Proceedings, edited by V. Strahonja, 247-258, Varazdin, HR: October 2-4. Faculty of Organization and Informatics, University of Zagreb, Zagreb, HR. http://archive.ceciis.foi.hr/app/public/conferences/2019/Proceedings/QSS/QSS3.pdf.
- Barbini, L., C. Bratosin, and T. Nägele. 2021. "Embedding Diagnosability of Complex Industrial Systems into the Design Process Using a Model-Based Methodology." *PHM Society European Conference* 6 (1):9. doi: 10.36001/phme.2021.v6i1.2806.
- Schuts,M., J. Hooman, I. Kurtev and D. Swagerman. 2018. "Reverse Engineering of Legacy Software Interfaces to a Model-Based Approach." *2018 Federated Conference on Computer Science and Information Systems (FedCSIS)*, 2018: pages 867-876. doi: 10.15439/2018F64
- Pil, A. 2022. "AI trains itself match fit on a digital twin." Bits&Chips, 2 June. https://bits-chips.nl/artikel/ai-trains-itself-match-fit-on-a-digital-twin/.
- Wesselius, J., J. van den Aker, R. Doornbos, T. Hendriks, J. Marincic, W. T. Suermondt. 2022. "MBSE in the High-Tech Equipment Industry, MBSE-Study of ESI and Partners – Observations and Conclusions." White paper, ESI (TNO). https://publications.tno.nl/publication/34639873/jgHNmz/TNO-R2022-R11504.pdf. TNO 2022 R11505/ESI 2022-10029

ABOUT THE AUTHORS

Wouter Leibbrandt is the science and operations director of TNO-ESI. Before joining ESI in 2016, Wouter was with NXP for ten years, managing the Advanced Applications Lab. Until 2006 he was with Philips Research labs for 14 years, managing various projects and departments in The Netherlands and abroad. Wouter holds a PhD in physics from Utrecht University.

Jacco Wesselius has been the business director of TNO-ESI since July 1, 2022. He joined ESI as a senior project manager in 2018. Before joining ESI, Jacco was with Technolution, where he was project manager, technology director, and business unit director from 2012 until 2018. Until 2012, Jacco was with Philips Healthcare for seventeen years, where he had various functions such as software engineer, systems architect, technology manager, and R&D director. Jacco holds a PhD in software engineering from the Delft University of Technology.

Frans Beenker was the ESI business director responsible for managing programs and activities within the high-tech industry. Before joining ESI at its start in 2001, Frans was with Philips, where he held several technical and management positions. Starting in 1982 at Philips Research, he moved to Philips Medical Systems in 1994, where he worked as a project manager on several large-scale product development projects. Frans holds a PhD in electrical engineering. In the summer of 2022, Frans transferred his responsibilities to Jacco Wesselius.

[Editor: Author biographies were current when the paper was initially published in 2022.]

INCOSE
www.incose.org

INCOSE Career Center

Scan QR code to visit www.careers.incose.org

The INCOSE Career Center connects top talent with leading organizations in the field. Explore job listings tailored for systems engineering professionals, post your résumé, and find the perfect role to match your skills and aspirations.

Whether you're a job seeker or recruiter, the INCOSE Career Center is your gateway to opportunities in the systems engineering community.

Connecting talent with opportunity

FEBRUARY 2026
VOLUME 29/ ISSUE 1

Modular Over-the-air Software Updates for Safety-critical Real-time Systems

Domenik Helms, domenik.helms@dlr.de; **Patrick Uven,** patrick.uven@dlr.de; and **Kim Grüttner,** kim.gruettner@dlr.de
Copyright ©2022 by Domenik Helms, Patrick Uven, and Kim Grüttner. Published by INCOSE with permission

■ **ABSTRACT**
Automotive software is undergoing a rapid change toward artificial intelligence and towards more and more connectedness with other systems. For both, an incremental design paradigm is desired, where the car's software is frequently updated after production but still can guarantee the highest automotive safety standards. We present a design flow and tool framework enabling a DevOps paradigm for automotive software development. DevOps means that software is developed in a continuous loop of development, deployment, usage in the field, collection of runtime data and feedback to the developers for the next design iteration. The software developers get support in defining, developing, and verifying new software functions based on the data gathered in the field by the previous software generation. The software developers can define contracts describing the time and resource assumptions on the integration environment and guarantees for other dependent software components in the system. These contracts allow a composition of software components and proof obligations to be discharged at design time through virtual integration testing and runtime through continuous monitoring of assumptions and guarantees on the software component's interfaces. An update package, consisting of the software component and its contracts, is then automatically created, transferred over the air, and deployed in the car. Monitors derived from the contracts allow for supervising the system's behavior, detecting failures at runtime, and annotating the situation to be included in a data collection, fueling the next design iteration.

■ **KEYWORDS:** DevOps; safety-critical; over the air updates; contracts; monitoring

WHY DOES THE AUTOMOTIVE INDUSTRY NEED UPDATES?

Updates for software have been around for almost as long as software exists. No matter how extensively software is tested before delivery, there can always be situations during operation that should have been considered during testing. Moreover, the environmental conditions for the software can always change over time, for example, through changes in the environment, new hardware, or new requirements for the program.

The procurement and installation of updates used to be a very time-consuming, partly manual process at the time of the first Windows programs. It has long since been solved, with the latest app marketplaces such as Google Play or the Microsoft Store allowing developers to post updates in these stores. The availability and compatibility of an update are then automatically identified. That update is automatically installed–depending on the software and hardware configuration–without user interaction. The separation of all high-level functionality into individual and separately updateable program parts (apps) is also known. It has been common practice for a long time on PCs and cell phones.

For non-safety-critical systems, updates are already commonplace today only because there are no life-threatening consequences in the event of an update error. The update software, of course, can have errors like any software. In addition, installing the new software or its interaction with existing software components can also lead to errors or complete system failure. There is currently no safe solution for safety-critical systems such as driving cars that rule out a system failure with potentially fatal consequences due to a faulty update or triggered by the update process itself, even for rare edge cases. Cars are susceptible because they run programs with complex real-time requirements, so certain programs must run in fixed time bounds; for example, to retrieve each new output of a sensor exactly once and then respond correctly and with a constrained delay to the sensor data.

A common practice in the example of the car is currently to apply updates and test the correctness of the updating process in the workshop during car maintenance in safety

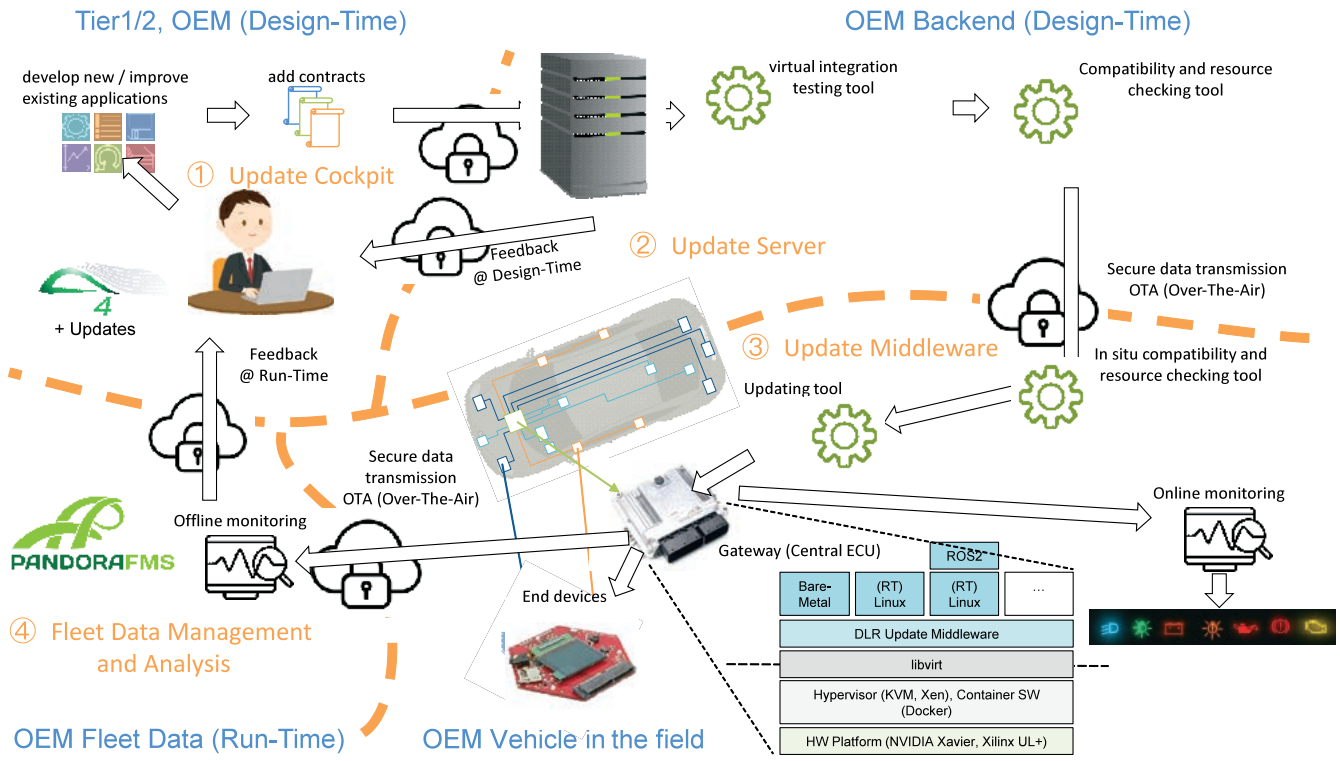


Figure 1. Concept for an automotive continuous development / continuous integration loop: ① The designer uses fleet data to develop an update for an automotive function and defines contracts to describe the system's time and resource requirements; ② The update is virtually tested for compatibility, timing, and resource demand. Monitors are generated from the contracts, which will subsequently safeguard the running function; ③ The update package is transferred via a wireless connection to the car, where it is again checked for compatibility and resource demand; Then it is installed without influencing other running functions using modularization and partitioning; Monitors constantly supervise the functions in accordance with the time and resource specifications; In the car, monitoring events can be used to circumvent critical situations outside the designer's initial assumptions; ④ Rare events in terms of timing, resource constraints, and rare functional behavior are transferred back to the manufacturer and collected for the next design iteration

to avoid these real-time problems. However, this already takes hours today, mainly because the entire vehicle software is updated at once to exclude accidental, malicious interactions with other program parts.

As vehicle software becomes more complex, the transfer and installation of a new complete update will take longer and longer, and more and more individual vehicle configurations (with/without lane assist, with/without parking assist) will also become more complex. On the other hand, new usage concepts such as car sharing will lead to ever higher utilization times, for example, fewer and fewer downtimes during which updates can be performed. Thirdly, with more and more driving functions, more and more updates will also become necessary. For AI-based systems, car manufacturers want the systems to detect rare and critical situations during operation and learn from them. The improved artificial intelligence is then to be distributed to all vehicles so that the entire fleet can benefit from the experience of individual vehicles. For systems that communicate with each other or with the cloud, it should also be possible to update them after delivery, as an update of one system may render updates to other communicating systems necessary.

All in all, updates will become more frequent, take longer, and vehicles will be out of service less time and less often to receive and implement such updates in a non-critical driving state.

SAFE UPDATES BY MODULARIZATION AND PARTITIONING

On the technical side, vehicles have long had a need and tendency to consolidate various software functions into a few central hardware units. In the past, almost every function executed in software, from the transmission of a switch position to the control of engine functions, was performed by a separate, independent hardware component. Still, for decades now, more and more functions have been consolidated into fewer and fewer individual hardware components. Consequently, this leads to mixed critical systems: hardware components that can simultaneously execute applications with burdensome real-time requirements, such as a brake assistant, and non-critical comfort functions, such as the navigation system software or air conditioning control.

To prevent mutual interference between the safety-critical components or between the safety-critical and non-critical components, it was proposed that all individual functions are separated by a basic operating software, a hypervisor, in such a way that the function of all other components is guaranteed even during an update of individual components (AGL 2018). To this end, the hypervisor creates an environment that appears to each function running on it as a separate hardware platform. The functions are executed in isolation from the remaining system and can only react with functions in other hypervisor partitions via defined interfaces. A set of functions can thus be modularized and partitioned into safety and non-safety relevant sets, implying different requirements on the hypervisor partition's capabilities in terms of timing and resource guarantees.

Modularization gives each component a clear and consistent outer limit in terms of functionality and interfaces to other functions. Partitioning through a hypervisor

further adds controllability and guarantees access to shared system resources such as memory or communication components. Also, in a system of modularized and partitioned functions, it is possible to predict the timing behavior of the overall system, even for different variants and system configurations before and after an update. Such tests are possible during design with a virtual prototype (virtual integration testing) and later before deployment in the existing system. At both points, it is possible to check whether the update can be integrated safely before it is implemented. Furthermore, partitions allow the independent update of functions residing in different partitions, even at run-time, which can become important for future systems.

In summary, automotive manufacturers and suppliers need to be able to let the car's hardware identify, download, and install updates on the fly and without user interaction, just as they do on a PC or cell phone. Because over-the-air uplink is limited in bandwidth, updates will be frequent in upcoming vehicle generations, modularized functions should be supported inside a partitioned HW/SW environment. Instead of all software having to be updated at once, individual software components and partitions can then be updated while others can remain as they are.

Depending on the manufacturer's requirements and the customer's acceptance, modularized and partitioned software components can keep running undisturbed while other components are updated. Therefore, neither the update process nor the updated software component itself must not pose a risk to the vehicle's safety at any time.

AN AUTOMOTIVE CONTINUOUS DEVELOPMENT/CONTINUOUS INTEGRATION LOOP.

Our design flow and software tool infrastructure that implements the above concepts of modularity and partition in combination with time and resource contracts and run-time monitors is visualized in Figure 1 and consists of four stages.

① *Update Cockpit*

The starting point to developing an update of an automotive function is an *Update Cockpit*, which is based on the App4MC software (Höttger et al. 2017). The update cockpit allows the definition of the application as an updatable unit plus contract-based metadata for the system integration. These metadata contain the interface definitions, the specification of timing behavior, split up in assumptions and guarantees (timing contracts), as well as the specification of resource requirements, split up in assumptions and guarantees (resource contracts). Furthermore, the update cockpit allows users to represent and modify the overall system configuration, for example, vehicle software for a specific vehicle platform. Modifications such as removing, modifying, or adding of software components trigger design time virtual integration and compatibility checks to confirm the validity of the new configuration (see ②). If the virtual integration and compatibility checks fail at design time, the new configuration is rejected so that it is not deployable in the field, and diagnostic trace information is delivered and visualized through the updated cockpit.

② *Update Server*

The update server provides simulation-based or analytical integration testing for timing contracts and a compatibility check for resource contracts (virtual integration testing). We use the MULTIC (Design Approach for Multi-Layer Time Coherency in ADAS and Automated Driving) software (Damm et al. 2019) to quickly and virtually test for various hardware and software configurations, which might be out in the field and which, for practical reasons, cannot be all available as actual hardware instances at the manufacturer's site. If hardware components are available, this can be optionally supported by the hardware-in-the-loop (HiL) tests of the target HW platform or the entire vehicle's electronics, network, wiring, and software architecture, for example, E/E architecture. If the integration or compatibility checks fail, the developer or system integrator delivers diagnostic traces as feedback. If the integration and compatibility checks are passed, an update packed for the vehicles in the field is prepared and submitted *Over-The-Air* (OTA) through a secured transmission into the vehicle.

③ *Update Middleware*

The updated middleware allows the update server (see ②) to securely transfer a new update package over a wireless connection to one central gateway ECU. The update may concern the ECU itself, or it may be targeted to one other subcomponent (end device) in the car, not directly connected to the outside world but connected to the gateway ECU. After the update is complete, it is validated for data integrity. Then, the compatibility and resource checking is repeated in the field against the hardware specifications of the end device's hardware to be updated to avoid catastrophic configuration failures. This includes verifying the proper hardware and software configuration, the system's health status, including the presence and severity of hardware aging, the configuration of the software container, and the associated monitor updates. Updates and monitoring of end devices are optionally supported and fully controlled through the gateway. The updated middleware uses libvirt (Ashley 2019) as a common abstraction layer above different state-of-the-art hypervisors (Kivity et al. 2007) (Barham et al. 2003) and containerization software (Rad et al. 2017) and HW platforms.

To implement the update, there are different options: For an **offline update**, the entire system is brought to a safe state and paused, the function to be updated is removed, and a function with the new software version is set up, started, and connected to the system. Such an update is feasible if the vehicle is at rest or parked. For an **online update**, only the function to be updated is stopped, while the other system functions remain up and running. As soon as the updated function is set up, it is reconnected to the remaining system. Such an update is feasible for uncritical system components. For a **runtime update** (zero delay update), the function to be updated remains operational, while a second instance with the update is set up and connected to its inputs while its outputs are ignored at first. Such a component then runs in a *shadow mode*, where it runs together with the entire system, and its functional behavior can be compared to the old version, which is still in charge of producing the required outputs. Between two executions, the control is handed over to the updated function by an atomic instruction. Afterward, the old function can be removed. Such an update is, in principle, able to perform updates even while driving, depending on the user's acceptance.

Once the update is implemented to the end device of the ECU gateway itself and the updated function is running again, the gateway ECU starts the monitors derived from the resource and timing contracts (see ①) included in the update package from the update server. The monitors supervise the function's timing behavior and resource demand and constantly compare it against the designer's assumptions described in the contracts. This can be accompanied by functional monitors, supervising rare events occurring and rare functional behavior of the system.

Functional monitoring may either be enabled by adding self-supervision to the function (such as novelty detection for neural networks) or by applying learning-based techniques to the system's behavior. A learning-based functional monitor does

observe relevant system parameters, such as the occurrence and timing of function calls. In an early learning phase (in the lab), it can be trained to predict the near future's parameters from the parameter's history. Such a system can be trained without human supervision just by observing the system acting in a typical environment. In the field, the monitor will constantly observe the system, predict the present behavior only by knowing the past behavior and then compare the prediction with the actual system behavior.

④ ***Fleet Data Management and Analysis***

On the one hand, the timing and resource contracts were defined by the designer under certain assumptions concerning the operating domains where the system is designed to operate, for example, its operational design domain. An operational design domain occurs with a pattern of external events, reaction timing of other systems, and performance of the own hardware, which might, in practice, undergo some degradations due to aging effects. On the other hand, automotive functions, especially AI-based ones, can detect their own uncertainty or the novelty of an occurring situation. In all cases of a violation of the operational design domain assumptions or functional misbehavior as detected by monitors (see ③), the past stimuli entering the system and its own internal state (history) together with long-term data of the hardware health status can be collected, compressed and submitted to the fleet data management server. We currently support the integration of the Pandora Flexible Monitoring System (PandoraFMS 2022). For example, Pandora FMS can detect when a network system is unresponsive, an application is defaced, or a memory leak has occurred. Pandora also monitors hardware components and operating systems. It can generate reports and send notifications to the developer team when problems occur. This way, the update cockpit is fed in information from the field to support the next update, reacting to the collected data by either redefining the operational design domain and updating specification of the timing and resource distribution through contracts or use collected stimuli to reimplement or retrain the function in order to address unexpected, rare events better.

CONCLUSION

We envision a seamless development and operation flow for safety-critical field devices, bringing together existing tools and bridging the gaps with our own proof-of-concept developments. Update server and update middleware were specified and developed in a series of industry-led research projects, existing as running hardware/software prototypes. The security of the update transmission and implementation was regarded together with safety. Nonetheless, there are still open issues coming from the very fact that the update process can initiate new software, potentially having unpredictable cross-influences with the existing software parts. The fleet data management and analysis are under intense research and development right now and are still in the conceptual phase. For the Update cockpit, all relevant information, tools, and methodology are already available, though it still must be integrated into a comfortable develop environment. ■

ACKNOWLEDGEMENTS

This work received support by the European Commission's Horizon 2020 program under the UP2DATE project (grant agreement 871465).

REFERENCES

- AGL 2018. "The AGL Software Defined Connected Car Architecture." *The Linux Foundation Automotive Grade Linux (AGL) Virtualization Expert Group (EG-VIRT)*
- Ashley, D. 2019. *Foundations of Libvirt Development*. Apress Berkeley, CA.
- Barham, P. 2003, "Xen and the art of virtualization." *ACM SIGOPS operating systems review* 37 (5): 164-177.
- Damm, W. 2019. "Multi-layer time coherency in the development of ADAS/AD systems: design approach and tooling." Paper presented at the Workshop on Design Automation for CPS and IoT (DESTION'19), Montreal, CA, April.
- Höttger, R. 2017. "APP4MC: Application platform project for multi- and many-core systems." *it – Information Technology*, 59 (5): 243-251.
- Kivity, A 2007, "kvm: the Linux virtual machine monitor." *Proceedings of the Linux symposium*, 1 (8): 225-230.
- PandoraFMS 2022. "Make smarter decisions with the data of your business." https://pandorafms.com/en/
- Rad, B.B. 2017, "An Introduction to Docker and Analysis of its Performance." *Intl J of Computer Science and Network Security*, 17 (3): 228-235.

ABOUT THE AUTHORS

Domenik Helms is the manager of the research group Deployments and Updates at the DLR Institute of Systems Engineering for Future Mobility. His background is in electronic design automation and embedded systems.

Patrick Uven is a researcher at the DLR Institute of Systems Engineering for Future Mobility. His background is in technical computer science and embedded systems.

Kim Grüttner is head of the department of System Evolution and Operation within the DLR Institute of Systems Engineering for Future Mobility. His background is in technical computer science and embedded systems.

[Editor: Author biographies were current when the paper was initially published in 2022.]

How Large Scale Agile Can Operate Systems Engineering in the Future

Laurent Alt, and **Mikaël Le Mouëlli,** lemouellic.mikael@bcg.com
Copyright ©2023 by Laurent Alt and Mikaël Le Mouëlli. Published and used by INCOSE with permission.

ABSTRACT
The significant shift happening today towards more connected, more automated, and more autonomous systems is bringing software inside all systems, and at the same time agile practices. Our experience of large-scale agile deployments in companies building or operating complex systems in automotive and aerospace shows that, whereas both approaches can easily coexist in isolated teams within the same company, major problems arise when coordinating them at the leadership level, where they are perceived as antagonist, and create misalignments, friction and quality issues. In this article, we propose to describe why it is important to make agile and systems engineering work together, how to do it, and how this impacts how we see value, systems, digital twins, and leadership. The following concepts of the FuSE agile roadmaps are addressed:
- Agility with long lead time components and dependencies
- Agility across organizations boundaries
- Orchestrating agile operations.

PERCEIVED DISCONNECT BETWEEN AGILE AND SYSTEMS ENGINEERING

In many industries, the increasing expectations on time-to-market, complexity, sustainability, regulations, and personalization are stressing the development processes in place, to make them more flexible and more adaptive. In addition, software is taking a large share of the added value of products, and forces organizations to expand their software development capabilities and adopt agile practices.

But there is a perceived opposition between systems engineering practices (which are often perceived as reliable but rigid) and agile (faster but permissive). Each approach has its benefits but also comes with constraints that seem to be at odds with the other.

So this "softwarization" trend raises the question of maintaining the existing processes in place, while putting in place agile ways of working. This also raises the question of which agile practices should be looked at, and if they should be adapted.

In the following, we consider agile in a very broad sense, including the end-to-end DevOps view based on feedback loops from real operations data, not constraining ourselves by any specific framework nor organizational implementation. We will simply consider agile as a collaborative and sustainable way of incrementally delivering value and learning, based on facts and data. This view encompasses both software startups and companies like SpaceX.

THE NEW CHALLENGES AHEAD

The best illustration of the numerous challenges that manufacturing industries are facing now is Tesla. It is true that the focus has mostly been put on the electrification side of the car market, but however significant this shift is for the economy, that technical challenge could be manageable for every original equipment manufacturer (OEM) by wisely using current engineering practices. It simply amounts to replacing one propulsion technology by another one.

But the problem is much bigger than that.

On top of electric propulsion, new important trends are actually impacting all automotive OEMs:
- more and more software in functions: for example, braking systems now heavily rely on sensors to trigger braking, and also make it possible to recover energy to the battery.
- new usage trends like ADAS (advanced driver-assistance systems) use numerous sensors, and many options can be turned on and off as a preference.
- independence of software from hardware, and increased platformization of the technology, in order to optimize reuse across car lines, especially due to the complexity of software.
- new business models based on user stickiness, increased connectivity, and more frequent updates.

- extension of the reach of software to a more global mobility scope, like charging stations.

These trends are not specific to automotive, of course, we can see them emerging in all markets, although not all at the same speed.

As a consequence of this, many organizations are adopting agile practices. But agile is often used in the information technology (IT) department and in teams doing the development of software applications. Besides, systems engineering practices are used for embedded software and hardware organizations. This situation is simply due to these organizations working in silos, each using what they are most familiar with. Both worlds are connected but do not operate consistently, and, consequently, they have a hard time defining shared priorities, speaking the same language, implementing requirements that are fulfilled by a mix of software and hardware, and usually discover quality issues too late.

Therefore, it is important to first explore how these agile and systems engineering teams can work together effectively, keeping the best of both worlds, in particular for products with long development lead times.

HOW AGILE AND SYSTEMS ENGINEERING CAN WORK TOGETHER

Without going against classical stage gate processes, which are designed to progressively derisk the delivery of complex products, here are a few guidelines to make the two systems work together.

- the agile iterations rhythm can be designed to match programs gates. One single agile cadence can be aligned on several programs, in order to deal properly with teams with fixed capacity.
- backlog items, which describe agile teams activities for example *user stories*, can be seen as studies with requirements as an input, and systems design documents as output (*definition of done*). According to the process stage we are in, the expected precision can be more or less precise (*acceptance criteria*). But they can also well be other types of activities, like testing, documenting, and so forth. A more detailed description of this incremental precision approach can be found in Krob (2019), for example.
- their *business value* can be used the usual way (related to the client), or reflect the expected gains from trade-offs (performance vs cost, for example), or even negatively as the risk of delaying them (Reinertsen 2009).

The benefit of this unifying approach is not local, it is global.

Locally, IT teams already working in agile will likely not be fond of formalizing requirements, until they must develop offboard features that are linked to onboard features. *Locally*, hardware teams will likely find little added value in slicing their work in small increments, until they become part of a mechatronic system that is evolving at the speed of software changes. But *globally*, the whole organization can communicate, develop, and integrate consistently.

Having now in mind an agile way of working that can cover all aspects of a complex, long lead time product, let's now look more closely at a few key aspects of product development.

DEFINING VALUE

The concept of "business value" is central to agile, since it governs how priorities are assigned to activities within a fixed capacity and fixed deadline constraint. Yet it is not so easy to define and manipulate, so much so it has sometimes been called "the elephant in the agile room" (Schwartz 2016). The question lies in how to define it, but also how to easily allocate it down to the level of teams and bring them actionable priorities very frequently to enable the so much needed teams empowerment.

Where is the problem? For agile teams in start-ups continuously delivering a service or an app to end users and being able to measure the outcomes of a new version on their business, it is easy to define some business value, and to connect it to everyone's daily work. But for large companies delivering complex products that take months or even years to deliver, it is different. A car or an aircraft are defined by several target attributes such as range, cost, NVH (noise, vibration, harshness), weight, and so forth, that are more or less strictly allocated across all the sub-systems upfront, and further design activities actually produce or refine trade-offs between those attributes. In addition, other properties as impact on manufacturability, sustainability, or delivery timeliness must be optimized too.

So, for complex systems, *value* is a multi-criteria concept that is defined in the context of a global product and organizational setting, and due to impacts, the overall convergence of the system design vs the target attributes can only be evaluated by integrating all the design elements. The more entangled the trade-offs, the more frequent updates we need.

Systems engineering practices make these choices possible. Frequency can be achieved by automation and intensive use of model-based systems engineering (MBSE) and simulation tools. But agile provides the incremental way of working that makes it possible to smoothly make the trade-offs converge as we move from upfront phases to more detailed design phases, and make them flow across the whole organization.

There is yet another aspect of value that must be considered.

Let's go back to the ever-increasing part of software in the design. The most important effect of this, is that that software is massively reused from a continuously evolving platform rather than specified against new requirements each time, increasing value even after launch through over-the-air (OTA) updates, hence creating the emergence of long-lived platforms that support whole product lines. Practically, this means that the concept of value must also include the contribution of an activity to a long-term architecture vision, and across several product lines, which must be balanced with short term priorities of single products.

LOOKING AT SYSTEMS AS PRODUCTS

The speed at which products are being delivered is progressively becoming a problem, especially as new constraints arrive. This is not new, since compliance to regulations have always been an important provider of requirements, and sustainability regulations are simply making the context more complex.

The problem is that, even with good systems engineering practices, many organizations already struggle with how to best balance innovation and carrying over existing design solutions in the context of new requirements. When requirements are cascaded too fast too early, this prevents proper reuse (or adaptation) of existing designs.

This is getting worse when one thinks of reusing across several product lines, where the variety of requirements is multiplied by the specifics of each line.

And even worse if we think that OEMs want to "own" the intelligence of their systems and therefore migrate the algorithmic part of their subsystems up to a vehicle software layer. And worse yet, considering the rapid evolution of sustainability regulations that force OEMs and suppliers to progressively have all their components fulfill requirements under more and more sustainability constraints, implying that components must follow a path towards more sustainability.

In a sustainable organization, this much change can only be managed with an *evolutionary* approach covering all aspects of the product, technology, knowledge, organization, testing facilities, and so forth.

A working mode where rigid expectations overcome reuse, and leave little room for trade-offs and adaptation, is called "project" mode in the agile world, as opposed to the "product" mode, where reuse and evolution are the default. This may sound counterintuitive, since agile is often believed to be very flexible and to encourage massive changes, but most of this agility comes from changing priorities, not from refactoring the technical stack nor disbanding teams and forming new ones. As Martin Fowler (one of the Agile Manifesto authors) puts it, design is not dead with agile, it simply becomes different (Fowler 2004). Enabling agility does not prevent good thinking upfront but is better achieved with the ability to make redesign possible when needed.

Therefore, agility is managed by integrating, at the core of the organization, the idea that each component must be considered as something that will evolve and that provides a service either internally or externally, that improves over time. This does not go against systems engineering, it simply means that organizations would better manage systems as products (meaning, systems associated with a vision, a development roadmap, a value delivered, competition, the means to deliver it, and so forth.

The solution consists in having the technical choices for a system be made in the context of a long-term vision of that system, a set of functions and structure progressively evolving along a roadmap, and assigning a product owner as the person who prioritizes the increments. This does not only hold for the system of interest, but also for sub systems on several levels.

Finally, organizing this way puts more emphasis on the importance of modeling and managing stable interfaces, how functions are fulfilled, how they evolve, in order to provide the necessary autonomy for product owners to manage their own work in an autonomous manner. Therefore, this product approach must reinforce good systems engineering practices.

DEVOPS AND DIGITAL TWINS

Agile is about working incrementally, and this ability to work by increments heavily relies on all stakeholders agreeing on shared facts. In software it is the famous principle of the Agile Manifesto "working software over comprehensive documentation." Everyone in a company doing a software product understands this software and what problems it solves for its users, so it is the best means to communicate and assess the work done. The word "working" has its importance too, since it means that whatever is considered done should have undergone a minimum of testing. This is the *raison d'être* for the DevOps chain.

In hardware, however, due to much longer cycle times, we cannot wait for a product to be delivered, even for a subsystem. So, one can rely on systems diagrams, 3D prints, digital mockups, prototypes, mules, etc.

However, if we look precisely at the verification and validation (V&V) part, and taking into account that the system of interest is a mix of software and hardware, if we wish to have the equivalent of a DevOps chain, the ideal artefact is a digital twin. This is so for two reasons. The first one is that the digital twin is agnostic regarding the kind of technology the system is using (hardware, embedded software, or software). The second one is that it can span the entire lifecycle of the product, even after launch, all the product configurations, and help simulate further software updates over the same hardware product.

An additional difficulty that will become more important is the integration of humans in the definition and operations of systems. Of course, this will definitely bring more complexity and more constraints for development and operations, but we also believe that this additional complexity will accelerate the use of data, the Internet of things (IoT), and artificial intelligence (AI), together with the definition of ethical principles.

Of course, it is a daunting task to model 100% of a system with MBSE. But there are ways to model things incrementally, by considering some of the components as black boxes, simulating their behavior with models, and progressively increasing the coverage of the whole system as needed.

LEADERSHIP AND CULTURE

It is well known that culture and leadership are the main issues in large scale agile transformations. Moving towards agile is probably the most difficult change in organizations since it implies a significant culture shift across all functions.

There are many aspects of this change: teams' empowerment, collaboration across teams and leaders, data driven decisions, value driven priorities, product mindset vs project mindset, being flexible on priorities, caring about organizational learning, focus on quality, etc.

Those can be found in the vast literature about agile leadership, but the most important thing about it is that the higher you go in hierarchy and responsibilities, the more difficult it is to make these changes happen due to the increasing pressure, time scarcity, and the longer history of control those leaders have acquired by reaching their position.

Here, we will simply mention one important point, that we find particularly relevant in the context of complex systems.

Since systems should be considered as products (or platforms) with a roadmap, leadership must have an increased ability to balance value and technical feasibility of these products more widely and more frequently. This does not mean that all leaders should master both skills, but rather that the organization should enable the seamless collaboration of value driven and technically savvy people in order to manage, on several levels of the system, balanced priorities between value delivery and technology feasibility.

In our experience, the pattern of disconnect between business and tech is pervasive. We have seen many situations where programme directives are so compelling from the start that they make reuse very difficult and impede the creation of technology roadmaps. But we have also seen many times systems architects spread across organizations, each having local influence but reporting to non tech-savvy leaders, or, on the contrary, being enough involved upfront to make important choices, but without having enough understanding of the business implications of their choices, due to the lack of ability to communicate across the leadership layers.

Of course, the agile ways of working make this connection more natural, since it is usually embedded in the sprint plannings and quarterly business reviews, or SAFe planning interval (PI) plannings. But when it is about executives, quarterly discussions are not enough to make a significant cultural change happen.

To us, this raises the question of how to structure communication, in a way that naturally connects both worlds. Systems engineering concepts need to be made more accessible, so that as many stakeholders as possible can be involved in decisions. We have successfully used high level "product maps" to align leaders on a shared understanding of a product view of their work, as opposed to simply delivering their work as a program. This has been successful in spreading a shared sense of the product, spot and share high level "invisible" dependencies at leadership level and early in the development plans.

Consequently, another topic that also needs to be addressed is the way leaders organize their agendas, in order to make these connection points as frequent as possible. One cannot simply expect to develop complex products and to change mindsets if the only alignment between stakeholders happens once a quarter. This point is addressed more in length in Alt-Le Mouëllic (2022).

CONCLUSION

With the increasing importance of software in complex products, agile ways of working are also becoming a standard in the development process. However, the perceived opposition between systems engineering and stage gate processes on one hand, and agile on the other hand, often creates dual organizations that struggle to work effectively.

Software organizations can be reluctant to adopt some systems engineering practices, as much as hardware organizations may not find a lot of added value in adopting agile. But the problem is not a local one, it is global. There are tremendous gains in unifying product development methods to encompass hardware and software, and this brings new insights in how agile should be considered at the scale of a company. ■

REFERENCES

- Alt, L., and M. Le Mouëllic. 2022. "How leaders can take ownership of their Agile Transformation." *LinkedIn*. https://www.linkedin.com/pulse/how-leaders-can-take-ownership-agile-transformation-laurent-alt/.
- Beck K. et al. 2001. Manifesto for Agile Software Development. https://agilemanifesto.org/.
- Fowler, M,. 2004. Is Design Dead? https://www.martinfowler.com/articles/designDead.html.
- Krob, D, 2019. "CESAF: an Iterative & Collaborative Approach for Complex Systems Development. Complex Systems Design & Management." Paris, FR. ffhal-02561389. https://hal.science/hal-02561389.
- Reinertsen, D. 2009. "The Principles of Product Development Flow: Second Generation Lean Product Development." Celeritas Publishing.
- Schwartz, M. 2016. *The Art of Business Value*. IT Revolution Press.

ABOUT THE AUTHORS

Laurent Alt is a seasoned leader (CTO, CEO) in technology development and innovation (notably at Dassault Systèmes and Lectra), and is now an expert in enterprise agility and systems engineering at BCG Paris, mainly supporting the move of automotive and aerospace companies towards software and agility, leveraging systems engineering practices.

Mikaël Le Mouëllic is a managing director and partner, at BCG Paris since 10 years, after 6 years in charge of production management at Vallourec. Mikaël is the head of BCG's R&D practice in Europe, helping automotive and aerospace companies in their transformation towards agile.

[Editor: Author biographies were current when the paper was initially published in 2023.]

FEBRUARY 2026
VOLUME 29/ ISSUE 1

Model-Based Systems Engineering as an Enabler of Agility

Sophie Plazanet, and **Juan Navas,** juan.navas@thalesgroup.com
Copyright ©2023 by Sophie Plazanet and Juan Navas. Permission granted to INCOSE to publish and use.

■ **ABSTRACT**

Model-based systems engineering (MBSE) with agility can help systems engineering programs which deal with both increasing complexity and frequent changes in environment and usages, shorter time-to-market, uncertainty of the needs, and more sophisticated industrial schemes. Agile approaches originated in software engineering can be extended and tailored to a certain extent to complex systems engineering and particularly to MBSE. Main benefits of agility are provision of a minimum viable product as early as possible in the schedule, early capture of changes of needs, enabling to deliver a system answering to the real needs, and securing of the value proposal. It includes also potential reduction in rework of the final system through regular customer feedback throughout development (left shift for the defect correction with early exposure), and efficiency of the use of resources. Concerning MBSE, the use of models as a single source of truth for completeness and consistency is useful to share and secure the design by improving communication within engineering teams and the building and support of the development strategy, and to help to automate some tasks such as model exchange and synchronization. In addition to the benefits of each approach, combining them may help to:

- **Organize and synchronize** the development and validation effort of one or multiple engineering teams.
- **Faster impact analysis** including trade-off studies/options and hence a **faster reaction to evolutions** in expectations and constraints, that is, the agility of systems.
- **Show regularly "end to end" value** to the customer and other stakeholders.

INTRODUCTION

In this article, we illustrate how model-based systems engineering (MBSE) may be an effective enabler of agility in systems engineering, focusing on dynamic learning and evolution (cf concepts of the INCOSE facilitated future of systems engineering (FuSE) roadmap), with a fictive testimony interview of a system engineer based on a fictive example (a drone-based product for inspection of electrical network), used to condensate experiences at Thales. Key concepts are presented, then the process of "warm-up/run/evaluation" is detailed, and we finish with the way to deal with an evolution request in a MBSE and agile context.

1. ***Could you tell more about your product and firm and yourself?***

I am a system engineer in the company Pythagoras. My firm develops and sells lightweight drone-based products for different markets: agriculture, aircraft exterior inspection, and public security enforcement. In addition to the drones themselves, these products embed mission control and data analysis software. These drone-based products feature manual and automated piloting, data acquisition using a wide range of technologies, live data processing, data recording, live, and post mission data analysis. After market analysis of the context, resulting in identification of the need of inspection of electrical networks, my firm has launched the development of a new product providing this service.

2. ***What means agility in systems engineering? And agility with MBSE?***

Agility in systems engineering refers to an engineering effort in which teams can adapt to new circumstances (for example changes in or new stakeholders needs) while meeting the customer expectations in terms of schedule, quality, and cost. Agile systems engineering is a principle-based method for designing, building, sustaining, and evolving systems when knowledge is uncertain and/or environments are dynamic.

Agility with MBSE refers to the use of key concepts favoring agility and co-engineering such as capabilities and functional chains, developed in an iterative and incremental way. And, to the use of system modelling tools to define these concepts, which allows additional engineering rigor and quality.

3. ***What are the key concepts you used for agile with MBSE?***

We focused on a subset of Arcadia concepts that are particularly useful in organizing the engineering effort and carrying value at the solution level: capabilities, functional chains, and scenarios. These engineering artefacts were the references for all engineering teams (systems, software, hardware, IVV, etc.).

We used agile concepts of increments, iteration, increment data packages, EPIC, and user stories.

- **Increment** is a working, tested subset of the system (or of the systems engineering artifacts) delivered regularly to the system stakeholders and built on top of an existing baseline. Value can be knowledge, risk reduction, new features, enhanced performance, etc.

  Each **iteration** is a standard, fixed-length timebox including several successive blocks of fixed duration, where agile teams deliver incremental value in the form of working, tested software and systems. The duration of these blocks may vary according to many factors, both system-related (for example, life-cycle phase, complexity of the capabilities included in the scope) and organization-related (for example, system or software engineering levels, available resources, and human resources policies)

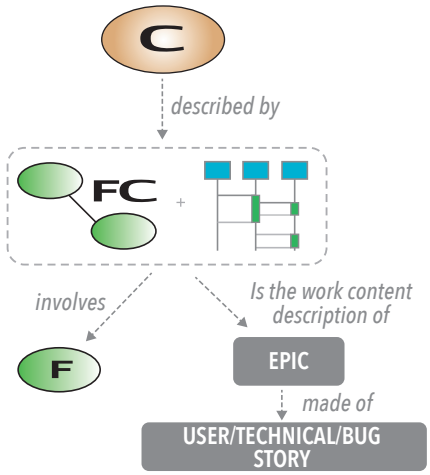


Figure 1. Relations between different concepts

- We refer to **EPIC** as an element of planning, which refers to a functional chain or scenario (or composition or pieces of them) and to other engineering data. It is used to define the expected content of a system increment and thus the value to be delivered to the user. It is defined in **user stories** that are developed in successive blocks. The content of the user stories was defined so that value (working software) was delivered after each software block (Figure 1).
- An **Increment Design Data Package** is a set of engineering data, related to an increment, that will be transferred as a baseline to either lower engineering levels teams (for example, a subsystem) or to other engineering teams (for example, verification and validation),

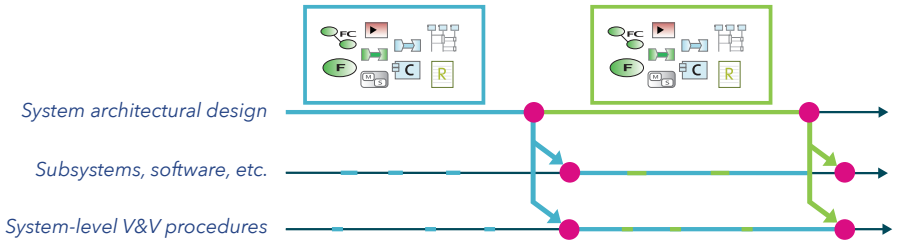


Figure 2. Increment packages dispatched to other agile teams at the end of iterations

becoming their inputs for subsequent iterations. In a MBSE context, it could be modelling artefacts (along the needs and contexts solution perspectives) described below, associated to textual and model requirements, constraints, justifications, and simulation-based analysis that are associated to these engineering artifacts (Figure 2).

4. ***How did you define the engineering workflow?***

To define the engineering workflow between the project's major milestones and associated reviews, we used a sports analogy. You need first to prepare your body (warm-up) before performing a continuous and strong effort (run), and then, if you want to improve, you need to measure and analyze your performance (evaluate).

Warm-up: The "warm-up" activities refer to tasks that will reduce the risk of the engineering efforts that will be done afterwards. It is about capture, selection, and prioritization of the work to be done to meet the objectives of the next milestone, providing the expected value to the stakeholders. It is also about the estimation of efforts required to do it and the definition of the schedule to do it.

Run: For an engineering team, the "run" activity is made of iterations or blocks, aiming at implementing product capabilities. This includes (non-exhaustive list) the detailed definition of product functions and exchanges involved in the capabilities, the development of the system and subsystems' architecture, the development of the software and hardware implementing expected behavior, and the verification and validation of what will be delivered at the end of an increment and to the customer.

Evaluation: The evaluation focuses on ensuring that the whole product increment produced during the iteration can be released; the major part of the integration, verification, and validation effort is performed incrementally during the run iterations, and the evaluation focuses on ensuring that the whole can be released. During early stages, evaluation may include multi-viewpoints analysis (safety, security, performance, reliability, testability, etc.), the preparation of a review of experts or the execution of simulations. Later it may include the approval by the customer or

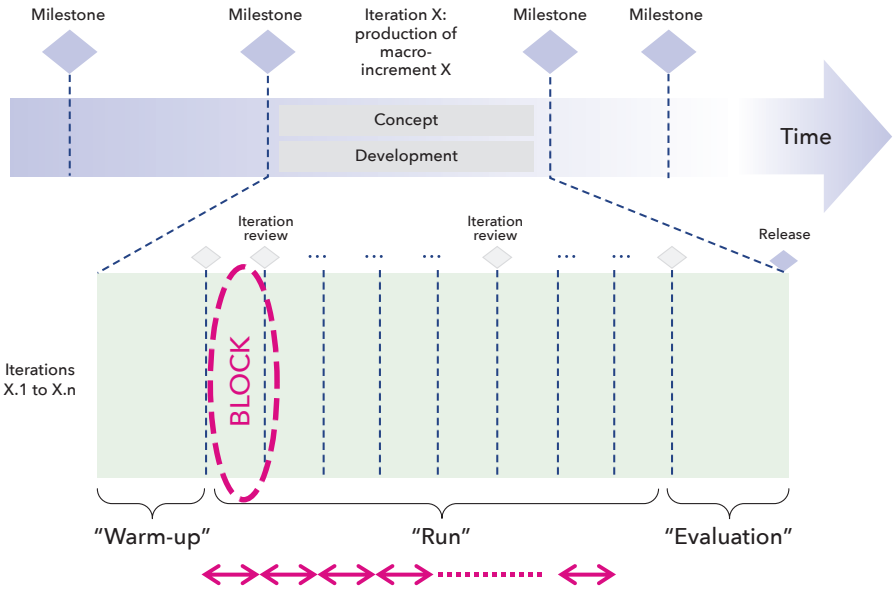


Figure 3. Engineering workflow of warmup, run, and evaluation, composed of several blocks/iterations

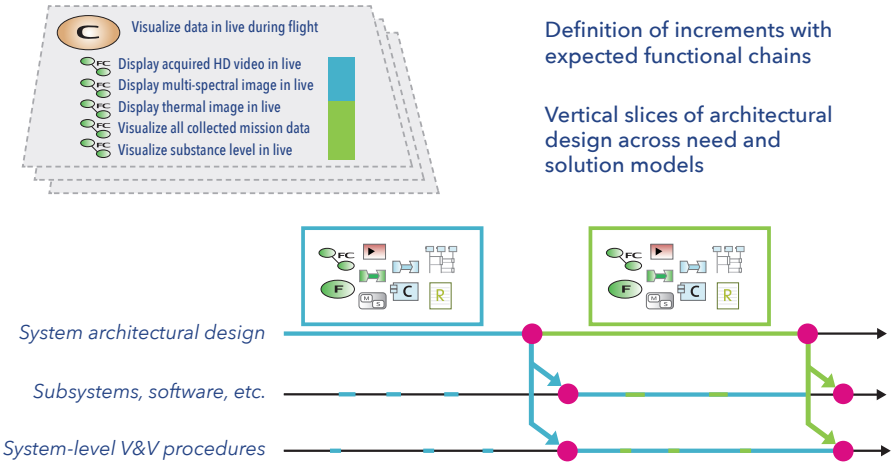


Figure 4. Example of the definition of increments for the capability "visualize data live during flight"

	ITERATION 1
	Acquire data (pictures, videos, scan...
	Acquire multi-spectral image
	Acquire thermal image
	Acquire 3D image
	Manually acquire data
	Manually trigger thermal image acquisition
	Manually trigger multi-spectral image
	Manually trigger 3D image acquisition
	ITERATION 2
	Acquire data (pictures, videos, scan...
	Acquire HD video of moving element
	Acquire HD video
	Acquire HD image
	Automatically follow a flight plan
	Automatically follow a moving target
	Visualize mission progress status

Figure 5. Extract of the repartition of functional chains in different iterations

the packaging of software and hardware releases. To evaluate how was produced the engineering effort, the engineering team members review the engineering practices, identify what went well and wrong, elucidate ways to improve the way they perform their engineering effort, including the dependencies with stakeholders, both outside and inside their organization (Figure 3).

Note that:

- The team defines the effort and time length allocated to the warm-up activity between two milestones.
- Warmup, run, and evaluate activities are not necessarily sequential, they can and are often executed in parallel: for example, some key members of the team can "warmup" by defining the scope, while others can "run" and pay technical debt that needs to be done at that moment.
- You can perform these activities several times.

5. ***How did you perform the warmup in the early stages of this development?***

We organize with the help of Pythagoras engineering coaches (agile, MBSE,…) orientation workshops, where all the teams have discussions to define the following points:

- **The articulation between engineering teams:** for example, what is a "contract" between engineering teams made of, what are the outputs from/inputs to each team, what is the development pace (length of iterations, for instance here 12 weeks was collectively decided)
- **The model-based engineering strategy:** what is the purpose of each model view? How will the views be structured? Are there existing building blocks to assemble? We defined a modelling plan.
- **The engineering tools and how they will be configured:** we chose the Team4Capella tool that integrates natively the MBSE Arcadia method and allows engineers to work concurrently, which was an enabler to co-engineering.
- The identification and selection of the scope of work and its schedule, with a first vision of the product/system to develop: We did this by selecting the capabilities that will be developed, validated, maintained or retired, along with their related functional chains and scenarios (Figure 4).

We defined which functional chains or scenarios (or composition or pieces of them) should be delivered for which iteration, as part of which scope of work of increment (Figure 5).

We defined in a model an intentional architecture of the system, that is, the architectural principles in which further architectural definition work will be based. In further run and evaluation phases, updates or complements of these assets (operational analysis, capabilities, architecture, etc.) were done. Thus, MBSE has accelerated learning by building and revising models of the intentional architecture.

- **Organization and exploitation of models:** Each capability of the system was assigned to a capability leader (cf below), who was accountable for the associated functional chains. The capability leader coordinated the co-engineering with integration, verification, and validation (IVV) and software teams on their capability iterations after iterations (Figure 6).

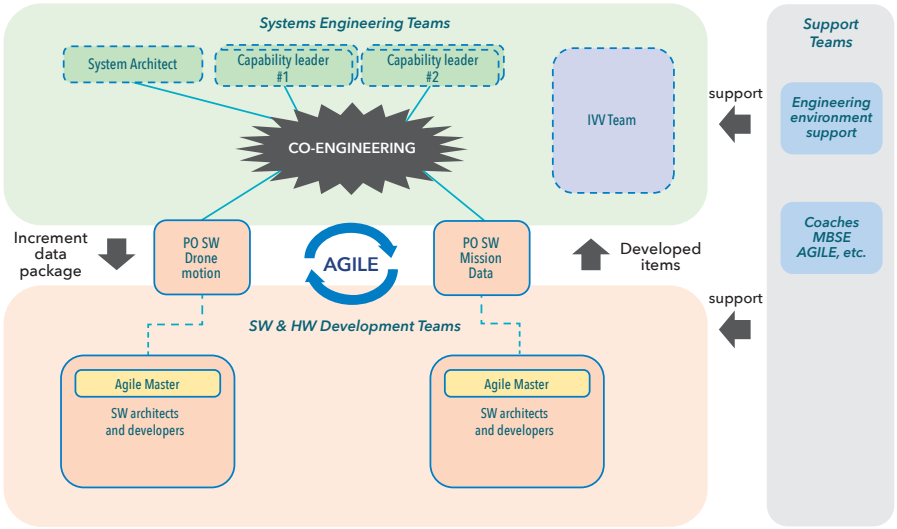


Figure 6. Example of Pythagoras organizational breakdown structure (OBS)

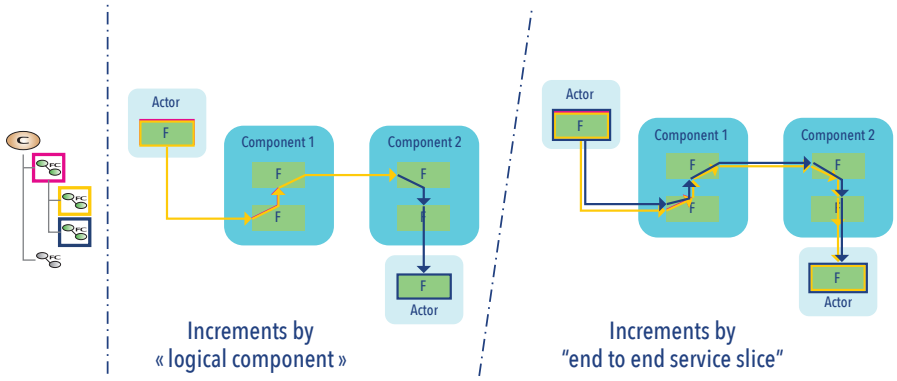


Figure 7. Increments by logical components or by end-to-end service slice

The results of these workshops as well as the model-based engineering strategy were then formalized in the systems engineering management plan.

Deal with the uncertainty of the needs: We identified while modelling what we don't know and variability points. We capitalized information in the model that could be for example inputs for decision to eliminate some architecture alternatives in a set-based approach.

Thanks to the testimonies of a previous project, we had learned the lessons that this warmup phase shall be performed with a sufficient scoping and not skipped.

6. ***How did you perform the run, designing, and implementing the increment design data packages?***

First, we defined the content of increments. It could be by for example an "end-to-end service" slice. As an alternative, it could be logical component, with the help of a simulator to simulate inputs/outputs and behaviors. Each slice, once implemented, is fully functional, creates value for the user, and makes user feedbacs possible. Taking care of the compatibility of interfaces between components is key especially for IVV work (with for example the delivery of interface data with identification of their version in increment data packages) (Figure 7).

Then we delivered the architectural design produced by systems teams to software and IVV teams, which was based on the capabilities and associated functional chains or scenarios describing them. In the example below, increment data package relates to the functional chain "manually control the drone motion."

Representative members of the software team participated to regular reviews of the increment data package current build by the systems team. Their role was to anticipate the feasibility and to make sure the system-level vision of the solution was compatible with the current software architecture. Participation of software architects in the agile co-engineering effort at system level was key for the developers to "accept" the models they will receive from the systems engineers. This effort helped to secure the design. In such reviews, there had been the presentation of physical architecture blank displaying the functional chain "manually control the drone motion," with the physical components involved (for example, micro controller, etc.), the expected behavior of these physical components, the operator external entity...

The development team received then for a run iteration n+1 the increment data package related to this functional chain, (cf Figure 8). Having inside the increment functional chain or scenarios (or composition or pieces of them) helped the team to better understand how the implementation of a piece of interface or a small feature fits in the product-wide picture. It helped them also to split in EPIC, applying the agile precepts, to refine the received EPICs in user stories that were developed in successive blocks. The content of the user stories was defined so that value (working software) was delivered after each block. For example, a block was about plugging the actual drone motion to the piloting graphical interface on the tablet. This increment data could contain system mode machine from solution perspective such as below, textual requirements associated to model elements, etc. (Figure 9).

The pace of the IVV team was aligned with the pace of the software development team. The releases were driven by IVV,

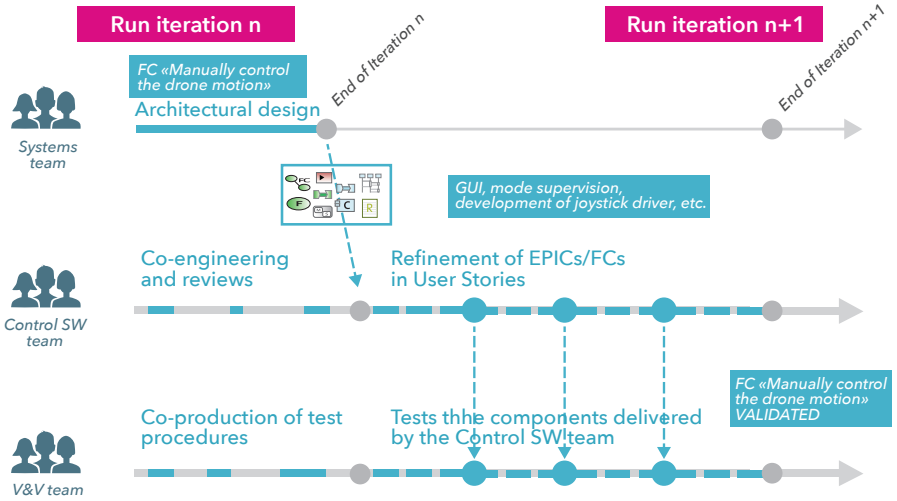


Figure 8. An increment package dispatched to other agile teams at the end of iterations

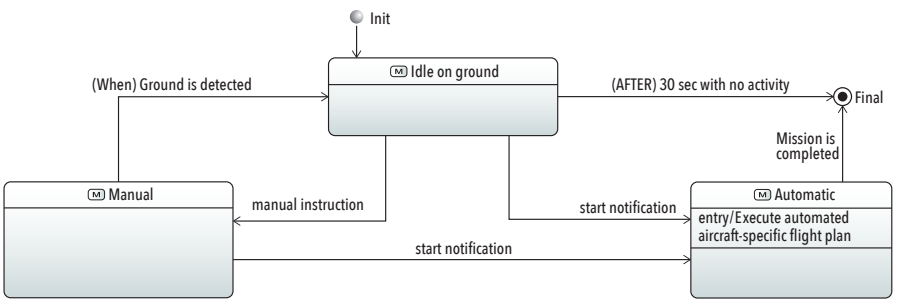


Figure 9. A piece of increment data package for the functional chain "manually control the drone motion"

Capability	% designed	% developed	% validated
Manually pilot the drone	40%	20%	10%
Automatically follow a flight plan	40%	20%	20%
Manually acquire data	30%	10%	10%
Automatically acquire data	60%	40%	40%
Visualize data during mission execution	70%	70%	30%
Visualize data after mission execution	100%	50%	40%
Analyze data during mission execution	0%	0%	0%
Analyze data after mission execution	50%	20%	0%

Figure 10. Progress status of design, development, and test per capability at the end of iteration 3

allowing to test end-to-end services with system integration. The IVV team integrated in the system the components delivered by the software team every third week and run the test procedures written collaboratively in co-engineering during the previous iteration. Each test procedure "tells the story" of its corresponding functional chain, the test steps roughly matching the steps of the functional chain. To obtain this result, IVV practitioners worked closely with the capability leaders in order to translate each need-perspective functional chain in a corresponding system-level test procedure. Models helped to share the design.

When a problem was encountered on a test step, finding the corresponding function or functional exchange in the model was straightforward. Using automated impact analysis, the investigation on the related data was also immediate. It is straightforward to locate the possible cause of a problem. This analysis of the model can have different outputs. If the model (need and corresponding solution) is correct, a defect is created on the faulty component. If the model is actually faulty, then a defect is created on the model itself, and an evolution request is created for the involved component.

7. ***How did you evaluate the design iteratively?***

For example, we organized a review of experts and the simulation about the product increment of the last iteration. We synthetize these results and run performed IVV results in a table such as in Figure 10 to evaluate and monitor the progress status. We also performed a retrospective to take a step back and improve our engineering practices.

8. ***When did the process of warm up/run/evaluation end?***

The process of warm up/run/evaluation ended when all the capabilities in Figure 2 were all released and accepted by the customer. Then the life cycle of the system transitions from development to full operation by the customer. The engineering organization is also an active actor of the evolutionary maintenance of the system: a "warm-up" iteration is currently performed to prepare the engineering teams for this new phase. We reuse both the previous work capitalized in the model and the previous process exposed.

9. ***How did you deal with an evolution request in a MBSE and agile context?***

Agile with MBSE helps to bring the value proposition and short loop for customer feedback. For example, during customer visibility milestones, the functional chains "manual drone control with joystick" and "manual drone control with tablet" were released and validated by the customer. Additional needs (obstacle avoidance and switch between automated piloting and manual piloting) expressed during this milestone were captured in the need-perspective model. We performed impact analysis with the help of queries or any other form of data extraction from the model to precisely compute the consequences of these evolution requests and consequently discussed with the customer to bring value corresponding to its request in further iterations.

CONCLUSION

In this article, we illustrated how MBSE may be an effective enabler of agility in systems engineering, focusing on dynamic learning and evolution (cf concepts of the FuSE roadmap). MBSE accelerates learning by building and revising models since the early stages and helps explore and get agreement on solutions when evolution is requested. Key concepts used in the contact of MBSE with agile were presented, then the process of "warm-up/run/evaluation" was detailed and we finished with the way to deal with an evolution request in a MBSE and agile context. Combining both approaches may help to:

- Organize and synchronize the development and validation effort of one or multiple engineering teams.
- Faster impact analysis including trade-off studies/options and hence a faster reaction to evolutions in expectations and constraints, that is, the agility of systems.
- Show regularly "end-to-end" value to the customer and other stakeholders. ■

ABOUT THE AUTHORS

Sophie Plazanet has been working in system engineering for several years, especially during the past 5 years in Thales. Passionate about MBSE, she joined Thales Corporate Engineering in 2021 where she is a MBSE coach, supporting the Thales engineering teams to adopt MBSE practices. She holds a Master of Engineering and Master of Research in advanced systems and robotics from Arts & Métiers ParisTech Engineering School.

Juan Navas is a systems architect with 15 years' experience on performing systems engineering activities and implementing innovative engineering practices in multiple organizations. He currently leads the modelling and simulation team at Thales Corporate Engineering and dedicates most of his time to expertise and consulting for Thales business units and other organizations worldwide, accompanying managers and architects when implementing MBSE practices. He holds a PhD in embedded software engineering, a MSc degree in control and computer science, and a degree in electronics and electrical engineering.

[Editor: Author biographies were current when the paper was initially published in 2023.]

INCOSE VOLUNTEER OPPORTUNITY

THE BEST ENGINEERS ALLOW FOR A LITTLE GIVE.

Become an INCOSE volunteer today!

incose.org/volunteer

INCOSE
A better world through a systems approach

Systems Engineering: The Journal of The International Council on Systems Engineering

Call for Papers

The *Systems Engineering* journal is intended to be a primary source of multidisciplinary information for the systems engineering and management of products and services, and processes of all types. Systems engineering activities involve the technologies and system management approaches needed for

- definition of systems, including identification of user requirements and technological specifications;
- development of systems, including conceptual architectures, tradeoff of design concepts, configuration management during system development, integration of new systems with legacy systems, integrated product and process development; and
- deployment of systems, including operational test and evaluation, maintenance over an extended life-cycle, and re-engineering.

Systems Engineering is the archival journal of, and exists to serve the following objectives of, the International Council on Systems Engineering (INCOSE):

- To provide a focal point for dissemination of systems engineering knowledge
- To promote collaboration in systems engineering education and research
- To encourage and assure establishment of professional standards for integrity in the practice of systems engineering
- To improve the professional status of all those engaged in the practice of systems engineering
- To encourage governmental and industrial support for research and educational programs that will improve the systems engineering process and its practice

The journal supports these goals by providing a continuing, respected publication of peer-reviewed results from research and development in the area of systems engineering. Systems engineering is defined broadly in this context as an interdisciplinary approach and means to enable the realization of successful systems that are of high quality, cost-effective, and trustworthy in meeting customer requirements.

The *Systems Engineering* journal is dedicated to all aspects of the engineering of systems: technical, management, economic, and social. It focuses on the life-cycle processes needed to create trustworthy and high-quality systems. It will also emphasize the systems management efforts needed to define, develop, and deploy trustworthy and high quality processes for the production of systems. Within this, *Systems Engineering* is especially concerned with evaluation of the efficiency and effectiveness of systems management, technical direction, and integration of systems. *Systems Engineering* is also very concerned with the engineering of systems that support sustainable development. Modern systems, including both products and services, are often very knowledge-intensive, and are found in both the public and private sectors. The journal emphasizes strategic and program management of these, and the information and knowledge base for knowledge principles, knowledge practices, and knowledge perspectives for the engineering of systems. Definitive case studies involving systems engineering practice are especially welcome.

The journal is a primary source of information for the systems engineering of products and services that are generally large in scale, scope, and complexity. *Systems Engineering* will be especially concerned with process- or product-line–related efforts needed to produce products that are trustworthy and of high quality, and that are cost effective in meeting user needs. A major component of this is system cost and operational effectiveness determination, and the development of processes that ensure that products are cost effective. This requires the integration of a number of engineering disciplines necessary for the definition, development, and deployment of complex systems. It also requires attention to the lifecycle process used to produce systems, and the integration of systems, including legacy systems, at various architectural levels. In addition, appropriate systems management of information and knowledge across technologies, organizations, and environments is also needed to insure a sustainable world.

The journal will accept and review submissions in English from any author, in any global locality, whether or not the author is an INCOSE member. A body of international peers will review all submissions, and the reviewers will suggest potential revisions to the author, with the intent to achieve published papers that

- relate to the field of systems engineering;
- represent new, previously unpublished work;
- advance the state of knowledge of the field; and
- conform to a high standard of scholarly presentation.

Editorial selection of works for publication will be made based on content, without regard to the stature of the authors. Selections will include a wide variety of international works, recognizing and supporting the essential breadth and universality of the field. Final selection of papers for publication, and the form of publication, shall rest with the editor.

Submission of quality papers for review is strongly encouraged. The review process is estimated to take three months, occasionally longer for hard-copy manuscript.

Systems Engineering operates an online submission and peer review system that allows authors to submit articles online and track their progress, throughout the peer-review process, via a web interface. All papers submitted to *Systems Engineering*, including revisions or resubmissions of prior manuscripts, must be made through the online system. Contributions sent through regular mail on paper or emails with attachments will not be reviewed or acknowledged.

All manuscripts must be submitted online to *Systems Engineering* at ScholarOne Manuscripts, located at:

https://mc.manuscriptcentral.com/SYS

Full instructions and support are available on the site, and a user ID and password can be obtained on the first visit.

Free Download:

"Enterprise Transformation Planning" with UAF and CATIA Magic eBook

DASSAULT SYSTEMES