31st Annual INCOSE international symposium
virtual event
July 17 - 22, 2021

Agility in the Future of Systems Engineering (FuSE) - A Roadmap of Foundational Concepts

Keith D. Willett
US Dept of Defense
Keith.Willett@incose.org

Rick Dove
INCOSE
dove@parshift.com

Alan Chudnow
Peraton
Alan.Chudnow@Peraton.com

Rusty Eckman
Northrop Grumman
Emmet.Eckman@ngc.com

Larri Rosser
Raytheon
larri.rosser@raytheon.com

Jennifer S. Stevens
NASA
jennstevens.se@gmail.com

Robin Yeman
Lockheed Martin
robin.yeman@lmco.com

Mike Yokell
Raytheon
smichael.yokell@raytheon.com

Abstract. The *Future of Systems Engineering* (FuSE) is an INCOSE led multi-organization collaborative activity focusing on many initiatives to identify and shape the future of systems engineering. The *FuSE Agility* collaboration identifies and elaborates a roadmap for an initial set of foundational concepts to further the integration of agility into the systems engineering lifecycle. This paper identifies four objectives for agility integration in people, process, technology, and environment and aligns nine foundational concepts to advance thinking and practice in agile-systems engineering (solutions) and agile systems-engineering (process).

Introduction

The *Future of Systems Engineering* (FuSE) is an INCOSE led multi-organization collaborative activity focusing on many initiatives (Figure 1) to identify and shape the future of systems engineering and contribute toward the practical realization of INCOSE's *Vision 2025*. The FuSE Agility collaboration identifies and elaborates a roadmap for an initial set of foundational concepts to further the integration of *agility* into the systems engineering lifecycle. This paper identifies nine foundational concepts and aligns them to four objectives. The objectives are to integrate agility in 1) agile operations, 2) agile-systems engineering (technology), 3) agile systems-engineering (process, development), and 4) agile-workforce (people). The intent of these foundational concepts is to instigate and inspire thought, research, development, and implementation of agile systems.

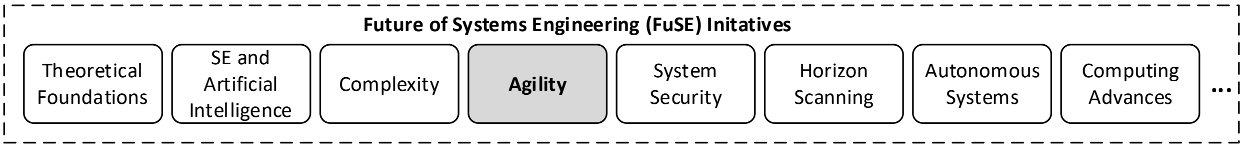

Figure 1: FuSE Agility in Context of FuSE Initiatives

The FuSE Agility project purpose is to provide a roadmap for how systems engineering can improve and advance the effectiveness of system agility in the near term. The problem is traditional systems engineering does not clearly enable or facilitate agility. The need is for strategies and plans for actionable concepts to advance agility and to remove barriers to agility. The intent is to provide concept descriptions sufficient to inspire and instigate individual concept development and implementation action in the broad-based systems engineering community.

Figure 2 shows the charter that guided the team's work during calendar year 2020, evolving with lessons learned. On the left side are three timeframes: eventual, near term, and immediate. The initial focus in 2020 was to identify a reasonable and actionable list of foundational concepts to support achievement of the near-term and eventual goals.

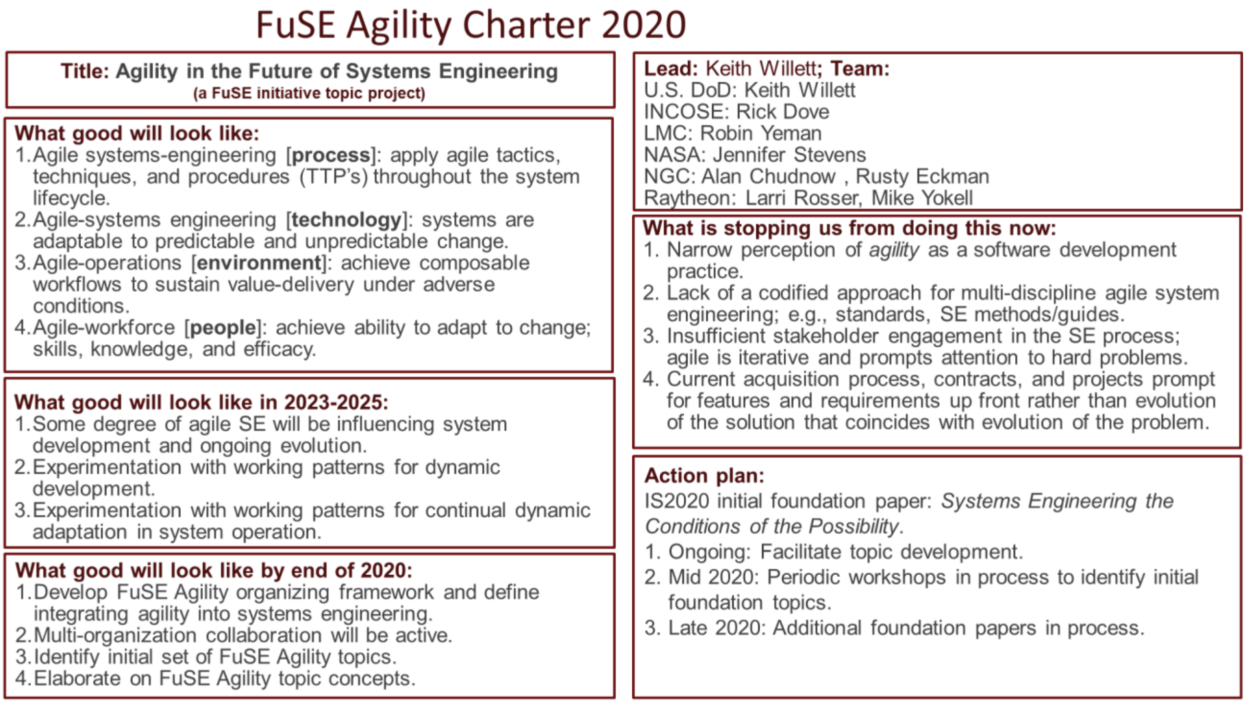

Figure 2: FuSE Agility Charter 2020

Framing Agility

Agility is the ability to move quickly and easily; speed with quality. For software development, the agile movement transitioned away from waterfall models to new techniques for more rapid product creation and for more effective adaptation to changes in the environment and marketplace. A FuSE Agility goal is to apply the concept of agility to all engineered systems across people, process, technology, and environment – well beyond just software. Some key agile concepts include:

- Orchestrate planning, design, integration and test and other life cycle activities to balance against the uncertainty that many assumptions about the product/market fit will be incorrect.
- Get frequent feedback via capability demonstrations that validate assumptions with potential or actual users, quantify improvements, and measure product/market fit.
- Work in small batch sizes to allow for more frequent feedback and adapt to market changes.

- Use metrics and reflection to improve and adapt to new market changes.
- Enable and empower teams to manage their workflow efficiently and consistently.

Table 1 provides one organizing framework for agility with intent to prompt and focus discussion on the questions why be agile, what can or should be agile, and how to manifest agile across people, process, technology, and environment.

Table 1: Organizing Framework for FuSE Agility

Interrogative	**Description**
Why (goal)	• Reduce the time from system engineering concept to deployment and user. • Improve the ease and speed of adaptation to change in the environment, mission, and marketplace. • Improve the effectiveness of system engineers, systems engineering processes to adapt to change. • Improve the architecture and design of systems to reduce the cost and schedule required to change and adapt to change. • Sustain value-delivery; remain viable and relevant in the face of adversity. • Improve team workflow
How (process, tactics, techniques, procedures (TTP's))	OODA+C2 (observe, orient, decide, act, command, and control). Static options (playbooks), dynamic options (composability).
What (things, material, solutions)	That which constitutes a system. System characteristics: structure, behavior, contents, resources, environment, and value-delivery.
Who (roles, responsibilities)	Stakeholder roles; e.g., owner (stockholder), operator/user (pilot), beneficiary (passenger or cargo shipper). Terms that are meaningful to the stakeholder (stakeholder currency).
When (triggers)	Predictable and unpredictable events; stimulus/response, periodicity.
Where (environment)	Containing whole (system of systems), broader marketplace and environment (ecosystem).

For purposes of FuSE Agility, a system can be social (people), a process, technical (engineered system), environment (engineered or natural), or any combination; e.g., socio-technical, cyber-physical, or system of systems. That which can be agile includes people (agile-workforce), process (agile-development), technology (agile-systems, adaptable solutions), and environment (agile operations, adaptable workflows) which correspond to the FuSE Agility Charter (Table 1) 'What good will look like.' Any system characteristic can be agile including structure (organization of parts (composition), state), behavior (function, functional exchange), content (real (cargo, people), virtual (data)), resources (raw material, energy/fuel), environment (containing whole, ecosystem), and value-delivery (viable, relevant).

The primary goal of *all* systems is to provide value-delivery under nominal conditions. A goal of *some* systems is to sustain value-delivery under adverse conditions. This implies the need for the system to remain viable (capable of operating successfully) and relevant (conform to current conditions) (Willett 2020b). Agility is one of many domains that contribute to system viability and relevance.

Given a stimulus that threatens viability or relevance, the system may respond with some adaptation. We encode some responses into discrete systems with simple rules or logic gates. Some responses require governance and adjudication logic to provide broader consideration of a system-of-systems and interactions within the ecosystem. The concept of *orchestration* includes command and control. Command provides governance and adjudication logic/rules to actuate external behavior and perform internal behavior for adaptation. Control includes the messaging infrastructure and message set to carry command decisions to the constituent parts of workflow. Stimuli for agile actions include predictable and unpredictable. To accommodate unpredictability, encoding agility includes *observe* (monitor with intent to raise awareness), *orient* (understand within context), *decide* (identify and select among viable options), and *act* (do something). This may occur in the broader sense of operational workflow (distributed independence) or within command and control that guides workflow execution.

Focusing FuSE Agility

To help focus FuSE Agility, Figure 3 displays the Agile Systems Engineering Life Cycle Model pattern (Schindel and Dove 2016), designating three intimately related systems.

- **System 1**: the target system under development.
- **System 2**: the system that produces, supports, and learns about the target system. This is the logical system within which the target system will exist during its lifecycle.
- **System:3**: the process improvement system, called the system of innovation that learns, configures, and matures system 2. System 3 is responsible for situational awareness, evolution, and knowledge management – the provider of operational agility.

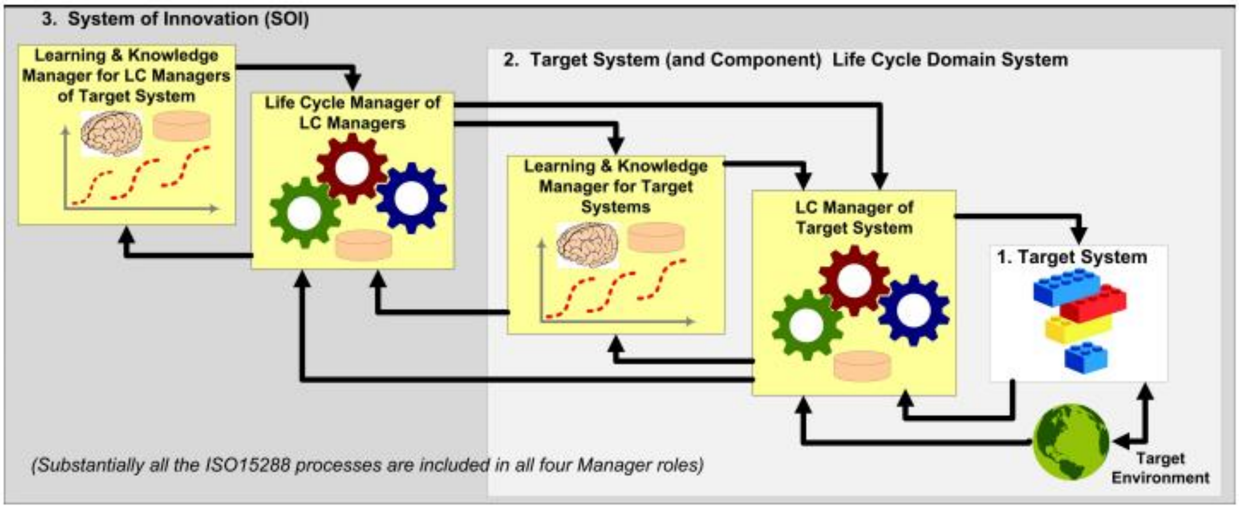

Figure 3: Agile Systems Engineering Life Cycle Model Pattern (Schindel and Dove 2016)

FuSE Agility Concept Descriptions

Expressing a concept as a what/why encourages multiple perspectives and different approaches on how to achieve it; i.e., there are multiple paths to the destination and people find varying inspiration to travel one path versus another. Criteria for FuSE Agility foundation concepts are:

- Has relevance to systems engineering considerations.
- Can provide new and useful value to the state of the practice.
- We can articulate concept value proposition in systems engineering terms.
- There is a referenceable knowledge base that supports the concept.
- Does not yet have sufficient published exposure for actionable systems engineering consideration.
- Is implementable now.

- Has sufficient ecosystem/infrastructure in place to support implementation.
- Is principally about what to do and why (strategic intent), rather than how (prescriptive tactics), though examples of how can augment understanding.

This section provides a brief description of each of the nine FuSE Agility foundational concepts (Table 2). The intent of the concept descriptions is to inspire and instigate open community concept development. The focus is on strategic intent, leaving ample room for various approaches.

Table 2: FuSE Agility Foundation Concepts

#	Concept
1	System of Innovation
2	Technical Oversight for Agile Projects
3	Effective Stakeholder Engagement
4	Agility Across the Value Stream
5	Orchestrating Agility with Long Lead Time Components
6	Continual Integration
7	Orchestrating Agile Operations
8	Option Management for Dynamic Adaptation
9	Harmonizing Risk in Agile Operations

Each concept description provides details on a common set of key point categories (Table 3) to guide concept elaboration, focus, interrelationships, and further investigation.

Table 3: Key General Concept Points

Problem	Problem addressed by the concept
Need	Need to solve the problem
Barriers	Description of that which stops us from achieving the concept
Intent	Strategies to address the need
Value	Values to realize using the strategies
Metrics	Metrics for measuring effectiveness of strategies
Notions	Example references to inspire strategy development

This round of foundational concepts finds motivation in the interests of the authors. Subsequent concepts will find similar motivation as well as the need to address new problems and to fill gaps that emerge from exploring the current set of concepts. The concept of *System of Innovation* is central to the current FuSE Agility concepts and contributes to all four objectives of agile systems-engineering (process), agile-systems engineering (solutions), agile operations, and agile workforce (people). Included in the *System of Innovation* is knowledge management to support technical oversight and project performance with formal knowledge capture and reuse of agile methods and lessons learned. *Technical Oversight for Agile Projects* examines flexible oversight for projects that by design will change requirements throughout. *Effective Stakeholder Engagement* highlights the need for stakeholder awareness and active participation in a process of ongoing adaptability. The value stream now consists of ongoing iterations where previous organizational silos must now collaborate and coordinate requiring *Agility across the Value Stream*. *Orchestrating Agile with Long Lead Time Components* addresses the influence of delays in receiving raw materials and components in an agile development process.

Continual Integration examines continual adaptation of fielded solutions (agile-systems) and continual adaptation of the use of those systems (agile operations). *Orchestration* begins to examine a path toward autonomous command and control. *Option Management for Dynamic Adaptation* is a survey of mathematical disciplines that contribute to encoding governance and adjudication logic in orchestration. *Harmonizing Operational Risk* examines the emerging concepts of loss-driven systems engineering (LDSE) and opportunity-driven systems engineering (ODSE) (Willett 2020b) as part of orchestration for agile operations.

Figure 4 shows the relationship of each concept to the four main objectives from the charter in Figure 2. Following the diagram are the concept descriptions for the nine foundational concepts.

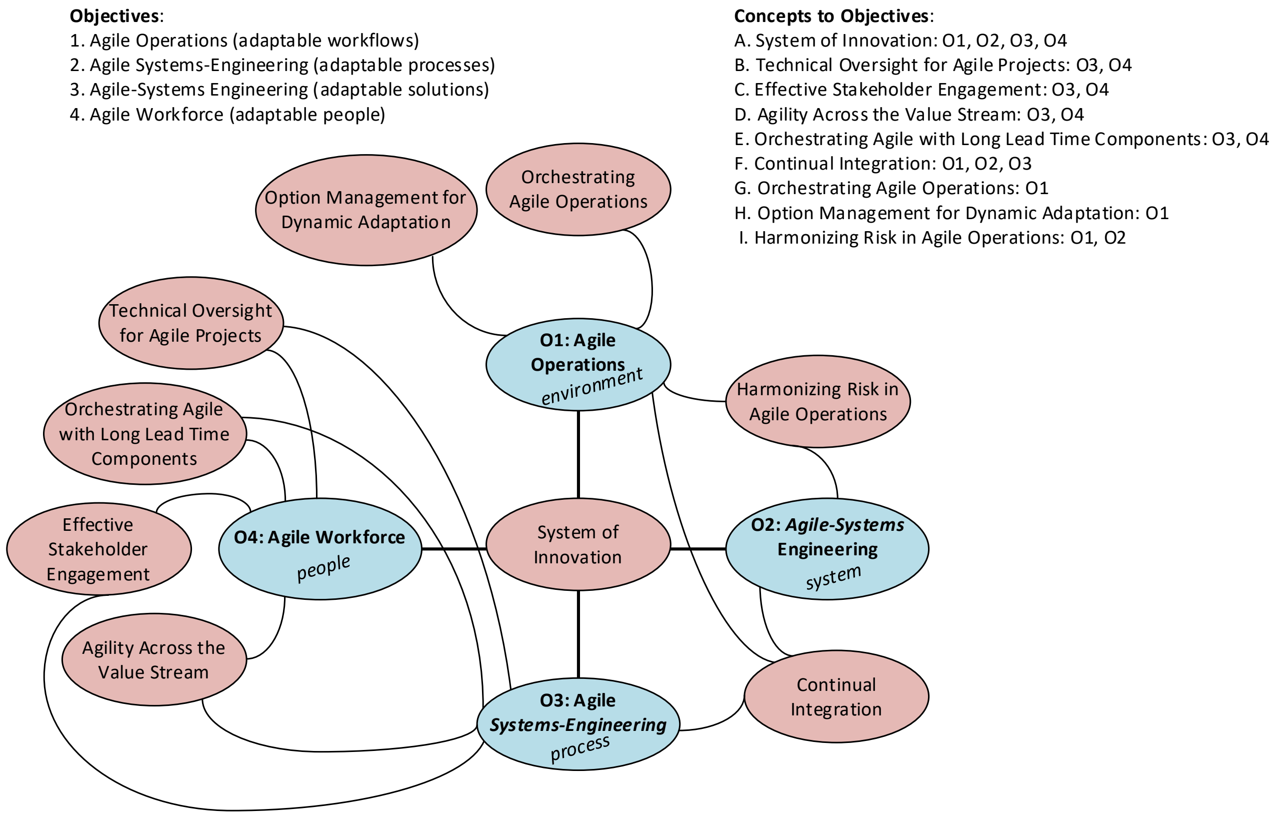

Figure 4: Mapping FuSE Agility Concepts to FuSE Agility Objectives

System of Innovation

Problem	Insufficient learning activity and knowledge management; barriers to knowledge application.
Need	Situational awareness and learning embedded in lifecycle processes; timely/affordable learning-application enabled; knowledge management.
Barriers	Unclear what to do or where to do it beyond learning ceremonies and contract obligation satisfaction. Comfort in static view of requirements and environment.
Intent	Explore the application of three core principles: sense, respond, and evolve.
Value	Less rework (cost/time); higher customer/user satisfaction; competency growth.
Metrics	Relevance of knowledge; impact of applied learning.
Notions	(Schindel and Dove 2016), (Schindel 2017), (Dove 2020).

Systems engineering codifies a collection of interrelated processes spread throughout system life-cycle stages (e.g., ISO/IEC/IEE 15288, ISO/IEC/IEE 24748). We may accomplish agility in systems engineering by integrating additional capabilities into these stages that enable contextually relevant responses; e.g., encoding and reusing knowledge. Collectively we refer to these agility capabilities as the *System of Innovation* (Schindel and Dove 2016).

The System of Innovation functions as the learning and lifecycle manager. Agile systems engineering embeds and distributes it throughout the lifecycle processes at relevant points. Its core is the central Situational Awareness stage that triggers entry into all other stages in the Agile Systems Engineering Lifecycle Model (Dove 2020).

The System of Innovation operationalizes agility with three basic principles: sensing, responding, and evolving.

Sensing (Observe, Orient)
- External awareness (proactive alertness)
- Internal awareness (proactive alertness)
- Evaluation (learning: outcome of sensing, process of sensing)

Responding (decide, act)
- Decision making (timely, informed)
- Action making (reconfigure process and product to fit the situation)
- Evaluation (learning: outcome of responding, process of responding)

Evolving (improve above with learned knowledge and capability)
- Experimentation (variations on process and product)
- Evaluation (learning: outcome of evolution, process of evolution)
- Memory (culture, capabilities, ConOps, knowledge management)

How these principles apply in a project context and in an institutional context are ripe for concept development.

All systems engineering processes have at least a tacit System of Innovation. Agile software development processes begin to get explicit with periodic retrospectives on both process and product work-in-process outcomes. But for complex SE projects we must identify what to monitor and sense to avoid undesirable outcomes, what response capability to affordably deploy, and how to coherently evolve in process and product; plus, identify where in the lifecycle processes these activities occur and the roles responsible to perform each activity.

Technical Oversight for Agile Projects

Problem	Current technical oversight approaches (e.g., Stage-Gates reviews) are not agile. They take too much calendar time, too much team effort, are not adequately responsive to continuous unpredictable change, and do not provide insight into gaps and risk on agile programs. The Waterfall model has a long lag between design reviews at the beginning and test reviews near the end.
Need	A light weight, interactive approach to technical oversight that provides insight in the form of good predictive feedback to agile programs with minimal burden of labor on the agile team. Balance reviews costs vs. schedule vs. benefits.
Barriers	Fixed expectations of the oversight process; contractual constraints; and the incorrect assumption that agile programs don't need technical oversight.
Intent	Make technical oversight agile; i.e., frequent, quick, useful feedback that provides insight into project performance against commitments, environmental change vs planned capabilities and schedule, and recommendations.

Value	Insight at the speed of relevance.
Metrics	Feedback relevance; feedback accuracy; feedback cycle time; oversight labor; ROI (OS labor: cost avoidance from oversight).

As product development becomes more agile, the speed and efficiency of external interactions can become governing factors on speed and efficiency of the program. This either limits the *response ability* of the process system, or discards the external interaction *not agile*, removing any benefits of the interaction from the process system.

External technical oversight provides a case in point. The goal of technical oversight is to provide insight into gaps and risks that the program team may miss. While this external assessment has proven value, the use of a stage-gate oversight approach adds drag to agile development. In some cases, this leads to longer cycle times and limitations on long term velocity. In others, it results in abandonment of technical oversight, which increases risk and rework along with lower product quality. In Waterfall approaches, key reviews happen early in the program. For many programs, there is often a long duration between design reviews (e.g., SRR, PDR, CDR) and testing. During this period, there is often little feedback. Because this period is lengthy, there can be significant change in the environment that impacts product/market fit.

The strategic goal of this activity is to re-envision technical oversight in a way that aligns with the principles and needs of agile projects. The new approach needs to ask the right questions to produce relevant proactive guidance in a way that does not constrain the velocity or cycle time of the program or require excessive labor on the part of either programs or overseers. The approach will likely focus on in-line continuous or short-cycle evaluation as opposed to external event-based reviews and should leverage existing attributes of agile programs such as automation and increment planning. The desired outcome is insight at the speed of relevance.

Provide insight and responsive forward-looking actionable guidance for agile programs using an approach that produces minimal drag and disruption and keeps pace with agile product development.

Effective Stakeholder Engagement

Problem	Timeliness, frequency, and depth of stakeholder collaborative engagement.
Need	Discovery of integration conflicts and true requirements as they evolve over time.
Barriers	Time involved; travel cost; inconvenient scheduling; lack of motivation.
Intent	Enable and facilitate compelling collaboration, cooperation, and teaming among all relevant stakeholders.
Value	Less rework (cost/time); higher customer and user satisfaction.
Metrics	Breadth and depth of stakeholder engagement; time and cost of rework. Lead time, cycle time, defect density.
Notions	(Dove, Schindel, Scrapper 2016); (Dove. Schindel, Garlington 2018).

Systems engineering benefits when the various stakeholders participate as a collaborating cooperative project-encompassing team. But participation comes in degrees of engagement. At the low end there is simple presence at occasionally scheduled work-in-process reviews. At the high end there is comprehension, inclusion, and contribution at frequent ad-hoc project progress and issue collaborations. The effectiveness of an agile systems engineering process depends on the timeliness and depth of engagement by stakeholders. This concept addresses core principles and common strategies for improving the effectiveness of stakeholder engagement in all forms.

An engagement process will have many different activities to satisfy different needs at different times for different stakeholders. Stakeholders of interest may include managers, system engineers, development engineers, subcontractors, producers, operators, maintainers, customers, and end users. Engagement is a social activity of collaborative exchange that may occur in a variety of ways, including synchronously and asynchronously, face-to-face and virtually, textually with wikis and commercial project status tools, and experientially with interactive demonstrations.

Every project includes a stakeholder engagement process consisting of a set of activities and procedures for conducting those activities. Stakeholder engagement activities and procedures generally are parts of other project processes, and not viewed collectively as a system with a common set of social requirements that are addressed by design strategies for effectiveness. A coherent engagement process facilitates collaboration for relevant information exchange among individuals, cooperation for optimal give and take among individuals, and teaming for collective endeavor toward common purpose. Engagement effectiveness depends on the experiential quality of the engagement activities of individual stakeholders. Effective engagement is comfortable, timely, and rewarding.

Agility Across the Value Stream

Problem	Multiple handoffs across organizational boundaries lead to slower lower quality products.
Need	Common language; minimize handoffs, product-based teams; common metrics
Barriers	Organizational silos
Strategic Intent	Enable customer -centric product-based delivery with low complexity and higher speed
Value Proposition	Adaptability to increase quality and speed, lower cost, and reduced risk
Metrics	Lead time, cycle time, defect density
Notions	Flow-based delivery; industrial DevOps

From the start of value creation through delivery requires multiple handoffs across many organizational silos each with different perspective, focus, language, motivations, and priorities. The increasing complexity levels of large-scale safety-critical systems magnify Conway's Law by virtue of so many organizational silos directly and indirectly influencing the value stream. The increase in product influence includes a decrease in intra-organizational communication, data, and general understanding across organizational boundaries.

Leveraging agile principles and practices throughout the value stream will help accelerate the rate of product delivery with higher quality and less cost. Expanding the *agility focus* to the *system of systems* level will magnify the results previously seen only in context of software development. This requires change to the system architecture as well as the culture:

System Context

- System structure
- Quantitative information flow
- Heterogeneous elements
- Emergent behavior
- Interfaces
- Nomenclature

Cultural Context

- Organizational structure
- Qualitative information flows
- Heterogeneous subculture
- Mental models
- Relationships
- Language

Orchestrating Agility with Long Lead Time Components

Problem	System under development needs to address components that can be developed quickly, components that take longer, and external dependencies. Components and external dependencies with long lead times complicate schedule coordination and disrupt technical performance.
Need	Scheduling and acquisition techniques that better align with agile-SE principles.
Barriers	[False] justification that long-lead items prohibit the use of agile-SE.
Intent	Clarify how agile-SE can accommodate long-lead time acquisition.
Value	Reduce long-term cost and risk; quicker time to market.
Metrics	Reduce non-productive wait time, integration effort, and rework.
Notions	Integrated master scheduling, giver/receivers, minimum viable product (MVPs), minimum viable capability delivery (MVCD) workarounds, trade studies, invest in alternatives.

Agile-development programs aim to get to market and mission quickly; and, to mitigate the risk that a changing environment will invalidate assumptions about product-market fit. Agility helps modulate capabilities and to match changing customer needs.

A long-lead item is a component or external dependency necessary for capability development and operations. Long-lead time is relative to the agile-development increments, epics, or sprint durations. Examples of long-lead items include hardware procurements, building construction, civil zoning permits, unique testing setups (e.g., rocket testing), accreditation and approvals (e.g., FDA, safety, cybersecurity accreditation, environmental impact), and very hard technical problems (e.g., nuclear fusion).

Components and external dependencies that take a long time significantly add to program risks. The long-lead items may not be available on schedule. The item may not meet technical expectations, complicating integration, test. Customer and user needs may change between program start and long-lead item delivery.

Traditional system engineers use several techniques today to mitigate these issues including Integrated Master Schedules (IMS), giver-receiver commitments, trade studies, and Risk and Opportunity Management (ROM) / risk mitigation investments. These techniques can work with or against agile principles: get to market quickly, work in small batch sizes, perform continuous integration, and provide frequent customer demonstration, experimentation, feedback). This concept explores TTPs that harmonize the agility of software development with the realities of needing long-lead times components.

Continual Integration

Problem	Late discovery of integration and requirements issues.
Need	Minimize risk and rework; maximize stakeholder engagement.
Barriers	Development effort and expense. Technologies for integrating/testing software prior to hardware being ready.
Intent	A Live-Virtual-Constructive platform for early and continual integrated testing and work-in-progress demonstrations.
Value	Less rework (cost/time); effective stakeholder engagement.

Metrics	Rework reduction; stakeholder value statements.
Notions	(Dove, Schindel, Garlington 2018); (Dove et al. 2020).

Component to component integration and system testing is often performed relatively late in the life cycle. Integrating components for system testing often reveals incompatibilities that cause rework. The later the rework in the development cycle, the higher the cost and time of rework. Mixed-domain cyber-physical systems have components with different development times, where many months for completion are typical.

This concept is about enabling integration as early as possible before the completion of system components. Integration issues caused by incompatible or insufficient specifications can be revealed by integrating proxy components: component simulations, off-the-shelf lower fidelity components, component prototypes, and even component work-in-process (Dove, Schindel, Garlington 2018).

Component proxies can evolve through states of increasing maturity. Stakeholders can see demonstrations of incremental work in process and interact with early-stage work.

Setting up integration-test events and demonstrations could be facilitated by an LVC-like Platform that can configure an integrated system from a selection of proxy, intermediate, and completed components. System components in whatever form might include a mix of locally present and remotely connected.

Orchestrating Agile Operations

Problem	Disparate solutions operate independently.
Need	Tightly coupled coordinated dynamic operations in real-time.
Barriers	Ability to encode self-learning, adaptive logic as decision-support to people and for autonomous decision making.
Intent	Elaborate orchestration as command and control for a system; and advance thinking on *command*.
Value	Fast adaptable system operation.
Metrics	Increase in autonomous system defense. Less people in-the-loop.
Notions	Integrated Adaptive Cyberspace Defense (IACD) – JHU Applied Physics Laboratory.

For systems engineers who design decision support to people directing systems and/or autonomous self-directing systems, the concept of *Orchestrating Agile Operations* explores governance and adjudication logic and rules to guide dynamic adaptable workflows (agile operations). Sustaining system viability and relevance may require adapting an individual system, adapting many members of a system of systems, or may require workflow redesign in real-time to avoid, withstand, or recover from an adversity. Governance logic addresses strategic goals (interests over time) and tactical objectives (immediate interests) and is the logic behind *Control*. Adjudication logic addresses tensions between strategic and tactical; and, among contending domains like viability, relevance, performance, safety, security, resilience, survivability, and sustainability. Successful resolution of tension is context dependent; i.e., the same stimulus may result in a different response within varying contexts.

Orchestration guides workflow execution to provide value-delivery under nominal conditions and sustain value-delivery under adverse conditions. Orchestration consists of *command*-and-control features and functions. *Control* is a messaging infrastructure and message set to direct constituent parts of the SoI. The messages are simple like START, STOP, BLOCK, and ALLOW among others. *Command* is the governance and adjudication logic and rule sets to make decisions to the current and ever-changing context of the SoI. Currently, *command* is predominantly manual with automation consisting of simple rule sets or logic gates. As artificial intelligence (AI) and complementary technology and methods become increasingly viable, we may engineer systems with greater sophistication in what to *observe* (monitor), how to *orient* (understand in context), and how *decide* (identify and select among viable options). Command then uses control messages to actuate activity within the SoI (*act*).

The logic of command consists of algorithms and axioms. Algorithms are a set of rules in the form of processes or sequence of events. Axioms represent the spirit of intent that guide action in the event of algorithmic failure or unpreparedness; e.g., *do no harm* (the solution should not cause more harm than the problem), *minimize unintentional harm* (acknowledges harm may occur), and *minimize intentional harm* (acknowledges harm will occur).

Orchestrating agile operations will consist of invoking static solutions in the form of redundancy or known sequence of events (playbooks); and dynamic solutions in the form composition (invoking modules) or dynamic development (real-time production of new solution or solution variations). Encoding of orchestration will draw upon multiple mathematical disciplines working in harmony; i.e., formal quantification of agility.

Option Management for Dynamic Adaptation

Problem	Lack of autonomy in orchestration; dependency on people in-the-loop.
Need	Continual dynamic adaptation within *cyber-relevant time*.
Barriers	Complicatedness of encoding autonomous governance and adjudication logic and rules; situational awareness that provides necessary inputs.
Intent	A foundation of technology and mathematical disciplines to quantify agility.
Value	Contribute to realization of continual dynamic adaptation in operations.
Metrics	Orchestration performance metrics.
Notions	Many patterns throughout the mathematical disciplines, per discussion below.

For systems engineers who design autonomous self-directing systems, the concept of *Option Management for Dynamic Adaptation* explores a hypothetical set of mathematical disciplines and technologies for continual dynamic adaptation in cyber-relevant time. Quantifying agility contributes to autonomous orchestration. Orchestration exists today with predominantly human intervention for understanding, decision-making, and action. People *in-the-loop* are smarter and more flexible than computers but slow. As we increase autonomy and have increasing need for understanding and decision-making in cyber-relevant time, we need sophisticated autonomous orchestration.

There are many existing and emerging mathematical disciplines that may contribute to encoding *Command* logic for continual dynamic adaptation of operations. *Set-Based Design* enumerates options readily available. *Category Theory* provides for set relationships. *Compositionality Theory* facilitates dynamic composition of readily available modules vs. dynamic development (code generation). *Combinatorics* helps manage compositional options and variations. *Quantum Cognition* provides for modeling human decisions; integrating the socio- aspect to system modeling. *Artificial Intelligence* and *machine learning* provide for simulated learning and adaptation from experience. *Distributed Ledger Technology* contribute to mechanistic trust and the encoding of techno-social contracts. *Bayesian Belief Networks* quantify dependency and degree of causality. *Uncertainty Quantification* quantify degree of accuracy. *Portfolio Theory* helps maximize return for given level of risk; dynamic optimization for autonomous contingency planning and proactive or preemptive adaptation (seek gain). *Network Theory* helps safeguard against weaponizing interconnectedness. *Viable Systems Theory* contributes to the evolution of dynamic systems.

Complementary to algorithms are *axioms*. The encoding of principles that capture the *spirit of intent* to guide actions in the event algorithms fail or are unprepared to accommodate current circumstances; e.g., do no harm vs. minimize unintentional harm (acknowledges harm may happen) vs. minimize intentional harm (acknowledges harm will happen).

Option Management for Dynamic Adaptation complements other concepts on *orchestration* (contributes to the logic of command) and *situational awareness* (necessary inputs to command).

Harmonizing Risk in Agile Operations

Problem	Operational risk predominantly focuses on potential loss.
Need	Expand awareness and operational realization of both the negative side of risk (loss) and the positive side of risk (opportunity, seek gain, optimize).
Barriers	Silo-thinking and predominance of looking at risk only in terms of loss.
Intent	Establish agility's role in sustaining system viability and relevance including proactive contingency planning, continual optimization, and seeking gain.
Value	Holistic approach to risk; dynamic adaptation in explore / exploit.
Metrics	Mean Time Between Failure (MTBF), Mean Time Between Repair (MTBR); uptime, value-delivery quantity and quality (time, accuracy, efficiency); consistency (dependability).
Notions	INCOSE INSIGHT December 2020 on loss-driven systems engineering (LDSE); opportunity-driven systems engineering (ODSE), and System Dynamics Modeling archetypes relevant to explore, exploit.

For systems engineers who design autonomous systems, the concept of *Harmonizing Risk in Agile Operations* explores the negative side of risk (loss) and the positive side of risk (gain) with intent to sustain system viability and relevance in the face of adversity. *Harmony* is an emergent order; "harmony resides in a reality to be created each and every time" (Sundararajan 2013, p.2). Harmony is not uniformity; rather, "harmony is a relational term which entails diversity and difference" (Sundararajan 2013, p.2). Harmony is a holistic perception, an overall sense of things rather than focusing on any particular thing (Lu 2004). Harmony is a dynamic equilibrium (The Doctrine of the Mean 1971). *Agility* is necessary to achieve harmony.

Risk includes a negative side (loss) and a positive side (opportunity). The negative side includes *avoid*, *withstand*, or *recover* from loss. The positive side includes *seek gain* or *sustain optimal outcomes*. While not mutually exclusive, the predominant focus of loss-driven is on *viability* and to *fight-through-adversity*; the predominant focus of opportunity-driven is on *relevance* and *stakeholder satisfaction*. Loss-driven domains include *reliability*, *sustainability*, *survivability*, *resistance*, *resilience*, *safety*, *security*, and *agility*. Opportunity-driven domains remain largely unexplored and include concepts like *explore*, *inquire*, *optimize*, *gain*, *acquire*, and *achieve*. Both loss and opportunity use modeling, contingencies, probabilities, and uncertainty.

The INCOSE INSIGHT October 2020 issue theme introduces and elaborates on loss-driven systems engineering (LDSE). This foundational topic provides a baseline from which to explore the complementary concept of opportunity-driven systems engineering (ODSE) and harmonize the two approaches throughout the system lifecycle.

Discussion

FuSE Agility met the objectives for 2020 in so far as this paper provides an organizing framework for agility, starts to define integration points for agility into systems engineering, shows active multi-organizational collaboration, defines and elaborates on an initial set of agility concepts. These accomplishments set up the pursuit of three to five-year objectives for influencing system development (integrating agility into SE doctrine) and establishing ongoing evolution throughout the system lifecycle. Each topic identifies notional references to begin exploration into patterns that will help capture and reuse known solutions for agile-development (design patterns), agile-solutions (architecture patterns), agile operations (decision patterns), and agile-workforce (performance patterns).

FuSE Agility begins to overcome obstacles for realizing the agility vision by expanding the perception of agility beyond just software development and by setting up many areas within which to explore codifying systems engineering practices; e.g., formalizing influences on ISO/IEC 15288 and the INCOSE Systems Engineering Handbook and Systems Engineering Body of Knowledge. The foundational concepts explore ways to engage stakeholders actively to represent their changing needs and help resolve tradeoff decisions throughout the lifecycle.

Next steps include using the FuSE Agility concepts herein and their expanded versions in complementary papers and articles to engage the systems engineering community in feedback to refine these concepts and define new concepts for 2021 and beyond.

Conclusion

We introduced four objectives and nine foundational concepts to advance thinking and practice for integrating agility into the systems engineering lifecycle. These begin a roadmap for how systems engineering can improve system agility in the near term and set up for future research. The INCOSE Vision 2025 refers to composable design methods, collaborative enterprise engineering, and multi-disciplinary automated workflows in support of agile programs. FuSE Agility elaborates on that vision by identifying need for principle-based learning embedded in all lifecycle stages (applied knowledge management), stakeholder engagement throughout the process including viable technical oversight to guide continual integration. FuSE Agility suggests ways to achieve agility across organizational boundaries; and explores solutions for long lead times that may disrupt the agile process. Transcending Vision 2025, FuSE Agility explores the concept of agile operations and adaptable workflows including the concepts autonomous orchestration, quantifying continual dynamic adaptation to achieve agility in cyber-relevant time, and harmonizing risk to address *loss* to sustain viability and opportunity to sustain relevance.

Our vision beyond 2025 includes systems engineering facilitating the lifecycle of autonomous systems including continuous dynamic adaptation to sustain value-delivery under nominal and adverse conditions. Continuous adaptation optimizes tradeoffs among *loss-driven domains* and *opportunity-driven domains* for the system to remain *viable* (capable of producing desired results) and *relevant* (compatible with the changing stakeholder needs and at times anticipating stakeholder needs). The future of systems engineering facilitates the complementary nature of LDSE and ODSE to design reactive and proactive emergent behavior in systems and operations; *reactive* (encountered X, learned, adapted) and *proactive* (anticipated X, affirmed, adapted). A future very much dependent on formalizing agility in systems engineering.

References

Confucius and his Disciples: *Doctrine of the Mean, 1971* (Courier Dover Publications, NY).

Cohen, Mike. 2018. Incorporating Governance or Oversight into an Agile Project. Mountain Goat Software Blog (https://www.mountaingoatsoftware.com/blog/incorporating-governance-or-oversight-into-an-agile-project).

Dove, R. and R. LaBarge. 2014. Fundamentals of Agile Systems Engineering – Part 1. International Council on Systems Engineering IS14, Las Vegas, NV, 30-Jun-03Jul. www.parshift.com/s/140630IS14-AgileSystemsEngineering-Part1.pdf .

Dove, R., W. Schindel, C. Scrapper. 2016. Agile Systems Engineering Process Features Collective Culture, Consciousness, and Conscience at SSC Pacific Unmanned Systems Group. Proceedings International Symposium. International Council on Systems Engineering. Edinburgh, Scotland, 18-21 July. <*www.parshift.com/s/ASELCM-01SSCPac.pdf*>

Dove, R., W. Schindel, K. Garlington. 2018. Case Study: Agile Systems Engineering at Lockheed Martin Aeronautics Integrated Fighter Group. Proceedings International Symposium. International Council on Systems Engineering. Washington, DC, 7-12 July. www.parshift.com/s/ASELCM-04LMC.pdf

Dove, R., D. Kemp, K. Lunney, T. McDermott, B. Papke, R. Yeman. 2020. Issues, impediments, and Inspiration for Continuous Integration in Mixed Discipline Development Projects. International Council on Systems Engineering. Virtual International Symposium, 20-22 July. www.parshift.com/s/200722CipPanel-85min.mp4

Dove, R. 2020. Agile SE Processes 203: Agility as a System. INCOSE Webinar #143. September 16. www.parshift.com/s/AgileSE-203.pdf

Hayes, W. 2017. Agile Metrics: A New Approach to Oversight. SEI CMU Blog (https://insights.sei.cmu.edu/sei_blog/2017/12/agile-metrics-a-new-approach-to-oversight.html).

International Council on Systems Engineering (INCOSE). 2014. Systems Engineering Vision 2025. July 2014

Lu, R. R. 2004. Zhung-guo gu-dai xiang-dui guan-xi si-wei tan-tao (Investigations of the idea of relativity in ancient China). Taipei: Shang ding wen hua

Schindel, W., R. Dove. 2016. Introduction to the Agile Systems Engineering Life Cycle MBSE Pattern. Proceedings International Symposium. International Council on Systems Engineering. Edinburgh, Scotland, July 18-21. www.parshift.com/s/160718IS16-IntroToTheAgileSystemsEngineeringLifeCycleMBSEPattern.pdf

Schindel, W. 2017. Innovation, Risk, Agility, and Learning, Viewed as Optimal Control & Estimation. Proceedings International Symposium. International Council on Systems Engineering. Adelaide, Australia, July 17-20.

Sundararajan, L. 2013. The Chinese notions of harmony, with special focus on implications for cross cultural and global psychology. The Humanistic Psychologist, 41, 1–10.

Willett, K.D. 2020a. Harmonizing the Domains of Loss-Driven Systems Engineering. INCOSE INSIGHT October 2020.

Willett, K.D. 2020b. Systems Engineering the Conditions of the Possibility. Proceedings IS2020, International Council on Systems Engineering. Virtual conference conducted on Cape Town, South Africa time**,** July 20-22.

Biography

Dr. Keith D. Willett is a senior strategist and enterprise security architect for the United States Department of Defense**.** He has a PhD in systems engineering from Stevens Institute of Technology. He is co-chair for the INCOSE working groups on Systems Security Engineering and Agile Systems & Agile SE. Dr. Willett is sole author of *Information Assurance Architecture* and coauthor of three books including *Multisystemic Resilience – Adaptation and Transformation in Contexts of Change* (Oxford Press) February 2021.

Rick Dove is an independent researcher, systems engineer, and project manager generally focused in the system security and system agility areas; and an adjunct professor at Stevens Institute of Technology in the School of Systems and Enterprises. He chairs the INCOSE working groups for System Security Engineering, and for Agile Systems and Systems Engineering. He is an INCOSE Fellow, and author of *Response Ability – the Language, Structure, and Culture of the Agile Enterprise*.

Mr. Alan Chudnow is a Tech Fellow Emeritus in the Peraton Cyber Mission Sector. He is an INCOSE-certified Expert Systems Engineering Professional with over 35 total years' experience. He has worked on SAR processing, developed special purpose FFT hardware, and served as the chief architect for several large signal processing and planning systems. Mr. Chudnow has a B.S. in Electrical Engineering from Wayne State University, an M.S. in EE from University of Southern California; and a B.A. in Philosophy from the U of MD.

Emmet C. Eckman III ("Rusty") has 30 years' experience in government, civilian contractor, and US Army reservist. He serves as a Northrop Grumman Corporation's Program Director on the Ground Based Strategic Deterrent Engineering. He is a Northrop Grumman *Emeritus* Technical Fellow and a PMI certified PMP, an INCOSE ESEP and the INCOSE Certification Advisory Group chair. He is also a SAFe® Product Consultant (SPC) a certified SCRUM Master and eXtreme Programming practitioner and an avid SCUBA diver.

Larri Rosser is a Raytheon Certified Architect in Raytheon Intelligence, Information and Services. She has 30 years industry experience in aerospace, defense and computing technology, multiple patents in the area of human-computer interface and a BS in Information Systems and Computer Science from Charter Oak State College. Currently, she holds the role of Chief Engineer within in the IPx mission area, where she practices Agile Systems Engineering with a cross functional team.

Jennifer Stevens is a systems engineering subject matter expert for NASA and currently the Chief Knowledge Integrator at the Marshall Space Flight Center (MSFC). She joined NASA in 2000 as a propulsion systems engineer in the MSFC Engineering Directorate and led the Propulsion SE&I RSRM Element Integration team in the Space Shuttle Program Office from 2006 until program closeout.

Robin Yeman is a Lockheed Martin (LM) Senior Fellow Agile / DevSecOps Enterprise Coach. She has 25+ years in software engineering with focus on Agile/DevSecOps building large complex solutions across multiple domains from submarines to satellites. She has certifications in SAFe Fellow, SPCT (Candidate), CEC, PMP, PMI-ACP, and CSEP and advocates for continuous learning. She is a PhD candidate at Colorado State researching best practices for complex safety critical solutions. Supports multiple government working groups and is a DevOps ambassador at the DevOps Institute.

Dr. Mike Yokell is a Systems Engineering Director at Raytheon Technologies. He has been the US representative to International standards-setting bodies for Systems and Software Engineering and was the project editor for two new international standards on Systems of Systems Engineering. Mike is certified as an expert systems engineering professional by INCOSE. He holds multiple US and European Patents for Model Based Systems Engineering (MBSE) and large-scale immersive virtual reality.