28th annual INCOSE international symposium
Washington, DC, USA
July 7 - 12, 2018

Case Study: Agile Systems Engineering at Lockheed Martin Aeronautics Integrated Fighter Group

Rick Dove
Paradigm Shift International
rick.dove@parshift.com

William (Bill) Schindel
ICTT System Sciences
schindel@ictt.com

Ken Garlington
Lockheed Martin Corporation
kennie.e.garlington@lmco.com

The Lockheed Martin Aeronautics Integrated Fighter Group (IFG), in Fort Worth, Texas, was motivated to move to an agile system engineering (SE) development methodology by the need to meet urgent defense needs for faster-changing threat situations. IFG has and is tailoring a baseline Scaled Agile Framework (SAFe®) systems engineering process for a portfolio of mixed hardware/software aircraft weapon system extensions, involving some 1,200 people in the process from executives, through managers, to developers. Process analysis in October 2015 reviewed two years of transformation experience, updated in this article to 2017 status. Notably, the SE process is facilitated by a transformation to an Open System Architecture aircraft-system infrastructure, enabling reusable cross platform component technologies and facilitating faster response to new system needs. The process synchronizes internal tempo-based development intervals with an external mixture of agile/waterfall subcontractor development processes. This article emphasizes the manifestation of agility as the purpose and outcome of an embedded *system of innovation*, and introduces concepts of information debt, process instrumentation, and a preliminary systems integration lab for early customer demonstrations and discovery of potential difficulties.

Introduction

INCOSE's Agile Systems Engineering Life Cycle Model (ASELCM) project has published three case studies of effective agile systems engineering in a variety of applications, collectively covering agile software, firmware, hardware, and people-ware systems engineering in experienced practice (Dove, Schindel, Scrapper 2016, Dove, Schindel 2017, Dove, Schindel, Hartney 2017). The objective of the ASELCM project is to discover agile life cycle model fundamentals and the underlying requirements for enabling and manifesting agility in multi-discipline system engineering. This article is the fourth case study, and benefits from the learnings of the prior three. Specifically, this article reveals the central and critical role that systemic, activity-based, continuous innovation plays in enabling and delivering system engineering agility.

This fourth case study is based on a three-day analytical workshop, held October 20-22, 2015 at Lockheed Martin Aeronautics Integrated Fighter Group (IFG) in Fort Worth, Texas, and subsequent verbal updates since then. That workshop analyzed a two-year-in-process evolving transformation from a waterfall approach to an IFG-tailored version of the Scaled Agile Framework (SAFe®) (Scaled Agile 2016). SAFe is a systems engineering process for large projects and for portfolios of multiple projects[1]. This article does not focus on the SAFe process, but rather on the means of transformation to, and the evolution of, a SAFe-like process that fits the nature of IFG's contract environment. For brevity, IFG's tailored SAFe process will be referred to here as IFG-TS, and is depicted as an operational model in Figure 1.

There is a lead systems engineer for each project, attached to the "Project Management" bubble and interfacing with the Value Stream (VS). Their responsibilities include project-specific technical management activities. Other systems engineering responsibilities are distributed: Value Stream Engineer (VSE), working with the Release Train Engineers and Scrum Masters, provides a VS-level interface in areas such as project assessment; the Solution Manager, with support from Product Managers and Product Owners, has the VS leadership for requirements elicitation and definition, in conjunction with the Chief Engineer. The Solution Architect, working with the System Architects, provides a similar function for architecture; the Shared Services bubble includes VS-wide support for activities such as risk/opportunity management and configuration management; and a specific Agile Release Train provides SMEs for mission analysis, system analysis, and design.

IFG is focused on upgrading existing aircraft in need of new weapons, weapons control, and avionics systems. IFG develops software internally, and selects and manages suppliers and subcontractors for weapons hardware and avionics.

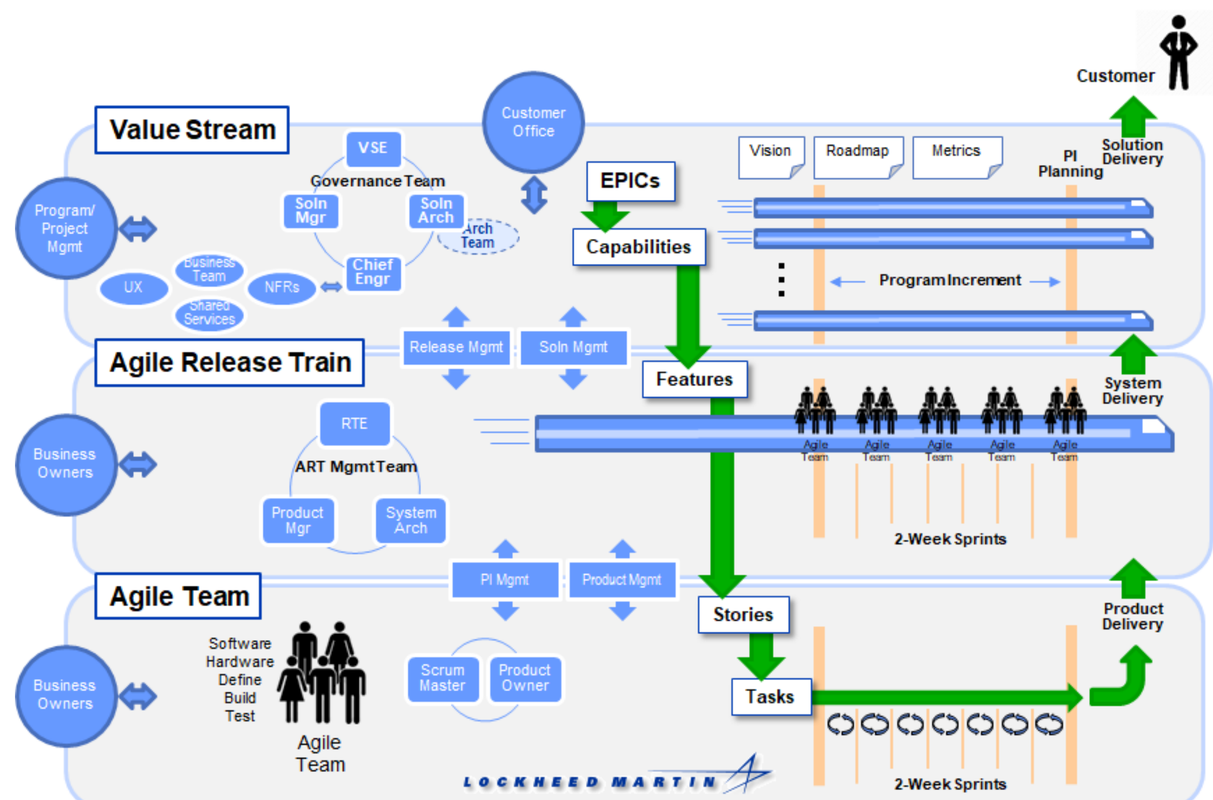


Figure 1. IFG-TS operational model.

[1] SAFe and Scaled Agile Framework are registered trademarks of Scaled Agile, Inc.

The focus of this article is on the "system" of innovation, responsible for managing innovation in both the system engineering process and the system engineered product. The system of innovation is a logical behavior-based system distributed throughout the system engineering process. Seemingly contradictory, Figure 2, referred to as the ASELCM pattern (Schindel and Dove 2016), shows the system engineering process embedded in the system of innovation – this is simply a difference between physical and logical boundaries, and recognition that the system of innovation is the source and driver of agility in the systems engineering process. The ASELCM pattern establishes a set of three logical-system reference boundaries, defined by their behavior, not their physical separation.

- System 1: The Target System, the subject of innovation over managed life cycles of development, deployment, and support.
- System 2: The Target System Life Cycle Domain System, including the entire external environment of the Target System—everything with which it directly interacts, particularly its operational environment and all systems that manage the life cycle of the Target System. This includes the external environment of the operational target system(s), as well as all the (agile or other) development, production, deployment, support, security, accounting, performance, and configuration management systems that manage System 1.
- System 3: The System of Innovation, which includes System 1 and 2 along with the systems managing (improving, deploying, supporting) the life cycle of System 2. This includes the systems that define, observe, analyze, improve and support processes of development, deployment, service, or other managers of System 1.

All three systems are (or at least should be) happening simultaneously, effectively an organic complex system motivated to survive and thrive by evolving suitably in an uncontrolled operational environment.

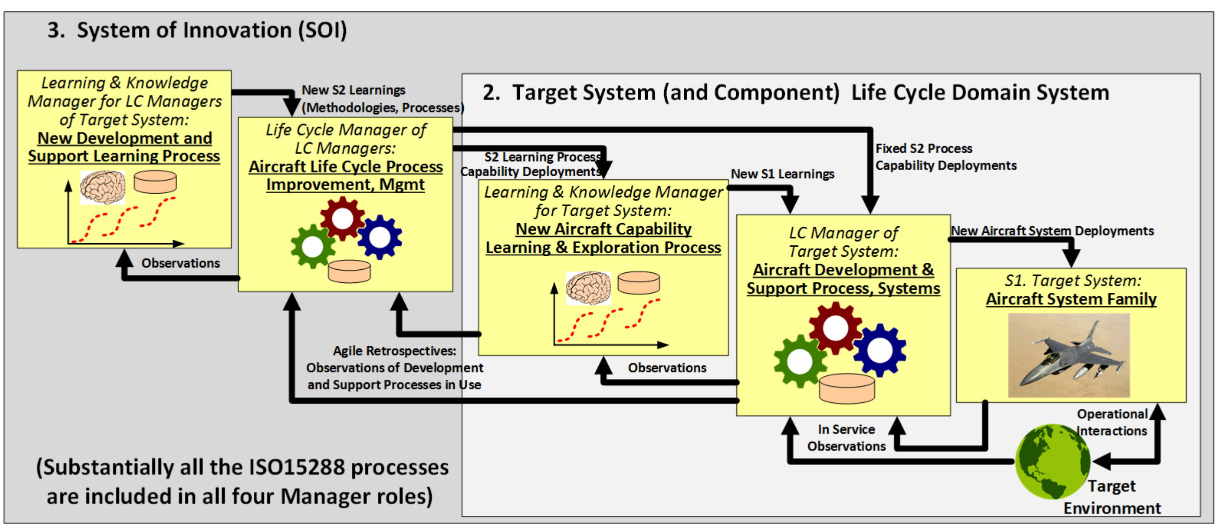


Figure 2. ASELCM Pattern system reference boundaries (Schindel and Dove 2016), configured for this case study.

Process Environment Characterization and Response Requirements

IFG customers are experiencing threat situations that are changing on ever shorter cycles, necessitating shorter-cycle, more frequent, counter-response. The traditional waterfall approach couldn't be counted on to meet urgent schedules. The need for a different approach was clearly evident.

Experience with agile software practices started at IFG in 2003. Results encouraged management to expect that value could be gained with a transition on a larger scale.

The choice of a tailored SAFe process was informed by experiences in other domains across the Lockheed Martin Corporation. After some study, IFG recognized SAFe as generally well aligned with their situation, and was encouraged by expressions of interest and assistance from customers. Good insights are recognized in other scaling models; but IFG places high core value on *Principles of Product Development Flow* (Reinertsen 2009), embedded in the SAFe approach.

This section first characterizes the environment-imposed *needs*, then characterizes the necessary *intent* of *response capability* to address those needs. Subsequent sections show selected *operational features* that fulfill intent.

Agile SE processes are necessary and justified when the engineering environment has characteristics of caprice, uncertainty, risk, variation, and evolution (CURVE). IFG characterized their systems engineering CURVE environment as shown in Table 1.

Table 1. SE process-environment CURVE characterization

Caprice (Unpredictability): Unknowable situations CC1: Urgent pre-emptive customer needs, sometimes called Quick Reaction Notice[2] events CC2: Changes in business environment, e.g., congressional funding commitments or legal requirements CC3: Project scope change **Uncertainty:** Randomness with unknowable probabilities CU1: Effectiveness of process tailoring CU2: Contract/customer compatibility with agile approach CU3: Management support/engagement in agile approach CU4: Team-member engagement with agile approach **Risk:** Randomness with knowable probabilities CR1: Cultural incompatibility CR2: Ability to keep and attract talent CR3: External stakeholder schedules (e.g. certification) CR4: Systems of Systems requirements changes	**Variation:** Knowable variables and ranges CV1: Multiple-project resource conflicts (e.g. test facilities, key people) CV2: Subcontractor development compatibility CV3: System of Systems integration integrity CV4: Requirements of differing importance levels **Evolution:** Gradual Successive Development CE1: OSA/OMS emphasis[3] CE2: Customer mission needs CE3: New compelling technology availability

Fleshing out the uncontrolled problem space in the CURVE Framework is a necessary first and continuous activity toward developing effective IFG-TS response requirements. Key selected

[2] For an example see: www.edwards.af.mil/News/Article/1226376/f-22-quick-reaction-test-and-modernization-efforts-lead-to-national-recognition

[3] For US Air Force example see: www.lockheedmartin.com/us/products/OSA.html, for US Navy example see: htpp://www3.opengroup.org/news/press/open-group-releases-future-airborne-capability-environment-face%E2%84%A2-technical-standard

response requirements recognized in the IFG-TS analysis are shown in Table 2, with parenthetical links to the CURVE elements they address. Many others that surfaced during the analysis are ignored here, as the ones shown are sufficient to make the intended case study points. The table is arranged according to the Response Situation Analysis framework (Dove and LaBarge 2014) that was employed in the workshop discovery activity.

Table 2. Response Situation Analysis

Proactive Response Requirements	Reactive Response Requirements
What must the process be creating or eliminating in the course of its operational activity? RC1: A safe environment for people to take prudent risks (CR2) RC2: Risk identification and mitigation plans at project and functional level (CC2, CC3, CU4) RC3: Loading plans with spare capacity for unknowns/inaccurate planning (CV1) RC4: Architectural planning/development horizon to accommodate variation (CC3, CV4, CE2) RC5: Experience accumulation (CU1) What performance will the process be expected to improve during operational life cycle? RI1: System level development optimization vs. local/functional optimization (CU1, CR1, CU4) RI2: Responsiveness to customer needs (CC1) RI3: Stakeholder, developer, and supplier alignment (CU2, CU3, CR1, CR3, CV2) RI4: Customer acceptance rate from acceptance testing events (CC1) RI5: Agility of existing integrated system (CU1, CE1) RI6: Awareness of evolving process effectiveness (CU1) RI7: Effectiveness of distributed knowledge exchange (CU1, CR2, CV2) What major events coming down the road will require a change in the process infrastructure? RM1: Evolution of customer missions (CE2) RM2: Cybersecurity and related standards (CC3, CU2, CR3) RM3: DoD Open Missions approach (CE1) What modifications in employable resources might need to be made as the process is used? RA1: Personnel that make up a team (CV1, CR2, CV4) RA2: Test infrastructure to maintain throughput (CV1) RA3: Modification in project-specific details of the operational model (CU1) RA4: Addition of subcontractor with new technology and/or process expertise (CE3) RA5 Reallocation of work between prime contractor and other entities (CC1, CV1)	What can go wrong that will need a systemic detection and response? RW1: Leadership and stakeholder churn that change vision and expectations (CC2, CC3, CU3) RW2: Non detection of variances (CU4. CV1, CV3) RW3: Insufficient identification and management of opportunities and risks (CR1, CR4) What process variables will need accommodation? RV1: Tailored process self-improvement and policing (CU1, CU4) RV2: Alignment and coordination of PI Planning (CC1, CC3, CU1, CV4) RV3: Organizational acceptance and adoption of tailored process (CU3, CU4, CR1) What elastic-capacity will be needed on resources/output/activity/other? RE1: System test capacity (CV1) RE2: Development capacity band to avoid disruption when work is more than expected in volume or difficulty (CC1, CC3, CV3, CV4) What types of resource relationship configurations will need changed during operation? RR1: Team-personnel assignments among multiple weapon systems (CC1, CR2, CV1) RR2: Work reassignments to match team capacities (CU1, CR2, CV1) RR3: Priorities for requirements (CC3, CV1, CV4) RR4: Acquisition procedures/policies/contract for situational and objectives reality (CC1, CU2, CE2, CE3)

Enabling, Facilitating, and Sustaining Agility

The Agile Architecture Pattern (AAP) for systems and processes that successfully deal with CURVE operational environments is used here for its succinct descriptive effect. The AAP in Figure 3 displays the principal architectural structure and strategy as a graphic representation that depicts what enables and facilitates agility in the IFG-TS process.

Briefly, the architecture contains three principal elements: a pool of resources that can be configured to address the necessary activity of the moment, a passive infrastructure with common rules for enabling ready interaction of these resources, and an active infrastructure with responsibilities for enabling sustainment of process agility by evolving and maintaining the resources, providing internal and external environmental awareness, assembling activities from available resources, and evolving the active and passive infrastructures.

The architecture is structured to configure a variety of process activities with personnel and other resources as and when needs arise. Four key activities are depicted that will be discussed later.

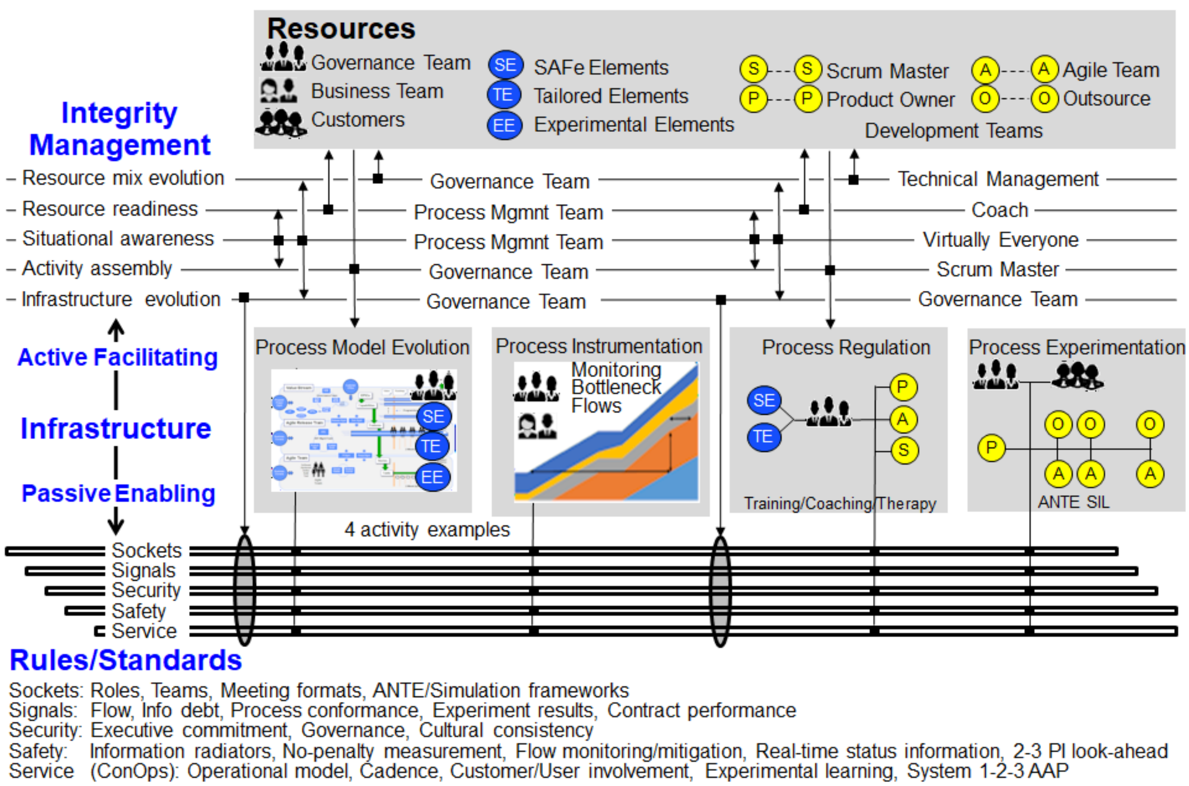


Figure 3. Agile Architecture Pattern instantiation for IFG-TS.
(Process Conformance activity is depicted as it was during 2015 transformation)

The AAP calls out the principal resources that are employed in assembling process-activity configurations:

- Governance team – This team includes the Chief Engineer, the Solution Architect, Solution Manager, and Value Stream Engineer (called Solution Train Engineer in current SAFe framework). Collectively they provide overall process governance and evolution, external technology awareness, OSA/OMS evolution and the integrity of a growing inventory of reusable componentry, and program increment planning. The Governance Team at the time of the 2015 analysis workshop included a Process Management Team – a team of four full-time people as principal process owners with another 2-6 part-time people, and value stream governance included the process coaches.

- Business Team – This team is responsible for external market awareness, and plans and manages large, typically cross-cutting customer-facing initiatives that encapsulate new development necessary to realize certain business benefits including those related to reusable componentry.
- Customers – Customers from the various programs collaborate with the Process Management Team on base-line and evolving process concepts that require contract accommodation.
- SAFe elements – The standard SAFe framework consists of many elements, which are base-line candidates for the evolving IFG-TS operational model.
- Tailored elements – These elements consist of modifications, additions, and eliminations to the standard SAFe framework elements.
- Experimental elements – These elements may be short term or limited employment concepts under experimental test for efficacy, eventually promoted to a tailored element or added to the population of negative-effect lessons learned.
- Development Teams –
  - Scrum Master – tactical agile team manager of work in process.
  - Product Owner – strategic agile team manager of work in process.
  - Agile Team – software and hardware developers and testers.
  - Outsource – subcontractors responsible generally for developing operational devices composed of hardware, software, and firmware, such as avionics.

Infrastructure consists of passive and active sections. The passive section includes the resource interconnection standards that enable effective process-activity assembly. The active section designates responsibilities for sustaining and evolving process agility.

Passive Enabling Infrastructure?

Figure 3 at the top shows the principal System 3 resources that can be assembled into process-activity configurations for specific situations. The ability to drag-and-drop these resources into plug-and-play configurations is enabled by the passive infrastructure, so called because it encompasses the fairly stable rules that enable effective resource interconnection.

Sockets – physical interconnects:

- Roles – descriptions of interaction standards for every role depicted in the process operational model.
- Teams – descriptions of interaction standards for every team depicted in the process operational model.
- Meeting formats – descriptions of interaction standards during various meeting types.
- ANTE/Simulation frameworks – descriptions of interface standards in the Agile Non-Target Environment (ANTE) preliminary SIL for interconnecting supplier device simulations, IFG-procured low-fidelity COTS devices, and IFG-developed work-in-process – discussed later.

Signals – data interconnects:

- Flow metrics – real-time process-flow monitoring – discussed later.
- Information debt – progress and status monitoring of required documentation – discussed later.
- Process conformance – IFG-TS knowledge assimilation and employment monitoring.
- Experiment results – data confirming/denying effectiveness of process experimentation.
- Contract performance and conformance – monitoring of performance against contracted process requirements and expectations.

Security – trust interconnects:
- Executive commitment – executive process-training participation and subsequently walking-the-talk and supporting the transition.
- Governance – Process Management Team open and consistent communication.
- Cultural consistency – training, coaching, awareness, and therapy.

Safety – of process users, process, process environment:
- Information radiators – prominent posters showing visual project status.
- No-penalty team measurement – team productivity is monitored for transition-learning purposes, but not exposed publicly.
- Flow monitoring and mitigation – process flow predictive measurement – discussed later.
- Real-time status information – daily-updated computer accessible project detail (VersionOne).
- Look-ahead 2-3 Program Increments for early awareness of pending architectural issues.

Service – process Concept of Operations (ConOps):
- Operational Model (IFG-TS process framework) – an evolving visual representation of the ConOps.
- Cadence – maintaining a consistent iteration tempo in Program Increments.
- Customer/User involvement – IFG-TS has a unique milestone called Program Backlog Review (PBR). After a contract is received all requirements are decomposed into capabilities (a lesson learned, considered better than lower-level feature decomposition). Then a PBR is held with the customer to get concurrence that the contract scope has been appropriately prioritized, with a clear understanding of the work for the next six months and less granularity beyond. Only one PBR is held for a program, as subsequent refinement is attended to in ongoing ceremonies. Both customers and users are involved in program increment completion testing.
- Experimental learning – process experimentation activity designs, implements, and evaluates limited-impact trials of promising process tailoring.
- Systems 1-2-3 AAPs – Learning in each system requires architecturally enabled application in the next lower system. System 3 agility is enabled by System 2 agility, which in turn is enabled by System 1 agility.

Active Facilitating Infrastructure

The active infrastructure is what sustains the agility of an SE process, and encompasses five responsibilities: the roster of available resources must evolve to be always what is needed, the resources that are available must always be in deployable condition, the assembly of new activity configurations must be effectively accomplished, and both the passive and active infrastructures must evolve in anticipation and/or satisfaction of new needs. These five responsibilities are outlined in standard role descriptions, assigned to appropriate personnel, and embedded within the process to ensure that effective process-activity is possible at unpredictable times.

The AAP depiction of responsibilities calls out general roles that get fulfilled by different people depending upon the specific activity of interest:

- Resource mix evolution – ensures that existing resources are upgraded, new resources are added, and inadequate resources are removed, in time to satisfy needs. This responsibility is triggered by situational awareness, and dispatched as shown in two of the Figure 3 activity examples.
- Resource readiness – ensures that sufficient resources are ready for deployment at unpredictable times. This responsibility is ongoing, and dispatched as shown in two of the Figure 3 activity examples.
- Situational awareness – monitors, evaluates, and anticipates the operational environment in relationship to situational response capability. This responsibility is ongoing, and dispatched as shown in two of the Figure 3 activity examples.
- Activity assembly – assembles process-activity configurations. This responsibility is triggered by situational events, as and when needed, and dispatched as shown in two of the Figure 3 activity examples.
- Infrastructure evolution – evolves the passive and active infrastructures as new rules and roles become appropriate to enable response to evolving needs. This responsibility is triggered by situational awareness, and dispatched as shown in two of the Figure 3 activity examples.

Innovation and Experimental Learning

Figure 3 instantiated the IFG-TS process in the Agile Architecture Pattern, depicting four activities (of many more) that play key roles in the transformation to, and sustainment of, process agility.

- Activity #1, Process Evolution: Process operational model evolution moved from an original framework resembling the SAFe 4.0 model to that which is depicted in Figure 1. Evolution of the IFG-TS process was an expected, managed, and facilitated hallmark of the transformation strategy from the beginning. At the time of the 2015 workshop, concept testing included a capability-based work breakdown structure for one aircraft platform with a wait-and-see on others (now adopted), 12-week program increments (now variable at 12-14 weeks), long-term teams (now adopted partially), weighted-shortest-job-first prioritization (a SAFe concept that proved inappropriate for the IFG environment), and the “preliminary” systems integration laboratory discussed below. An update in mid-2017 provided the parenthetical status shown above and affirmed continued process evolution in areas where specifics were declined, but were generally outlined as unforeseen process changes, unpredicted changes in contracting approaches that support evolving process agility, and unexpected but favorable evolution in team-member engagement with the agile approach.

- Activity #2, Managed Workflow: Process instrumentation for cumulative work flow awareness and pending-bottleneck predictive capability is provided at IFG by VersionOne (www.versionone.com) agile-process management software. The IFG-TS team considers effectively managed work flow as the critical factor in avoiding bottlenecks that threaten schedule. Figure 4 depicts the measurement of queue size as the predictor of test facility cycle time, a frequent bottleneck that can be mitigated by managing queue size. Queue size for tasks awaiting attention by development (build) teams can also guide team loading to favor task assignment to less-loaded teams. See Reinertsen (2009) for the concepts and math behind flow management.

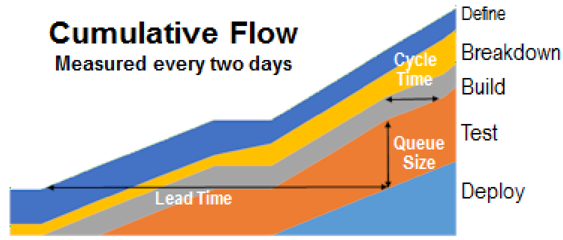


Figure 4. Automated cumulative processs-flow metrics, with queue size predicting cycle time in a test facility.

- Activity #3, Transformation Training and Coaching: Process regulation puts emphasis on training, coaching, and therapy for process conformance. At the time of the 2015 workshop 1200 people at IFG had been trained on SAFe and IFG-TS. Training started at the executive level and worked its way down the chain. Executives had to understand the transformation being made and their responsibility for leading the change. IFG's Vice President supported this at the highest level. Every time a release train started new team members were trained in all the roles. There was a dedicated transformation team consisting of four full-timers and two-to-six part-timers as needs arose. This team considered themselves an "external" group, as they were not involved "internally" with contract-fulfillment operations. By late 2016 responsibilities for process understanding and conformance had been defused and distributed "internally," and the transformation team was eliminated. Process ownership transitioned to an internal group called the Engineering and Technology group. The emphasis put on coaching was important in the first few years, and successful to the point that little is necessary now as the concepts have been assimilated and acculturated. An update in mid-2017 acknowledges explicit training continues for newly-assigned current members of the organization as well new hires, while recognizing the valued emergence of peer-peer informal means of knowledge distribution and coaching.

- Activity #4, Facilitated Experimentation: Process experimentation with a "preliminary" system integration lab (SIL) is of particular note . At the time of the 2015 workshop IFG was in early experimentation with this preliminary SIL concept, which they call the Agile Non-Target Environment (ANTE). The ANTE is conceptually similar to a Live, Virtual, Constructive (LVC) environment, and is used to compose integrated systems consisting of real devices, simulated devices, IFG software work-in-process, and operators. When useful for integration testing, the ANTE also employs lower-fidelity open-market devices with similar capability but lower performance than what is eventually expected from subcontractors. Subcontractors are required to provide device simulations to IFG ANTE specs. In contrast, the target system testing environment includes both traditional SIL and test-aircraft platforms employed at the end of program increments. An update on ANTE evolution in mid-2017 declared it a successful experiment based on customer feedback that values the early and incremental demonstration of working concepts and advanced exposure to difficulties in need of attention.

System of Innovation and the Information Balance Sheet

The above discussion of Figure 2 summarized high level ASELCM Pattern reference boundaries, System 1, 2, and 3 (Schindel and Dove, 2016); the same diagram also shows six subsystems:

System 3:

- Learning & Knowledge Management for Life Cycle (LC) Managers (for this case study example, the Learning Process for New Development and Support Processes).
- System 3 Life Cycle Manager of LC Managers (for this case study example, the Improvement and Maintenance Process for New Development and Support Processes).

System 2:

- Learning & Knowledge Manager for Target System (for this case study example, the New Aircraft Capability Learning & Exploration Process).
- LC Manager of Target System (for this case study example, the Aircraft Development & Support Process and Systems).
- Target Environment (for this case study example, the Aircraft Operational Environment, depicted as the green globe in Figure 2).

System 1:

- Target System (for this case study example, the Aircraft System Family).

The behaviors of these six subsystems were described in terms of the ISO15288 Life Cycle Processes and Agile Scrum Processes in (Schindel and Dove 2016). The Response Situation Analysis (RSA) Requirements of Table 2 describe aspects of that behavior, with particularly relevant case study examples illustrated in Table 3.

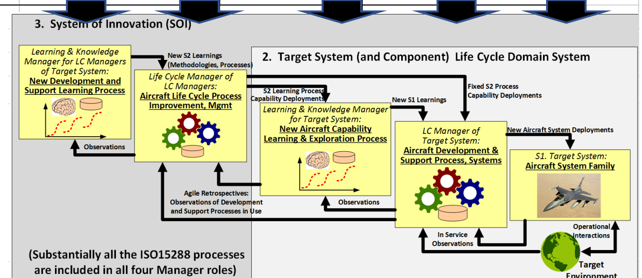
Table 3. Case Study Examples of RSA Requirements

RSA Req (Figure 5)	Lockheed Martin Case Study Example
RA3	Working with customer on evolution of acceptable agile methods, including contract issues
RC2	Higher level of attention to general LMC competitive capabilities
RA4	Subcontractor performance monitoring and adaptation
RC2	CONOPS of evolving LMC Aircraft Operations & Maintenance
RC5	Accumulation of agile methods
R16	Evolving customer appreciation and assimilation, with contract accommodation evolution
RA3	Variation of configuration of cadence increment time length
RC1	Training class on SAFe framework and LMC Aircraft Operations and Maintenance
RA3	Selection of Scrum versus Kanban at the team level
RV1	Training, coaching, and therapy for processes
RE1	Adjust cycles to accommodate shared facility resources
RE2	Loading of flow across enterprise to manage bottlenecks

The full set of RSA Requirements in Table 2 is shown in Figure 5, projected into those six subsystems with five columns aligned with the subsystem locations in the center graphic. The five columns of Figure 5 categorize the division of their interacting agility roles into Performers (whose performance is to be made more agile), Managers (which manage the life cycles of Performers, based on what is currently known), and Learners (which accumulate new knowledge based on experience, for future use by Performers through Managers). Performers, managers, and learners are roles filled by agents that may be people or processes.

	S3L	S3M	S2L	S2M	S1
RC1: A safe environment for people to take prudent risks (CR2)	>>	>		*	
RC2: Risk identification and mitigation plans at project and functional level (CC2, CC3, CU4)	>>	>		*	
RC3: Loading plans with spare capacity for unknowns/inaccurate planning (CV1)	>>	>	*	*	
RC4: Architectural planning/development horizon to accommodate variation (CC3, CV4, CE2)			>>	>	*
RC5: Experience accumulation (CU1)	>>		>>		
RI1: System level development productivity vs. local/functional optimization (CU1, CR1, CU4)	>>	>		*	
RI2: Responsiveness to customer needs (CC1)	>>	>		*	
RI3: Stakeholder, developer, and supplier alignment (CU2, CU3, CR1, CR3, CV2)	>>	>		*	
RI4: Customer acceptance rate at initial operational test (CC1)	>>	>		*	
RI5: Agility of existing integrated system (CU1, CE1)			>>	>	*
RI6: Awareness of evolving process effectiveness (CU1)		>	*	*	
RI7: Effectiveness of distributed knowledge exchange (CU1, CR2, CV2)	>>	>	*	*	
RM1: Evolution of customer missions (CE2)			>>	>	*
RM2: Cybersecurity and related standards (CC3, CU2, CR3)			>>	>	*
RM3: DoD Open Missions approach (CE1)			>>	>	*
RA1: Personnel that make up a team (CV1, CR2, CV4)	>>	>	*	*	
RA2: Test infrastructure to maintain throughput (CV1)	>>	>	*	*	
RA3: Modification in project-specific operating model (CU1)	>>	>	*	*	
RA4: Addition of subcontractor with new technology expertise (CE3)	>>	>	*	*	
RA5: Reallocation of work between prime contractor and other entities (CC1, CV1)	>>	>	*	*	

>>	Learner	>>
>	Manager	>
*	Performer	*



	S3L	S3M	S2L	S2M	S1
RW1: Leadership and stakeholder churn that change vision and expectations (CC2, CC3, CU3)	>>	>		*	
RW2: Non detection of variances (CU4. CV1, CV3)	>>	>		*	
RW3: Insufficient identification and management of opportunities and risks (CR1, CR4)	>>	>		*	
RV1: Tailored process self-improvement and policing (CU1, CU4)	>>	>		*	
RV2: Alignment and coordination of PI Planning (CC1, CC3, CU1, CV4)	>>	>		*	
RV3: Organizational acceptance and adoption of tailored process (CU3, CU4, CR1)	>>	>		*	
RE1: System test capacity (CV1)	>>	>		*	
RE2: Capacity changes to meet scope changes (CR2, CV1)	>>	>		*	
RR1: Team-personnel assignments among multiple weapon systems (CC1, CR2, CV1)	>>	>		*	
RR2: Schedules to match team capacity (CU1, CR2, CV1)	>>	>		*	
RR3: Priorities for requirements (CC3, CV1, CV4)	>>	>		*	
RR4: Acquisition procedures/policies/contract for situational and objectives reality (CC1, CU2,	>>	>		*	

Figure 5. Projection of RSA requirements against ASELCM Pattern.

Performer, manager, and learner roles explain the difference between *adaptation* (managed change in the capabilities of a performer) and *learning* (information gain by a learner). Adaptation can occur (within limits) without new learning, as when a flexible aircraft systems architecture is already in place and is exploited by a managing system, to rapidly adapt performance characteristics in response to environmental changes. By contrast, learning is illustrated by accumulating new information about potential aircraft system architectures and environmental threats. This example is seen in RSA Requirement RC4 of Figure 5. The same difference can be seen in even single cell living systems, whose existing DNA is variably expressed to adapt to environmental changes, without additional "learning" accumulated in changes to DNA over evolutionary time.

By "learning", we mean accumulation of experience in the form of information, and by "adaptation" we mean change in performance using only what is already known. So, an already well-informed system may, without learning new things, demonstrate agile adaptation within a given envelope, but advancing beyond that envelope demands learning new information – a form of "debt" analyzed in the IFG case study workshop and described in the next sections.

Managing Information Debt: Balance Sheet Model of Learning

IFG-TS process evolution had to address differences between government-customer contractual requirements for process artifacts (e.g., documents) and the information generation of the (evolving) agile process used by IFG. A review of IFG's approach led to a discussion of what we termed "information debt" by the ASELCM project team during the analysis workshop. While "debt" has a specific meaning in finance, its use in agile methods ("technical debt") has been quantitative but not the whole picture of the "balance sheet" analogy across system life cycles.

Information Debt was added to the ASELCM Pattern's stakeholder features (Schindel and Dove 2016), expressing the difference between the information currently available and the information needed to (not only deliver but also) support the life cycle of a system, and this includes uncertainty. It is in contrast to the better-known technical debt ("the extra development work that arises when code that is easy to implement in the short run is used instead of applying the best overall solution" [www.techopedia.com/definition/27913/technical-debt]). Information debt as an explicit concept helps us address the perceived tension between Agile Software Development methods and traditional Systems Engineering methods—but also an earlier and more basic challenge of justifying systems engineering of any kind.

Figure 6 (a) reminds us of the familiar (to systems engineers, if not others) fact of life—during the early project stages of lower accumulated cost, most of the future costs of a project become committed, by decisions (explicit or implicit) of a systems engineering nature. This is one of the traditional arguments for early stage systems engineering investment. Figure 6 (b) adds the idea of information debt, which is the not-yet-generated information necessary to deliver and sustain the system, and illustrates three different scenarios of information debt reduction scenarios. As pointed out by (Thomas 2016), there are effective "interest" costs paid by projects that don't pay off their information debt early enough, and the higher the risks involved, the greater the interest rate penalty to be expected. Scenario 3 of Figure 6 (b) illustrates a case of particular worry to traditional

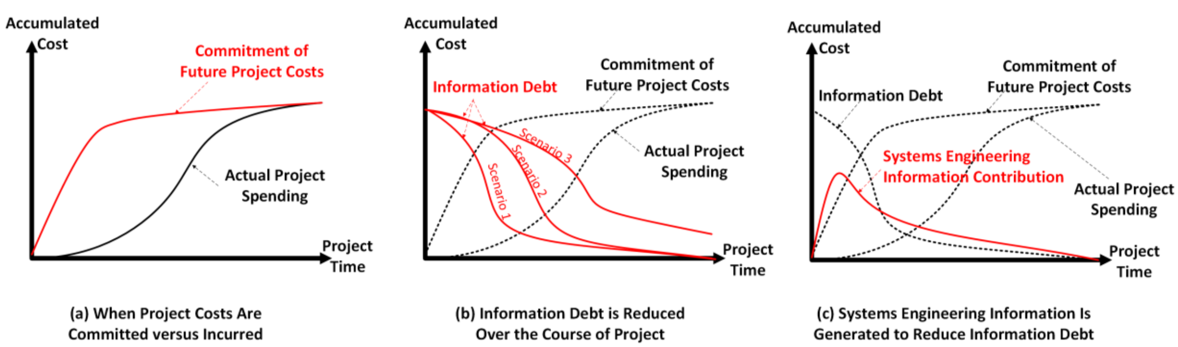


Figure 6. Financial flows—accumulated project costs, information debt,
and SE information contribution.

systems engineers considering agile methods: Does the Agile Manifesto mean that the project will end with remaining information debt outstanding, leaving us with a “working system” but an ongoing interest penalty caused by a shortage of needed information?

Figure 6 (c) illustrates the idea that systems engineering information must be generated (e.g., requirements, design architectures, risk assessments, etc.) early enough in the project to drive down information debt early enough and completely enough. The other side of the related controversy is the agile community’s concern that top-down documentation generation they associate with systems engineering can have its own risks, in too-late discovery of misunderstandings concerning stakeholder needs and expectations, the efficacy of design approaches, etc. Both of these opposite concerns are valid, and an objective means is needed to find the right middle ground—that is the purpose of the concept of information debt. It forces us to decide what information is really required by the subsequent life cycle of a system. It also sets the stage for recognizing that there are both real Balance Sheet (asset and liability) and Income Statement (revenue and expense) issues at stake, described further in the next section.

System 2 Learning Observed: Explicit System 1 Patterns as Balance Sheet Assets

Learning can be seen as discovery of regularities (patterns) that apply repeatedly over otherwise varying instances. The ability to rapidly develop and support System 1 aircraft configurations that dynamically respond to a range of “different” instance conditions is improved when System 2 recognizes and exploits these underlying patterns. For IFG and other enterprises, this takes the form of System 1 platform architectures that provide a framework and component family, discussed earlier as IFG’s OSA aircraft infrastructure, which became a “learned” part of the formal models discovered and maintained by System 2, describing System 1, shown in Figure 7.

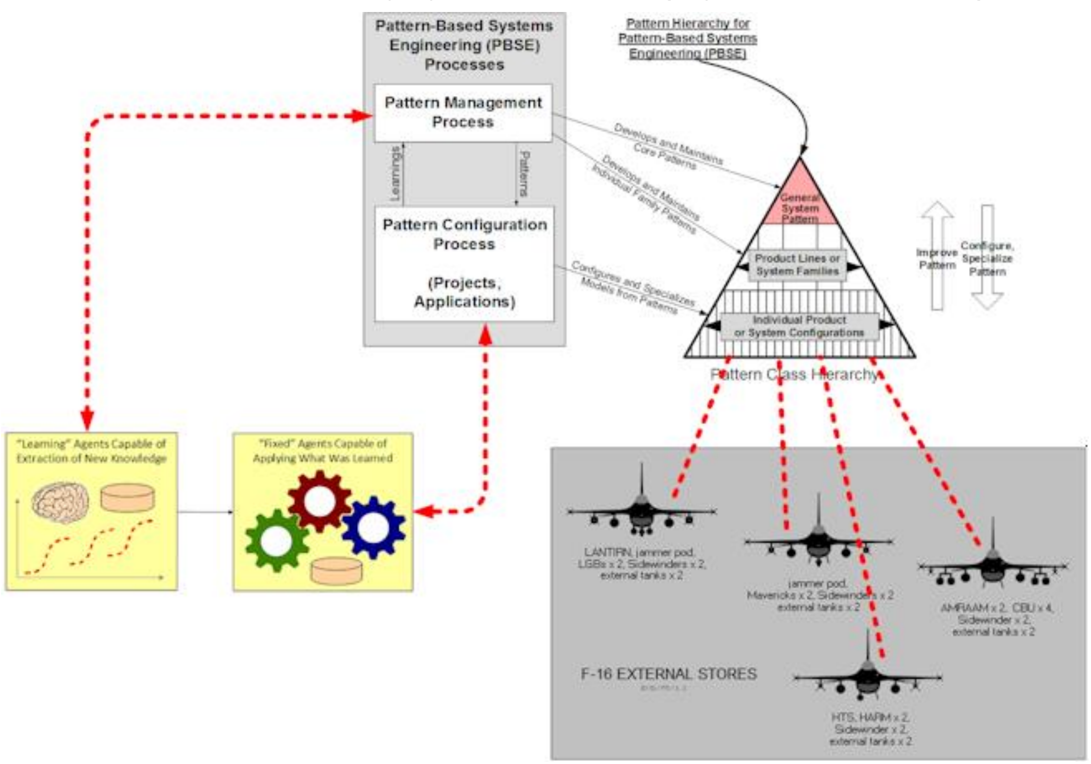


Figure 7. Platform architectures increase agility.
(Goebel, G. F-16 Variants www.airvectors.net/avf16_2.html])

On their face, both traditional and agile systems engineering would appear to build in enough "process" to address a "green field" or "clean sheet" situation, in which a project begins with no prior knowledge of requirements, design, or otherwise, and processes are provided to discover that information. In practice this is rarely the case, because nearly all system projects begin in the context of a large existing base of knowledge. Until recently, both traditional and agile SE methods offered scant theory in how these existing "assets" (prior knowledge) should be used, other than general guidance to consult and make use of standards, technical readiness levels, etc.

Historically, agile methods in particular emphasize learning by humans, but focus more on optimizing for human learning, not a general theory for accumulation and use of what is learned, and the sharing of this knowledge across a learning organization. The ASELCM Pattern recognizes prior knowledge in both human and other (e.g., stored data) forms, as learned System Patterns, whether in informal human expertise or formal representations shared between humans and information systems—in both cases, these are subsequently applied when the past learning is needed. Figure 8 is the subset of the ASELCM Pattern recognizing those aspects.

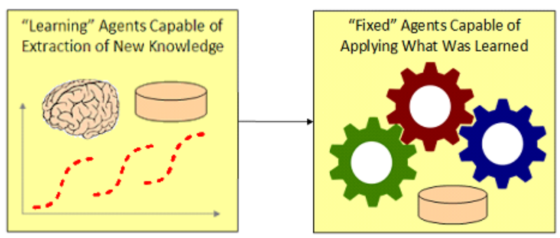


Figure 8. ASELCM human or other learning processes, learned assets, and their use.

Now that we have related information debt as a "balance sheet" entry, separate from the revenue and expense ("income statement") view of development, we can now turn to the positive side of that balance sheet. We observe that learned system patterns can be viewed as capital assets. In fact, they can be used to offset information debt on the balance sheet.

Moreover, this approach can be used to greatly strengthen the argument for early stage systems engineering during projects, because the information contribution curve of Figure 6 (c) can be generated without an equivalent surge in systems engineering expense, an income statement variable. This is accomplished by discovering and maintaining system pattern assets, then applying them during the early stage of a project as IP assets to generate information and pay off information debt—analogous to paying for a new house by using an existing asset.

This is the approach that was observed at IFG, where the OSA architectural platform pattern was used to effectively increase rapid response flexibility by lowering the cost of early stage Information Debt reduction, using this asset.

Financial standards (e.g., Financial Accounting Standards Board) do not typically provide for capitalization of human expertise, but patterns that are learned and explicitly stored are effectively software IP, which can be capitalized financially (Sherey 2006). This moves us from a metaphor to an actual financial model portion of the ASELCM Pattern. The use of system patterns in a full Product Line Engineering model of agility was the subject of a separate case study by the same ASELCM project (Dove, Schindel, Hartney 2017).

Concluding Remarks

The transformation period to agile systems engineering at IFG is over. Learning and process evolution continues. Asked in early 2018 if IFG feels this new agile approach is noticeably better than their prior approach, the answer demurred on releasable details, but summarized with this comment: "After the inevitable growing pains, the introduction of this new approach has been beneficial. Incremental releases and the release planning process permits earlier detection/correction of potential technical and programmatic issues. Major milestones have been accomplished on time, and customer response has been positive." A 2017 cleared-for-public-release presentation does offer some details: www.dau.mil/locations/midwest/Documents/F-22 Scaled Agile Framework SAFe (Cleared for Public Release)2.pdf.

The illumination of information debt and its role in systems engineering agility is a direct outcome of the IFG-TS analysis and case study development. In retrospect, explicit attention to information debt was present in the prior ASELCM project analysis workshops, but went unacknowledged in the case study articles. Information debt expresses and highlights the difference between the information currently available and the information needed to deliver and support the life cycle of a system. As an explicit concept this helps address the perceived tension between Agile Software methods and traditional Systems Engineering methods.

IFG-TS also illuminated the role of actionable process instrumentation, focused in this case study article on the role played by automated flow metrics that enable early-warning mitigation. This too, in retrospect, was present in the prior project analysis workshops but went unacknowledged in the case study articles. As a more general concept, process instrumentation provides awareness that puts project success accountability on the process owners rather than the development managers.

IFG has respect for SAFe as a tailorable framework appropriate for its general fit to their systems engineering activity. They recognize the differences between, as they express it, the commercial software target environment of SAFe origins and the different needs of weapon-system contract reality; and have found the SAFe framework accommodating to necessary tailoring. They can't make frequent short-cycle deliverable releases, but they can do short-cycle iterative and incremental development learning and learning-application.

The ANTE preliminary SIL is of particular note. Agile systems engineering, as opposed to agile software development, has difficulty in demonstrating short cycle incremental improvement and progress. The ANTE concept enables asynchronous subsystem testing with progressive subsystem improvement from initial simulation to prototype delivery to final delivery. The ANTE can also employ low-fidelity COTS devices as well as finished devices and simulated devices to stand-up a completely integrated prototype system for early integration issue revelation and mitigation.

This article is the fourth case study from INCOSE's Agile Systems Engineering Life Cycle Model project. This article's focus is different than the first three, having benefited from emergent understandings in the project's search for fundamental concepts necessary for mixed-discipline systems engineering agility. Highlighted in this article is the central role played by the embedded *System of Innovation* – a behavioral, not logical, system boundary within the agile systems engineering life cycle. This article and its three predecessors are case studies, intended to support the fundamental conclusions that will be explicitly presented in a final report, work that has already begun.

References

DoD OSA Data Rights Team. 2013. DoD Open Systems Architecture – Contract Guidebook for Program Managers v.1.1. Defense Acquisition University. https://acc.dau.mil/adl/en-US/631578/file/73333/OSAGuidebook%20v%201_1%20final.pdf

Dove, R., R. LaBarge. 2014. Fundamentals of Agile Systems Engineering – Part 1 and Part 2. International Council on Systems Engineering, International Symposium, Las Vegas, NV, 30Jun-3Jul. www.parshift.com/s/140630IS14-AgileSystemsEngineering-Part1&2.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. 2017. Case Study: Agile SE Process for Centralized SoS Sustainment at Northrop Grumman. Proceedings International Symposium. International Council on Systems Engineering. Adelaide, Australia, 17-20 July. www.parshift.com/s/ASELCM-03NGC.pdf

Dove, R. 2017. Agility in Systems Engineering – Findings from Recent Studies. Working paper originally distributed 15-April-2017 with ongoing augmentation and revision. www.parshift.com/s/ASELCM170415-AgilityInSE-Findings.pdf

Dove, R., W. Schindel, R. Hartney. 2017. Case Study: Agile Hardware/Firmware/Software Product Line Engineering at Rockwell Collins. Proceedings 11th Annual IEEE International Systems Conference. Montreal, Quebec, Canada, 24-27 April. www.parshift.com/s/ASELCM-02RC.pdf

Reinertsen, D. G. 2009. *The Principles of Product Development Flow: Second Generation Lean Product Development*. Celeritas Publishing, Redondo Beach, CA, USA.

Scaled Agile. 2016. SAFe® 4.0 Introduction. A Scaled Agile, Inc. White Paper. July. www.scaledagileframework.com/?wpdmact=process&did=MTI1LmhvdGxpbms

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, 18-21 July. www.parshift.com/s/160718IS16-IntroToTheAgileSystemsEngineeringLifeCycleMBSEPattern.pdf

Sherey, J. 2006. "Capitalizing on Systems Engineering," in Proc. of INCOSE 2006 International Symposium, Orlando, FL, 9-13 July.

Thomas, J. 2016. "Technical Debt – Thoughts from a Systems Perspective for the Leadership Team", in Proc. of INCOSE 2016 Great Lakes Regional Conference, Mackinac Island, MI, 18-21 September.

Welby, S. P. 2014. Modular Open Systems Architecture in DoD Acquisition. 17th Annual NDIA Systems Engineering Conference. Springfield, VA. October 29, 2014. www.acq.osd.mil/se/briefs/16943-2014_10_29_NDIA-SEC-Welby-MOSA-vF.pdf

Whittle, R. 2015. Lt. Gen. Otto Commits Air Force To More Open Mission Systems. Breaking Defense. http://breakingdefense.com/2015/10/lt-gen-otto-commits-air-force-to-more-open-mission-systems/

Acknowledgements

Garry Roedler, Lockheed Martin, led the effort to find an appropriate process for analysis at Lockheed Martin. Stephen Cornwell, Lockheed Martin, agreed to provide the process for analysis and the necessary resources for conducting the analysis workshop. Stacy Berthelson, Lockheed

Martin, organized and coordinated the workshop facility and logistics. The process material for review was assembled, presented, and discussed by Ken Garlington, Lockheed Martin IFG Principle Systems Engineer and lead systems engineer for fighter aircraft avionics modernization, and Gerald Turner, Lockheed Martin IFG Systems Engineering Senior Staff and Agile Transformation Leader and Coach.

The SE process analysis described in this article is the collective output of a workshop team that includes (alphabetically): Stacy Berthelson (Lockheed Aeronautics), Kevin Bryniak (Lockheed Engineering), Stephen Cornwell (Lockheed Aeronautics), Michael Coughenour (Lockheed IS&GS), Rick Dove (Paradigm Shift International), John Davidson (Rockwell-Collins), Robert Eisenberg (Lockheed IS&GS), Kevin Forsberg (OGR Systems), Ken Garlington (Lockheed Aeronautics), Greg Howard (MITRE), Bud Lawson (Lawson Konsult), Michael Liggan (MITRE), Todd McGall (Lockheed MFC), James (Craig) Pickel (Rockwell-Collins), Jack Ring (OntoPilot), Garry Roedler (Lockheed Corporate), Bill Schindel (ICTT Systems Sciences), Bennett Sproul (Rockwell-Collins), Gerald Turner (Lockheed Aeronautics), Mike Yokell (Lockheed Aeronautics).

Biographies

Rick Dove is CEO of Paradigm Shift International, specializing in agile systems research, engineering, and project management; and an adjunct professor at Stevens Institute of Technology teaching graduate courses in agile and self-organizing systems. He chairs the INCOSE working groups for Agile Systems and Systems Engineering, and for Systems Security Engineering, and is the leader of the INCOSE Agile Systems Engineering Life Cycle Model project. He is an INCOSE Fellow, and author of *Response Ability, the Language, Structure, and Culture of the Agile Enterprise*.

Bill Schindel is president of ICTT System Sciences. His engineering career began in mil/aero systems with IBM Federal Systems, included faculty service at Rose-Hulman Institute of Technology, and founding of three systems enterprises. Bill co-led a project on Systems of Innovation in the INCOSE System Science Working Group, co-leads the Patterns Working Group, and is a member of the lead team of the INCOSE Agile Systems Engineering Life Cycle project.

Ken Garlington is a Principal Systems Engineer in the Integrated Fighter Group. He has worked as a software engineer and systems engineer at Lockheed Martin for 32 years on various weapons systems. His current role is as a lead systems engineer for fighter aircraft avionics modernization efforts. His interests include safety related systems development and the application of Agile and Lean principles to the systems engineering discipline. He holds Master’s degrees in software engineering and systems engineering from the University of Texas at Arlington.