Case Study: Transition to Scaled-Agile Systems Engineering at Lockheed Martin's Integrated Fighter Group

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

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

Abstract Lockheed Martin's Integrated Fighter Group (IFG), in Fort Worth, Texas, was motivated to move to an agile system engineering (SE) development process by the need to meet urgent defense needs for faster-changing threat situations. IFG is tailoring the base-line Scaled Agile Framework (SAFe®) 4-Level systems engineering process, and have trained 1,200 people on the process from executives, through managers, to developers. Process analysis in October 2015 reviewed two years of learning, experimentation, and transformation experience on a portfolio of mixed hardware/software aircraft weapon system extensions. Notably, the SE process is facilitated by a transformation to an Open System Architecture aircraft-system infrastructure, enabling re-usable 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 case study emphasizes the transformation to the IFG-tailored SAFe approach with learning processes, lessons learned, and optimal process control.

Introduction

INCOSE project-in-process is seeking a generic Agile Systems Engineering Life Cycle Model (ASELCM), and is doing this by analyzing and building case studies of effective agile systems engineering in a multifaceted variety of applications, collectively covering agile software, firmware, hardware, and people-ware systems engineering processes in experienced practice (Dove, Schindel, Scrapper 2016, Dove, Schindel, Kenney 2017, Dove, Schindel, Hartney, 2017). The objective of the project is to discover and justify process principles as repetitively employed patterns, which are necessary and sufficient for any system engineering process that must contend effectively with an unpredictable, uncertain, and evolving engineering environment; and to document case studies that employ these principles in the context of different engineering process environments. The succinct articulation of those principles must await sufficient multi-case analysis.

The fourth three-day workshop, held October 20-22, 2015 at Lockheed Martin's Integrated Fighter Group (IFG) in Fort Worth, Texas, analyzed a two-year-in-process evolving transformation to an IFG-tailored version of the Scaled Agile Framework (SAFe®) 4.0 for Lean Software and Systems

Engineering (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.

Lockheed Martin's 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.

For brevity, Lockheed Martin's IFG-tailored SAFe process will be referred to here as IFG-TS.

To set context, IFG is primarily working with two existing families of fighter aircraft, sustaining useful life by evolving their weapons and avionics systems to meet evolving threats. IFG has approximately 1,200 SAFe-trained employees involved with the IFG-TS process across multiple programs. Rather than looking at programs as isolated contract development work, the business strategy seeks leverage across multiple programs with reusable components supported by an OSA [2]/OMS [3] infrastructure. This is a Product Line Engineering strategy that enables and facilitates shorter cycle times for urgent program needs, lower costs where reuse is appropriate, and competitive technology upgrade and insertion. The OSA/OMS infrastructure development and evolution is done with Independent Research and Development (IR&D) funds, of no financial burden to the programs.

Figure 1 depicts the scope of the development requirements tree, driven at the top with IFG Business Epic (product line strategy) requirements. Business Epics are served by multiple Program Epics, which are composed of customer-required fixed Capabilities achieved with IFG-variable Features detailed as User Stories that encompass the lowest level development Tasks. The Integrated Master Schedule includes Epic through Feature items.

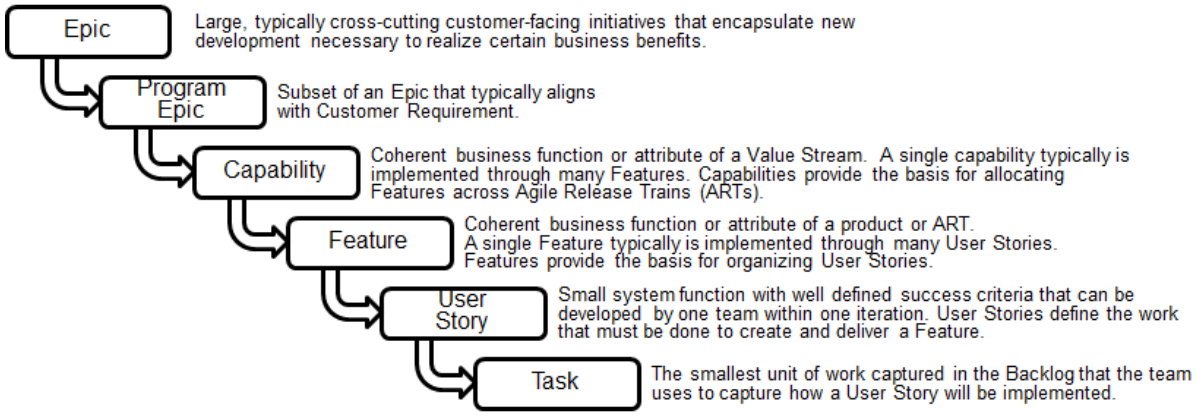

Figure 1. Typical backlog decomposition, where ART refers to a SAFe Agile Release Train

Notable process concepts that will be discussed include:

- Experimental learning processes for evolutionary process tailoring.
- Process instrumentation for development work flow awareness and optimal mitigation.

[1] SAFe and Scaled Agile Framework are registered trademarks of Scaled Agile, Inc.
[2] OSA: Open Systems Architecture (DoD OSA Data Rights Team 2013, Welby 2014).
[3] OMS: Open Mission Systems (Whittle 2015).

- Process instrumentation for information debt awareness and mitigation for sustainable systems and minimizing field disruption of development teams.
- Emphasis on training, coaching, and therapy.
- Synchrony with suppliers using waterfall and some who are transforming to incremental deliveries.
- Preliminary Systems Integration Laboratory (SIL) of low fidelity Commercial Off-The-Shelf (COTS) product to get early-testing results.
- OSA (aircraft infrastructure) transformation.
- Test-driven requirements specification

IFG began their transformation to agile SE slowly in 2003, with some individual software teams employing agile development processes. In 2011 early customer collaboration concepts were added. Management saw the value in an agile SE approach, and the evolution to what IFG has today began, with SAFe taking root in 2013. 2003 was also when IFG began using middleware for the target-system infrastructure, with the vision back then of evolving into Open Systems Architecture (DoD OSA Data Rights Team 2013) for system infrastructure and component interfaces. Transformation is an evolutionary process.

SE Process Overview

Agile SE processes are necessary and justified when the engineering environment has characteristics of capriciousness, uncertainty, risk, variation, and evolution (CURVE). IFG characterized their systems engineering CURVE environment as follows:

Capriciousness (Unpredictability): Unknowable Situations
- Urgent operational needs
- Diminishing manufacturing sources

Uncertainty: Randomness With Unknowable Probabilities
- Funding (e.g. Sequestration)
- Solution feasibility
- Regression impacts – the effects of integrating new development with prior development

Risk: Randomness With Knowable Probabilities
- Competition losses
- Attract/keep talent
- Systems of Systems requirements changes
- Schedule/external stakeholder timelines (e.g. certification)

Variation: Knowable Variables And Associated Ranges
- Multiple projects competing for bottlenecks (e.g. ground/flight test)
- System Of Systems integration

Evolution: Gradual Successive Development
- Planned modernization/sustainment increments
- Open Mission Systems evolution

Figure 2 shows s general outline of IFG-TS, depicting where some key tailored elements are located in the hierarchy, as well as other elements that provide context but won't be discussed.

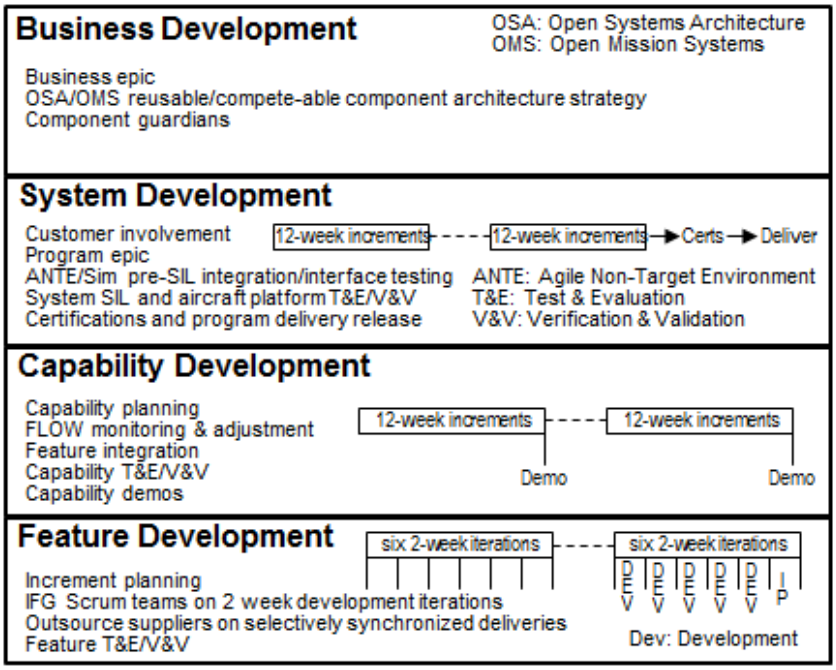

Figure 2. General outline of 4-level IFG-TS, with 12 week Program Increments, each with five 2-week development iterations and one 2-week Innovation and Planning (IP) iteration to reflect, adapt, plan, and buffer against iteration overrun.

This article is not about SAFe, nor much about IFG's tailored SAFe, but rather about the process of evolving a scaled-agility systems engineering process appropriate to IFG that leverages the SAFe framework as a starting point.

This article will not explain SAFe in any detail, and assumes the reader is either familiar with SAFe or can refer to the SAFe documentation (Scaled Agile 2016) if desired.

Spanning two years, IFG had evolved process tailoring in three increments at the time of the 2015 analysis.

Enabling, Facilitating, and Sustaining Agility

An Agile Architecture Pattern (AAP) for systems and processes that successfully deal with CURVE operational environments is used here for its succinct descriptive effect (Dove and LaBarge 2014). AAP displays the principle architectural structure and strategy as a graphic representation that depicts what enables and facilitates agility in a process or system. It is a framework for customer and management communication, for training new team members, for capturing lessons learned, and for maintaining a current central understanding of the process' key operational concepts as they evolve. It serves well as a single-graphic road map for the operational concept.

For purposes of describing the relevant systems engineering process issues unambiguously, we recognize three distinct systems of interest, distinguished as Systems 1, 2, and 3. System 1 is the target system under development – in this case new aircraft weapon systems to evolve efficacy and usefulness of existing aircraft. System 2 is the systems engineering process for developing, sustaining, and evolving new system capability – in this case the IFG-TS process that produces System 1. System 3 is the innovation process that evolves System 2 – in this case IFG's systematic evolutionary tailoring of the initial SAFe baseline framework.

Figure 3 depicts the AAP for System 3, as the focus at IFG is on tailoring and evolving the base-line SAFe framework for appropriate and effective employment in the IFG systems engineering environment. Briefly, the architecture contains three principle 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 System 3 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.

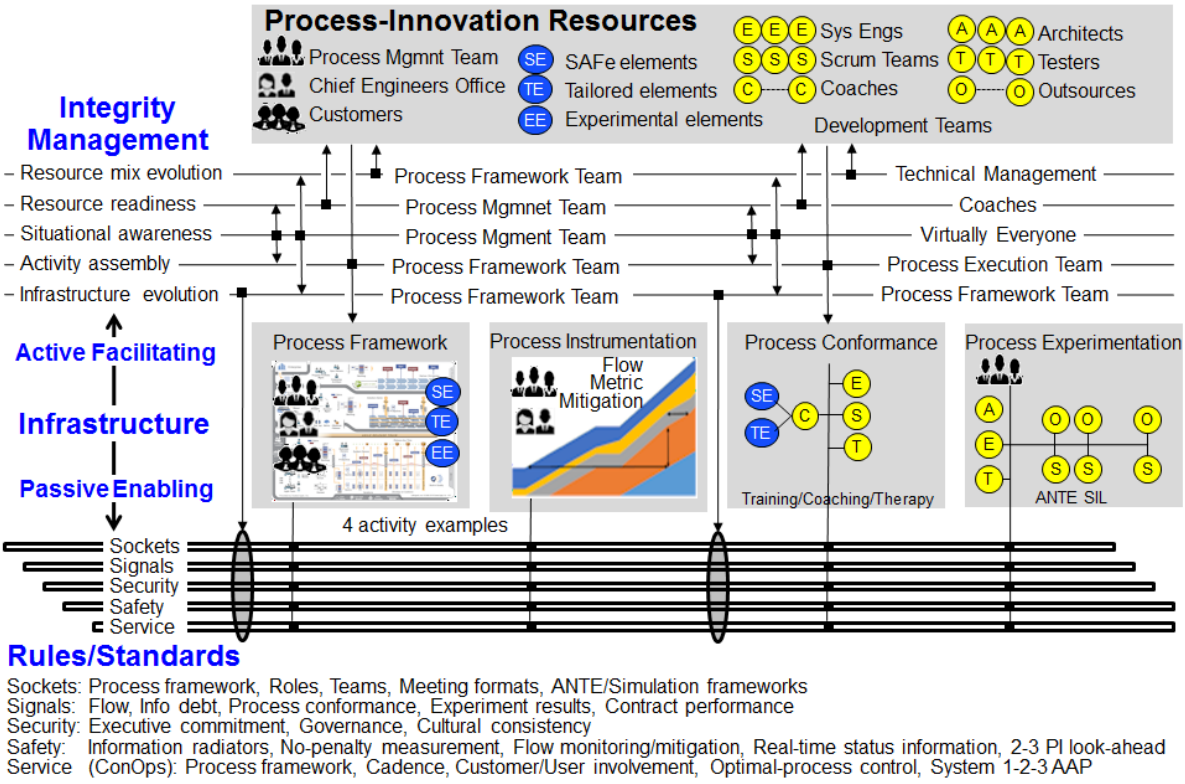

Figure 3. Process-innovation (System 3), the learning and evolution AAP.

The AAP instance of the IFG-TS System 3 process in Figure 3 serves to frame the discussion of *key* System 3 elements and their relationships. The architecture is structured to configure a variety of process activities with personnel and other resources as and when needs arise. Agility in System 3 enables and facilitates effective and timely adjustment as transition learning reveals needs.

The four System 3 activities depicted in the AAP are instructive examples from many, but are the key activities that guide the evolution of System 2. All will be discussed in more detail later, but in general:

- The Process Framework activity updates and captures snapshots of the evolving System 2 process framework.
- The Process Instrumentation activity monitors embedded process-performance sensors and triggers performance adjustments as appropriate.

- The Process Conformance activity trains and coaches personnel in process concepts, and provides therapy when needed.
- The Process Experimentation activity designs, implements, and evaluates limited-impact trials of promising process tailoring.

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

- Process Management Team – A team of four full-time people with another 2-6 part-time people. The four full-timers are the principle process owners.
- Chief Engineers Office – In addition to the traditional technical duties of Chief Engineers, this team collaborates with the Process Management Team for consensus on IFG-TS tailoring evolution.
- Customers – Customers from the various programs collaborate with the Process Management Team on base-line and evolving process concepts that require contract accommodation.
- SAFe framework elements – The standard SAFe framework consists of many elements, which are base-line candidates for the evolving IFG-TS framework.
- Tailored framework elements – These elements consist of modifications, additions, and eliminations to the standard SAFe framework elements.
- Experimental framework elements – These elements may be short term or limited employment concepts under experimental test for efficacy, eventually promoted to a tailored framework element or added to the population of negative-effect lessons learned.
- Development Team Resources – Resources as named are common to software development processes and self-explanatory, though it should be noted that Outsources are 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 principle 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:

- Process framework – structural relationships among roles, teams, and process activities depicted visually as current state of evolving process architecture.
- Roles – descriptions of interaction standards for every role depicted in the process framework.
- Teams – descriptions of interaction standards for every team depicted in the process framework.
- 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.
- Real-time status detail – daily-updated computer accessible project detail (VersionOne).
- 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.
- Look-ahead 2-3 Program Increments for early awareness of pending architectural issues.

Service – process Concept of Operations (ConOps):

- IFG-TS process framework – an evolving visual representation of the ConOps.
- Cadence – maintaining a consistent iteration tempo in Program Increments.
- Customer involvement – IFG-TS has a unique milestone called Program Backlog Review (PBR). After a contract is received all requirements are decomposed into features. Then a PBR is held with the customer to get concurrence that work can begin, 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.
- Optimal-process control – with flow measurement and mitigation the key driving factor.
- System 1-2-3 AAP – Learning in each system requires facilitated 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 Process Owners of System 3, during the transition period, consider themselves an "external" group, as they are not involved "internally" with System 2 process execution. In this article we call them the Process Framework Team, consisting of four full-timers and two-six part-timers as needs arise. During the transition there is also a Process Management Team, consisting of the same core four but dispatching different responsibilities of management rather than framework-evolution responsibilities. That the same people are involved in both teams provides an intimate awareness of process evolution effects unavailable with arms-length feed-back loops.

The AAP depiction of responsibilities is felt to be self explanatory and unnecessary of further explanation.

Key Operational Process Aspects

- Experimental learning on processes tailoring concepts is a hallmark of the IFG-TS framework and ConOps evolution. Experiments are generally implemented as test cases limited in deployment until evidence confirms or denies efficacy. Examples include a capability-based work breakdown structure for one aircraft platform, with a wait-and-see on others; single-server Flow modeling for cycle-time prediction; 12-week program increments; long-term teams; development-resource pairing, and the ANTE/Simulation preliminary SIL.
- Process instrumentation for cumulative work flow awareness and pending-bottleneck predictive capability is provided 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.

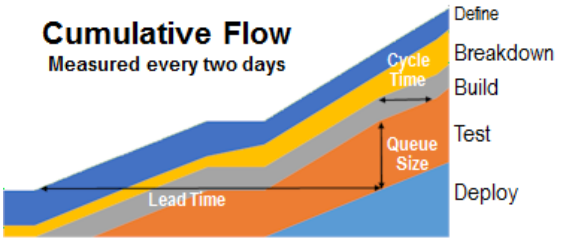

Figure 4. Automated cumulative processs-flow metrics, with queue size predicting cycle time in a test example

- Process instrumentation for information debt awareness and mitigation. IFG has three motivations for managing the production of sustainment documentation: 1) customer wants fully future-capable field sustainability without reliance on IFG, 2) IFG wants to eliminate the cost and time of providing information later when original people aren't there or forget what they did and how and why, and 3) certain certification activity needs such documentation for security, air worthiness, interoperability, etc. Instrumentation for measuring info debt is nothing out of the ordinary: the needs are written into contracts and engineering plans and reviewed for progress as requirements artifacts incrementally and periodically.
- Emphasis on training, coaching, and therapy (for process conformance). As of October 2015, two years into the transformation, 1200 people at IFG had been trained on SAFe. Training started at the executive level and worked its way down the chain. The executives

had to understand the change being made and their responsibility for leading the change. IFG's Vice President supports this at the highest level. Every time a release train is started, new team members are trained on all the roles. These three-day training courses are structured for engagement with hands on activity.

- Synchrony with outsource suppliers using waterfall and some who have transforming to incremental deliveries. Waterfall suppliers must deliver with Program Increment end timing. Some outsource suppliers have voluntarily opted to deliver in synchrony with iteration-end timing.
- A preliminary SIL, referred to by IFG as ANTE, is configured with low fidelity COTS avionics devices and supplier provided device-to-be simulation software. The ANTE provides an early integration test bed for IFG work-in-process software. Simulations are provided by the device suppliers to IFG specs, and low-fidelity COTS are IFG-procured devices with similar capability but lower performance than what is eventually expected from suppliers. In contrast, the target system testing environment includes both traditional SIL and test-aircraft platforms employed at the end of a Program Increments.
- System-1 OSA (aircraft infrastructure) transformation (with OMS) and evolution to enable and facilitate reusable components across aircraft platform portfolios. OMS provides the customer and IFG with ability to compete functional components. Hydraulics and cabling are receiving OSA attention as well as device components.
- Test-driven requirements specification. With customer concurrence, specification shall statement are written after demonstrated/tested program increment completion. After a contract is received all requirements are decomposed into features. Then a review is held with the customer to get their concurrence that work can begin, 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 of the work beyond the first six months is attended to in the ongoing ceremonies.

On Choosing the IFG-TS Process

IFG customers were (and are) experiencing threat situations that were (and 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 encourage by Corporate. After some study, IFG recognized SAFe as being well aligned with the 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.

Lessons Learned and Experimental Learning

Some outsource suppliers voluntarily experimented with deliveries in synchrony with program increment iterations and, contrary to their initial expectations, learned that their overall project costs went down rather than up.

Traditionally IFG has moved people around into different teams, but now there is a mix that keeps some teams together, based on evidence and some metrics of the value of keeping a team together. Teams that have stayed together appear to have higher work-enjoyment and higher productivity. IFG cites industry findings that people are happier in long term teams, and IFG has a concern about retaining talent.

The IFG-TS team is focused on process efficiency, but some development teams think that if they do their work fast and first it will influence the work of others. IFG intentionally makes some teams less productive for the sake of making the process more productive. Systems test is a traditionally bottleneck at IFG, so some times they slow a team down to make the process more effective. They also recognize that producing as fast as possible can unknowingly produce more defects.

A new milestone for IFG-TS was created, called PBR – Program Backlog Review. After a contract is received all requirements are decomposed into features. A review is then held with the customer to get concurrence that work can begin. Program Increment planning is done with the customer, to confirm that this is what they want built in the next 12 weeks. The statement of work specifically refers to cadence based reviews. Recent contracts have specs that actually read like user stories – the customer is no longer writing shall statements. At the PBR the customer expects a good cut at the next 6 months of work. There is less granularity in the things over 6 months away, and these get firmed in the ongoing ceremonies without another PBR. Technical reviews are implemented as part of the Program Increment demos if the customer wants to see specific data. If an outsource supplier practices waterfall, a PDR and CDR is employed.

IFG introduced a role they call component guardian, responsible for consistency within an individual component, as multiple teams modify components that combine to a feature. Guardians look for conflicts in stories, and features are created to be as independent as possible.

Pattern-Based Model View of Key Operational Aspects

The ASELCM Pattern

The ASELCM Pattern is a formal Model-Based Systems Engineering (MBSE) reference model describing the framework of system life cycle management from an agility perspective, providing a non-prescriptive reference emphasizing the principles of agility, for analysis purposes. It is described further in Schindel and Dove (2016). Figure 5 is one view of that model, summarizing three key system boundaries, configured for this IFG case study:

- **System 1:** The Target System, subject of innovation over managed life cycles; in this IFG case study, the Aircraft System(s) being developed, deployed, supported.
- **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. In this IFG case study, this includes all the external environment of the operational aircraft system(s), as well as all the (agile or other) development, deployment, support, security, accounting, performance, and configuration management systems that manage the (System 1) Aircraft Systems.

- **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. In this IFG case study, this includes the systems that define, observe, analyze (as in agile Process Retrospective) improve and support the processes of development, deployment, service, or other managers of System 1.

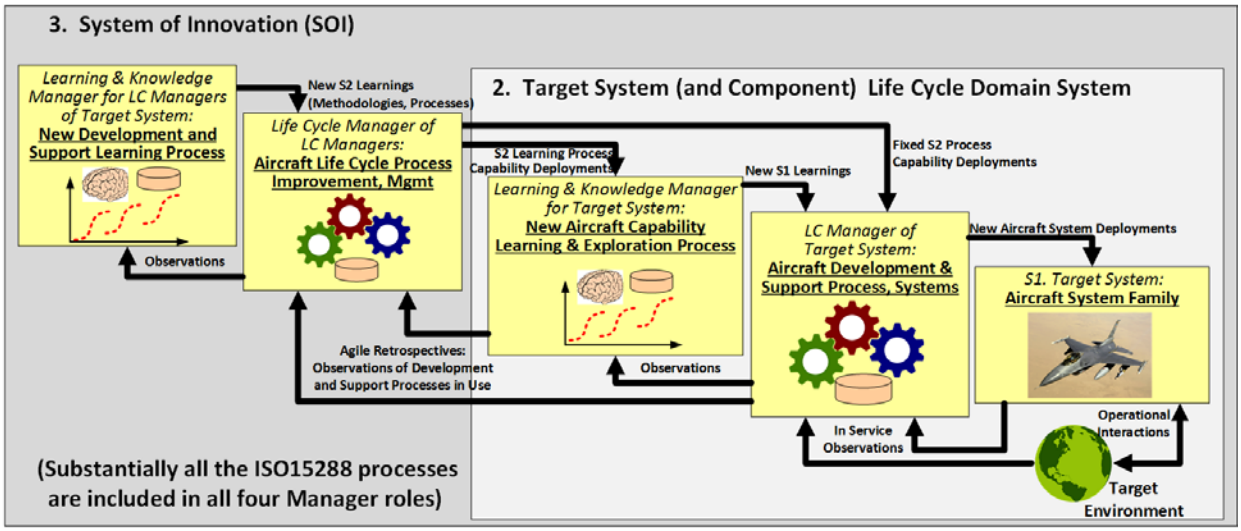

Figure 5. ASELCM pattern system reference boundaries, configured for this IFG aircraft systems development and support case study.

The observations discussed in this paper and at the IFG analysis workshop are further expressed using the ASELCM Pattern, in the following.

Toward Optimal Flow in System 2: System 1 Agile Trajectories

The most basic evolution over time of System 2 observed in the IFG case was the transformation toward agile methods. In subsequent agile operations of System 2, incremental agile development movements are made in System 1 Stakeholder Features Subspace, Technical Requirements Subspace, and Physical Architecture Subspace, repeating this in a sequence of agile sprints, until a release occurs. A single such increment in those System 1 model subspaces may be seen as (at least partly concurrent, but outcome synchronized) rectilinear "around the block" steps in the different subspaces, compared to the "as the crow flies" net result, as shown in (a) of Figure 6 below.

In the language of "flow" (Reinertsen 2009; Cskiszentmihalyi 1990), (a), (b), and (c) of Figure 6 illustrates the effect of different "batch" sizes for System 1 changes during development. (We use "batch" here as Reinertsen does, whether it refers to the amount of software developed during an agile software Sprint, or the amount of any grouped, synchronized set of developmental deliverables from a single agile increment, as well as production batches in a lean production situation.) The ultimate in large batches is the "waterfall", consisting of a single batch, shown in (a) of Figure 6. When compared to waterfall or other traditional large "batch" approaches, the agile trajectory is seen to use smaller (internal cadence synchronized) rectilinear increments. The resulting "smoothing" of trajectory is closely related to the idea of "flow" in systems engineering and other processes. It is also a consequence of the underlying objective function surface being

pursued—that surface is nearly always non-linear (non-planar). As a result, (b) and (c) of Figure 6 illustrate that "dead reckoning" in a single shot from the point of origination in Figure 6 does not in general produce the same outcome as smaller batch increments.

The related literature on "flow", beginning with psychological flow experiences by individual humans and progressing to effectiveness of industrial processes, describes a collection of heuristics or observations on the nature of what is sometimes referred to as "optimal" performance,

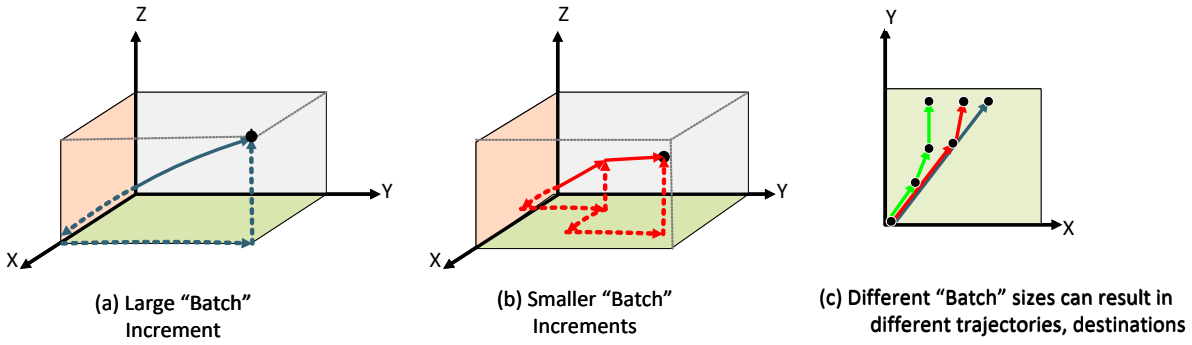

Figure 6. Smaller batch sizes can result in different configuration trajectories.

because it can be seen to be better in certain dimensions. For example, (Reinertsen 2009) describes this phenomenon in terms of Queues, Batch Size, Cadence/Synch/Flow Control, Fast Feedback, and other underlying characteristics. These are associated with better coupling the trajectory to the forces inherent in System 1's stakeholder needs, external operating environment, and internal architecture, versus the System 2's business process constraining forces. The resulting sense of improved "flow" is because of that improved coupling. Importantly for agility, the resulting overall trajectory outcome itself can be different (improved) when smaller batches are used, reflecting outcomes of better-informed exploration of stakeholder needs, technical requirements, and design solutions, and accompanying trajectory times. When these informal descriptions are moved into the formal MBSE model, the opportunity is opened for an improved understanding of optimal innovation trajectories, based on the established theory of optimal control (Schindel 2016, Pontryagin et al 1962).

Managing Information Debt: Balance Sheet Model of Learning

LM IFG's 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. This is not the only issue of timing as it relates to the generation of systems engineering information, even in more traditional methods.

A review of IFG's approach led to a discussion of "Information Debt" by the ASELCM Project Team during the analysis workshop. While "debt" has a specific meaning in finance, its ("Technical Debt") use in agile methods has been quantitative but not the whole picture of the "balance sheet" analogy in systems. In the ASELCM Pattern, "Information Debt" has been added, and defined to measure the difference between the information currently available and the information needed to (not only deliver but also) support the life cycle of a system. This is in comparison to the better-known Technical Debt, defined to measure "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 methods and traditional systems engineering methods—but also an earlier and more basic challenge of justifying systems engineering of any kind.

Figure 7 (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 7 (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 7 (b) illustrates a case of particular worry to traditional 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?

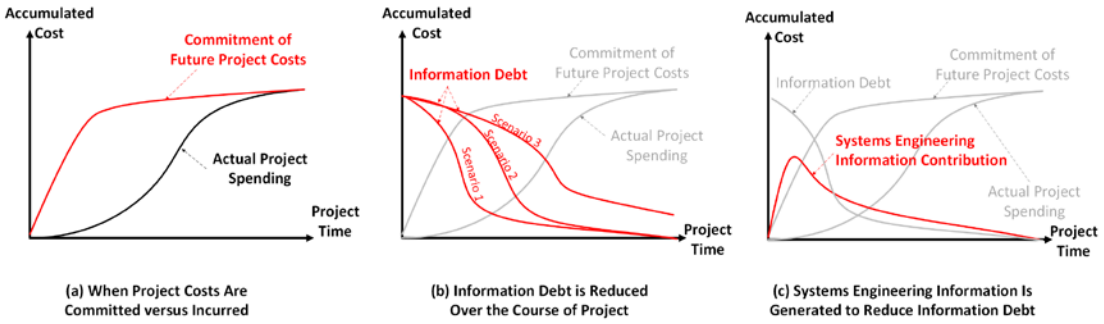

Figure 7. Financial Flows—Accumulated Project Costs, Information Debt, and SE Information Contribution

Figure 7 (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 too heavy-handed, 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 <u>really</u> required by the subsequent life cycle of a system. It also sets the stage for recognizing there are both Balance Sheet (asset and liability) and Income Statement (revenue and expense) issues at stake, and this is 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 below in Figure 8.

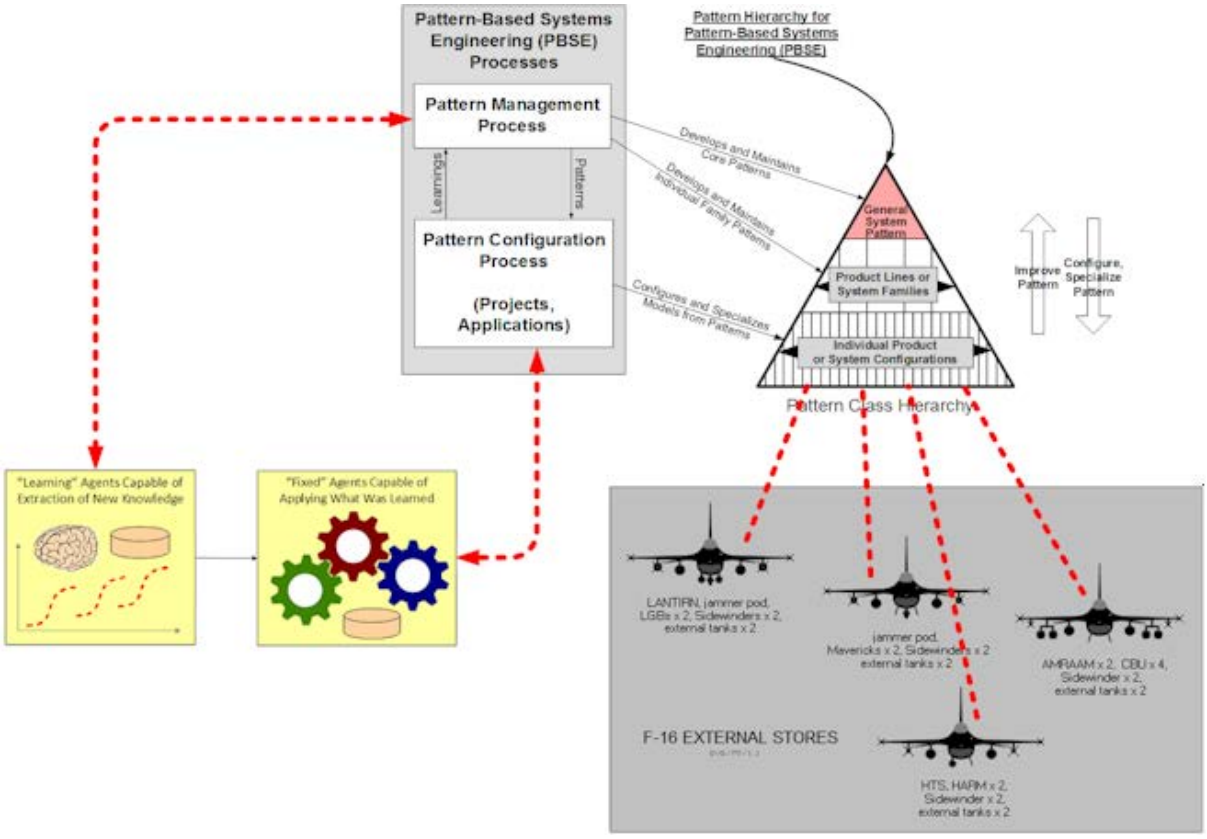

Figure 8. Platform architectures increase agility
Goebel, G. F-16 Variants. www.airvectors.net/avf16_2.html

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 passing 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 9 is the subset of the ASELCM Pattern recognizing those aspects:

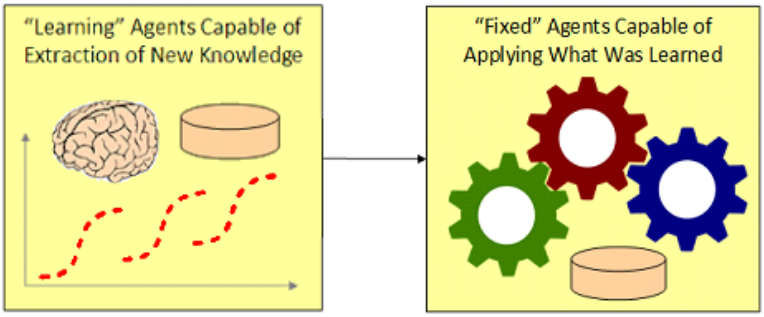

Figure 9. ASELCM human or other learning processes, learned assets, and their use

The previous section introduced Information Debt as a "balance sheet" entry, separate from the revenue and expense ("income statement") view of development. In this section, we add to the positive side of that balance sheet, by observing that Learned Systems Patterns can be viewed as capital assets. In fact, they can be used to offset Information Debt.

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 7 (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.

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 2005). 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 will be the subject of a separate case study by the ASELCM Project.

Concluding Remarks

The ASELCM analysis team developed a high level of respect for the IFG-TS process leadership team – their depth of process understanding, their focus on the intent rather than the prescription of procedural concepts, their high respect for process-flow management, their proactive attention to controlled evidence-based experimentation, and their people-focus on organization-wide training, coaching, and rationalized understanding-assistance during the transition.

A first for the ASELCM analysis team was exposure to meaningful and actionable process instrumentation, focused in this case study article on the role played by automated flow metrics that enables early-warning mitigation. As a more general concept, process instrumentation

provides awareness that puts project success accountability on the process owners rather than the development managers, and lends itself to the beginnings of scaled agility with minimal overhead.

Another first for the ASELCM team was the illumination of information debt. This concept is not new, as all projects are aware of their documentation requirements. But the needs of IFG to decouple long-life product operational support and sustainment from development resources quantifies the cost of this debt and motivates managed awareness and mitigation.

IFG appreciates the business, customer, and project advantages enabled by a System 1 OSA approach. This loosely-coupled modular open architecture provides agility in System 1 that facilitates agility in System 2, enabling faster and higher quality project completion through component reuse and affordable learning-driven reconfiguration.

IFG has respect for SAFe as a tailorable framework rather than a dogmatic procedure, a framework accepted for its general fit to their systems engineering situation rather than its popularity of the moment. 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.

IFG's tailored SAFe approach is a moving target as it evolves – this article presented the state of affairs at the time of the October 2015 observation. A lot has evolved since. Two developments relative to the focus of this article warrant mention. The "external" centralized Process Framework and Process Management Team responsibilities are a year later defused and distributed "internally". The "Process Owner" is now 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.

References

Csikszentmihalyi, M. 1990. *Flow: The Psychology of Optimal Experience,* Harper & Row.

Dove, R. and 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.

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, 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, M. Kenney. 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. 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

Pontryagin, L. S., V. G. Boltyanskii, R. V. Gamkrelidze, E. F. Mishchenko. 1962. The Mathematical Theory of Optimal Processes, English translation, Interscience.

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. and 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.

Schindel, W. 2016. "Innovation, Risk, and Agility, Viewed as Optimal Control & Estimation", in Proc. of INCOSE 2016 Great Lakes Regional Conference, Mackinac Island, MI, Sept.

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

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, Sept.

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 is 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 Discovery Project. He is an INCOSE Fellow, and the 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.