11th ANNUAL IEEE INTERNATIONAL SYSTEMS CONFERENCE Montréal, Québec, Canada, April 24-27, 2017

Case Study: Agile Hardware/Firmware/Software Product Line Engineering at Rockwell Collins

Rick Dove, *Fellow/Member INCOSE*, *Member IEEE*, William Schindel, *Fellow/Member INCOSE*, Robert (Will) Hartney

(This version was updated 1-June-2018 for typos and replaced a 2017 working paper reference with a 2018 published version)

Abstract—Rockwell Collins, in Cedar Rapids, Iowa, is using a product line approach for a family of radios they produce for military and international markets, with an agile systems engineering process tailored individually for synchronized software, firmware, and hardware development. This mixed-discipline engineering group encompasses some 350 employees working on multiple projects simultaneously for multiple customers. The product line approach provides an agile systems target-context for the agile systems engineering process – re-using common product-line hardware, firmware, and software whenever possible, evolving the product line with both internally-funded development and opportunities presented by customer project work, and accommodating special customer needs that grow the knowledge and capability base for potential later use in product-line extension. This article focuses on the mixed-discipline agile systems engineering process reviewed in the September, 2015, INCOSE Agile Systems Engineering Life Cycle Model (ASELCM) discovery workshop, and doesn't address the proprietary strategy and detail of the product line architecture. Notable elements that will be discussed include a parallel evolving product line roadmap as the strategic driving function, a facilitating central relationship-management role for systems engineering, synchronization of mixed engineering disciplines, an enabling infrastructure for agile hardware development, strong opportunity management complimenting traditional risk management, and comparison to the ASELCM Model-Based Systems Engineering (MBSE) pattern.

Keywords—**agile systems engineering, agile architecture pattern, ASELCM pattern, military radio, product line engineering.**

I. Introduction

An INCOSE project-in-process is seeking a generic Agile Systems Engineering Life Cycle Model (ASELCM), and is doing this by workshop-review of effective agile systems engineering (SE) in a variety of applications, collectively covering agile software, firmware, hardware, and people-ware systems-engineering processes in experienced practice. Selected case studies of these reviews are written to support the eventual ASELCM and its core underlying principles, which must await sufficient multi-case analysis (Dove, Schindel, Scrapper 2016, Dove, Schindel, Kenney 2017, Dove, Schindel, Garlington 2017).

This case study article is based upon the September 21-23, 2015 workshop at Rockwell Collins in Cedar Rapids, Iowa, that reviewed the Communications Engineering group's ARC-210 product-family systems engineering process, which they call RC Agile. ARC-210 encompasses a family of airborne radios for US and international military markets. Rockwell Collins has evolved a 1990 legacy heritage into an integrated agile systems engineering approach, with coupled incremental development for software, firmware, and hardware development, tailored individually for each discipline, with inspiration from SAFe®[1] and Scrum[2] agile software development processes.

To set context, the RC Agile program serves a highly competitive government market, with customers that often ask for unreasonable technical specifications cherry-picked from the best technical performance features available anywhere, which can't always exist together as a coherent system. The US DoD agencies may supply certain software and firmware non-developmental items (NDI) owned by DoD for required employment. The international military market is prohibited from employing ITAR protected technology, and requires different standards than the domestic market, especially for security features that international customers want independent of US government standards.

The competitive environment for military radios is already changing in the domestic market. Competition from established defense contractors is now augmented by competition from commercial firms, enabled by new government acquisition policies that keep the competition active even after initial contracts are established (http://breakingdefense.com/2016/03/armys-new-radio-strategy-is-unrelenting-competition/).

Typical clean-sheet projects average 3.5 years, with as many as 14 circuit boards, and 2-4 chasses. The Communications Engineering group encompasses some 250 engineers, including about 20 systems engineers. Figure 1 depicts the central role played by systems engineers and others in providing relationship management among the key elements of the process.

Notable process concepts that will be discussed include:

- Product line architecture and strategy, as agility-enabling concept.
- Active SE management of all relationships, as agility-facilitating concept.

Paper submitted for review 12/23/2016. This work was supported in part by the International Council on Systems Engineering (INCOSE).

Rick Dove is with Paradigm Shift International, Taos County, NM 87556 USA (e-mail: dove@parshift.com).
Bill Schindel is with ICTT Systems Sciences, Terre Haute, IN 47803 USA. (e-mail: schindel@ictt.com).
Will Hartney is with Rockwell Collins, Cedar Rapids, IA 52498 USA. (e-mail: robert.hartney@rockwellcollins.com).

1 Leffingwell, Dean. 2007. Scaling Software Agility: Best Practices for Large Enterprises. Addison-Wesley Professional. SAFe and Scaled Agile Framework are registered trademarks of Scaled Agile, Inc.

2 Schwaber, Ken and Jeff Sutherland. 2013. The Scrum Guide. www.scrum.org.

- Active external awareness evolving the Product-Line Market Requirements Document.
- Coupled cross-discipline agility.
- Agile hardware development platform infrastructure.
- Active opportunity management.

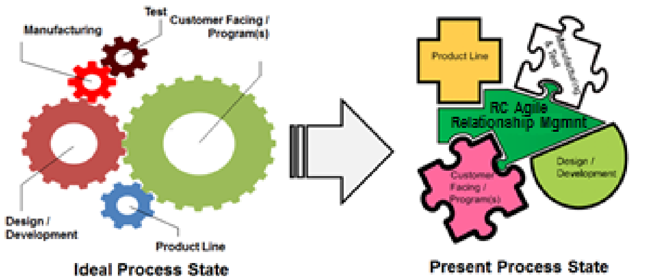

Fig. 1. Enabling dynamic coupling internally and loose coupling externally

II. SE Process Overview

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

Capriciousness (Unpredictability): Unknowable Situations

- International and DoD Markets have long and volatile acquisition cycles.

Uncertainty: Randomness With Unknowable Probabilities

- Feature Most Important Requirements (MIRs) are subjective and not clearly defined, leads to chasing an ever-moving competitive landscape.
- Unknown and Emerging Stakeholders / Users / and even ConOps.

Risk: Randomness With Knowable Probabilities

- Firmware/Hardware architecture may not be adaptable enough to address future requirements causing churn.
- Highly complex, highly regulated standardizations in an NDI competition results in significant investment with no guarantee for return.
- Unrealistic expectations of some customers exceed current technology envelope.
- Product development without DoD Sponsorship brings risk from 3rd party evaluations.

Variation: Knowable Variables And Associated Ranges

- Market-Based approach ties tightly to evolving industry needs.

Evolution: Gradual Successive Development

- Customers expect improvements in Space, Weight and Power - Cost (SWAPC) and new functionality which causes evolution of the design.

Figure 2 depicts the incremental development cycles for software, firmware, and hardware on a single project. Hardware consists of circuit boards and chasses. Engineers are typically working on multiple projects simultaneously.

III. 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

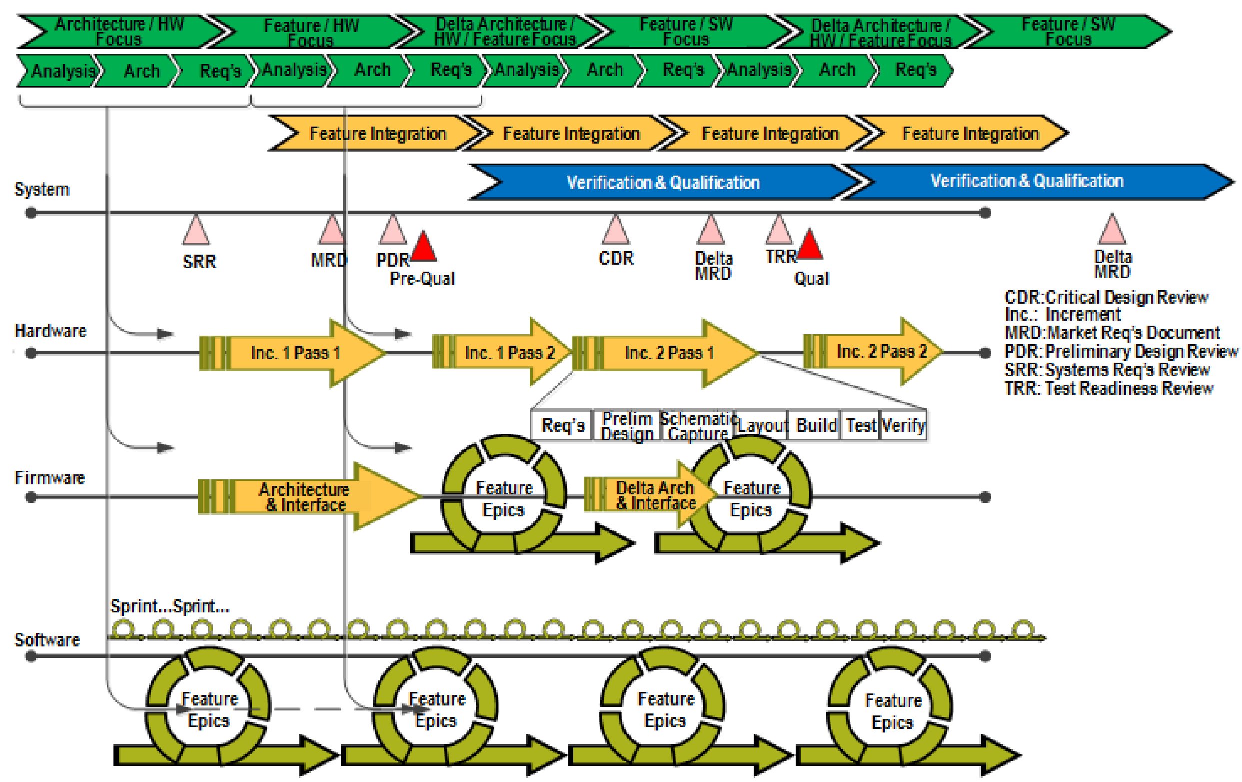

Fig. 2. Software works to a 3-month program-increment (Epic) cadence, with hardware and firmware integration of most recent increment completion.

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, three systems of interest are recognized, distinguished as Systems 1, 2, and 3. System 1 is the target system under development – in this case a military radio system as well as components that may become part of the product line. System 2 is the systems engineering process for developing, sustaining, and evolving new system capability – in this case the RC-Agile process that produces System 1, a new radio and possible product line component evolution. System 3 is the innovation process that evolves System 2 – briefly discussed in this article.

Figure 3 depicts case-study featured elements of the AAP for System 2. 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 2 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. Selected activities are traditionally depicted to show some differences in responsibilities associated with specific activities, and illustrate the employment of resources.

The AAP instance of the RC Agile process in Figure 3 depicts *key* System 2 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 2 enables and facilitates competitive cost/schedule/feature project delivery, multiple simultaneous projects within the ARC-210 program, and evolution of the ARC-210 Product Line.

A. *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. Page limitations imposed on this article don't permit detail explanations for the content of each of the five passive infrastructure categories. It is hoped that the content labels are sufficiently indicative.

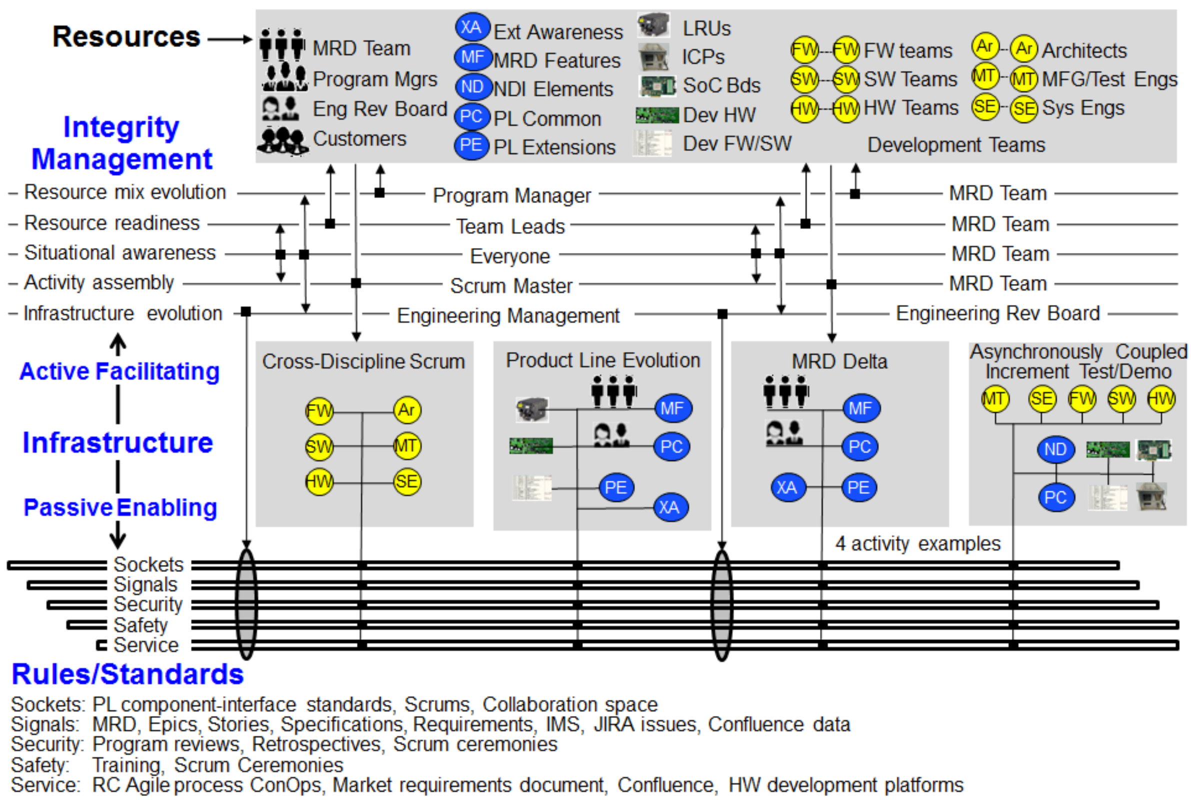

Fig. 3. RC Agile SE process for project delivery and product line evolution.

The five categories and what they encompass are:

- Sockets – physical interconnects.
- Signals – data interconnects.
- Security – trust interconnects.
- Safety – of process users, process, process environment.
- Service – process ConOps.

B. *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 is felt to be self explanatory and unnecessary of further explanation.

IV. Key Operational Process Aspects

A. *Product line architecture and strategy, as agility-enabling concept.*

Agile systems engineering processes are enabled by an agile architecture in the products they produce, as the agile product architecture permits the development process to reconfigure and augment the work-in-process as incremental learning occurs. The ARC-210 product family architecture is guided by four tenets: modularity, commonality, scalability, and standardization. Reusable modules in the product line include common boards, common firmware, common software, common requirements, common test cases, and common test platforms. The MRD Team and the Engineering Review Board establish the product family architecture and its interface standards. Development tries to maximize the space of commonality to evolve the product line. The MRD looks ahead to the future evolution of the product line, with selected future planned features brought into scheduled development as projects present opportunities. Customer requirements and features that fall outside the current product line component catalogue and the future evolutionary roadmap are welcome as competitive differentiation. The product line strategy allows new projects to reuse or modify elements of prior development, providing a competitive advantage that shortens project time and lowers project cost.

B. *Active SE management of all relationships, as agility-facilitating concept.*

In Figure 1 the RC Agile Relationship Management "green arrow" represents the process leaders that facilitate timely and effective communication between all of the process elements surrounding it in the depiction. Process leaders include the technical program manager, scrum masters, program managers, architects, and systems engineers. The purpose of this leadership group is to enables dynamic coupling internally and loose coupling externally.

C. *Active external awareness evolving the Product-Line Market Requirements Document.*

Figure 4 depicts the incremental development of the product line roadmap, aka the MRD. This is an internal document produced by the MRD Team, not shared widely even internally, beyond the architects. Their focus is to manage the product line. The Engineering Review Board assesses the market value, cost, and execution criteria. This activity is funded by internal IR&D. This is not aligned with development work. Sometimes the delta MRDs are ad hoc, find something a current customer needs, holds an emergency session and develops requirements that are brought into specifications under development.

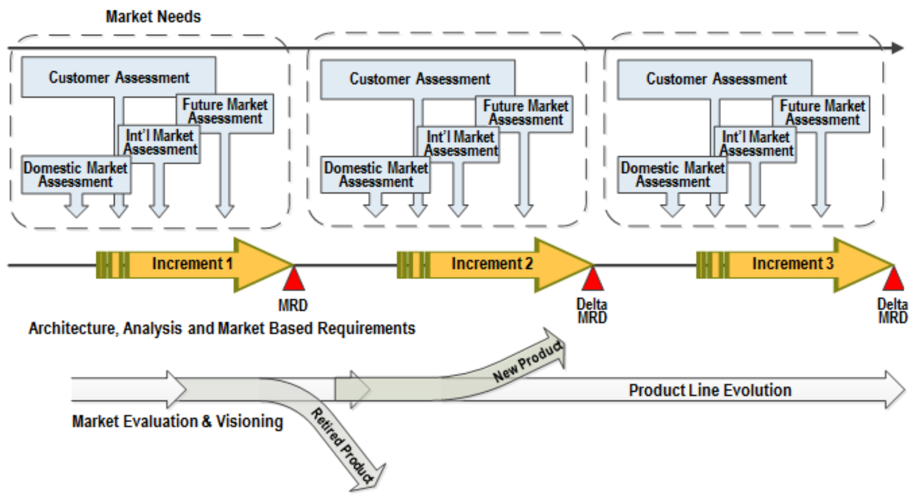

Fig. 4. Active external awareness incrementally updates the Market Requirements Document (MRD).

D. *Coupled cross-discipline agility.*

Figure 2 shows various forms of incremental development practiced by software, firmware, hardware, and system development teams. Epics at the software and firmware level are three-month increments that attempt to align, but the hardware level doesn't lend itself so readily to a constant fixed cadence. Nevertheless, hardware development, which includes circuit board fabrication and chasses fabrication, do proceed in successive increments that incorporate the most recently completed increments of software and firmware. Multi-discipline increments are asynchronously aligned for test and demonstrations that make use of the latest discipline-completed increment.

Some while ago the Communications Engineering group changed its facility layout to have a common collaboration area where all disciplines are co-located, with desks in low-rise cubicles that permit a standing engineer to see everyone that is present. This common space has multiple meeting rooms on the perimeter fully-outfitted to support ad-hoc cross-discipline discussions and presentations. Cross discipline scrum and scrum-of-scrum meetings make use of these meeting spaces. Also on the perimeter are entrance ways to discipline-specialized labs for engineering development that requires equipment support and security separation.

E. *Agile hardware development platform infrastructure.*

Software development generally employs commercially available object-oriented platforms that facilitate iterative and incremental development readily accommodating changes to work done in prior increments and iterations. Firmware development also has object oriented techniques and development platforms to facilitate incremental and iterative development. Hardware, however, doesn't have commercially available development platforms with this agility-supporting flexibility; so Rockwell's Communications Engineering group developed their own techniques and equipment support. The principal focus is on the firmware-containing circuit cards needed by software development for incremental testing during sprint iterations and especially at three-month increment testing events. Hardware has four platforms sequentially employed, in general, to accommodate this: commercially available system-on-chip prototype boards, Rockwell-developed circuit boards, a Rockwell-built integrated computing platform (ICP), and line replaceable units (LRUs) which are the target packaging chasses. The Product Line component inventory makes Rockwell-built circuit cards readily available as either actual end-product reusable cards or sufficiently similar to accommodate early software interface testing. The ICP is a Rockwell-built scalable circuit card rack with supporting power and cabling that can accommodate multiple circuit cards for early and incremental system testing. The LRU chasses are either drawn from the product line inventory or developed newly if necessary, but employment of an inventoried LRU permits early and incremental system testing as the next step up from the ICP.

F. *Active opportunity management.*

Risk management activity at Rockwell Collins includes opportunity management explicitly. Opportunity management is done by systems engineering. Much of this opportunity management is focused on product line evolution. Product line feature-addition opportunities, as Rockwell-funded extensions to a project feature requirement, are prime considerations. Risk management allocates a percentage of budget for mitigation strategies that burn down risk. Opportunity management analyzes Rockwell costs against Rockwell gains for doing something more than required to meet customer project expectations.

Opportunity management also draws from the product line component-employment opportunities that offer potential to accomplish something faster or with less budget than anticipated. Traditional risk mitigation also benefits from the product line strategy when multiple customers with the same risk make mitigation affordable that for a single customer would otherwise be unaffordable as a cost-benefit tradeoff.

Opportunity identification comes in more forms than adding product line features. For instance, outsource testing at lower costs than can be done in house is an example taken advantage of often.

V. Pattern-Based Model View of Key Operational Aspects

A. *The ASELCM Pattern*

The ASELCM Pattern is a formal 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 here for the Rockwell Collins case study:

- **System 1:** The Target System, subject of innovation over managed life cycles; in the Rockwell Collins case study, the radio system(s) being developed, deployed, and 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 the case

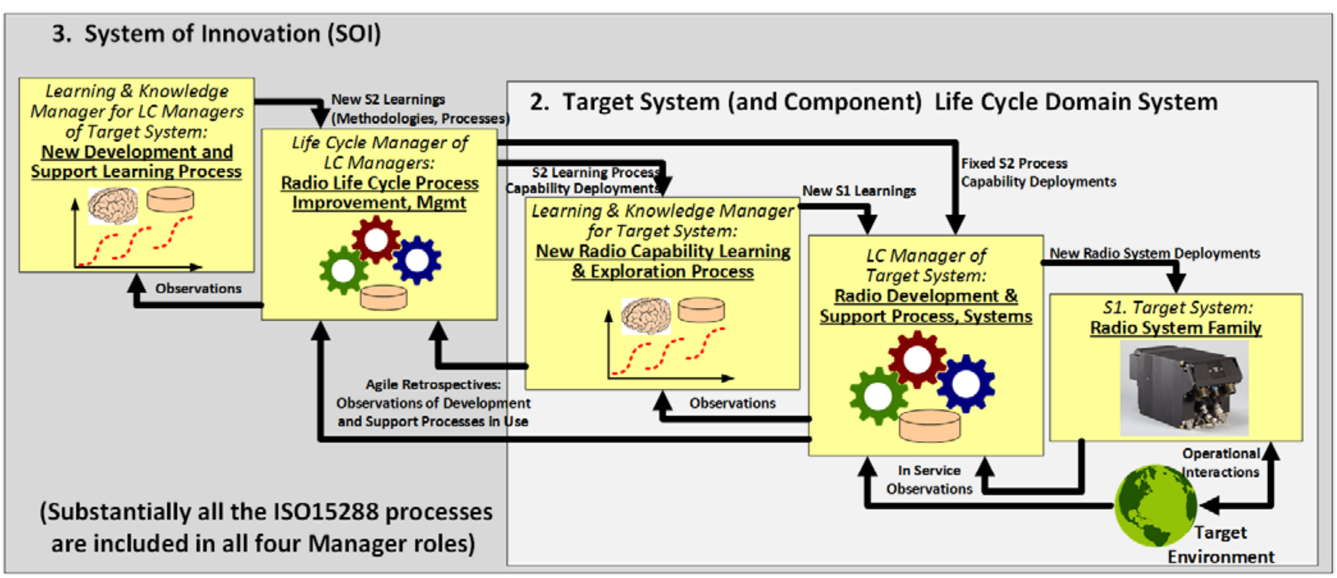

Fig. 5. ASELCM pattern system reference boundaries, configured for Rockwell Collins systems development and support case study.

studied, this includes all the external environment of the operational radio system(s), as well as all the (agile or other) development, production, deployment, support, security, accounting, performance, and configuration management systems that manage the (System 1) radio 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 the case studied, this includes the systems that define, observe, analyze (as in agile Process Retrospective) improve and support processes of development, deployment, service, or other managers of System 1.

The case study observations discussed in this paper and the Rockwell Collins workshop are further expressed using the ASELCM Pattern shown in Figure 5.

B. *Observed ASELCM System 1 Product Line Configurable Subset*

The ASELCM Workshop team observed recognition of the product line family management issues of several aspects of the ASELCM Target System reference model. Examples of radio-related aspects are illustrated in Figures 6 and 7.

C. *Observed ASELCM System 2 and 3 Product Line Subsets*

Beyond the System 1 product line, the Rockwell Collins discovery workshop noted recognition of evolutionary product line aspects of ASELCM Systems 2 and 3, including in particular the systems of test, as further described in (Cook and Schindel, 2015). Figure 8 illustrates that all the life cycle management processes of ISO 15288 are potentially subject to that agile product line oriented perspective.

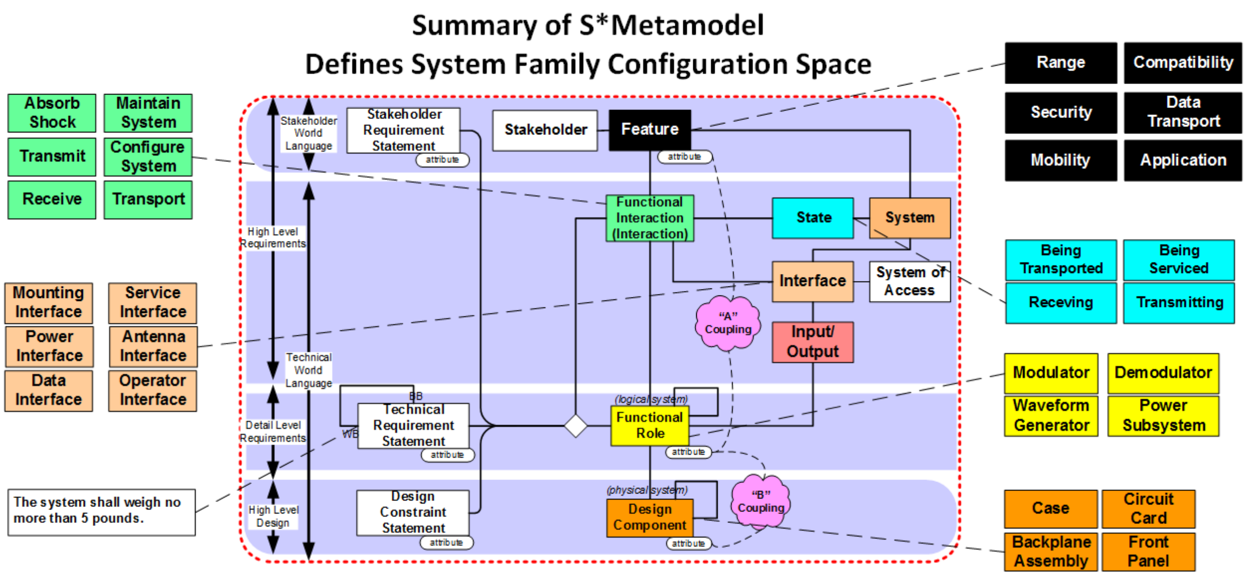

Figure 6: Product line family issues ultimately include the minimal system model issues.

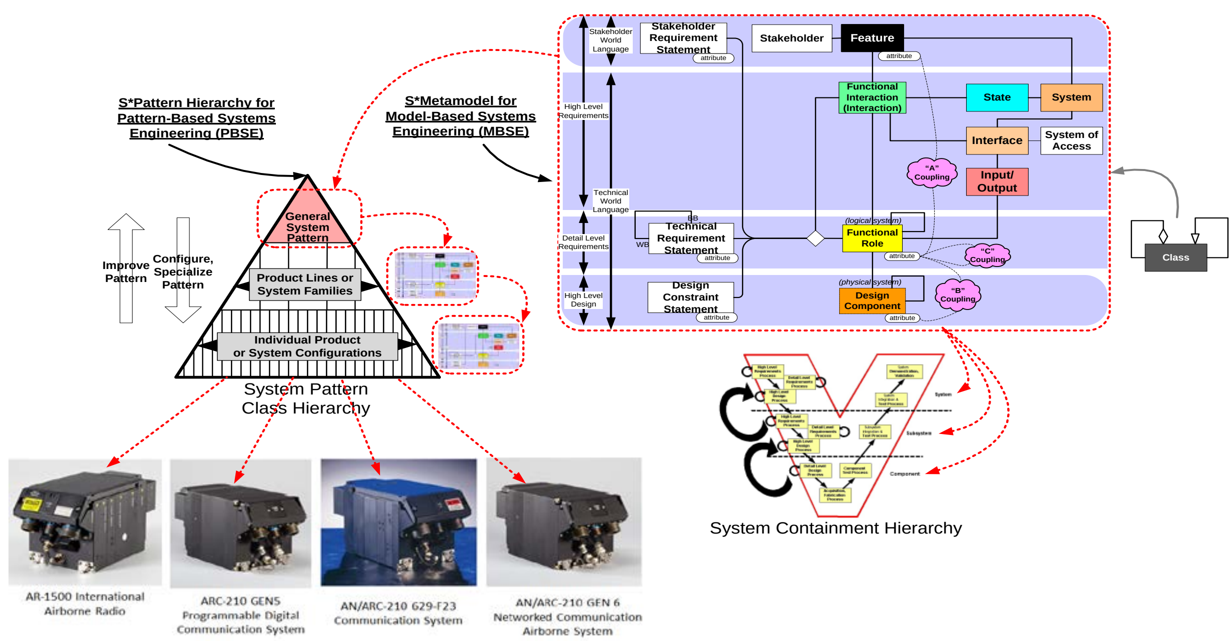

Fig. 7. Product lines configure varying products.

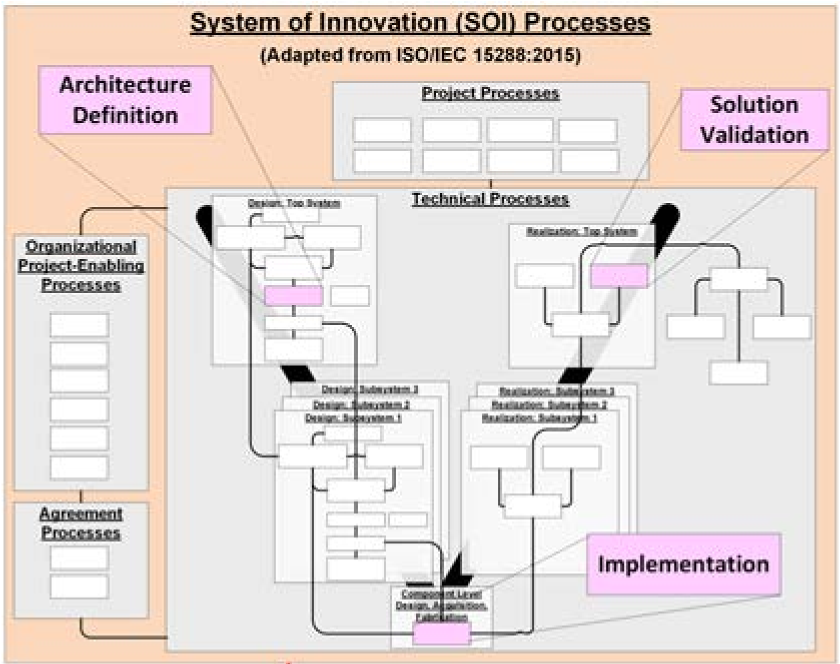

Fig. 8. All ISO15288 life cycle processes are candidates for Product Line Engineering configurability

Northrop Grumman's GCSS-J program designed and developed software componentry parameterized for reuse in similar but different applications (Dove, Schindel, Kenney 2017). Lockheed Martin's Integrated Fighter Group (IFG) was implementing Open System Architecture in its aircraft platforms, with a strong emphasis on reusable capability-modules (Dove, Schindel, Garlington 2017) to satisfy DoD-customer requirements and needs for urgent project completion. Navy's SpaWar System Center Pacific (SSC-Pac) was more overt in their architecture-enabled implementation of unmanned ground vehicle technology that could be shared among multiple project sponsors (Dove, Schindel, Scrapper 2016).

Northrop Grumman GCSS-J, Lockheed Martin IFG, and Navy SSC-Pac viewed the systems they produced as ones that would be perpetually evolving, and benefit would come from a System 1 agile infrastructure design that would facilitate reconfiguration, augmentation, and addition of modular capability at minimized cost and maximized speed. In all cases the emphasis is on perpetual sustainment and evolution enabled by a Product Line architectural approach.

VI. Concluding Remarks

This case study is based on the third workshop review in the ASELCM project, and sensitized the authors to the realization that three of the other workshop reviews also revealed core Product Line concepts without acknowledgement.

VII. References

Cook, D., Schindel, W., Utilizing MBSE patterns to accelerate system verification, in Proc. of the INCOSE 2015 International Symposium, Seattle, WA, July, 2015.

Dove, R., R. LaBarge. 2014. Fundamentals of agile systems engineering – part 1 & 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, and 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, and K. Garlington 2017. Case study: Case Study: Agile Systems Engineering at Lockheed Martin Aeronautics Integrated Fighter Group. (originally referenced working paper updated here to final published paper) www.parshift.com/s/ASELCM-04LMC.pdf.

ISO/IEC. 2013. ISO/IEC 26550:2013 – Software and Systems Engineering – Reference Model for Product Line Engineering and Management. ISO.

ISO/IEC. 2015. ISO/IEC 15288:2015 – Systems Engineering—System Life Cycle Processes. ISO.

INCOSE Product Line Engineering Working Group. 2015. www.incose.org/ChaptersGroups/WorkingGroups/analytic/product-lines.

INCOSE/OMG Patterns Working Group. 2016. www.omgwiki.org/MBSE/doku.php?id=mbse:patterns:patterns.

Meyer, M. and A. Lehnerd. 2011. *The Power of Product Platforms*. Free Press.

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. www.parshift.com/s/160718IS16-IntroToTheAgileSystemsEngineeringLifeCycleMBSEPattern.pdf

Schindel, W. 2011. What is the smallest model of a system? Proc. of the INCOSE 2011 International Symposium, International Council on Systems Engineering.

Schindel, W., T. Peterson. 2013. Introduction to pattern-based systems engineering (PBSE): Leveraging MBSE techniques. Proc. of INCOSE IS2013, Tutorial, June.

Schindel, W., V. Smith. 2002. Results of applying a families-of-systems approach to systems engineering of product line families. SAE International, TR 2002-01-3086.

Schindel, W., et al. 2015. MBSE methodology summary: Pattern-Based Systems Engineering (PBSE), based on S*MBSE models, V1.5.5A, INCOSE Patterns Working Group, retrieved 2015 from: www.omgwiki.org/MBSE/doku.php?id=mbse:pbse.

VIII. Acknowledgements

The SE process review leading to the case study described is this article was conducted by a workshop team that includes (alphabetically): John Davidson (Rockwell Collins), Rick Dove (Paradigm Shift International), Renee Frazier (Rockwell Collins), Kevin Forsberg (OGR Systems), Brendan Getz (Rockwell Collins), Michael Gries (Rockwell Collins), Kevin Gunn (MITRE), Robert (Will) Hartney (Rockwell Collins), Terry Hrabik (Rockwell Collins), David Lempia (Rockwell Collins), Gregg Lind (Rockwell Collins), James Pickel (Rockwell Collins), Jack Ring (OntoPilot), Ryan Sanger (Rockwell Collins), Bill Schindel (ICTT Systems Sciences), and Jim Sproul (Rockwell Collins).