JUNE 2018
VOLUME 21 / ISSUE 2

Synergy: Agile Systems Engineering and Product Line Engineering at Rockwell Collins

Rick Dove, dove@parshift.com

■ **ABSTRACT**

Rockwell Collins, in Cedar Rapids, US-IA, uses a product line approach for a family of radios they produce for military and international markets, and employs an agile systems engineering process that asynchronously aligns software, firmware, and hardware development increments. This mixed-discipline engineering group encompasses some 350 employees working on multiple projects simultaneously for multiple customers. Key concepts enabling and facilitating agility include a product line architecture, active forward-looking needs awareness and opportunity management, organizational relationship management, cross-discipline development increment coupling, and an infrastructure platform for hardware and firmware development.

INTRODUCTION

This article is based on an agile systems engineering process analysis conducted in 2015 at Rockwell Collins in Cedar Rapids, US-IA (now being acquired by UTC), that reviewed the communications engineering group's ARC-210 product line systems engineering process, which they called 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.

The RC Agile process 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 cannot always exist together as a coherent system. US Department of Defense (DoD) agencies may supply certain software and firmware non-developmental items (NDI) owned by the DoD for required employment. The international military market is prohibited from employing International Traffic in Arms Regulations (ITAR)-protected technology, and requires different standards than the domestic market, especially for security features that international customers want to be independent of US government standards.

Typical clean-sheet projects average three and a half years, with as many as fourteen circuit boards, and two to four chasses. The communications engineering group encompassed some 250 engineers, including about twenty systems engineers. Figure 1 depicts the central role systems engineers and others played in providing relationship management among the four key stakeholders involved in project success.

Notable process concepts that we will discuss include:

- Product line architecture and strategy, as an agility-enabling concept
- Active systems engineering management of all relationships, as an agility-facilitating concept
- Active external awareness evolving the product line roadmap
- Coupled cross-discipline agility
- Agile hardware development platform infrastructure
- Active opportunity management.

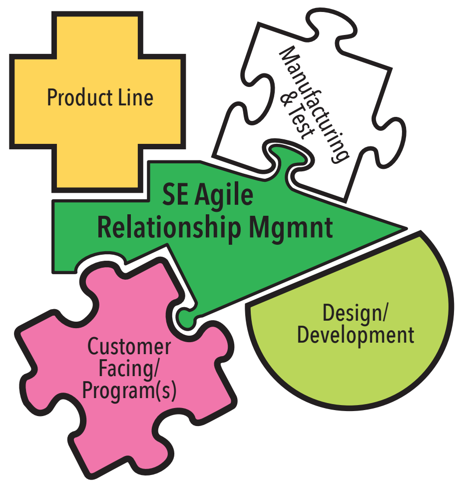

Figure 1. Systems engineering staff plays central role in stakeholder coupling.

Agile systems engineering processes are necessary and justified when the engineering environment has characteristics of caprice, uncertainty, risk, variation, and evolution (CURVE). See (Dove 2018) for more on CURVE in general. Rockwell Collins characterized their systems engineering CURVE environment as follows:

- Caprice: Unknowable situations

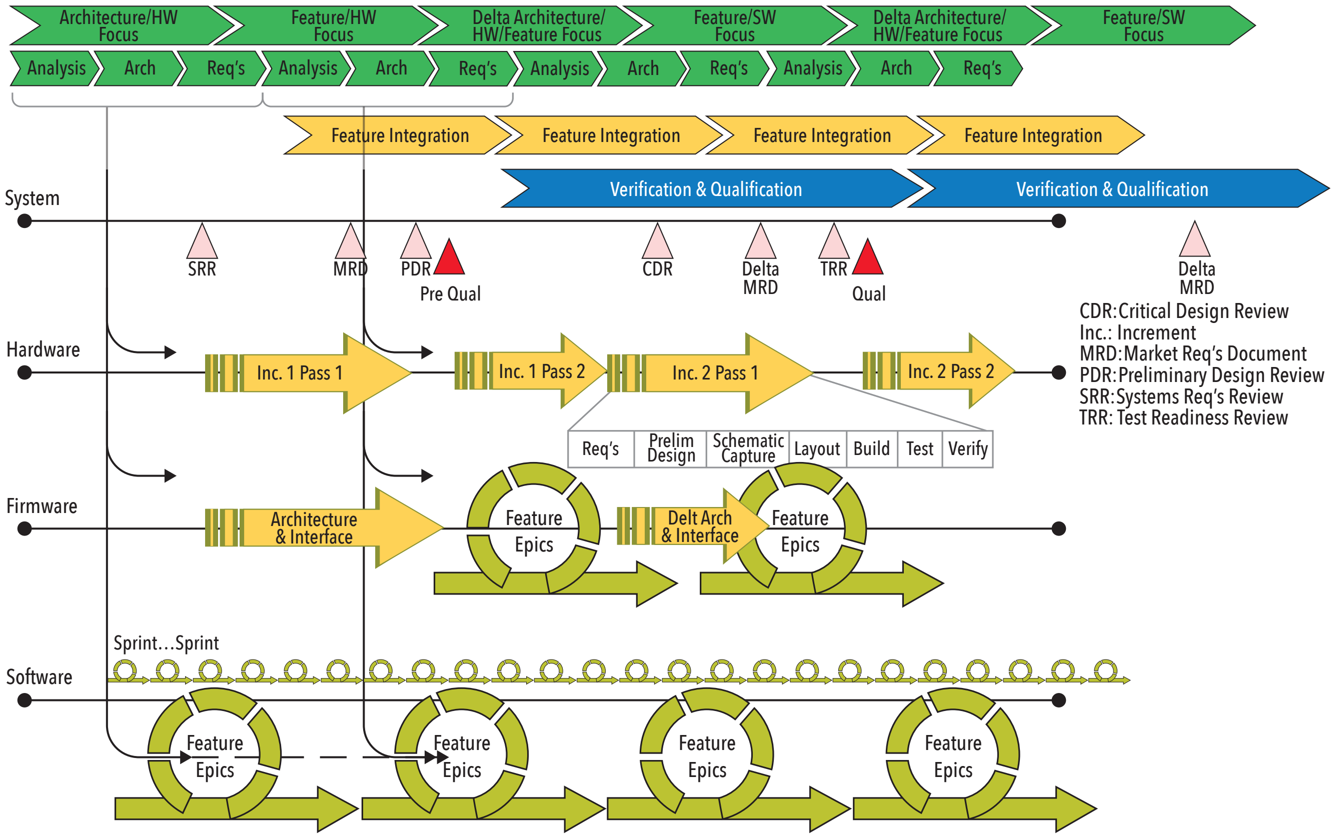

Figure 2. Software works to a three-month program-increment (epic) cadence, with hardware and firmware integration of most recent increment completion

- International and DoD markets have long and volatile acquisition cycles.
- Uncertainty: Randomness with unknowable probabilities
  - Feature most important requirements (MIRs) that are subjective and not clearly defined, which 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 a non-development item provided by the customer competition results in significant investment with no guarantee for return.
  - Unrealistic expectations of some customers exceed the current technology envelope.
  - Product development without DoD sponsorship brings risk from third 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, power, and cost 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. Software development follows a scrum- and SAFe-inspired approach, with the principle agile-process carryover to other disciplines being incremental development and frequent demonstrations of working product in process.

KEY OPERATIONAL PROCESS ASPECTS

Product Line Architecture and Strategy, as an Agility-Enabling Concept

An architectural pattern of reusable, loosely coupled, encapsulated resources enables agility in systems and processes (Dove and LaBarge 2014). Product line engineering inherently employs this agile architecture pattern. Four tenets guide the ARC-210 product line architecture. These four tenets are: 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 inherent product agility enables effective process agility as the agile product architecture permits the development process to affordably reconfigure and augment the work-in-process as incremental learning occurs. The product line architecture and its interface standards are a joint effort of a roadmap team and the engineering review board. Product development then tries to maximize the space of commonality to evolve the product line. 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.

Active Systems Engineering Management of All Relationships as an 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 the process elements surrounding it in the

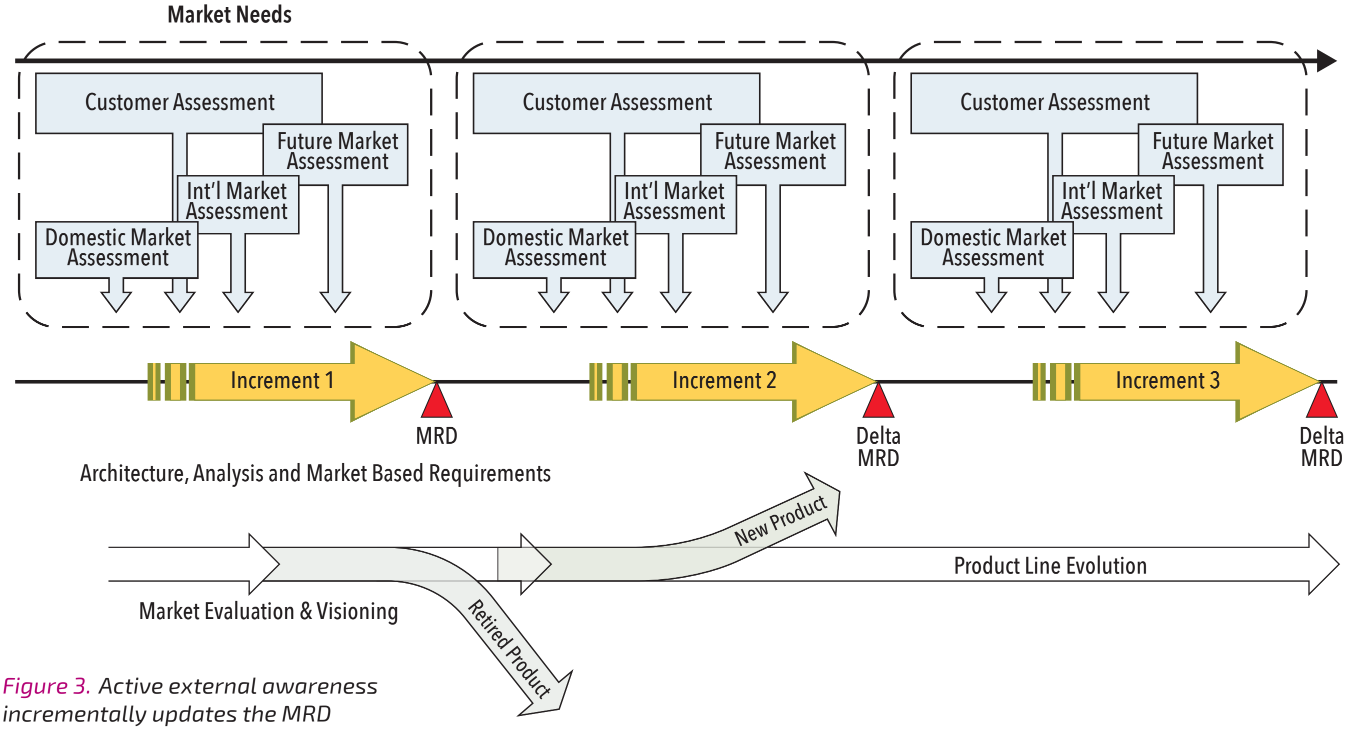

Figure 3. Active external awareness incrementally updates the MRD

depiction. Process leaders include the technical program manager, scrum masters, program managers, architects, and systems engineers. The purpose of this leadership group is to enable dynamic coupling internally and loose coupling externally.

Active External Awareness Evolving the Product Line Roadmap

Figure 3 depicts the incremental development of the product line roadmap, which we call the market requirements document (MRD). 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. The MRD team produces this internal document, which they do not share widely, even internally, beyond the architects. The MRD team works with the engineering review board to assess market value, cost, and execution criteria. Independent research and development funds this activity, which is not internal and not aligned with development work. Sometimes the delta MRD teams are ad hoc. In these cases they may find something a current customer needs, hold an emergency session, and develop requirements that they bring into specifications under development.

Coupled Cross-Discipline Agility

Figure 3 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 does not 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. Engineers asynchronously align multi-discipline increments for tests and demonstrations that make use of the latest discipline-completed increment.

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.

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, does not have commercially available development platforms with this agility-supporting flexibility; 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 sequentially employed platforms, 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.

Active Opportunity Management.

Risk management activity at Rockwell Collins includes opportunity management explicitly. Systems engineering achieves opportunity management. Much of this opportunity management focuses 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, which 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 that can be done in house is an example engineers take advantage of often.

IN CONCLUSION

Agility in systems engineering is a capability to respond organically to CURVEd opportunity and threat (Dove 2018). Active awareness of the external and internal environments drives this capability which an agile architectural pattern of both process and product enables.

Product line engineering provides an agile product architecture, which enables an agile systems engineering process that can evolve the product line faster and more effectively. The result is synergy in the dynamic interaction of product line engineering and agile systems engineering at Rockwell Collins.

Software has commercially available and open source object-oriented development platforms, which inherently structures an agile software product and facilitates incremental development, test, and demonstration evolution. But hardware in its many different forms has no such common platform although exceptions exist in some specific domains, such as semiconductor design and development. The product line engineering approach at Rockwell Collins helped engineers recognize a common set of needs across projects for firmware and electronic board development. Agile software development approaches helped the broader engineering group appreciate the values of incremental demonstrations, integration, and test of work in process. Product line architecture provides commonality in architecture, interface, and form factor that engineers have leveraged at Rockwell Collins to design and construct an infrastructure platform for integrating, testing, and demonstrating work-in-process of electronic, firmware, and software development. Again, this is a synergistic result of combining product line engineering with agile systems engineering.

Active awareness and evaluation of market and technology evolution is a hallmark of innovative product line engineering. At Rockwell Collins, the market requirements document team periodically reviews the external environment and the internal skill environment for both opportunity and risk. This team incrementally monitors and analyzes domestic, international, and future markets, and evaluates and analyzes customer satisfaction. The engineering review board assesses the market value, cost, and execution criteria. Internal independent research and development funds this activity to sustain a product line that that anticipates future requirements. Though project development project needs do not drive this activity, the discovery of new information is often related to a customer's current project needs and immediately brought down into specifications under development facilitated by the agile systems engineering capability. Again, we see synergy at work.

This synergy is also evident in other ASELCM project case studies. Northrop Grumman's GCSS-J program designed and developed software components parameterized for reuse in similar but different applications (Dove, Schindel, and Kenney 2017). Lockheed Martin's integrated fighter group is implementing open systems architecture in its aircraft platforms, with a strong emphasis on reusable capability modules (Dove, Schindel, and Garlington 2018). Navy's SpaWar System Center Pacific (SSC-Pac) employs architecture-enabled sharing of unmanned ground vehicle technology among multiple project sponsors (Dove, Schindel, and Scrapper 2016).

Rockwell Collins, Northrop Grumman, Lockheed Martin, and Navy SSC-Pac view the systems they produce as ones that will be perpetually evolving with benefit from the synergy of combining product line engineering and agile systems engineering concepts. ■

ACKNOWLEDGEMENT

This article draws upon and improves selected material from a broader coverage of the RC Agile process, first published as (Dove, Schindel, and Hartney 2017).

REFERENCES

- Dove, R., R. LaBarge. 2014. "Fundamentals of Agile Systems Engineering – Part 1." Paper presented at the International Council on Systems Engineering, International Symposium, Las Vegas, US-NV, 30 June-3 July. www.parshift.com/s/140630IS14-AgileSystemsEngineering-Part1.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." Paper presented at the International Symposium. International Council on Systems Engineering. Edinburgh, GB, 18-21 July. www.parshift.com/s/ASELCM-01SSCPac.pdf.
- Dove, R., W. Schindel, R. Hartney. 2017. "Case Study: Agile Hardware/Firmware/Software Product Line Engineering at Rockwell Collins." Paper presented at the 11th Annual IEEE International Systems Conference. Montreal, CA, 24-27 April. www.parshift.com/s/ASELCM-02RC.pdf.
- Dove, R, W. Schindel, M. Kenney. 2017. "Case Study: Agile SE Process for Centralized SoS Sustainment at Northrop Grumman." Paper presented at the INCOSE International Symposium. Adelaide, AU, 17-20 July. www.parshift.com/s/ASELCM-03NGC.pdf.
- Dove, R. 2018. "Enabling and Practicing Systems Engineering Agility – Prelude." *INSIGHT* 21 (2).
- Dove, R., W. Schindel, K. Garlington. 2018. "Case Study: Agile Systems Engineering at Lockheed Martin Aeronautics Integrated Fighter Group." Paper presented at the INCOSE International Symposium, Washington, US-DC, July 7-12. www.parshift.com/s/ASELCM-04LMC.pdf.

ABOUT THE AUTHOR

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*.