Systems Engineering Agility in a Nutshell

Rick Dove, dove@parshift.com, Kerry Lunney, kerry.lunney@thalesgroup.com.au,
Michael Orosz, mdorosz@isi.edu, Mike Yokell, mike.r.yokell@gmail.com

Abstract

Systems engineering must necessarily have the agility to anticipate and effectively respond to an increasingly dynamic and uncertain environment. Agile systems engineering, agile software engineering, and agile any-kind-of engineering share common goals and leverage common agility-enabling strategies. This article succinctly describes eight strategic aspects with application discussions at the systems engineering level.

Cracking the Shell

Agile software development has pioneered and proliferated methods for managing software projects (e.g. Scrum et al.) and engineering software products (e.g. XP et al.) when knowledge is uncertain and environments are dynamic. The success of these approaches is challenging other engineering disciplines to find better ways to navigate their development activities through similar uncertainties and dynamics.

Agile software development methods (process tactics) necessarily leverage the nature of software engineering. A software product is created by engineers who are supported by an integrated hierarchy of many tools (computers, code compilers, user interfaces, development platforms, et al.) that gives them fast turn-around control over design, fabrication, and verification. Piecewise functional prototypes can be created and tested in minutes, and deployed into evolving user product in hours and days.

Contrast that with electronic printed circuit board (PCB) development – procured parts, separate design and fabrication engineers, custom mechanical enclosure designs, procurement interaction, and supply chain issues. Oversimplified, but the nature of engineering activity and concerns is clearly very different. Making a PCB engineering process more agile would necessarily use different methods than software development. Nevertheless, those methods would have the same fundamental goals: reduce the adverse effects of uncertain knowledge and dynamic environments.

While tactical methods necessarily vary among different engineering domains (the how part), strategies for achieving common goals (the what and why parts) are domain independent. This article offers eight strategic aspects (Figure 1) that individually can improve the agility of engineering in any domain as well as at the systems engineering level. The next two pages present all eight strategic aspects, each in terms of needs (why) and behaviors (what), with a discussion of application from the systems engineering point of view.

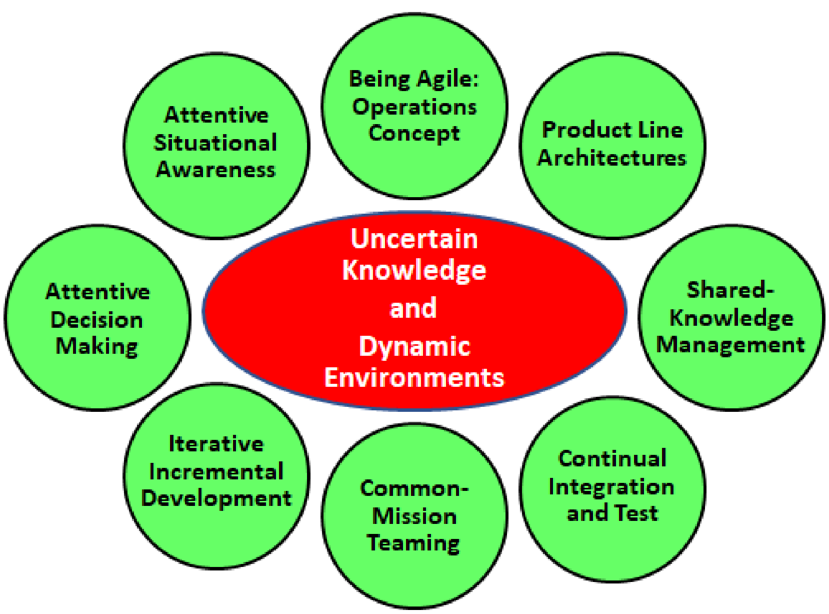


Figure 1: Eight strategic aspects of Agile Engineering

Insight 26 (2):11-14. International Council on Systems Engineering, June 2023

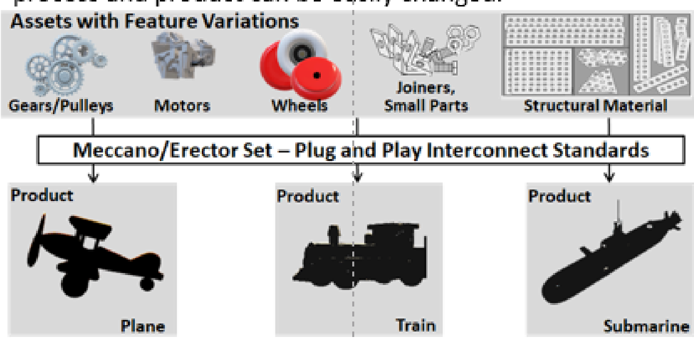
Product Line Architectures

Needs: Facilitated product and process experimentation, modification, and evolution.

Behaviors: Composable and reconfigurable product and process designs from variations of reusable assets.

Discussion: One fixed process approach won't fit all projects, so an appropriate process should be easy to compose and evolve according to context and usage experience. Variations of reusable assets are built over time as features are modified for different contextual usage.

A hallmark of agile systems engineering is iterative incremental development, which modifies work in process as suitability is repetitively evaluated. The agility of the process depends upon the agility of the product – so both process and product can be easily changed.



Notional Agile Architecture Pattern

Attentive Situational Awareness

Needs: Timely knowledge of emergent risks and opportunities.

Behaviors: Active monitoring and evaluation of relevant internal and external operational-environment factors.

Discussion: Are things being done right (internal awareness) and are the right things being done (external awareness)? Having the agile capability for timely and cost-effective change does little good if you don't know when that ability should be exercised. Situational awareness can be enhanced with systemic methods and mechanisms.

Alert in-the-moment constant attention

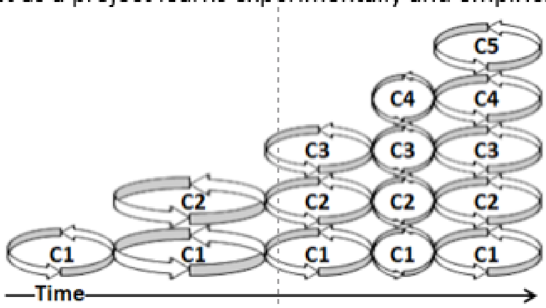
Iterative Incremental Development

Needs: Minimize unexpected rework and maximize quality.

Behaviors: Incremental loops of building, evaluating, correcting, and improving capabilities.

Discussion: Generally increments *create* capabilities and iterations add and augment features to *improve* capabilities.

- Increment cycles are beneficially timed to coordinate events such as integrated testing and evaluation, capability deployment, experimental deployment, or release to production.
- Increments may have constant or variable cadence to accommodate management standards or operational dynamics.
- Iteration cycles are beneficially timed to minimize rework cost as a project learns experimentally and empirically.



Iterative capability improvements (looping) and incremental capability additions (successive development periods)

Attentive Decision Making

Needs: Timely corrective and improvement actions.

Behaviors: Systemic linkage of situational awareness to decisive action.

Discussion: Empower decision making at the point of most knowledge. As a counter example, technical debt (a term for knowing something needs correction or improvement but postponing action) is situational awareness without a causal link to prompt action.

John Boyd's OODA loop

Common-Mission Teaming

Needs: Coherent collective pursuit of a common mission.

Behaviors: Engaged collaboration, cooperation, and teaming among all relevant stakeholders.

Discussion: Collaboration, cooperation, and teaming are not synonymous, and need individual support attention. Collaboration is an act of relevant information exchange among individuals, cooperation is an act of optimal give and take among individuals, and teaming is an act of collective endeavor toward a common purpose.

Tightly integrated coherent operation

Continual Integration & Test

Needs: Early revelation of system integration issues.

Behaviors: Integrated demonstration and test of work-in-process.

Discussion: Discovering integration issues late in development activities can impact cost and schedule with major rework. Synchronizing multiple domain engineering activities via continual integration and test provides faster and clearer insight into potential system integration issues.

SpaWar iteratively evolving unmanned technology integration platform.

Shared-Knowledge Management

Needs: Accelerated mutual learning and single source of truth for internal and external stakeholders.

Behaviors: Facilitated communication, collaboration, and knowledge curation.

Discussion: There are two kinds of knowledge to consider. Short time frame operational knowledge: what happened, what's happening, what's planned to happen. Long time frame curated knowledge: what do we know of reusable relevance, e.g., digital artifacts, lessons learned, and proven practices.

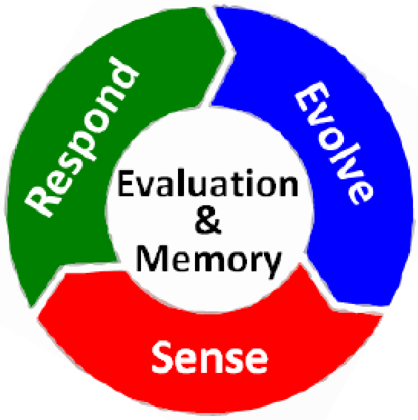
Depicted books represent information containers of any kind; but typically digital

Being Agile: Operations Concept

Needs: Attentive operational response to evolving knowledge and dynamic environments.

Behaviors: Sensing, responding, evolving.

Discussion: Agile systems engineering is not about doing Agile, it is about being agile. Being agile is a behavior, not a procedure – a behavior sensitive to threats and opportunities in the operational environment, decisive when faced with threat or opportunity, and driven to improve these capabilities. Deciding how to implement any of the core aspects, even this one, should be done with sense-respond-evolve principles in mind as aspect objectives.



Three principles that operationalize agility

The succinctness of the descriptions and the display on two pages is done with purpose. Descriptive content attempts to be sufficient to inform and direct application intent without overly constraining approaches compatible with culture, organizational readiness, and possible contract constraints. This two-page brief can function as a personal things-to-consider scope reminder or as a whole-picture guide for collaborative discussion or improvement.

Each of the aspects can individually improve capability to deal with uncertain knowledge and dynamic environments in any engineering process; but to have something intended as an agile engineering process at either domain or system level requires multiple aspects operating in concert. Individual aspects are strategic concepts that can tactically manifest over a range of intensity. Thus, the degree of agility is a product of how many of these aspects are operational as well as how effectively each one contributes to the agility required by the operating environment.

These eight aspects in their current form have emerged from the pooled knowledge of the authors of this article – knowledge gained from their experiences in case study work, university research work, and responsibilities for organizational systems engineering processes and practices. None of these aspects are new concepts. What is new is the amalgamation organized as domain independent fundamental strategies for engineering when knowledge is uncertain and operating environments are dynamic.

Figure 2 depicts the relationship between the eight strategic aspects presented here and the nine foundational concepts in the roadmap developed for Agility in the Future of Systems Engineering (Willett et al. 2021). Maturing and evolving the concepts on the right side will leverage the aspects on the left side.

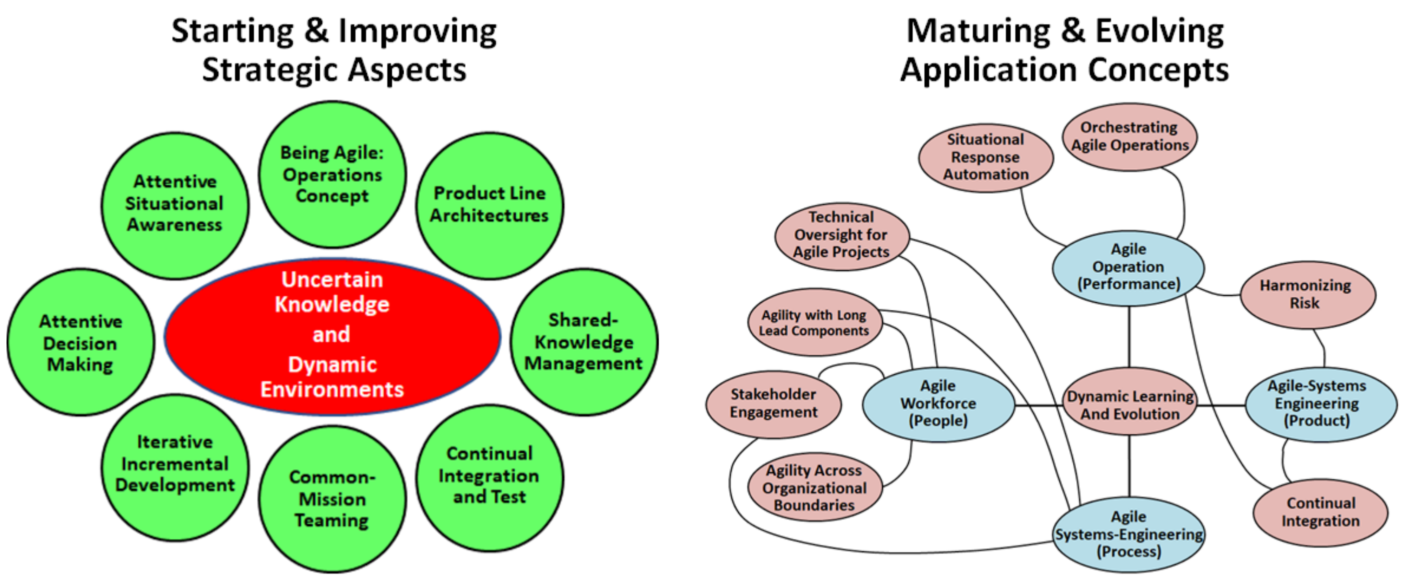


Figure 2. Large organizations likely have units working in both early and advanced stages

Whether your organization is down the road already or just thinking about the values of being more agile, each of the aspects likely has some form of practice in place already. One way to inspire actionable awareness of the collective view beyond theory is to develop and share a short case study – one that shows each aspect in real practice instances somewhere in your organization.

References

Dove, R., Lunney, K. Orosz, M, and Yokell, M. 2023. Agile Systems Engineering – Eight Core Aspects. Proceedings International Symposium. International Council on Systems Engineering. Honolulu, HI, July 15-20. www.parshift.com/s/230715IS23-AgileSE-EightCoreAspects.pdf

Willett, K. D., R. Dove, A. Chudnow, R. Eckman, L. Rosser, J. S. Stevens, R. Yeman, and M. Yokell. 2021. Agility in the Future of Systems Engineering (FuSE) – A Roadmap of Foundational Concepts. Paper Presented at the 31st Annual INCOSE International Symposium, Virtual: July 17-22.

Bios

Rick Dove is an independent researcher, systems engineer, and project manager generally focused in the systems agility and systems security areas. He chairs the INCOSE working groups for Agile Systems and Systems Engineering and for Systems Security Engineering. He leads both the Agility and Security project areas for INCOSE's Future of Systems

Engineering (FuSE) initiative. He is an INCOSE Fellow, and author of *Response Ability – the Language, Structure, and Culture of the Agile Enterprise*.

Kerry Lunney is the Country Engineering Director and Chief Engineer in Thales Australia. She has extensive experience developing and delivering large system solutions, working in various industries including ICT, Gaming, Financial, Transport, Aerospace and Defence, in Australia, Asia and USA. She also participates in a number of global working groups and research projects. Kerry is a Past President INCOSE, and holds the Expert Systems Engineering Professional (ESEP) qualification. She is also an INCOSE Fellow, and a Fellow of Engineers Australia with the status of Engineering Executive and Chartered Professional Engineer.

Dr. Mike Orosz directs the Decision Systems Group at the University of Southern California's Information Sciences Institute (USC/ISI) and is a Research Associate Professor in USC's Sonny Astani Department of Civil and Environmental Engineering. Dr Orosz has over 30 years' experience in government and commercial software development, systems engineering and acquisition, applied research and development, and project management and has developed several successful products in both the government and commercial sectors.

Dr. Mike Yokell is a leader in Systems Engineering in the US Aerospace and Defense Industry. He has been the US representative to international standards-setting bodies for Systems and Software Engineering and was the project editor for two new international standards on Systems of Systems Engineering. Mike is certified as an expert systems engineering professional by INCOSE. He holds multiple US and European Patents for Model Based Systems Engineering and large-scale immersive virtual reality.