Systems Engineering Agility – A Guide Book for Systems Engineers

October 28-31, 2024

NDIA Systems & Mission Engineering Conference, Norfolk, Virginia

Presenter: Rick Dove

Collaborators: Rick Dove, Kerry Lunney, Mike Orosz, Mark Vriesenga, Jennifer Whitby, Mike Yokell

Abstract **Caveat: Dense small-font text slides are provided as visuals, not for reading comprehension**

Agile systems engineering can design, build, sustain, and evolve purpose-fulfilling creations when knowledge is uncertain and operational environments are dynamic. That is the promise – but what does it look like and how does it do that?

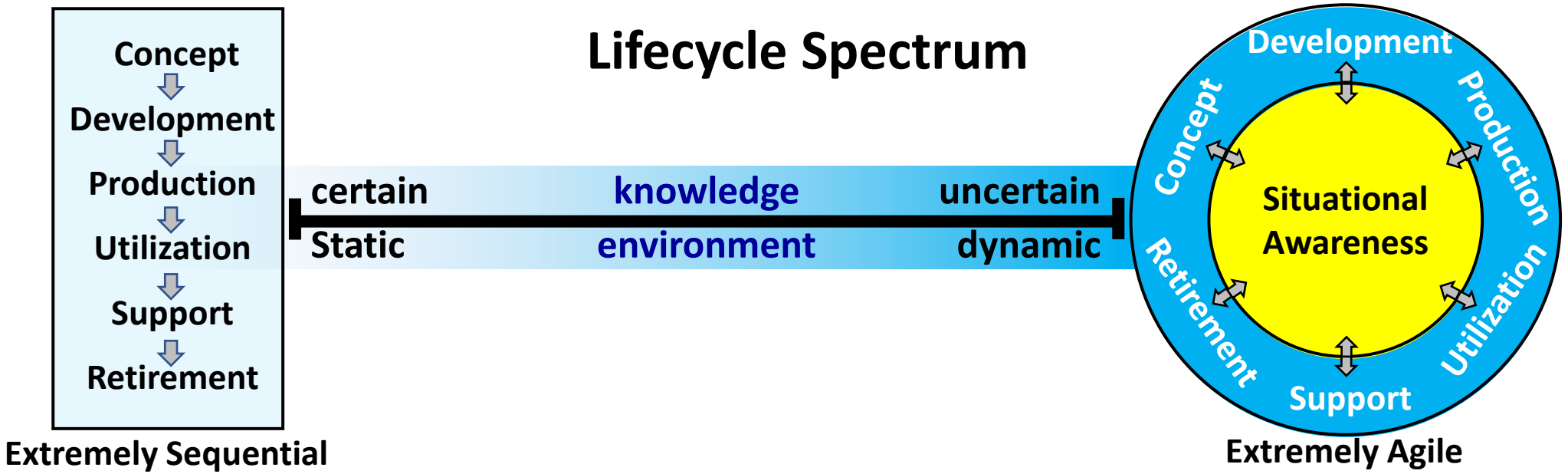
INCOSE is producing a 4-page Systems Engineering Agility Primer as part of its Future of Systems Engineering (FuSE) initiative, with publication expected mid-2024. The Primer focuses on the what (behavior) and why (needs) of 8 strategic aspects that enable and facilitate agile systems engineering. The Primer is intended as an introductory and motivational overview suitable for individual reading as well as support for group discussions, workshops, or tutorials that want to explore the strategic aspects in more detail with the aid of a succinct desk-top reference.

This presentation will include a quick review of the Primer content and then focus on how that material is being expanded into a more detailed 50-60 page Guide. The Guide is a current work in process with an INCOSE production target date of mid-to-late 2025. An October 2024 presentation will show some finished treatment as well as expose some open questions about alternatives and offer opportunities for involvement in review and completion of the work.

The Guide is currently developed as a five-section document, each of which will be discussed in the presentation. The Guide opens with sections on Purpose and Context, and then a third section devotes a few pages to each of the eight aspects – Adaptable Modular Architectures, Iterative Incremental Development, Attentive Situational Awareness, Attentive Decision Making, Common-Mission Teaming, Shared-Knowledge Management, Continual Integration and Test, and Being Agile: OpsCon. A fourth section discusses Design and Employment Considerations for each of the aspects, and a fifth section closes with a series of Case Stories on how each of the aspects is employed in a diversity of project domains.

Bio: Rick Dove is an independent researcher, systems engineer, and project manager generally focused in the system agility and system security areas. He chairs the INCOSE working groups for Agile Systems & Systems Engineering and for System Security Engineering; and leads INCOSE's Future of Systems Engineering (FuSE) project areas for both systems engineering agility and systems engineering security. He is an INCOSE Fellow, and book author of *Response Ability – the Language, Structure, and Culture of the Agile Enterprise*.

Context



Agile systems engineering is systems engineering as known in:

- **ISO/IEC/IEEE standards**
- **Vee model**
- **INCOSE Systems Engineering Handbook**
- **however it is practiced by organizations that design, build, and sustain systems.**

What distinguishes it as "agile" systems engineering is its:

- **leverage of situational awareness**
- **enablement of continual system evolution**
- **intent to satisfy mission rather than plan.**

Agile systems engineering

is a strategy-based method

for designing, building, sustaining, and evolving systems

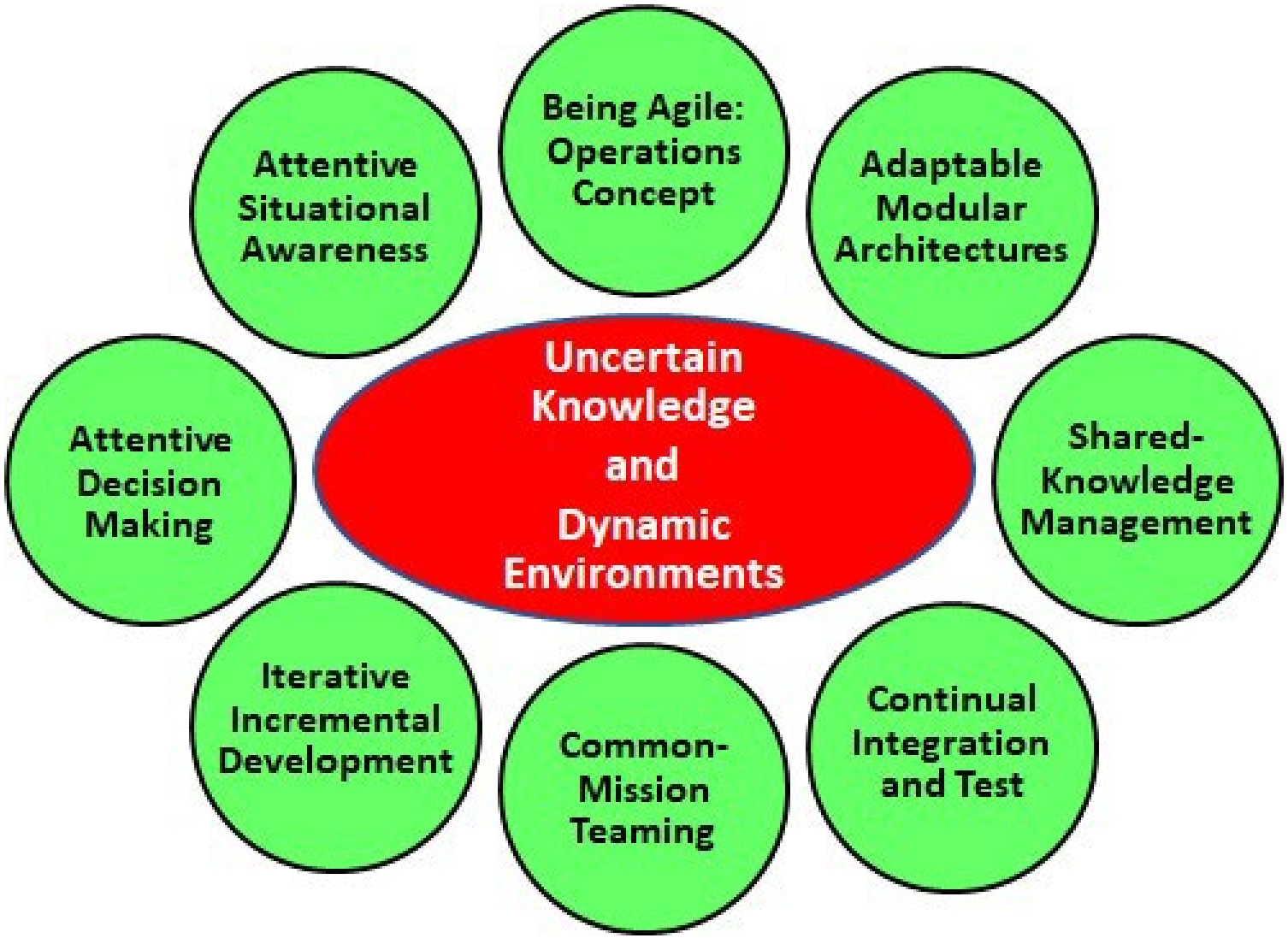
when knowledge is uncertain and/or environments are dynamic

Agile System Engineering is a what, not a how.
There are many hows, principally focused on the development phase,

and also many focused on a single (software) engineering discipline,

Eight Strategic Aspects That Enable Agility

www.researchgate.net/publication/373092973



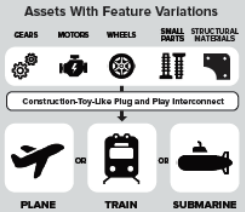
Adaptable Modular Architectures

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.

Iterative Incremental Development

Needs: Minimize rework, maximize quality, facilitate innovation.

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).

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 test and demonstration 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.

Iteratively evolving self-driving technology integration platform.

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.

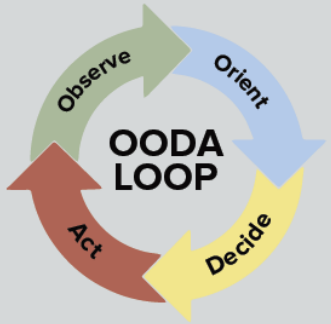
Alert, in-the-moment, constant attention.

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.



Responsible attention may take time, but never pauses.

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.

Information containers of any kind, available to all, and 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.

Primer Link: https://www.incose.org/publications/products/agile-primer

Working Outline

1. **Purpose and Nature of the Guide – 1 page (what to expect and not to expect, how to use this guide)**
2. **Context – 2 pages: why and when is an agile SE approach needed, its objectives, its nature, and its benefits**
3. **Eight Strategic Aspects – 2 pages each**
    1. **Adaptable Modular Architectures**
    2. **Incremental Iterative Development**
    3. **Attentive Situational Awareness**
    4. **Attentive Decision Making**
    5. **Common-Mission Teaming**
    6. **Shared-Knowledge Management**
    7. **Continual Integration & Test**
    8. **Being Agile: Operations Concept**
4. **Considerations – 2-3 pages**
   Values (minimizing rework, maximizing quality, facilitating innovation) and social considerations for each Aspect
5. **Case Story Examples – 4 pages each**
   (how Aspects have been employed in a variety of environments):
    1. **Uninhabited Vehicle Technology – (US Navy)**
    2. **Aircraft Radio Product Line – (Collins)**
    3. **Vehicle Innovation Product Family – (Tesla)**
    4. **High Performance Sports Car – (McLaren)**
    5. **Space System – (Multi-project amalgamation).**
    6. **Lottery System – (Multi-project amalgamation).**
    7. **(and others … possibly).**

Total: ~50 pages

A taste of each section follows

Chapter 1: Purpose and Nature

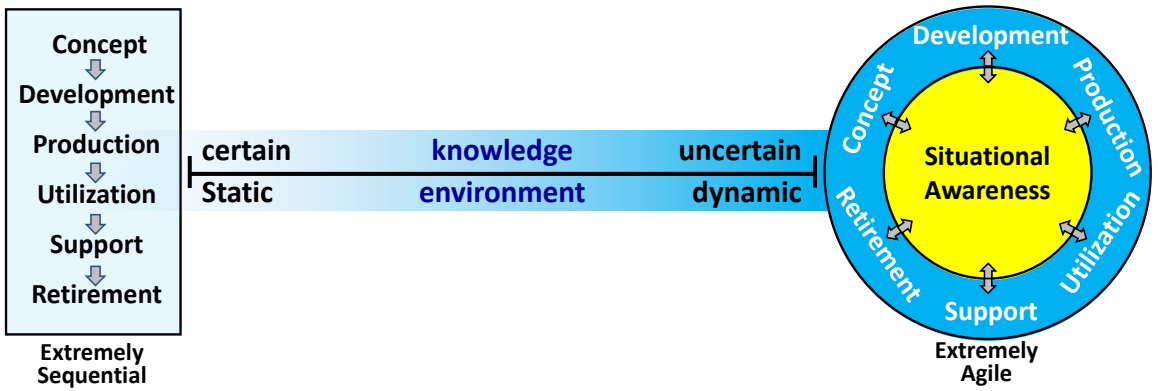
There is a general wish for one or more standard Agile Systems Engineering processes to choose from, prescriptive like Scrum or framework-ish like Safe.

“I want to think about engineering the product, not the process; that should be a repeatable procedure.”

But … You have to think about the process – because knowledge is uncertain and the environment is dynamic

This Guide explores the breadth and depth of 8 strategic aspects, portraying look (with theory) and feel (with examples).

Chapter 2 – Context



INCOSE's Vision 2035 expressed a fundamental **need**:
"Systems engineering anticipates and effectively responds to an increasingly dynamic and uncertain environment."

Ashby's law expresses a timeless **need**:
"When the variety or complexity of the environment exceeds the capacity of a system the environment will dominate and ultimately destroy that system."

Asynchronous Stage Activity

- You're using a personal computer in the morning = utilization stage.
- In the afternoon an SSD (Solid State Drive) is installed = production/deployment stage.
- Which replaces the Hard Drive = retirement.
- Next day the BIOS are adjusted for optimal SSD performance = support stage.
- Supplier is creating a new widescreen monitor based on market demand research = development stage.
- Supplier is always dreaming up product line extensions = concept stage.

Concurrent Stage Activity

- You're driving a year-old Tesla = utilization stage.
- Simultaneously Tesla is downloading an AI upgrade = production/deployment stage.
- Simultaneously that upgrade is replacing an older capability = retirement stage.
- Simultaneously engineers are creating a market-desired faster charge capability = development stage.
- Simultaneously engineers are dreaming up tomorrow's upgrade = concept stage.
- Simultaneously Tesla is asking to schedule a part-replacement house call based on monitored stats = support stage.

Chapter 3 – Strategic Aspects (outline)

Look and Feel of 8 Aspects at 2 Pages Each

Why/Needs (why this strategy is needed)
- xxx

What/Behaviors (what this strategy looks like as an outcome)
- xxx

Informative similar concepts
- xxx

How/OpsCon (Sense-Respond-Evolve triplet)
- xxx

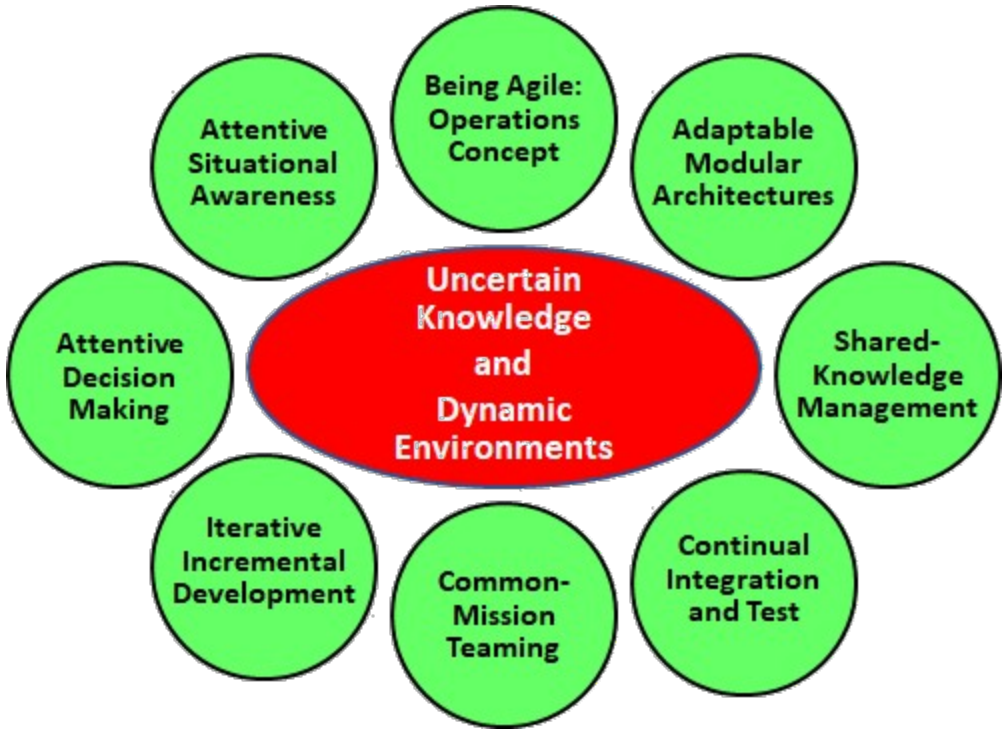
Practice Examples (expanded selections from Chapter 5 Case Stories)
- xxx

Theory (how it mitigates uncertain knowledge and dynamic environments)
- Uncertainty: xxx
- Dynamics: xxx

Value (how it minimizes rework, maximizes quality, facilitates innovation)
- xxx

Social considerations (what can make it desirable/embraceable/engaging)
- xxx

Comparative Assessment Metrics (quantitative and qualitative evaluation)
- xxx



Preliminary material has been developed for all outline areas,

Chapter 4 – Considerations[1/2]

Table below is a single-bullet concept example – multiple bullets for each Aspect is in process.

Explicitly identifying the values wanted from each Aspect gives measurable purpose and evaluation criteria to Aspect design, operation, and improvement.

Aspect	Minimize Rework	Maximize Quality	Facilitate Innovation
AMA	• Reusable modules have proven operational history.	• Improved modules can be swapped in transparently.	• Experimental modules can be tried and replaced easily if they fall short.
IID	• Discovers need for rework early and affordably.	• Iterations converge on better solutions.	• Iterations and increments enable affordable innovation experiments.
ASA	• Reveals need for rework early.	• Reveals new value opportunities.	• Reveals innovation opportunities.
ADM	• Curtails work in wrong direction as soon as possible.	• Adjusts requirements for new values as soon as possible.	• Seizes the opportunity – carpe diem.
CMT	• Avoids misdirected correction-needed work effort.	• Improves outcomes within constrained schedule.	• Considers more option within aligned thinking.
SKM	• Reveals development conflicts early.	• Moves otherwise lost rework time into improved outcomes.	• Shared cognitive knowledge instigates/fosters innovation.
CIT	• Reveals issues early for affordable rework.	• Conserves funds for performance improvement.	• Enables wip demonstrations to suggest innovations.
BAO	• Optimizes expenditure with tight sense/respond/evolve loops.	• Instills trust with unequivocal sense/respond/evolve loops.	• Creates delight with venturesome sense/respond/evolve loops.

AMA: Adaptable Modular Architectures
IID: Iterative Incremental Development
ASA: Attentive Situational Awareness
ADM: Attentive Decision Making
CMT: Common-Mission Teaming
SKM: Shared-Knowledge Management
CIT: Continual Integration & Test
BAO: Being Agile OpsCon

Chapter 4 – Considerations$^{2/2}$

Below is a conceptual start that will be expanded.

Aspect	Social Considerations
AMA	• Coherent easy methods for finding and employing relevant reusable assets.
IID	• Celebration and reward for improvement and innovation.
ASA	• Readily available data (transparency) makes trustworthy environment.
ADM	• Empower decision making at point of most knowledge.
CMT	• Clear, compelling mission. • Opt-in teaming and recruitment rather than task assignment. • No penalty for corrected misalignment.
SKM	• Transparency and knowledge equality evokes a sense of inclusion and facilitates higher personal performance.
CIT	• Incremental wip test and demonstration provides progress visibility and frequent encouraging feedback.
BAO	• Incentivize learning and sharing and reasonable risks w/o penalty for trying that fails. • Trust is necessary among all project participants, else agility is impeded. • Engender a sense of mission-based meaningful involvement with a like-minded team. • Get suppliers and subs to work toward mission as opposed to contract.

AMA: Adaptable Modular Architectures
CMT: Common-Mission Teaming
IID: Iterative Incremental Development
SKM: Shared-Knowledge Management
ASA: Attentive Situational Awareness
CIT: Continual Integration & Test
ADM: Attentive Decision Making
BAO: Being Agile OpsCon

Chapter 5 – Case Stories

Introduction

The prior section dealt with the eight strategic aspects from a theory point of view, what they look like. This section deals with the eight aspects from a practice point of view, what they feel like.

What they look like was described in terms of needs and behaviors – the context-independent why and what of the strategic aspect. What they feel like is necessarily described as how these aspects manifest as context-dependent tactical implementations.

Five case stories follow, each is 4 pages long, intended as:

- Inspiration from in-practice examples
- Enticement as succinct short reads
- Variety across application domains
- Diversity of implementation methods
- Reference for revisits over time

The examples provided in each case story are from actual practice; but are referred to as case stories rather than case studies because:

- they contain examples selected for aspect employment diversity rather than comprehensive coverage
- some case stories are amalgams from multiple but similar project activities rather than from a single project or organization

(Then … A single short paragraph for each case story, positioning its nature and diversity within the group)

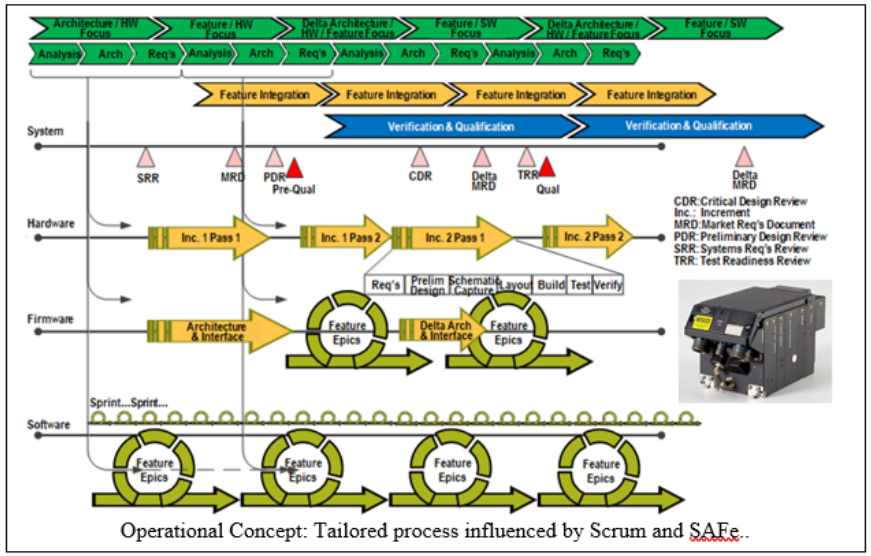
Aircraft Radio Product Line

Systems Engineering Agility Case Story (draft 240825)

Introduction

Rockell Collins, circa 2015, was a supplier of radios to military and commercial aviation markets when they invited a team from INCOSE to study their agile systems engineering approach in a three-day workshop. Examples in this case story are drawn from notes, recordings, and published papers associated with that study (Dove, Schindel, Hartney 2017).

The figure below is Rockwell's depiction of their overall process. This figure will aid in appreciating how all the moving parts relate to each other as they are subsequently explained in the framework of the eight aspects.



Operational Concept: Tailored process influenced by Scrum and SAFe..

High Points

Adaptable Modular Architectures:

- Product line architectural tenets: modularity, commonality, scalability, standardization.
- Engineers could easily switch among 3 or so projects to accommodate resource availability.

Iterative Incremental Development:

- Software and firmware attempted to align on 30-day increments.
- Hardware budgeted for two versions of completed capabilities for feedback and variations.

Attentive Situational Awareness:

- External awareness drove a continually evolving Market Requirements Document (MRD).
- Internal awareness looked for project opportunities to extend the product line.

Attentive Decision Making:

- Opportunity-hungry decision making – MRD-driven project requirements development.
- Facility reconfigured to bring all technical teams together to facilitate fast decision making.

Common-Mission Teaming:

- Common corporate mission, culture, and self identity was strongly embraced by everyone.
- SE cross-group relationship management ensured common project mission understandings.

Shared-Knowledge Management:

- Evolving product line and reusable assets provided long-term knowledge management.
- Evolving MRD provided mid-term, backlog-like, knowledge management.

Continual Integration & Test:

- Hardware team provided prototype FPGA boards to software team early on as a start.
- ICP (integrated computing platform) evolved to final product "like an erector set construction."

Being Agile: OpsCon

- Systems engineering managed sense/respond/evolve real-time coherence among stakeholders.
- Corporate values and product line establishment provided agility foundation.

Adaptable Modular Architectures

Reusable modules (assets) in a product line approach included common boards, common firmware, common software, common requirements, common test cases, and common test platforms.

Product line strategy allowed new projects to reuse or modify elements of prior development, providing a competitive advantage that shortened project time and lowered project cost.

SW/FW/HW engineers were involved in three or so projects at any one time, switching easily among projects with common process architecture to accommodate resource availability and evolving priorities.

Product line architectural tenets were: modularity, commonality, scalability, standardization.

Iterative Incremental Development

Increments at the software and firmware level were three-months long that attempted to align.

Hardware planed and budgeted two version with two iterations each, each with completed capabilities that provided learn-by-doing user feedback and product line variations (different power spec, connectors, etc).

Hardware development of circuit board, RF equipment, power supply, human interface, and chasses iterate within their few ad-hoc increments; incorporating the most recently completed increments of software and firmware.

Iterations and increments of engineering efforts were integrated asynchronously into an evolving integrated test configuration that eventually became the finished product.

Attentive Situational Awareness

Situational awareness was the engine that drove a continually evolving Market Requirements Document (MRD) – a closely held roadmap of product line evolution plans.

External situational awareness data sources included customer interviews, technology evolution monitoring, and national and international needs development in military and commercial markets.

Related internal situational awareness sources included workforce skills and competency monitoring, contracts with potential to fund MRD items, and urgencies that might prompt IR&D funding.

Opportunity awareness was proactive: assessing values that could be gained if exceeding customer requirements could enable use of a product line feature; and looking for net gains to the product line if a non-required cost was incurred.

Attentive Decision Making

There was an opportunity-driven tight coupling between the MRD team's situational awareness and the Systems Engineering team's requirements development.

"For instance," they explained, "We just got some really hot intel. This customer we're pursuing needs this feature. We'll hold an emergency session and develop requirements that are instantaneously brought down in to our specifications under development. This is highly responsive."

The facility was reconfigured to bring all technical teams together in one place to facilitate fast decision making when an event required a multi-party response agreement.

Common-Mission Teaming

Their high level mission was embraced by everyone: "to be the most trusted source of communication and aviation electronics solutions."

SE took an active role in cross group relationship management to ensure common mission understanding and sustainment – this activity wa credited as the leading enabler and facilitator of SE agility.

The facility was reconfigured for entire workforce collaboration space, bringing multiple disciplines together in one central location so that information could be shared in real time quickly and dynamically. The technical teams were all at one end of the building. They had coffee together.

They had huddle rooms for ad hoc meetings. At any instant they could get 4 -6 people together with all the resources needed for an impromptu meeting. They also had closed labs they can go into and work.

Shared-Knowledge Management

Systems Engineering viewed their primary role as the real-time knowledge management and coherence channel (connective tissue, middleware) between customer/program execution, design/development execution, manufacturing/test execution, and product line execution.

Lessons learned were incorporate in next phase of development and kept in a lessons learned data base.

Product line reusable assets and active catalog management was a principle part of their long term knowledge management.

Continual evolution of the Market Requirements Document (MRD) was mid-term (backlog-like) knowledge management.

Typical commercial software aps supported real-time, development-in-progress, knowledge management.

Continual Integration & Test

Hardware engineering provided software engineering with something to work with early on. This started initially with prototype boards, and then graduated to what they called an ICP (integrated computing platform) – described as looking like an erector set construction in continual evolution.

The ICP capability evolved iteratively and incrementally through multiple physical stages over time until it transformed into the real product.

The engineering manager believed the key enabler to their agility was being able to allow development early on for different phases of hardware, getting hardware pieces out to software engineering on a just-in-time basis as the feature set matured.

The ICP even had some of the manufacturing needs incorporated to make transition to production easier.

Being Agile: Operations Concept

Present was a very well organized and managed framework of practice guidelines/procedures/policies that enabled coherent/consistent/safe (risk free-ish) pursuit of business and engineering activity. Everybody articulated and understood what these were. It was acculturated – a cornerstone of the accepted/respected culture, not spoken of as "these are the rules we are required to follow," but rather as these are our values that guide how we conduct our affairs.

As they explained it: "Being agile involves building adaptable relationships to manage through environmental uncertainty and dynamics…we consciously enable dynamic coupling internally and lose coupling externally."

As they explained it: "A key agility-enabling concept is the active management of all relationships, more so than product line engineering, which just provides context."

As they explained it: "Rather than applying a cook book Agile Manifesto we leverage corporate values and drive for product line establishment."

MRD situational awareness activity is the sense/respond/evolve innovation engine.

References

Dove, R., Shindel, W., Hartney, R. 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.

Dove, R. 2018. Synergy: Agile Systems Engineering and Product Line Engineering at Rockwell Collins. Insight 21(2):43-46. International Council on Systems Engineering, June.

High Points – Aircraft Radio Product Line (Collins)

Adaptable Modular Architectures:
- **Product line architectural tenets: modularity, commonality, scalability, standardization.**
- **Engineers could easily switch among 3 or so projects to accommodate resource availability.**

Iterative Incremental Development:
- **Software and firmware attempted to align on 30-day increments.**
- **Hardware budgeted for two versions of completed capabilities for feedback and variations.**

Attentive Situational Awareness:
- **External awareness drove a continually evolving Market Requirements Document (MRD).**
- **Internal awareness looked for project opportunities to extend the product line.**

Attentive Decision Making:
- **Opportunity-hungry decision making – MRD-driven project requirements development.**
- **Facility reconfigured to bring all technical teams together to facilitate fast decision making.**

Common-Mission Teaming:
- **Common corporate mission, culture, and self identity was strongly embraced by everyone.**
- **SE cross-group relationship management ensured common project mission understandings.**

Shared-Knowledge Management:
- **Evolving product line and reusable assets provided long-term knowledge management.**
- **Evolving MRD provided mid-term, backlog-like, knowledge management.**

Continual Integration & Test:
- **Hardware team provided prototype FPGA boards to software team early on as a start.**
- **ICP (integrated computing platform) evolved to final product "like an erector set construction."**

Being Agile: OpsCon
- **Systems engineering managed sense/respond/evolve real-time coherence among stakeholders.**
- **Corporate values and product line establishment provided agility foundation.**

Uninhabited Vehicle Technology

Systems Engineering Agility Case Story (draft 240825)

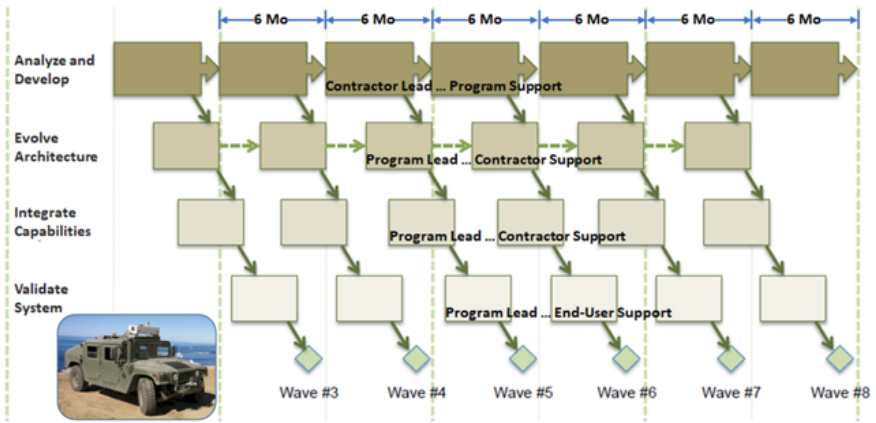
Introduction

This project did continuous agile development and evolution of uninhabited, off-road vehicle technology.

In 2015, a team from INCOSE studied the project's 6-month overlapping-increment wave-model process, and published it as a case study (Dove, Schindel, Scrapper 2016). Examples in this case story focus solely on the project's employment of the eight strategic aspects of systems engineering agility. Most notably, the process put a prime emphasis on enabling and facilitating team effectiveness: creating an embraced culture of engagement, a collective consciousness emerging from comprehensive real-time information support, and a team conscience on a mission for the end users.

The project ran multiple sub-projects with multiple sponsors simultaneously. At study time, the project employed 6 subcontractors with 4-5 engineers each for device development and 60 government employees for continuous process and product systems engineering and project support.

The figure captures the project's overall wave-model process. This figure shows how the project's engineering tasks relate to each other as they are subsequently explained in the framework of the eight aspects.



Operational Concept: Iterative incremental development with overlapping six-month waves

High Points

Adaptable Modular Architectures:

- Architecture and interface standards designed, evolved, and enforced by program management.
- Task composability facilitated by common interactive interfaces for contractors and internal resources.

Iterative Incremental Development:

- Six month increments balanced development time with sponsor-demanded progress evidence.
- Minimum of two iterations within increments demonstrated progress with integrated testing.

Attentive Situational Awareness:

- Full-day user table-top exercises at each increment end verified and revealed appropriate needs.
- Active engagement by internal and external personnel was constantly monitored for sufficiency.

Attentive Decision Making:

- The shared culture expected and demanded timely attentive closure to open issues.
- Clarity of vision and objective provided coherent criteria for rapid resolution and appreciation.

Common-Mission Teaming:

- Contract and program personnel on equal footing as team members; expected to interact family-like.
- Team meetings were opened with taxpayer and user stories of empathetic needs.

Shared-Knowledge Management:

- An interactive real-time knowledge base orchestrated the interaction of all project participants.
- A collective consciousness emerged, sensitive to in-the-moment activity coherence and conflict.

Continual Integration & Test:

- Three operational test vehicles were fitted with component interaction monitoring and data gathering.
- Work-in-process was experimentally installed on operational vehicles to reveal integration challenges.

Being Agile: OpsCon

- Real-time shared design and decision knowledge orchestrated interaction of engineers and managers.
- Leadership and culture placed highest value on full team, in-the-moment, situational awareness.

Adaptable Modular Architectures

The modular open architecture was developed for the systems as a key enabler for sustaining the agile process capability, which depended upon the ability to easily replace and upgrade individual system components in successive evolutionary waves. The open architecture was planned and evolved before each systems engineering wave began, under the principle that architectural refactoring incurs time and cost that should be avoided.

Project objectives included modular, reconfigurable executable mission modules, with the government retaining design, control, and ownership of the product architecture to ensure that developed components conformed to this architectural standard.

The wave-model process used a modular task architecture, providing structure and flexibility to accommodate multiple-sponsor diverse organizational and programmatic issues, and lower costs to all sponsors with re-usable product modules across projects.

Iterative Incremental Development

The wave-model process offered meaningful progress feedback in project-appropriate 6-month cycles, long enough to accommodate incremental new-capability development time, and short enough to demonstrate frequent progress to sponsors and allow learning and affordable re-planning and corrective action when needed.

Within the six month increments development contractors were encouraged and expected to bring work-in-process iterations in for occasional and informal integrated testing.

Attentive Situational Awareness

A formal Experimentation and Test Plan was structured to accumulate evidentiary information for feedback into the development cycle, and ensures critical capability objectives and functional requirements are being met.

User full-day workshops held at the end of every wave that include project engineers, contract developers, and marines from different units with different needs. These workshops explained the technology, responded to survey questions, and conducted table top exercises on different types of missions that often revealed tactical operational procedures not anticipated by the systems engineering team.

Quality of process-engagement was actively monitored for both internal personnel and external developers, which triggered inquiry if insufficient, and then mitigation by removing causes, or personnel replacement. This was subjectively measured by leadership in frequency and quality of communications, wiki-contributions, and meeting contributions.

Risk burndown adapted the Scrum burndown chart approach to focus on progressive and continuous timely reduction of project risk that tracked known risks. Burndown of risks was monitored in a series of technical reviews and test events analyzing evidentiary information.

Attentive Decision Making

A clarity of vision and objectives provided common coherent criteria so that everyone brought their knowledge to bear within a common relevant context. Disagreements caused by different criteria were avoided, and interactions were decisive rather than unresolved. This approach drove rapid closure among multiple perspectives.

The culture did not tolerate inaction. Where the culture didn't fit an individual (or vice-versa), that individual either moved on or adjusted.

Common-Mission Teaming

Outside contractors were completely integrated with inside personnel as full team members, forming a family-like relationship of all-for-one and one-for-all.

A prime emphasis was placed on enabling and facilitating team effectiveness, creating an embraced culture of engagement, a collective consciousness emerging from comprehensive real-time information support, and a team conscience on a mission for the end users.

Full and active engagement with the systems engineering process intent and the systems engineering project objectives was expected. All team members were expected to share the common mission, supporting and supported by all other team members, at all times. The nature of the systems engineering process, its leadership, and the transparency of comprehensive real-time project status provided team-engagement sensitivity.

Every team meeting began with short, personalized stories of the taxpayer and the warfighter, as customers and users, in relationship to expected project outcomes. Affordability, usability, and effectiveness were typically the focus of these stories; a common team sense of empathy and conscience was the objective.

Shared-Knowledge Management

An interactive, real-time, shared knowledge repository designed, built, and constantly evolved by internal staff was considered a necessary enabling cornerstone of project success. This federated amalgamation of COTS (commercial off-the-shelf) applications was called the Continuous Integration Environment (CIE) in recognition of its functional purpose. CIE orchestrated the interaction of engineers, managers, and external contract developers.

CIE provided open-process visibility and access to all project status and information affecting collective ability to meet project objectives and individual ability to perform. Leadership and culture demanded and practiced open communication on all team issues, with no private conversations allowed.

The intent was to facilitate a real-time collective consciousness, where all team members were plugged in to all information associated with full project success, as well as to the information of relevance to their specific responsibilities and tasks. New data, new decisions, new issues, new test results, wherever and whenever they occurred, rippled through the relevant federation of CIE components and CIE user views immediately. This collective consciousness manifested for the team much like it does for musicians in a symphony orchestra, where off notes and bad timing are immediately sensed by all.

Continual Integration & Test

Three operational uninhabited vehicles were used as evolving continual integration and test platforms.

To mitigate late revelations an experimentation test task was added to the testing procedure about mid-way through contractor development. This task requires developers to put their work-in-process on the vehicle to interact with everything else, providing an early indication of the integration challenge.

Integration platforms were fitted with an evolving collection of real-time, on-board, data gathering tools to monitor network loading, CPU loading, variable values, et al. to identify potential contributors to unexpected performance or interfere with other devices.

Being Agile: Operations Concept

The power of the project's process emerged from a core focus on team effectiveness: creating and sustaining a culture of engagement, a collective consciousness of shared knowledge, and a shared conscience of providing affordable value to the warfighter and the taxpayer. Leadership and culture placed highest value on full team, in-the-moment, situational awareness.

A custom built Continuous Integration Environment (as they called it) provided real-time knowledge evolution that orchestrated the interaction of engineers, managers, and external contract developers. This shared design and decision knowledge minimized rework, collaborative progress meetings maximized quality, and intentional experimentation facilitated innovation.

Reference

Dove, R., Schindel, W., Scraper, C. 2016. Agile Systems Engineering Process Features Collective Culture, Consciousness, and Conscience at SSC Pacific Unmanned Systems Group. INCOSE International Symposium. Edinburgh, Scotland, UK, 18-21 July. www.researchgate.net/publication/308083752

High Points – Uninhabited Vehicle Technology (US Navy)

Adaptable Modular Architectures:
- **Architecture and interface standards designed, evolved, and enforced by program management.**
- **Task composability facilitated by common interactive interfaces for contractors and internal resources.**

Iterative Incremental Development:
- **Six month increments balanced development time with sponsor-demanded progress evidence.**
- **Minimum of two iterations within increments demonstrated progress with integrated testing.**

Attentive Situational Awareness:
- **Full-day user table-top exercises at each increment end verified and revealed appropriate needs.**
- **Active engagement by internal and external personnel was constantly monitored for sufficiency.**

Attentive Decision Making:
- **The shared culture expected and demanded timely attentive closure to open issues.**
- **Clarity of vision and objective provided coherent criteria for rapid resolution and appreciation.**

Common-Mission Teaming:
- **Contract and program personnel on equal footing as team members; expected to interact family-like.**
- **Team meetings were opened with taxpayer and user stories of empathetic needs.**

Shared-Knowledge Management:
- **An interactive real-time knowledge base orchestrated the interaction of all project participants.**
- **A collective consciousness emerged, sensitive to in-the-moment activity coherence and conflict.**

Continual Integration & Test:
- **Three operational test vehicles were fitted with component interaction monitoring and data gathering.**
- **Work-in-process was experimentally installed on operational vehicles to reveal integration challenges.**

Being Agile: OpsCon
- **Real-time shared design and decision knowledge orchestrated interaction of engineers and managers.**
- **Leadership and culture placed highest value on full team, in-the-moment, situational awareness.**

Vehicle Innovation Product Family

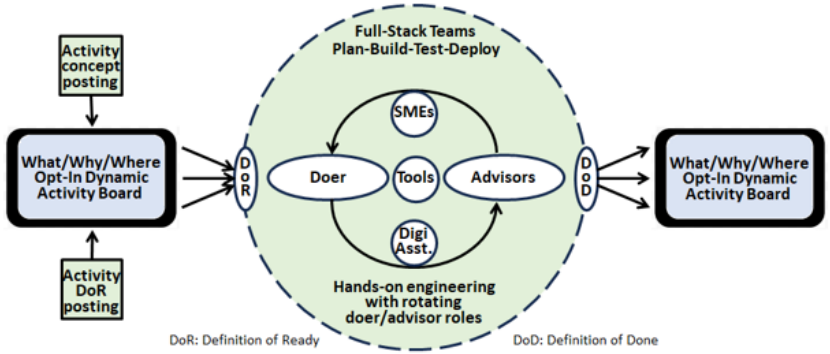
Systems Engineering Agility Case Story (draft 241010)

Introduction

Examples in this case story are drawn from what was publicly available about engineering agility at Tesla Motors in its automotive business. Source material is principally a selective amalgamation from many differently focused video accountings of Joe Justice's engagement at Tesla (circa 2020) as an employee and his subsequent continued monitoring through 2023.

This accounting is purposely written in the past tense as it is based on information from a specific time period; but the chosen examples have timeless value whether still in practice at Tesla or not.

Tesla operated at the extreme right of the life cycle spectrum, where all systems engineering phases occurred concurrently and systems engineering was a continuous activity affecting all vehicles in development, in production, and in operation. Group experiential engineering was practiced, where full-stack teams rotate all members periodically through roles of hands-on doer and advising to the current doer. Design eventually emerges from successful building, modifying, and testing.



Operational Concept: Digitally assisted hands-on group experiential engineering. Output may be a new increment of capability or an iterative improvement on an existing capability.

High Points

Adaptable Modular Architectures:
- Engineering process, product, and factory composed of modules with stable interfaces.
- Rapid opt-in modular teaming enabled by culture and personnel-interaction handbook rules.

Iterative Incremental Development:
- 60 part changes a day average in 2021/22 reflected constant improvement iteration pace.
- Automated safety certification testing enabled continuous iterative development.

Attentive Situational Awareness:
- Ubiquitous AI tools provided awareness of engineering, product, and production improvement trends.
- Software called Autobidder rapidly found alternative suppliers to resolve supply issues.

Attentive Decision Making:
- DSM tool eliminated human manager decision delays by answering questions directly.
- Service options and scheduling driven by operational monitors and digital twin profile.

Common-Mission Teaming:
- Opt-in mission-pulled team composition.
- Clear and monitored mission focus to minimize/eliminate multi-mission switching costs.

Shared-Knowledge Management:
- DSM-AI did long term knowledge curation and cross-company dissemination.
- WIP knowledge continuously evolved on personal phones, computers, and factory screens.

Continual Integration & Test:
- Each vehicle on the production line was treated as an automated integration & test platform.
- Production line cars ran software doing integration regression testing as parts were added.

Being Agile: Operations Concept
- Continual innovation mission on product and production performance, efficiency, and cost.
- Production processes and tooling were designed to facilitate rapid/frequent change.

Adaptable Modular Architectures

Tesla used modular architectures with interconnect specifications for everything: product, process, facility, production, tooling, and teams. Adaptable modular architectures appear to be a dominant mental pattern for all types of systems at Tesla.

Interconnect specs were allowed to evolve asynchronously with backward compatible adaptors.

Service personnel were rapid module-swap replacement trained rather than repair trained (repairs done at the factory provided a continuous learning experience).

Culture and anti-handbook handbook (2.5 pages) provided personnel collaborative interface rules for rapid adaptable opt-in modular teaming.

Tesla built what they called Gigafactories, as these factories cover a larage amount of geography. Each successive factory build took less time then the previous build as they employed a modular factory-design approach that learned and improveed on both design and construction techniques of replicated factory modules.

Iterative Incremental Development

At Tesla every car on the production line had capabilities that could be improved at any time, and new capabilities that could be added at any time.

Tesla was making 60 part changes a day on average in the 2021/22 time frame. Not a fixed number, but indicative of the pace of constant improvements. Part variation among cars was not a service issue as the modular architecture maintained stable interfaces to allow transparent replacements.

At Tesla full self-driving was an incremental capability under development in 2023, expected to be an optional upgrade in the future for cars in service.

Body sections that began as hundreds of welded & bolted parts were iteratively improved (same capability) over time to become 2 castings.

Speed of safety certification dictated iteration speed, so every car drove itself through an in-factory, non-destructive, certification test, and registered that result with NHTSA (National Highway Traffic Safety Adminsitration).

Attentive Situational Awareness

Ubiquitous data-driven Digital Self Management (DSM) AI tools provided continuous situational awareness of enginering, product, and production improvement trends at every factory station.

DSM-AI, for instance, measured paint quality on a dozen dimensions. Its data set was very high resolution images of every car's painted surfaces. If an improvement was tried it could tell immediately if it was better.

With full-team, mission-based, engineering motivated to innovate, many eyes, ears, and sensitivities monitored global information sources for opportunities and threats.

Vehical's had on-board monitoring for how it was being used by passengers and drivers, how it was functioning as a machine, and how suitable it was for continued trouble-free service.

Tesla-developed software, referred to as Autobidder, could do mass polls in seconds to find suppliers with prices, capabilities, and track records if supply reliability became an issue. An excellent example circumvented the 2021 chip shortage issue that heavily impacted other auto manufactures.

Attentive Decision Making

Tesla had replaced, as much as possible, human decision points with apps that could render a decision immediately; believing that speed of decision making fundamentally determines the speed of product development and response. As a concept, Digital Self Management questions why some other human is needed to decide issues that delay personal action.

DSM-AI attentive awareness provided data for immediate deployment decisions. Data authorized the decision, no other authority was required. If an automated test was passed, immediate deployment needed no further authorization.

Customer service options and scheduling decisions were driven by on-board product operational monitors and digital twin profiles – no human was in the loop.

Common-Mission Teaming

Teaming opportunities were posted dynamically to personal phones and on ubiquitous factory screens to attract immediate opt-in participation, much like Open Space conference techniques (join us for what's happening at location X).

A 3.5-page employee handbook established guardrails and behavior expectations for collaborative opt-in teaming.

Teams functioned like so-called Mobs in software development, i.e., everyone worked on the same thing at the same time in the same space, with rotating roles of driver (doer) and navigators (advisors).

The work-day mission was constant innovation to improve performance of everything. Job descriptions included expectations to reduce the number of parts, process steps, and lines of code constantly.

Shared-Knowledge Management

Digital Self-Management (DSM) at Tesla refers to AI/ML (artificial intelligence, machine learning) software that learns, evolves, and provides data for personal decision making, for design improvement assistance, and for product improvement evaluation. Evolving data sets are shared among all factories.

Tesla kept a complete digital-twin CAD model of every car, and updated it with operational wear or changes based on real-time analytics from each car.

Each station in the factory has DSM, which creates a real-time instant feedback loop for skill and job improvement, and for product and production improvement.

As to project status radiators: There were no product backlog or Kanban boards as employed in agile software engineering; there was only a doing column on ubiquitous monitor boards and on all phones. Most projects are extremely short so the focus is on what is happening now.

Continual Integration & Test

Every car had potentially new components, so each component and car on the production line was a work-in-process, automated-test, integration platform.

At Tesla every car put itself through automated testing of every non-destructive compliance and certification test. A car on the production line ran software in what was referred to as "factory mode," which ran and displayed resuts of constant regression testing as the car was being assembled.

Tesla autopilot was originally developed so that cars could drive themselves through certification testing.

Being Agile: Operations Concept

At Tesla the metric that mattered above all else was the pace of innovation – which meant frequent change, enabled by culture and facilitated by infrastructure; e.g., Enterprise Resource Planning (ERP) infrastructure software was custom designed and built as a central nervous system to connect customer experience information to relevant staff to reveal problematic issues immedialy.

Frequent change was enabled by cultural expectation in job descriptions demanding innovation, by financial rewards for delivered innovation, and by peers expecting collaborative innovation.

Frequent change was facilitated by infrastructural support from all strategic aspects described in this document, and from tight sense/respond/evolve loops supported by DSM tools.

With constant innovation as the operational mission, agility in everything was a natural. side effect.

Any approval that waited for a manager was automated by software, which largely eliminated decision delay and replaced management, allowing nearly 100% of staff to be improving products every day.

Suppliers and supplies changed a lot and rapidly, so production processes had high variability to deal with – a concept counter to what tradition considers necessary. The score was on outcome, not on conformance to process; and on outcome improvements, not outcome consistency. Key-component suppliers (e.g. Bosch, Panasonic) contract-required to improve their components weekly.

Reference

Dove, R. 2023. Agile Systems Engineering … It's Not Your Father's Oldsmobile. INCOSE Webinar 167. 20 September.

High Points – Automotive Vehicle Product Line (Tesla)

Adaptable Modular Architectures:
- **Engineering process, product, and factory composed of modules with stable interfaces.**
- **Rapid opt-in modular teaming enabled by culture and personnel-interaction handbook rules.**

Iterative Incremental Development:
- **60 part changes a day average in 2021/22 reflected constant improvement iteration pace.**
- **Automated safety certification testing enabled continuous iterative development.**

Attentive Situational Awareness:
- **Ubiquitous AI tools provided awareness of engineering, product, and production improvement trends.**
- **Software called Autobidder rapidly found alternative suppliers to resolve supply issues.**

Attentive Decision Making:
- **DSM tool eliminated human manager decision delays by answering questions directly.**
- **Service options and scheduling driven by operational monitors and digital twin profile.**

Common-Mission Teaming:
- **Opt-in mission-pulled team composition.**
- **Clear and monitored mission focus to minimize/eliminate multi-mission switching costs.**

Shared-Knowledge Management:
- **DSM-AI did long term knowledge curation and cross-company dissemination.**
- **WIP knowledge continuously evolved on personal phones, computers, and factory screens.**

Continual Integration & Test:
- **Each vehicle on the production line was treated as an automated integration & test platform.**
- **Production line cars ran software doing integration regression testing as parts were added.**

Being Agile: Operations Concept
- **Continual innovation mission on product and production performance, efficiency, and cost.**
- **Production processes and tooling were designed to facilitate rapid/frequent change.**

Wrap Up

You don’t need a boxed Agile Engineering Procedure to reap the benefits of agility.
Tesla and SpaceX don’t speak about agility or Agile.

BE-ing agile is different than DO-ing Agile.

However, tailoring a COTS framework (e.g., SAFe), might be organizationally expedient, and more effective if you leverage the roles played by the 8 Aspects.

========== ==========

The INCOSE SE Agility Guide team welcomes
practical-minded, mission-oriented, assistance
as collaborators, writers, reviewers.

contact
rick.dove@incose.net