

Webinar

# Agile SE Processes 206: Agile Systems Engineering ... It's Not Your Father's Oldsmobile

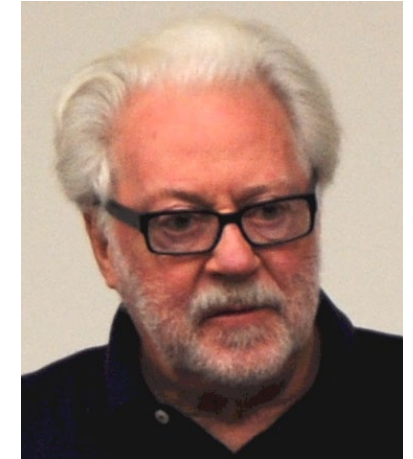
20-Sep-2023

**Rick Dove**

Anthem, AZ, [dove@parshift.com](mailto:dove@parshift.com), 575-770-7101

Independent Operator

Chair: INCOSE WG for Agile Systems & Systems Engineering



Agile 206 webinar slides: [Agile SE ... It's Not Your Father's Oldsmobile](#)  
Agile 205 webinar slides: [Agile SE in the Future of Systems Engineering](#)  
Agile 204 webinar slides: [Agile SE Life Cycle Model](#)  
Agile 203 webinar slides: [Agile SE Agility as a System](#)  
Agile 202 webinar slides: [Agile SE Continuous Integration](#)  
Agile 201 webinar slides: [Agile SE Problem Space Requirements](#)  
Agile 106 webinar slides: [Agile System/Process as Risk Management](#)  
Agile 105 webinar slides: [Agile System/Process Operational Awareness](#)  
Agile 104 webinar slides: [Agile System/Process Engagement Quality](#)  
Agile 103 webinar slides: [Agile System/Process Design Principles](#)  
Agile 102 webinar slides: [Agile System/Process Design Requirements](#)  
Agile 101 webinar slides: [Agile System/Process Architecture Pattern](#)  
(updated asynchronously from time-to-time)

# Agile Systems Engineering ... It's Not Your Father's Oldsmobile\*

The Manifesto for Agile Software Development and related processes like Scrum are well known, and well suited to software engineering.

But software engineering is very different than electronic engineering, mechanical engineering, and systems engineering.

Engineering today needs the agility to anticipate and effectively respond to increasingly dynamic and uncertain environments.

While tactical methods (the how part) for engineering agility necessarily vary among different engineering domains, strategies (the what and why parts) for achieving these common goals are domain independent.

Research has shown us fundamental common strategies that enable agility in any domain; but each domain needs to express these patterns differently to fit the nature of that domain.

This presentation will profile eight strategic aspects  
~~and contrast examples of their tactical manifestations in systems engineering and software engineering.~~  
with application examples of extreme agility as practiced at Tesla and SpaceX.

\*Oldsmobile's 1989 advertising phrase attempting to appeal to a younger generation.

# Systems Engineering Agility

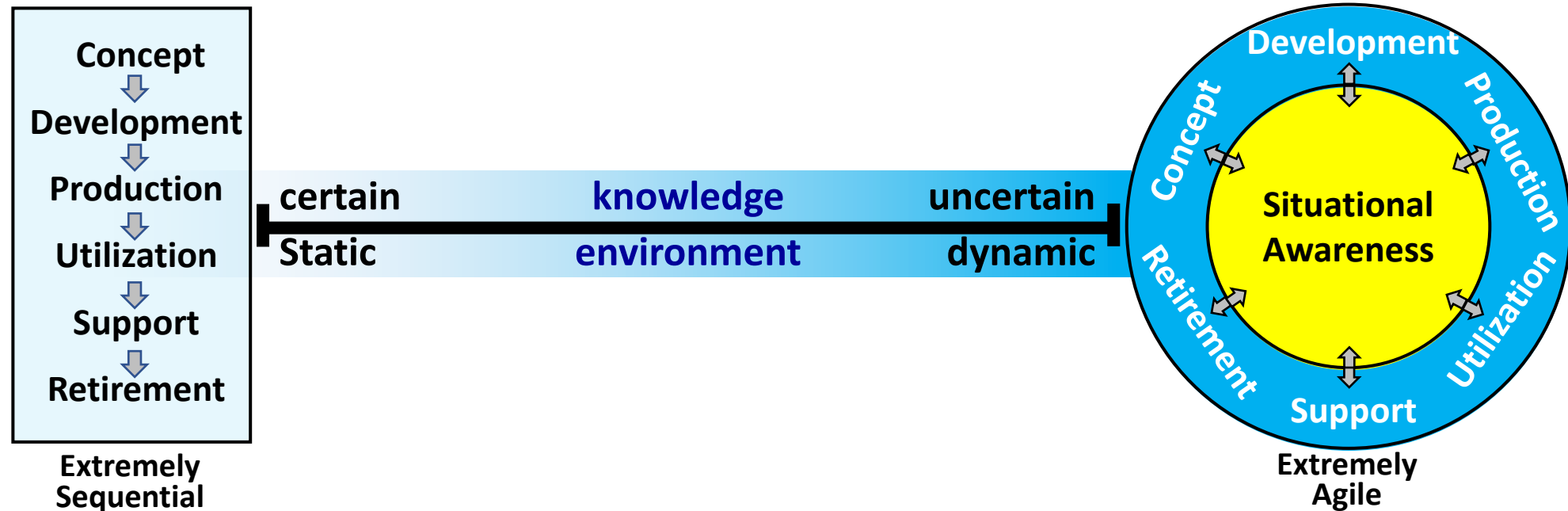
**Agile systems engineering is a strategy-driven method for designing, building, sustaining, and evolving systems when knowledge is uncertain and/or environments are dynamic.**

**Agile systems engineering is *being* agile, not *doing* agile.**

**Agile System Engineering is a what, not a how;  
a strategic intent, not a tactical method.**

**There are many different methods that can be adopted, adapted, or crafted to suite project contexts and organizational cultures;  
but all share the same goals and strategies for being.**

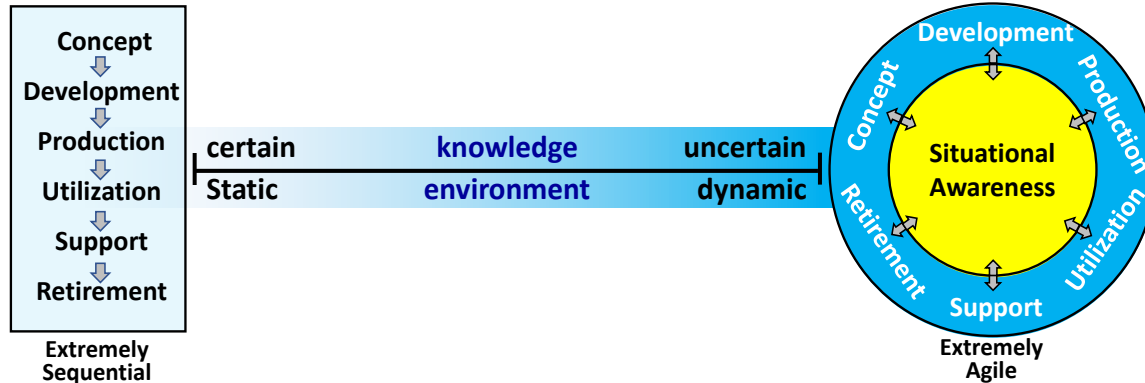
# Life Cycle Spectrum



Agile systems engineering is first and foremost systems engineering as it is known through its ISO/IEC/IEEE standards, the Vee model, the INCOSE Systems Engineering Handbook, and 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, its enablement of continual system evolution, and its intent to satisfy mission rather than plan.

# 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.

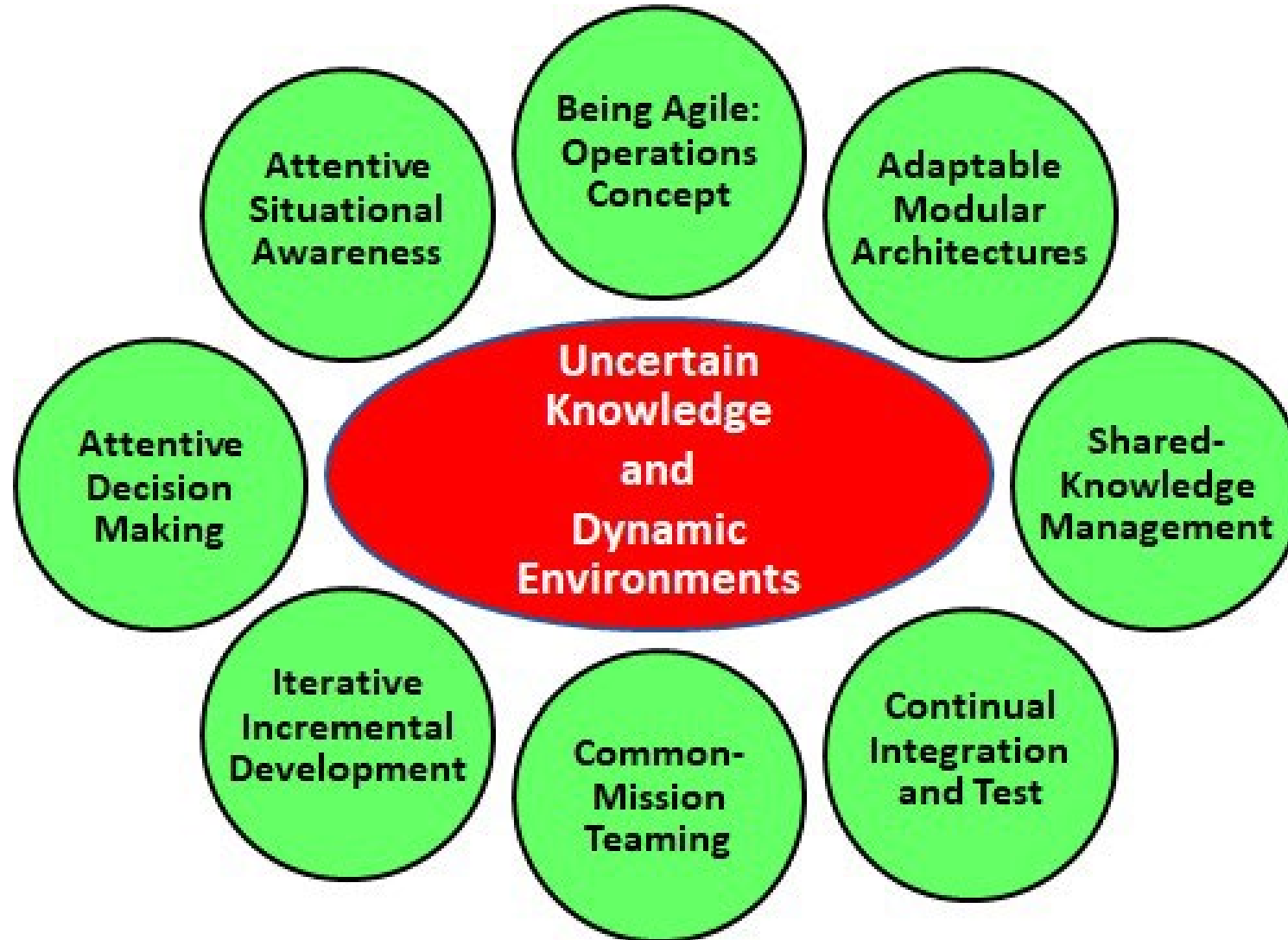


# Value Proposition for Agile <any kind of> Engineering

**Minimize rework**  
**Maximize quality**  
**Drive innovation**

# Eight Strategic Aspects That Enable Agility

[www.parshift.com/s/230715IS23-AgileSE-EightCoreAspects.pdf](http://www.parshift.com/s/230715IS23-AgileSE-EightCoreAspects.pdf)





## Adaptable Modular Architectures

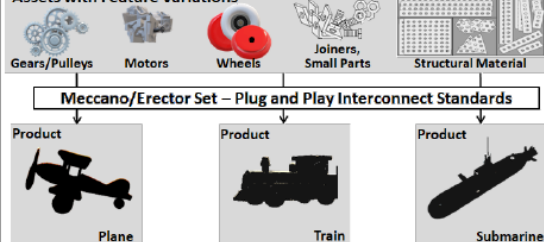
**Needs:** Facilitated product and process experimentation, modification, and evolution.

**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.

Assets with Feature Variations



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

## 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.



SpaWar iteratively evolving unmanned technology integration platform.

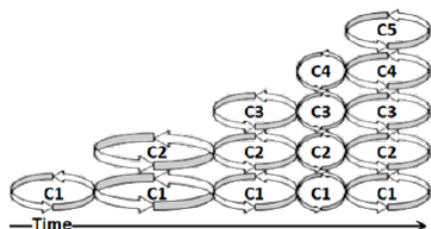
## Iterative Incremental Development

**Needs:** Minimize rework, maximize quality, drive 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)

## 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

## 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.



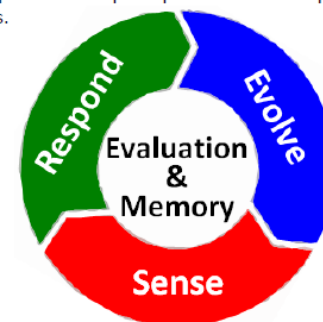
Depicted books represent information containers of any kind; but typically digital

## 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



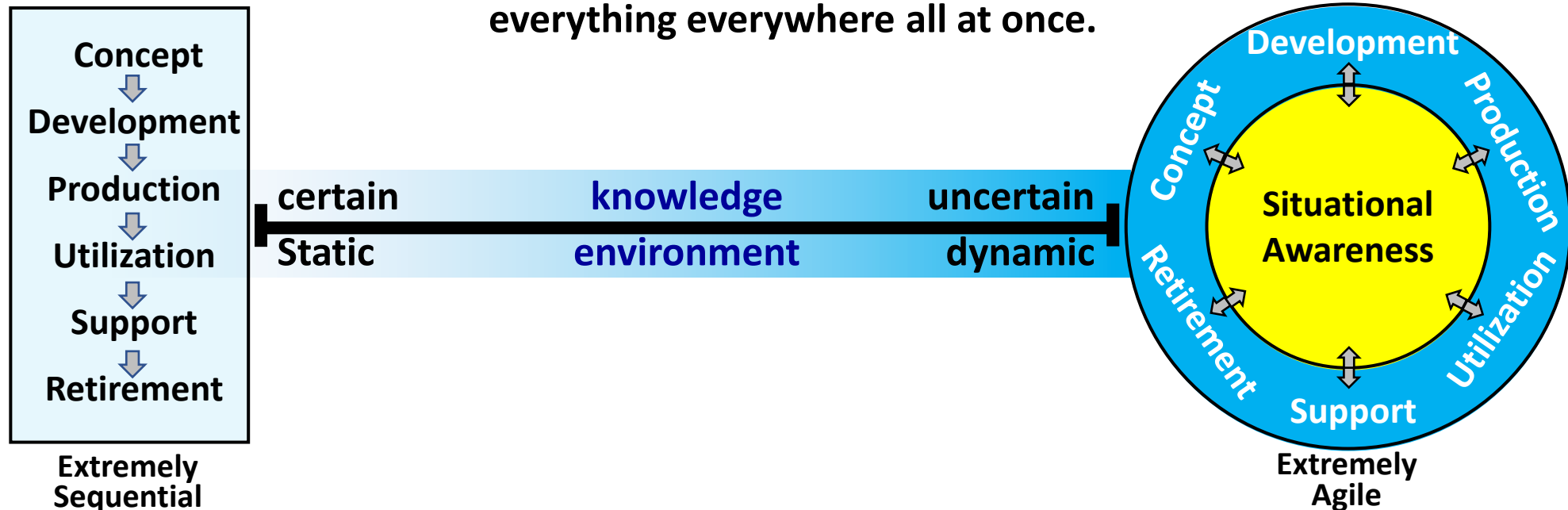
# Extremely Agile

Case examples will be drawn from what is publicly available about Tesla's engineering agility – mainly from video accountings of Joe Justice's engagements at Tesla as an employee and at SpaceX as an observer.

Caveat: drawn from ongoing product evolution engineering, not from develop-from-scratch engineering, i.e., not about cybertruck birth, rather about continuous daily innovation of the S3, X, and Y.

However, it is about  
an omnipresent pervasive cultural infrastructure  
that channels all engineering activity ...

everything everywhere all at once.



# Tesla – From a Distance

Vehicles coming off the production line evolve continuously for better performance and lower cost.

Production methods evolve continuously for higher quality, lower cost, and smaller footprint.

Factory construction methods evolve continuously for speed-to-operation.

Vertical integration is employed to drive constant innovation in key subsystems and avoid supply chain issues.

Key subsystem suppliers are required to improve delivered components weekly.

Production expects, and is designed to accommodate, constant change in all subsystems and parts.

Significant investment in, and usage, of AI/ML.

Flat organization – virtually no manager positions.

High-mission driven: accelerate the advent of sustainable energy (Patents are freely shared and license free).

Expected employee priority: Speed of innovation and dedication to immediate & long term missions.

Object-oriented modular architecture is a first principle and dominant pattern for everything: part design, process design, production design, facility design, teaming design, ...

Tesla & SpaceX – top 2 desires for engineering graduates.



# Adaptable Modular Architectures

**Why/Need:** Facilitated product and process experimentation, modification, and evolution.

**What/Behavior:** Composable and reconfigurable product and process designs from variations of reusable assets.

**How/ConOps:** Module interconnect specs and facilitation (process/tooling) to find, employ, and evolve available reusable assets.

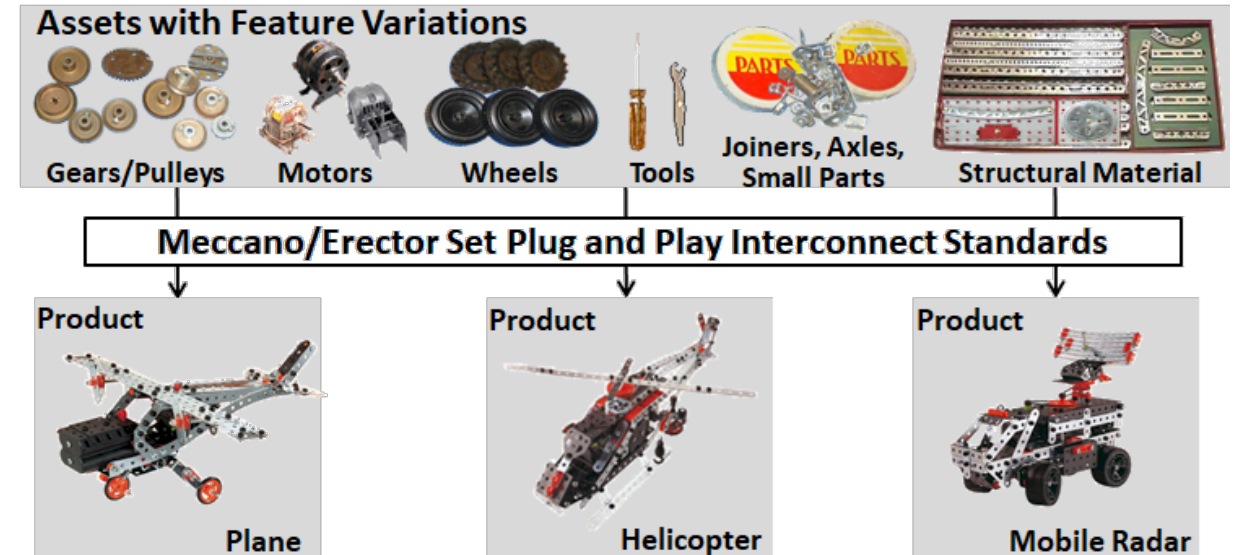
**Heuristic:** Encapsulate modules to preclude cross-module dependencies and enable independent evolution and innovation.

## Theory/Look:

- Uncertainty: Architecture enables affordable experimental learning to reduce uncertainty.
- Dynamics: Architecture facilitates timely composable response to opportunities and threats.

## Practice/Feel at Tesla:

- Tesla and SpaceX use modular architectures with module-interconnect specs for everything: product, process, facility, production, tooling, and people.
- Interconnect specs are allowed to evolve asynchronously with backward compatible adaptors.
- Adaptable modular architectures appear to be a dominant mental pattern for all types of systems at Tesla and SpaceX.



Iconic Agile Architecture Pattern

# Iterative Incremental Development

**Why/Need:** Minimize rework, maximize quality, drive innovation.

**What/Behavior:** Incremental loops of building, evaluating, correcting, improving, and delivering capabilities.

**How/ConOps:** Try something, evaluate it, incorporate it or try something else

**Heuristic:** build a little, test a little, learn a lot.

**Theory/Look:**

- Uncertainty: rapid experimental development, test, and evaluation resolves uncertainties.
- Dynamics: Iterations and increments enable and facilitate changing goals to fit changing needs.

**Practice/Feel at Tesla:**

- At Tesla every car on the production line is represented in the graphic below as a full column, where many capabilities are potentially improved and new capabilities may or may not be added.
- “Tesla makes 60 part changes a day. 60 new parts introduced in production and sold to customers every day.”

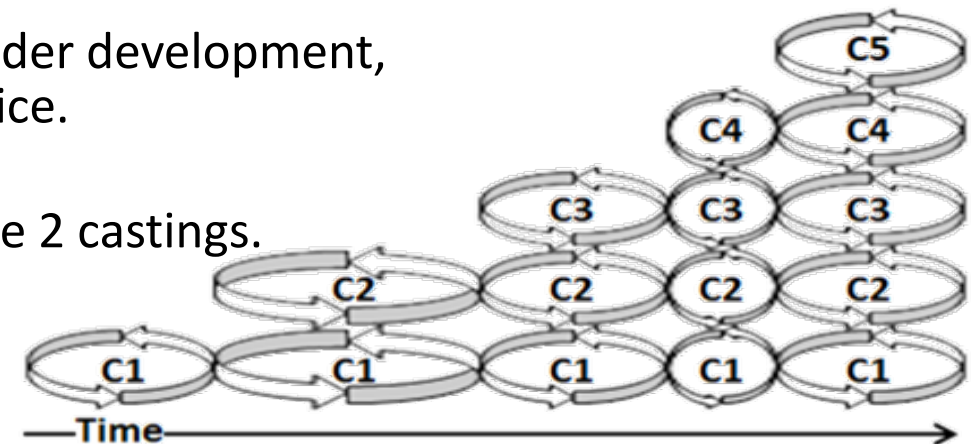
Joe Justice, 7-Jul-2022, Tesla’s Secret Process for Rapid Innovation, [www.youtube.com/watch?v=FE7OUGC4OB8](https://www.youtube.com/watch?v=FE7OUGC4OB8), 14:15

- At Tesla full self driving is an incremental capability currently under development, and expected to be an optional upgrade for existing cars in service.
- Body sections that began as hundreds of welded & bolted parts were iteratively improved (same capability) over time to become 2 castings.

- SpaceX: Raptor engines are built on a 2-day cycle and no two of them are the same as each build introduces innovations.

Joe Justice, 7-Jul-2022, Tesla’s Secret Process for Rapid Innovation, [www.youtube.com/watch?v=FE7OUGC4OB8](https://www.youtube.com/watch?v=FE7OUGC4OB8), 15:50

- Speed of safety certification dictates iteration speed, so every car drives itself through an in-factory certification test and registers that result with NHTSA.



Iterative capability improvements (looping) and incremental capability additions (successive columns)

# Common-Mission Teaming

**Why/Need:** Coherent collective pursuit of a common mission.

**What/Behavior:** Engaged collaboration, cooperation, and teaming among all relevant stakeholders.

**How/ConOps:** Team actively senses and corrects mission divergence.

**Heuristic:** Maintain team stability and mission focus until done to avoid switching costs.

## Theory/Look:

- Uncertainty: Team cognition and awareness greater than individual sum emerges through interactions.
- Dynamics: Mission dominance enables dynamic adjustments of methods.

## Practice/Feel at Tesla:

- Teaming opportunities are posted dynamically on personal phone and on ubiquitous monitor boards to attract immediate opt-in participation, much like *Open Space* conference technique (join us for what's happening at location X).
- 3.5 page employee handbook establishes guardrails and behavior expectations for collaborative opt-in teaming.
- Teams function much like *Mobs* in software development, i.e., everyone works on the same thing at the same time in the same space, with rotating roles of driver and navigators.
- Mission is constant innovation to improve performance (of everything) and reduce number of parts, process steps, lines of code (in all job descriptions).

Joe Justice, 9-Aug-2023, MobAI with Joe Justice, [www.youtube.com/watch?v=Fh\\_FDGiTvvg](https://www.youtube.com/watch?v=Fh_FDGiTvvg)





# Shared-Knowledge Management

**Why/Need:** Accelerated mutual learning and single source of truth for internal and external stakeholders.

**What/behavior:** Facilitated communication, collaboration, and knowledge curation.

**How/ConOps:** New knowledge processes – evaluate/codify/add. Old knowledge processes – find/access/prune.

**Heuristic:** Automate where possible for relevant acquisition, evolution, and distribution.

## Theory/Look:

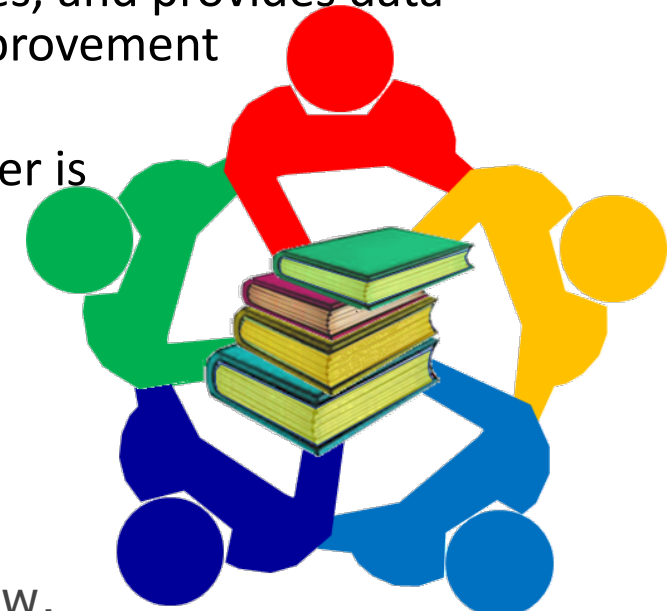
- Uncertainty: Access to relevant knowledge reduces uncertainty.
- Dynamics: Accessible reusable knowledge provides more options for quick response.

## Practice/Feel at Tesla:

- Digital Self Management (DSM) is AI machine-learning software that learns, evolves, and provides data for personal decision making, for design improvement assistance, and product improvement evaluation. Evolving DSM data sets are shared among all factories.
- “There are no bosses, your manager is data.” Any approval that waits for a manager is automated by software. This software has largely replaced management, allowing nearly 100% of staff to be an engineer, directly improving products every day.

Joe Justice, Digital self management, 9-Mar-2023, [www.youtube.com/watch?v=gypG12NUtgo](https://www.youtube.com/watch?v=gypG12NUtgo)

- 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: No product backlog or Kanban board, there is only a doing column on ubiquitous monitor boards and on all phones. Most projects are extremely short so the focus is on what's happening now.



Information containers of any kind; but typically digital



# Attentive Situational Awareness

**Why/Need:** Timely knowledge of emergent risks and opportunities.

**What/Behavior:** Active monitoring and evaluation of internal & external operational-environment factors.

**How/ConOps:** Data-sensing instrumentation, data-driven monitoring and decision triggers, and effectiveness-driven improvement.

**Heuristic:** Socialize and automate awareness to focus attention on doing things right and doing right things.

**Theory/Look:**

- Uncertainty: Data-driven feedback on operations and operational results reduces situational uncertainty.
- Dynamics: Attentive situation awareness is the enabling capability for sensing and responding to dynamics.

**Practice/Feel at Tesla:**

- Ubiquitous data-driven DSM-AI/ML for continuously-improved situational awareness at every factory station.
- DSM-AI, for instance, measures paint quality on a dozen dimensions. Its data set is very high res images of every car's painted surfaces. If an improvement is tried it can tell immediately if it is better.

Joe Justice, 14-Feb-2023, Ways of Working at Tesla, [www.youtube.com/watch?v=IGw3Z9Kk5lw](https://www.youtube.com/watch?v=IGw3Z9Kk5lw) 44:50

- Mob engineering motivated to innovate has many eyes, ears, and sensitivities monitoring global information for opportunity and threat.
- Car usage is monitored: how it is being used by people, how it is used as a machine, how suitable it is for sustained service.
- Autobidder (software) does mass polls in seconds to find suppliers with prices, capabilities, track records – solved chip shortage issues.

MunWai Consulting, 2023, Agile @ Tesla Audio Episode 10: "Agile", <https://munwaic.com/podcasts/agile-tesla/> 20:00



Alert in-the-moment constant attention

# Attentive Decision Making

**Why/Need:** Timely corrective and improvement actions.

**What/Behavior:** Systemic linkage of situational awareness to decisive action.

**How/ConOps:** Awareness triggered decision needs, decision debt monitoring, effectiveness-driven improvement.

**Heuristic:** Enable decision making at the point of most knowledge: local awareness or remote expertise.

**Theory/Look:**

- Uncertainty: A decision is a learning experience that reduces uncertainty whether right or wrong.
- Dynamics: Attentive decisions are timely responses to mitigate environmental dynamics.

**Practice/Feel at Tesla:**

- “Replacing human decision points with apps is the digital backbone of a modern company and fundamentally determines the speed of product development and response. Digital Self Management means saying ‘why would we ever ask a human to decide this?!’ And being honest.”

Joe Justice, Digital self management, 9-Mar-2023, [www.youtube.com/watch?v=gypG12NUtgo](https://www.youtube.com/watch?v=gypG12NUtgo) 9:18

- DSM-AI/ML is attentive awareness that provides data for immediate deployment decisions. Data authorizes the decision, no other authority is required.
- “You don’t have to ask Elon ‘Should I put the new heat pump in the car?’ If your automated tests passed that is your boss. You deploy immediately.”

Joe Justice, Digital self management, 9-Mar-2023, [www.youtube.com/watch?v=gypG12NUtgo](https://www.youtube.com/watch?v=gypG12NUtgo) 20:11

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



John Boyd's OODA loop

# Continual Integration & Test

**Why/Need:** Early revelation of system integration issues.

**What/Behavior:** Integrated test and demonstration of work-in-process.

**How/ConOps:** Automate testing and evaluation to drive iterative improvement

**Heuristic:** Incrementally and asynchronously integrate work-in-process and automated regression testing.

**Theory/Look:**

- Uncertainty: Reduces uncertainty of integration issues.
- Dynamics: Reveals conflicts among components and conflicts between system and operational situation.

**Practice/Feel at Tesla:**

- Every car has potentially new components so each component and car on the production line is a work-in-process, automated-test, integration platform.
- “Every single car and rocket puts itself through automated testing of every non-destructive compliance and certification test. This is called ‘Factory Mode’ and it is where autopilot came from. In Factory Mode cars run their own regression tests as they are being assembled, and display that information.”

Joe Justice, 9-Mar-2023, [www.youtube.com/watch?v=qypG12NUtgo](https://www.youtube.com/watch?v=qypG12NUtgo) 11:20

- “Modules of the car are individually tested and regression tested in Factory Mode, then tested again as soon as they are connected on the car.

Joe Justice, 9-Mar-2023, [www.youtube.com/watch?v=qypG12NUtgo](https://www.youtube.com/watch?v=qypG12NUtgo) 11:54

- Factory mode autopilot: the car drives itself through certification testing.



Iteratively evolving  
technology integration platform. 17

# Being Agile: Operations Concept

**Why/Need:** Attentive operational response to evolving knowledge and dynamic environments.

**What/Behavior:** Sensing, responding, evolving.

**How/ConOps:** Design and use a sense-respond-evolve SE ConOps process – as a system.

**Heuristic:** Keep your eyes on the road and your hands upon the wheel – for in-the-moment engagement.

**Theory/Look:**

- Uncertainty: Experimentally probe and prototype to test and resolve uncertain knowledge
- Dynamics: Sense-respond-evolve compatibly in concert with environmental changes.

**Practice/Feel at Tesla:**

- “Pace of innovation is the only thing that matters – not cost per unit, not management efficiency, no other metric is above pace of innovation.”

Joe Justice, 6-Apr-2022, Radical Unbossing at Tesla, Joe Justice, [www.youtube.com/watch?v=ozdBx1SG-vo](https://www.youtube.com/watch?v=ozdBx1SG-vo)

- The machine that makes the machine is where SE attention is principally focused.
- “The goal is for 100% of staff to be engineers in hardware and software. Any work that requires taking someone away from engineering is automated as much as practical.” Joe Justice, 9-Mar-2023, Digital self management, [www.youtube.com/watch?v=qypG12NUtgo](https://www.youtube.com/watch?v=qypG12NUtgo)

- Suppliers and supplies change a lot and rapidly, so processes have high variability to deal with – a concept counter to what tradition considers necessary. The score is on outcome, not on conformance to process, and on outcome improvements, not outcome consistency.

Joe Justice, 14-Feb-2023, Ways of Working at Tesla, [www.youtube.com/watch?v=IGw3Z9Kk5lw](https://www.youtube.com/watch?v=IGw3Z9Kk5lw)

- Key-component suppliers (e.g. Bosch, Panasonic, 3M) are required to improve their components weekly.



Three principles that operationalize agility

# Tesla – Up Close and Personal

## Adaptable Modular Architectures:

- Product, engineering process, production, and factory composed of encapsulated modules with stable interfaces.
- Process of Open Space mob-teaming supported by Digital Self Management awareness-decision tools.

## Iterative Incremental Development:

- Continual independent parallel evolution of encapsulated product, production, and facility modules.

## Attentive Situational Awareness:

- DSM-AI/ML tool has task-specific data sets to continually & immediately evaluate outcome results.
- DSM-AI/ML tool has answers for management questions – eliminating need for human manager.
- Autobidder (software) does mass polls in seconds to find suppliers with prices, capabilities, and track record.

## Attentive Decision Making:

- DSM-AI/ML tool has task-specific data sets to continually & immediately evaluate outcome results.
- DSM-AI/ML tool has answers for management questions – eliminating human manager functions and needs
- Customer service option and scheduling driven by on-board 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/ML curation and dissemination.

## Continual Integration & Test:

- Every car has potentially new components and each component and car is an automated integration & test platform.

## Being Agile: OpsCon

- Continual mission-driven innovation on performance, efficiency, and cost of product and production.
- Operational culture has no need or use for agility procedures and agility management processes.

Purpose of case story development was to see if Tesla uses strategic aspects beyond the eight. None were found.

A revelation, however, was that data driven tooling (DSM-AI/ML) is a key Tesla agility enabler functioning as an automated implementation and systemic connection of Attentive Situational Awareness to Attentive Decision Making.



# Take Aways

**that don't require radical reorganization or impossible cultural change – appropriate where benefits can be gained**

**Mob engineering – for critical engineering issues/opportunities, everybody necessary and useful in the same room at the same time working together until the mission is completed. Not as a transformed way of life, but it might grow in usage with successes and experience.**

**Management automation – not wholesale, but incremental identification of what waits for approvals and decisions that could be automated, with priority on low hanging fruit and high-pain delays.**

**Automated regression testing of incrementally-integrated work in process.**

**Incentivize cost/time/quality improvements on outcomes and methods – even if that's only in your personal cadre of influence (coworkers, direct/indirect reports – recognition, respect, reputation)**

**Engineering designers and production designers work together simultaneously – iteratively designing (or improving the design) of a subsystem/component/part and how to produce it.**

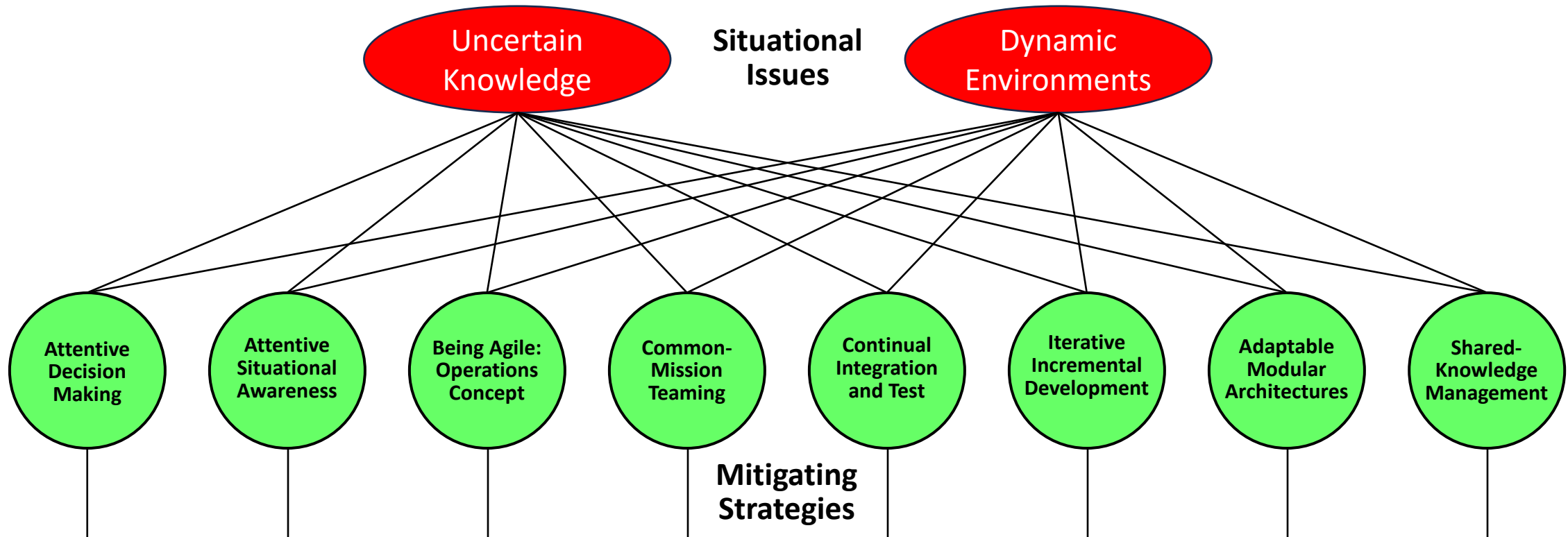
**Production line designed for, and comfortable with, part/component/subsystem variation**

**Continuous improvement required of key component/subsystem suppliers.**

...



# Wrap Up



## Employment Considerations

- Leveraging aspect synergies
- Avoiding aspect conflicts
- Organizational readiness
- Continuous iterative improvement

# Common Strategies, Different Tactics

Provocative examples: many contrasts can be listed in each row depending upon project context.

Strategic Aspect	Software Engineering	Systems Engineering
Adaptable Modular Architecture	<ul style="list-style-type: none"> <li>• Standard interface</li> <li>• COTS object-oriented integrated dev environment</li> </ul>	<ul style="list-style-type: none"> <li>• Proprietary interface</li> <li>• MOSA design effort</li> </ul>
Iterative Incremental Development	<ul style="list-style-type: none"> <li>• Tight</li> <li>• Test time stability</li> </ul>	<ul style="list-style-type: none"> <li>• Loose</li> <li>• Test facility conflicts</li> </ul>
Attentive Situational Awareness	<ul style="list-style-type: none"> <li>• Introspective</li> <li>• Frequent user feedback</li> </ul>	<ul style="list-style-type: none"> <li>• Extrospective</li> <li>• Sparse user feedback</li> </ul>
Attentive Decision Making	<ul style="list-style-type: none"> <li>• Simple</li> <li>• Few people</li> </ul>	<ul style="list-style-type: none"> <li>• Complicated</li> <li>• Many people</li> </ul>
Common-Mission Teaming	<ul style="list-style-type: none"> <li>• Homogeneous</li> <li>• Shared language</li> </ul>	<ul style="list-style-type: none"> <li>• Heterogeneous</li> <li>• Different languages</li> </ul>
Shared Knowledge Management	<ul style="list-style-type: none"> <li>• Code libraries</li> <li>• Integrated tools</li> </ul>	<ul style="list-style-type: none"> <li>• PLM</li> <li>• Federated tools</li> </ul>
Continual Integration and Test	<ul style="list-style-type: none"> <li>• Common platforms</li> <li>• Synchronous</li> </ul>	<ul style="list-style-type: none"> <li>• Proprietary platforms</li> <li>• Asynchronous</li> </ul>
Being Agile: Operations Concept	<ul style="list-style-type: none"> <li>• Do Agile</li> <li>• Many COTS Options</li> </ul>	<ul style="list-style-type: none"> <li>• Be agile</li> <li>• Home grown and tailored</li> </ul>

## Adaptable Modular Architectures

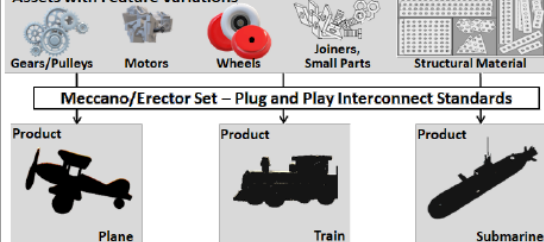
**Needs:** Facilitated product and process experimentation, modification, and evolution.

**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.

Assets with Feature Variations



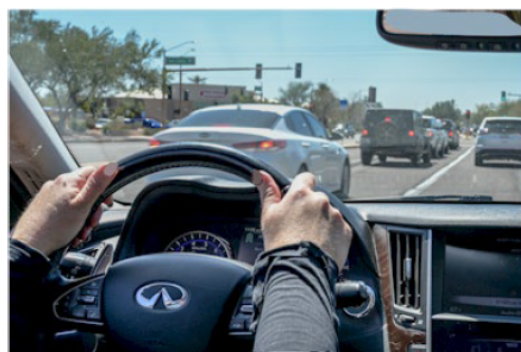
Notional Agile Architecture Pattern

## 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

## 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.



SpaWar iteratively evolving unmanned technology integration platform.

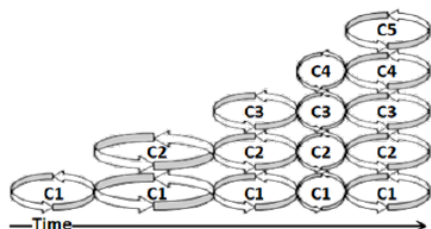
## Iterative Incremental Development

**Needs:** Minimize rework, maximize quality, drive 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)

## 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

## 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.



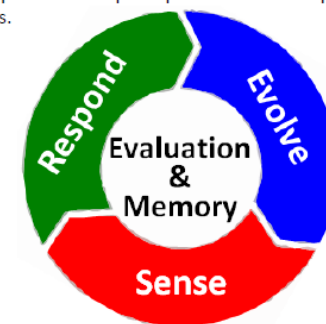
Depicted books represent information containers of any kind; but typically digital

## 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

# References

**Agile World - Joe Justice - Tesla's Secret Process for Rapid Innovation, 2022 July 7**

<https://www.youtube.com/watch?v=FE7OUGC4OB8>

**Agile at Tesla, 11 audio episodes covering how the company is organized and operates, MunWai Consulting, 2023**

<https://munwaic.com/podcasts/agile-tesla/>

**Joe Justice, MobAI with Joe Justice, 2023 August 9**

[www.youtube.com/watch?v=Fh\\_FDGiTvcg](https://www.youtube.com/watch?v=Fh_FDGiTvcg)

**Digital self management - Joe Justice keynote at the LPPDE Virtual Summit, 2023 March 9**

[www.youtube.com/watch?v=qypG12NUtgo](https://www.youtube.com/watch?v=qypG12NUtgo)

**Ways of Working at Tesla Talk by Joe Justice with Q&A, 2023 February 14**

[www.youtube.com/watch?v=IGw3Z9Kk5lw](https://www.youtube.com/watch?v=IGw3Z9Kk5lw)

**Radical Unbossing at Tesla, Joe Justice, 2022 April 6**

[www.youtube.com/watch?v=ozdBx1SG-vo](https://www.youtube.com/watch?v=ozdBx1SG-vo)

**Agile Systems Engineering – 8 Core Aspects, INCOSE Symposium, July 2023**

[www.parshift.com/s/230715IS23-AgileSE-EightCoreAspects.pdf](https://www.parshift.com/s/230715IS23-AgileSE-EightCoreAspects.pdf)

**Systems Engineering Agility in a Nutshell, INCOSE INSIGHT, June 2023**

[www.parshift.com/s/230601Insight-SystemsEngineeringAgilityInANutshell.pdf](https://www.parshift.com/s/230601Insight-SystemsEngineeringAgilityInANutshell.pdf)

## Full Series

Agile 206 webinar slides: [Agile SE ... It's Not Your Father's Oldsmobile](#)  
Agile 205 webinar slides: [Agile SE in the Future of Systems Engineering](#)  
Agile 204 webinar slides: [Agile SE Life Cycle Model](#)  
Agile 203 webinar slides: [Agile SE Agility as a System](#)  
Agile 202 webinar slides: [Agile SE Continuous Integration](#)  
Agile 201 webinar slides: [Agile SE Problem Space Requirements](#)  
Agile 106 webinar slides: [Agile System/Process Risk Management & Mitigation](#)  
Agile 105 webinar slides: [Agile System/Process Operational Awareness](#)  
Agile 104 webinar slides: [Agile System/Process Engagement Quality](#)  
Agile 103 webinar slides: [Agile System/Process Design Principles](#)  
Agile 102 webinar slides: [Agile System/Process Design Requirements](#)  
Agile 101 webinar slides: [Agile System/Process Architecture Pattern](#)  
(updated asynchronously from time-to-time)

Original webinars with recordings at:  
[www.incose.org/products-and-publications/webinars](http://www.incose.org/products-and-publications/webinars)

Webinar ID: Webinar 167 Dove 20 September 2023 Agile SE Processes 206  
Webinar ID: Webinar 161 Dove 21 September 2022 Agile SE Processes 205  
Webinar ID: Webinar 153 Dove 15 September 2021 Agile SE Processes 204  
Webinar ID: Webinar 143 Dove 16 September 2020 Agile SE Processes 203  
Webinar ID: Webinar 131 Dove 18 September 2019 Agile SE Processes 202  
Webinar ID: Webinar 116 Dove 19 September 2018 Agile SE Processes 201  
Webinar ID: Webinar 104 Dove 20 September 2017 Agile Systems & Processes 106  
Webinar ID: Webinar 092 Dove 28 September 2016 Agile Systems & Processes 105  
Webinar ID: Webinar 082 Dove 16 September 2015 Agile Systems & Processes 104  
Webinar ID: Webinar 067 Dove 17 September 2014 Agile Systems & Processes 103  
Webinar ID: Webinar 056 Dove 18 September 2013 Agile Systems & Processes 102  
Webinar ID: Webinar 045 Dove 19 September 2012 Agile Systems & Processes 101