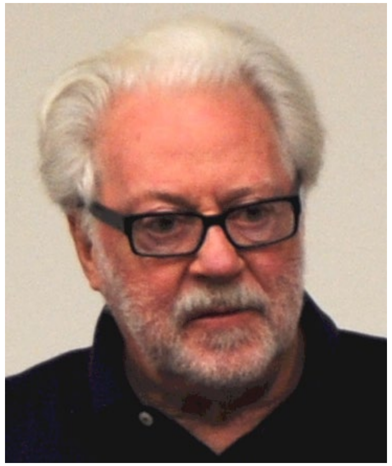
Webinar

Agile SE Strategies 301:
Agile SE Strategies – Purpose and Values

2-Oct-2024

Rick Dove
Chair: INCOSE WG for Agile Systems & Systems Engineering



Agile 301 webinar slides: Agile Strategies Purpose and Values
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)

Abstract

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

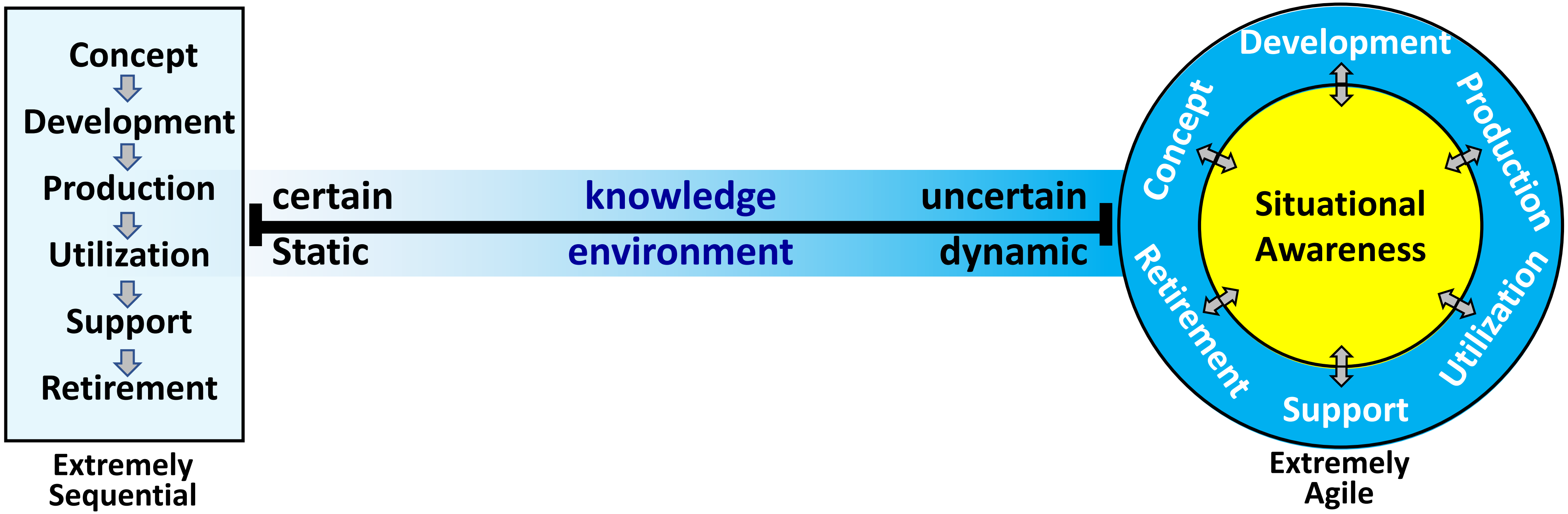
Eight operational strategies for achieving this purpose are identified in the Systems Engineering Agility Primer.

Each strategy delivers agility values by minimizing rework, maximizing quality, and facilitating innovation.

How these strategies are able to deliver these values and contribute to the purpose in each of the eight strategies is explored in this webinar and underscored with in-practice examples.

Intended purpose and values provide a means for assessing and improving effectiveness of strategies.

SE Life Cycle Spectrum



Agile System Engineering is a what, not a how.
Many different methods can be applied, but all share the same goals and strategies.

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.

Value Proposition for Agile <any kind of> Engineering

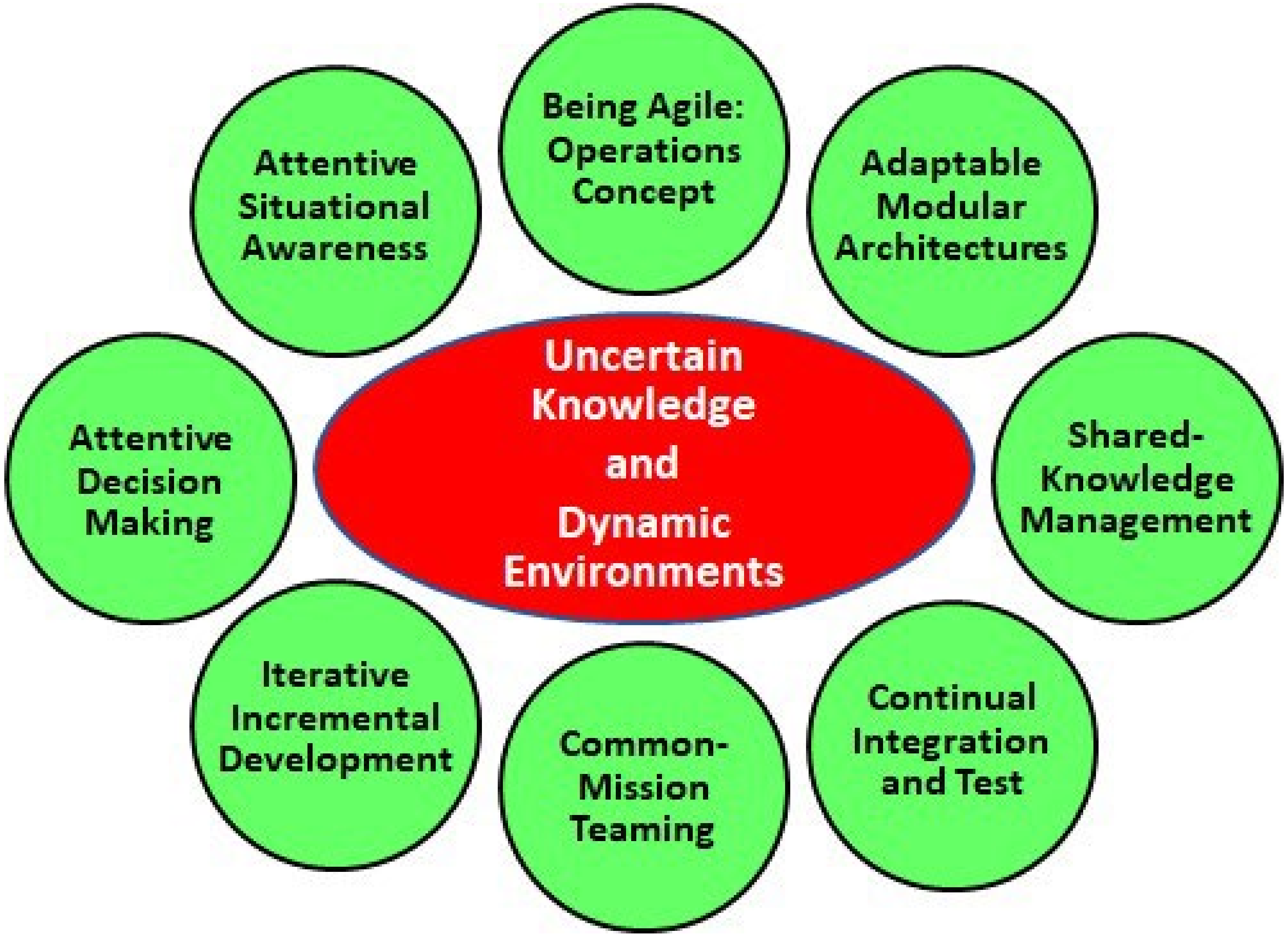
Minimize rework

Maximize quality

Facilitate innovation

Eight Strategic Aspects That Enable Agility

www.parshift.com/s/230715IS23-AgileSE-EightCoreAspects.pdf



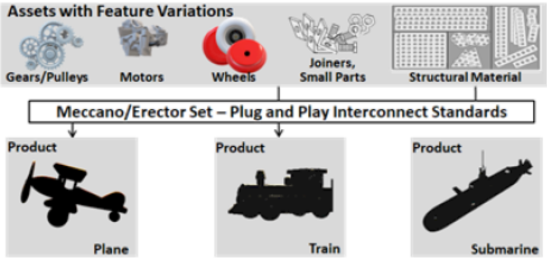
Product Line Architectures

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

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

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

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



Notional Agile Architecture Pattern

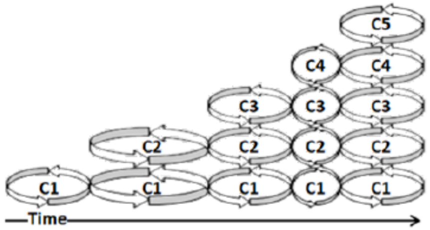
Iterative Incremental Development

Needs: Minimize unexpected rework and maximize quality.

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

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

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



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

Common-Mission Teaming

Needs: Coherent collective pursuit of a common mission.

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

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

Tightly integrated coherent operation

Continual Integration & Test

Needs: Early revelation of system integration issues.

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

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

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.

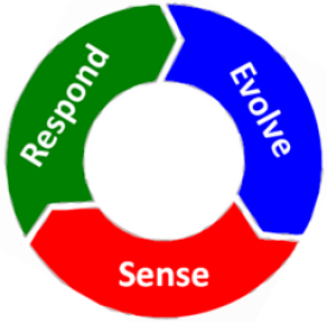
Information containers of any kind, available to all, and typically digital

Being Agile: Operations Concept

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

Behaviors: Sensing, responding, evolving.

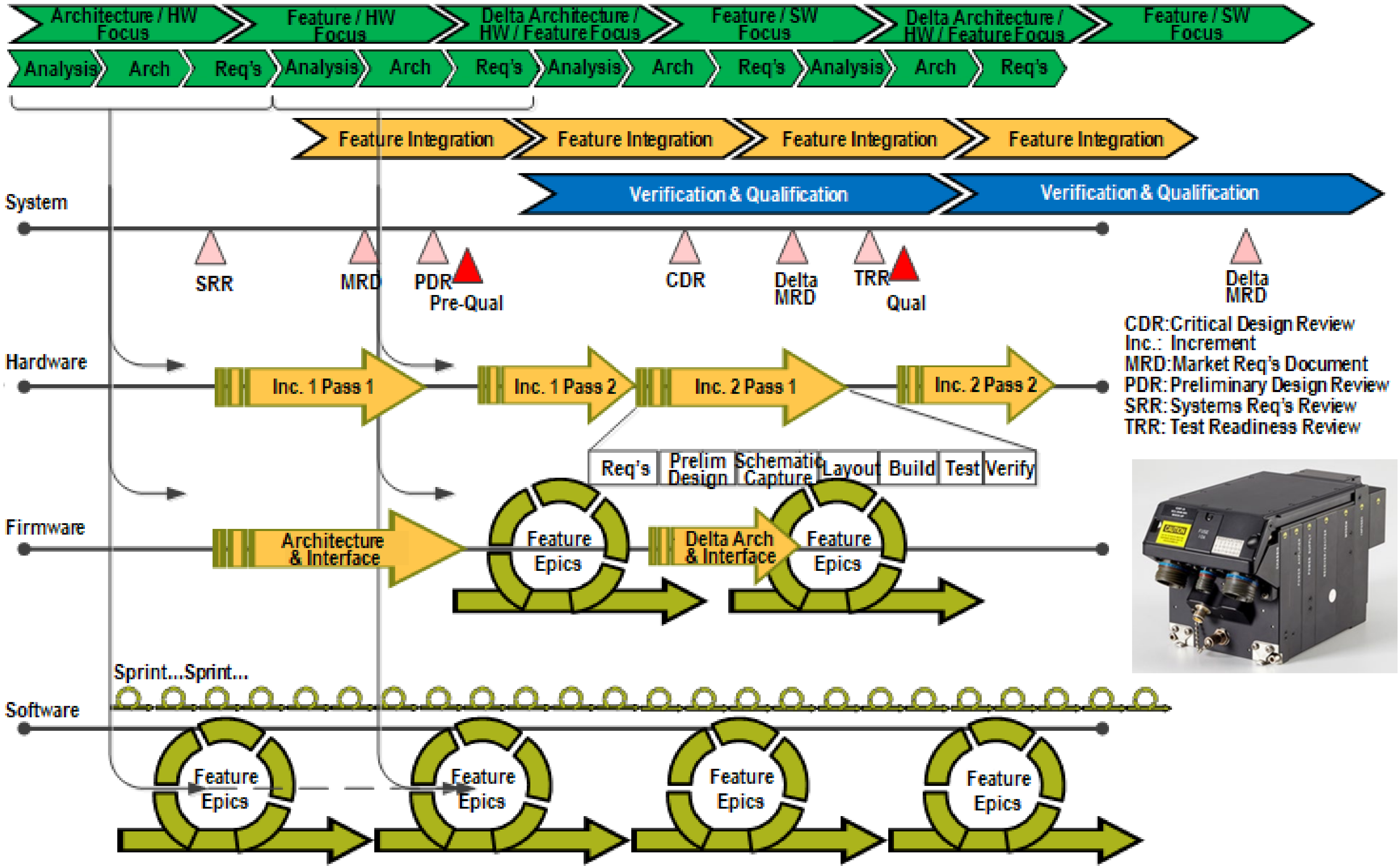
Discussion: Agile systems engineering is not about doing Agile, it is about being agile. Being agile is a behavior, not a procedure – a behavior sensitive to threats and opportunities in the operational environment, decisive when faced with threat or opportunity, and driven to improve these capabilities. 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

Aircraft Radios – Collins

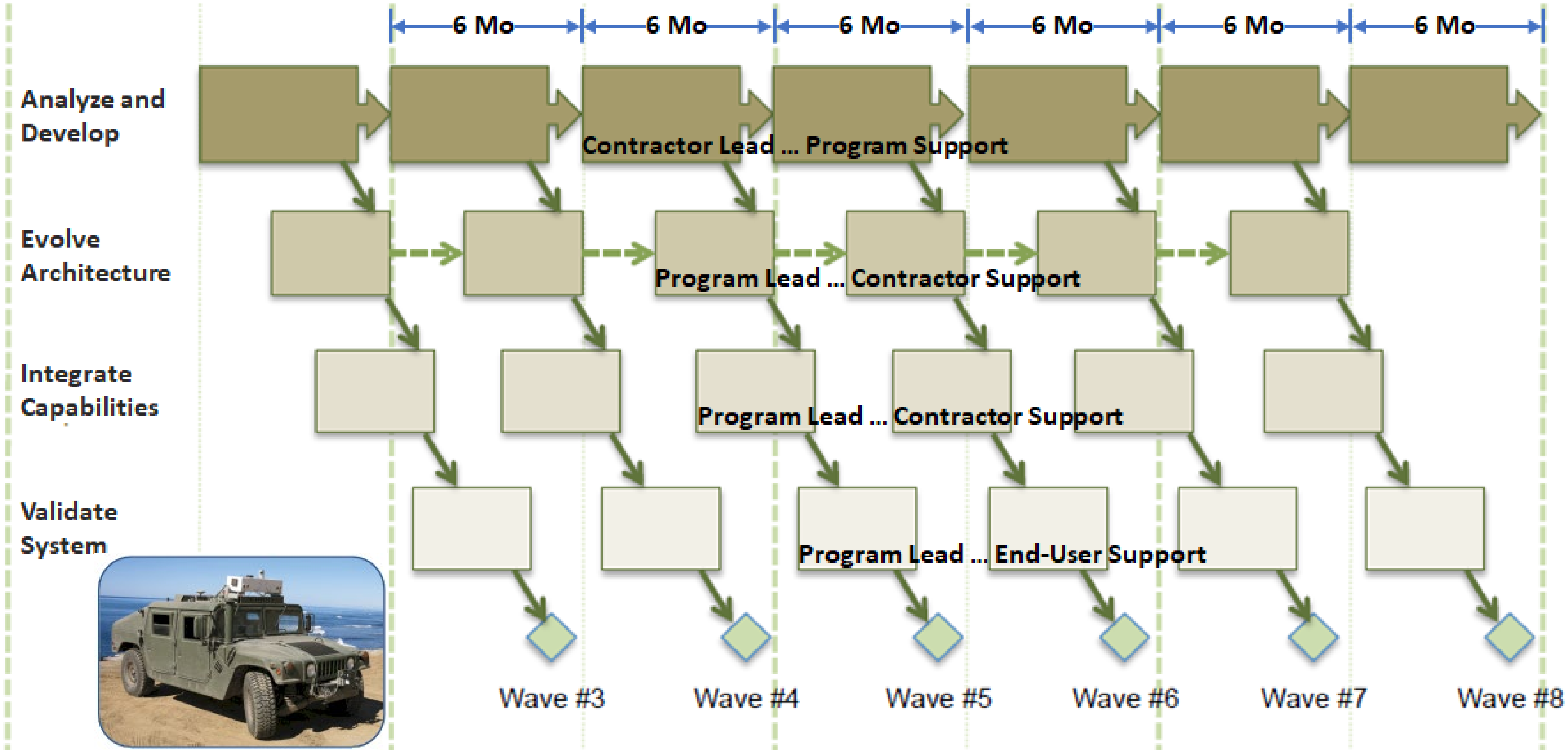
Tailored process influenced by Scrum and SAFe



CDR: Critical Design Review
Inc.: Increment
MRD: Market Req's Document
PDR: Preliminary Design Review
SRR: Systems Req's Review
TRR: Test Readiness Review

Self-Driving Off-Road Technology – USNavy

Wave Process: Modular Workstreams – Overlapping Increments

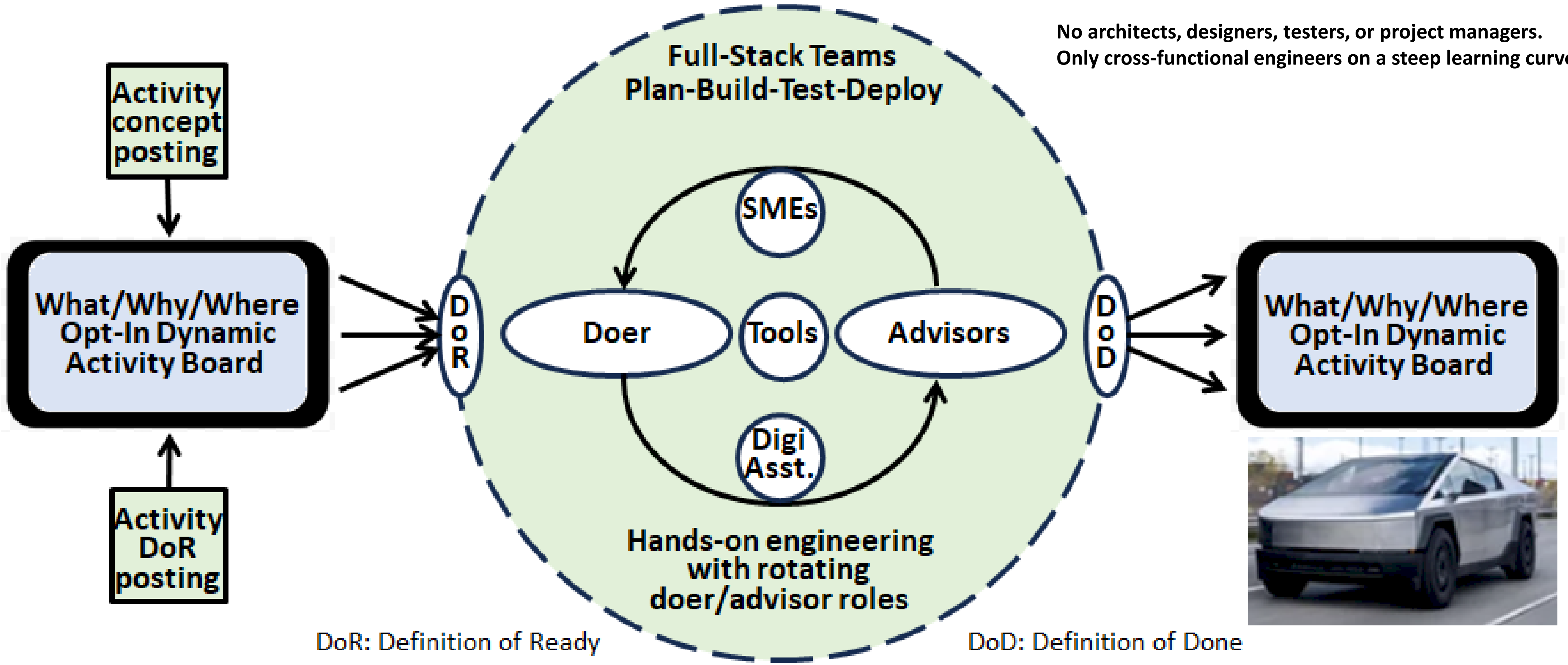


Electric Vehicles – Tesla

Group Engineering: Modularity Enables Parallel Independent Workstreams

Joe Justice. 2023. Tesla and SpaceX speed with MobAI. www.youtube.com/watch?v=N-JYhKZV4-A

No architects, designers, testers, or project managers.
Only cross-functional engineers on a steep learning curve.



Adaptable Modular Architectures

Why (Need): Facilitated <u>product and process</u> experimentation, modification, and evolution.

What (Behavior): Composable and reconfigurable product and process designs from variations of reusable assets.

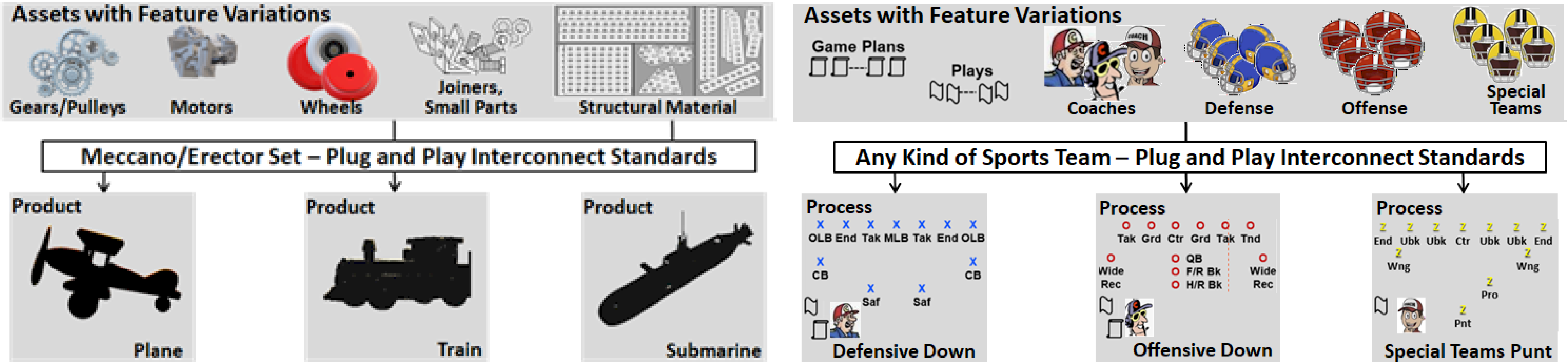
Purpose:

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

Values:

- Minimize rework – Reusable modules have proven operational history.
- Maximize quality – Improved modules can be swapped in transparently.
- Facilitate innovation – Experimental modules can be tried and replaced easily if they fall short.

Social Considerations: Coherent easy methods for finding and employing relevant reusable assets.



⬆ Product – Notional Agile Architecture Pattern – **Process ⬆**

Adaptable Modular Architectures in Practice

Collins:

- 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 typically involved in three or so projects at any one time, easily switching among projects with common process architecture to accommodate evolving priorities and resource availability.

USNavy:

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

Tesla:

- Modular architectures with stable interconnect specs were a dominant mental pattern for everything: product, process, facility, production, tooling, and people – evolvable with backward compatible adaptors.
- Rapid opt-in modular teaming was enabled by a strong core culture and a 3.5 page handbook focused solely on expected behavior and mission-based interaction.

Iterative Incremental Development

Why (Need): Minimize rework, maximize quality, facilitate innovation.

What (Behavior): Incremental loops of building, evaluating, correcting, improving, and delivering capabilities.

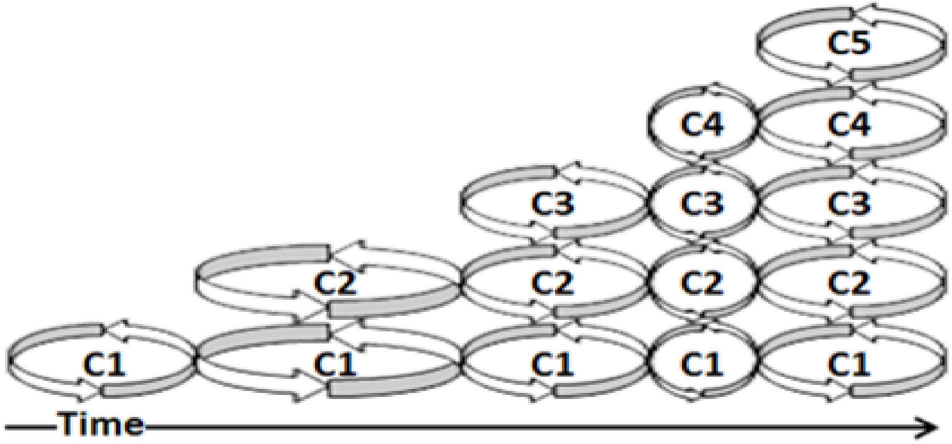
Purpose:

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

Values:

- Minimize rework – Discovers need for rework early and affordably.
- Maximize quality – Iterations converge on better solutions.
- Facilitate innovation – Iterations and increments enable affordable innovation experiments.

Social Considerations: Celebration and reward for improvement and innovation.



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

Iterative Incremental Development in Practice

Collins:

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

USNavy:

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

Tesla:

- 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

Why (Need): Timely knowledge of emergent risks and opportunities.

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

Purpose:

- 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.

Values:

- Minimize rework – Reveals need for rework early.
- Maximize quality – Reveal new value opportunities.
- Facilitate innovation – Reveal innovation opportunities.

Social Considerations: Readily available data (transparency) makes trustworthy environment.

Alert in-the-moment constant attention

Attentive Situational Awareness in Practice

Collins:

- External awareness was the engine that drove a continually evolving Market Requirements Document (MRD) – a closely held roadmap of product line evolution plans.
- Internal awareness monitored workforce skills and competency, contracts with potential to fund MRD items, and urgencies that might prompt IR&D funding.

USNavy:

- Full-day user table-top exercises at each increment included project engineers, contract developers, and marines from different units to verify and reveal a diversity of needs.
- Engagement by internal and external personnel was actively monitored for sufficiency by leadership in frequency and quality of communications, wiki-contributions, and meeting contributions.

Tesla:

- External awareness included Autobidder (software) that could do mass polls in seconds to find suppliers with prices, capabilities, and track records to solve shortage issues (2021 chip shortage did not cause a problem).
- Internal awareness included ubiquitous Digital Self Management (DSM) AI tools providing continuous situational awareness of engineering, product, and production improvement trends at every factory station.

Attentive Decision Making

Why (Need): Timely corrective and improvement actions.

What (Behavior): Systemic linkage of situational awareness to decisive action.

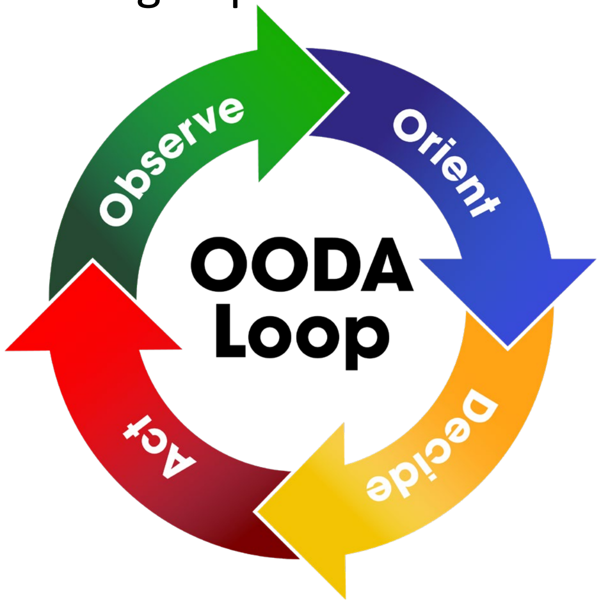
Purpose:

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

Values:

Minimize rework – Curtails work in wrong direction as soon as possible.
- Maximize quality – Adjusts requirements for new values as soon as possible.
- Facilitate innovation – Seizes the opportunity – carpe diem.

Social Considerations: Empower decision-making at point of most knowledge.



Responsible attention may
take time, but never pauses

Attentive Decision Making in Practice

Collins:

- There was an opportunity-driven tight coupling between the MRD team's situational awareness and the Systems Engineering team's requirements development.
- Facility was reconfigured to bring all technical teams together in one place to facilitates fast decision making when an event required a multi-party response agreement.

USNavy:

- The culture shared by internal personnel and subcontractors did not tolerate inaction, expecting and demanding timely attentive closure to open issues.
- Clarity of vision and objective provided common coherent criteria that drove rapid closure among multiple perspectives.

Tesla:

- Digital Self Management (DSM) software replaced most human decision points with apps that render a decision immediately; eliminating wait times that delay personal action.
- Customer service options and scheduling decisions are driven by on-board operational monitors and digital twin profiles – no human in the loop.

Common-Mission Teaming

Why (Need): Coherent collective pursuit of a common mission.

What (Behavior): Engaged collaboration, cooperation, and teaming among all relevant stakeholders.

Purpose:

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

Values:

- Minimize rework – Avoids misdirected correction-needed work effort.
- Maximize quality – Improves outcomes within constrained schedule.
- Facilitate innovation – Considers more options within aligned thinking.

Social Considerations: Opt-in teaming and recruitment rather than task assignment

Tightly integrated coherent operation

Common-Mission Teaming in Practice

Collins:

- SE's active role in cross-group relationship management ensured common project mission understanding – this activity was credited as the leading enabler and facilitator of SE agility.
- High level mission was embraced and practiced by everyone: to be the most trusted source of communication and aviation electronics solutions.

USNavy:

- Team meetings opened with taxpayer and user stories of customer needs; a common team mission of affordability, usability, and effectiveness was the objective.
- Contract and program personnel operated on equal footing as team members; expected to interact family-like, all-for-one and one-for-all, on a mission for the end users.

Tesla:

- All job descriptions describe common mission as constant innovation to improve performance of everything while reducing number of parts, process steps, and lines of code.
- Opt-in 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 team enforcement of mission focus.

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.

Purpose:

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

Values:

- Minimize rework – Reveals development conflicts early.
- Maximize quality – Moves otherwise lost rework time into improved outcomes.
- Facilitate innovation – Shared cognitive knowledge instigates/fosters innovation.

Social Considerations: Transparency and knowledge equality evokes a sense of inclusion and facilitates higher personal performance.

Information containers of any kind, available to all and typically digital

Shared-Knowledge Management

Collins:

- Product line reusable everything and active catalog management is principle part of long term knowledge management.
- Continual evolution of Market Requirements Document (MRD) is mid-term knowledge management, with typical commercial software aps supporting real-time development-in-process knowledge management.

USNavy

- An interactive real-time knowledge base designed, built, and constantly evolved by internal staff orchestrated the interaction of all project participants; was considered an enabling cornerstone of project success.
- As a never-ending project, past-increment knowledge persisted for long term usage while current increment ubiquitous transparency revealed off-note activity immediately.

Tesla:

- Digital Self Management (DSM) AI software learns, evolves, and provides long term knowledge for design improvement assistance and product improvement evaluation – shared among all factories.
- No product backlog or Kanban board exists, instead there is only a what’s-happening-now listing on ubiquitous monitor boards and on all phones.

Continual Integration & Test

Why (Need): Early revelation of system integration issues.

What (Behavior): Integrated test and demonstration of work-in-process.

Purpose:

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

Values:

- Minimize rework – Reveals issues early for affordable rework.
- Maximize quality – Conserves funds for performance improvement.
- Facilitate innovation – Enables wip demonstrations to suggest innovations.

Social Considerations: Incremental wip test and demonstration provides progress visibility and frequent encouraging feedback.

Iteratively evolving self-driving technology integration platform.

Continual Integration & Test

Collins:

- Hardware engineering provided software engineering with prototype boards early on, and then began constructing what they call an integrated computing platform (ICP) for integrated system testing.
- The ICP capability evolved iteratively and incrementally through multiple physical stages over time until it was transformed into the final product.

USNavy:

- Three operational vehicles functioned as continual integration and test platforms; running data gathering tools to monitor network loading, CPU loading, variable values, and other component interaction factors.
- Device developers were required to install and experimentally test work in process on the integration platforms to provide early warning of integration issues.

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 car on the production line had on-board software running in what is called Factory Mode, which does integration testing as already united-tested components are added to the vehicle.

Being Agile: Operations Concept

Why (Need): Attentive operational response to evolving knowledge and dynamic environments.

What (Behavior): Sensing, responding, evolving.

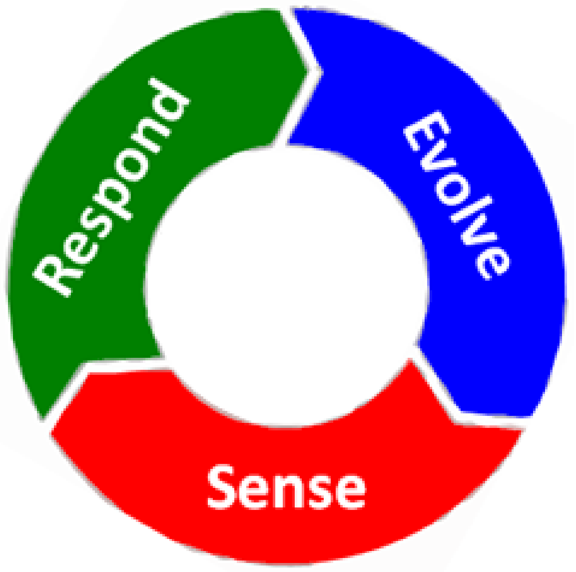
Purpose:

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

Values:

- Minimize rework – Optimizes expenditure with tight sense/respond/evolve loops.
- Maximize quality – Instills trust with unequivocal sense/respond/evolve loops.
- Facilitate innovation – Creates delight with venturesome sense/respond/evolve loops.

Social Considerations: Incentivize learning and sharing and reasonable innovation risks w/o penalty for trying that fails.



Three principles that operationalize agility

Being Agile: Operations Concept

Collins:

- 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: “Key agility-enabling concept is the active management of all relationships, more so than product line engineering, which just provides context.”

USNavy:

- 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.

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., ERP custom built as central nervous system.
- Suppliers and supplies changed a lot, so processes had high variability to deal with – score was on outcome, not conformance to past process, and on improvement, not consistency.

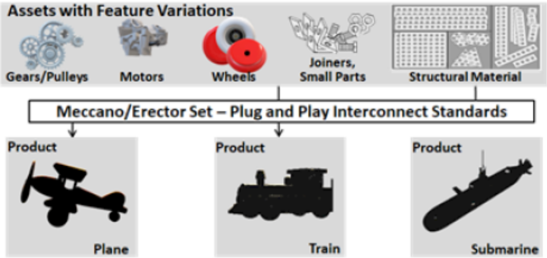
Product Line Architectures

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

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

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

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



Notional Agile Architecture Pattern

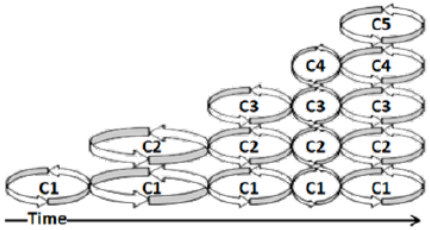
Iterative Incremental Development

Needs: Minimize unexpected rework and maximize quality.

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

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

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



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

Common-Mission Teaming

Needs: Coherent collective pursuit of a common mission.

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

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

Tightly integrated coherent operation

Continual Integration & Test

Needs: Early revelation of system integration issues.

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

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

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.

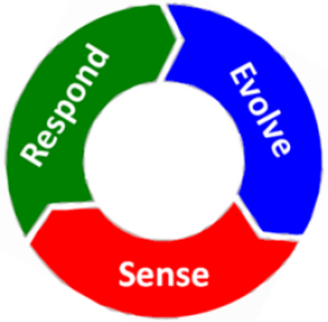
Information containers of any kind, available to all, and typically digital

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

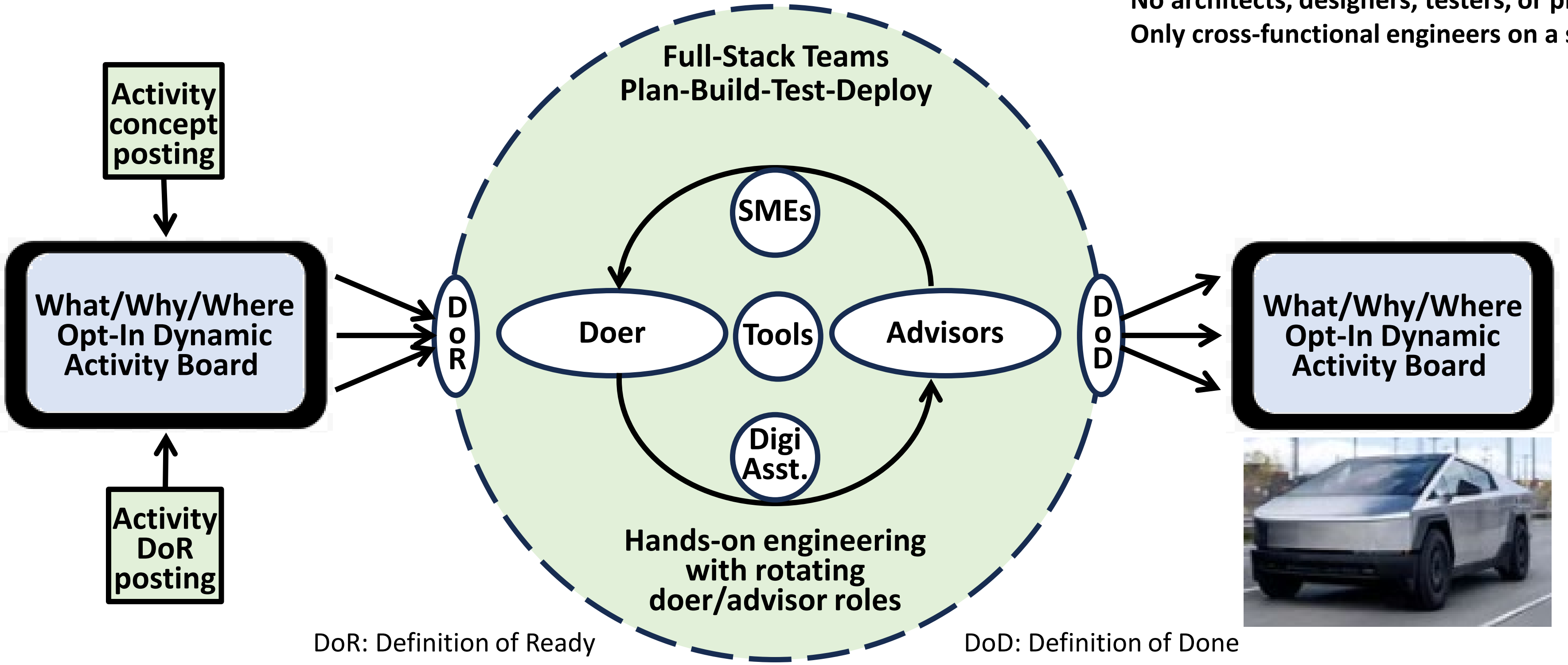
Reality …


CULTURE EATS
STRATEGY FOR
BREAKFAST.
Peter Drucker
wist.info

… and the Environment Shapes Culture

e.g., Group Experiential Engineering at Tesla

No architects, designers, testers, or project managers.
Only cross-functional engineers on a steep learning curve.



Enabler:

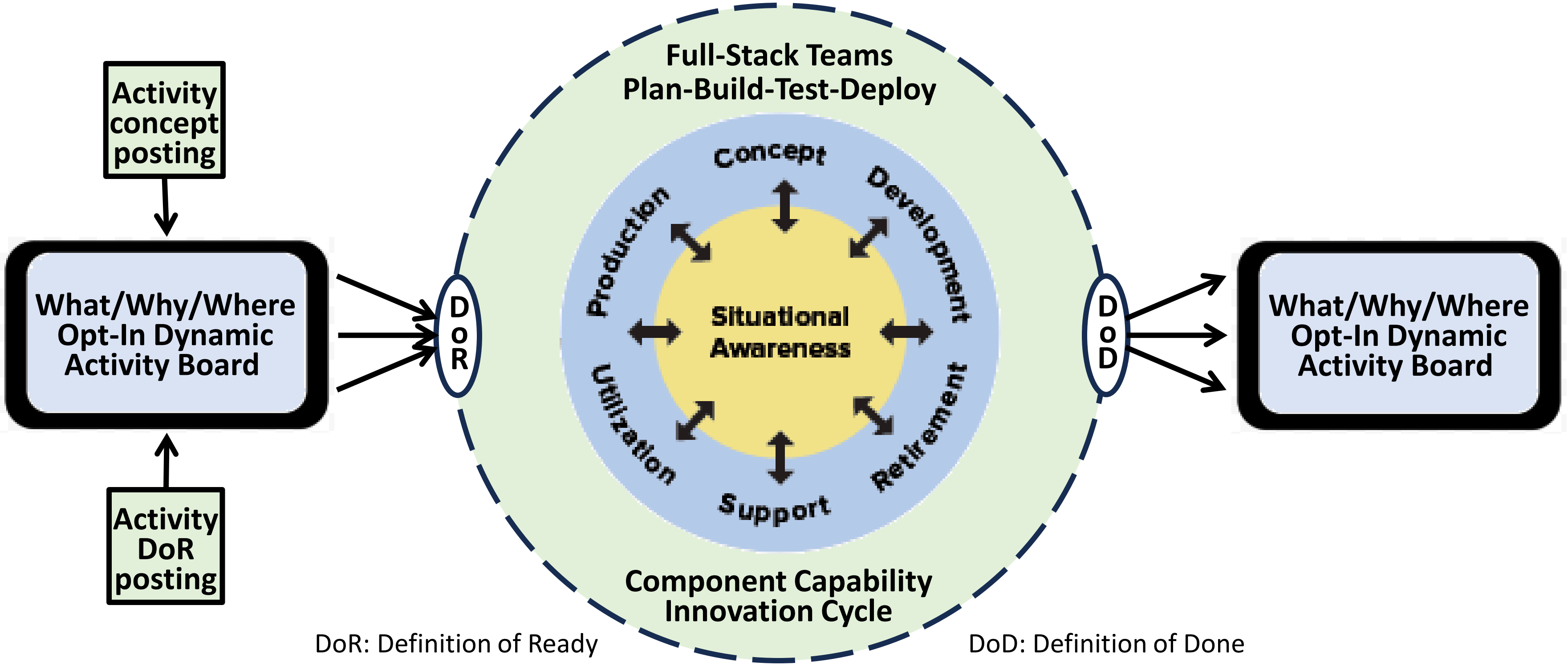
- Compelling long-term, mission-driving, never-ending, Innovation engineering

Facilitators (infrastructure):

- Modularity for parallel independent workstreams
- AI/ML data sets on everything
- Digital Self Management (DSM) as AI/ML application
- Comprehensive automated testing
- Real-time product usage feedback
- Digital twins
- Et al.

… and the Environment Shapes Culture

e.g., Group Experiential Engineering … Anywhere



This approach does not need
a Tesla-like umbrella
or an Elon Musk to hold it

Source Material

Agile Development at Tesla, 2021, March 12
www.agile-academy.com/en/organizational-development/agile-entwicklung-tesla-joe-justice

The Secrets to Agility at Tesla, Joe Justice, 2021, November 21
www.youtube.com/watch?v=AKqKCn4fheY

Radical Unbossing at Tesla, Joe Justice, 2022 April 6
www.youtube.com/watch?v=ozdBx1SG-vo

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/

Ways of Working at Tesla Talk by Joe Justice with Q&A, 2023 February 14
www.youtube.com/watch?v=lGw3Z9Kk5Iw

Digital self management - Joe Justice keynote at the LPPDE Virtual Summit, 2023 March 9
www.youtube.com/watch?v=qypG12NUtgo

Joe Justice, MobAI with Joe Justice, 2023 August 9
www.youtube.com/watch?v=Fh_FDGiTvcg

Tesla and SpaceX speed with MobAI, Joe Justice, 2023 October 18
www.youtube.com/watch?v=N-JYhKZV4-A

Agile Systems Engineering Process Features Collective Culture, Consciousness, and Conscience at SSC Pacific, 2016, July 18 [USNavy case study]
https://www.researchgate.net/publication/308083752

Case Study: Agile Hardware/Firmware/Software Product Line Engineering at Rockwell Collins, 2017 April 24 [Collins case study]
https://www.researchgate.net/publication/316280809

Agile Systems Engineering – 8 Core Aspects, INCOSE Symposium, July 2023
https://www.researchgate.net/publication/373092973

Full Series

Agile 301 webinar slides: Agile SE Strategies – Purpose, & Values
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:
INCOSE Professional Development Portal

Webinar ID: Webinar 176 Dove 02 October 2024 Agile SE Strategies 301
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