Welcome to the Webinar! Please note that we have moved to the ZOOM platform. Please join ZOOM audio (Voice over Internet) if you are able to connect. Otherwise, please see the webinar invitation for dial-in phone lines

INCOSE Webinar Series

Wednesday 15th September 2021 – Webinar 154

Agile Systems Engineering Life Cycle Model

Rick Dove

INCOSE

Webinar

Agile SE Processes 204:
Agile System Engineering Life Cycle Model

15-Sep-2021

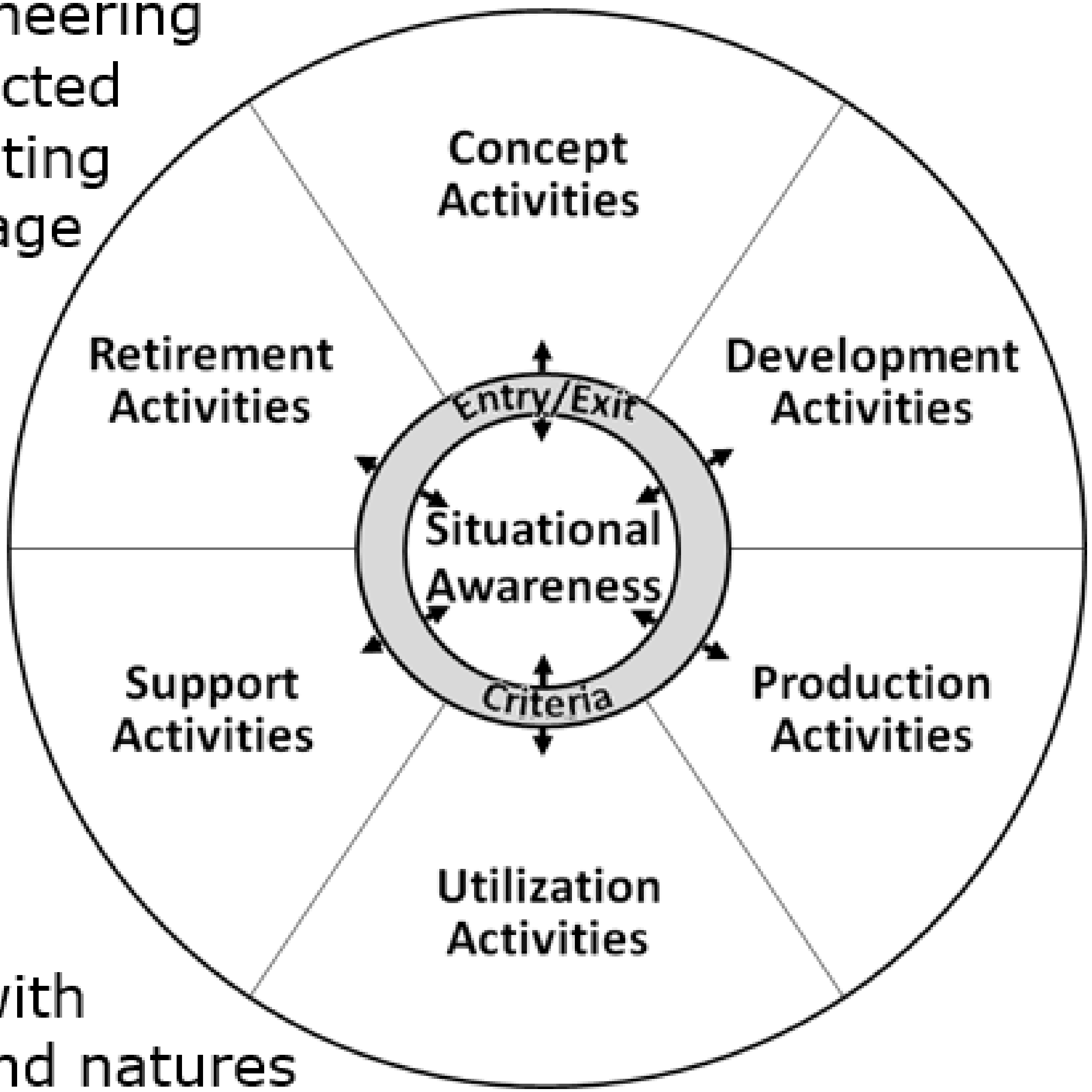
Rick Dove
Anthem, AZ, dove@parshift.com, 575-770-7101
Independent Operator
Adjunct Professor, Stevens Institute of Technology
Chair: INCOSE WG for Agile Systems & Systems Engineering

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: The Agile Systems Engineering Life Cycle Model is graphically depicted as a circle of life cycle stages radiating from a newly recognized center stage of situational awareness.

How can this depiction and its operational management implications help us get better performance from any agile systems engineering process?

This webinar will explore the management implications of the circular agile SE life cycle model, with an emphasis on the critical roles and natures of situational awareness and knowledge management.



Agility Knowledge Development

**In the ‘90s we analyzed hundreds of
real-world <u>systems and processes</u>
that exhibited agility, asking how they did that, and
converged on fundamental structural patterns that fit facts.**

**Recently we have analyzed
real-world <u>SE processes</u>
that exhibit agility, asking how they do that, and
converging on fundamental behavior patterns that fit facts.**

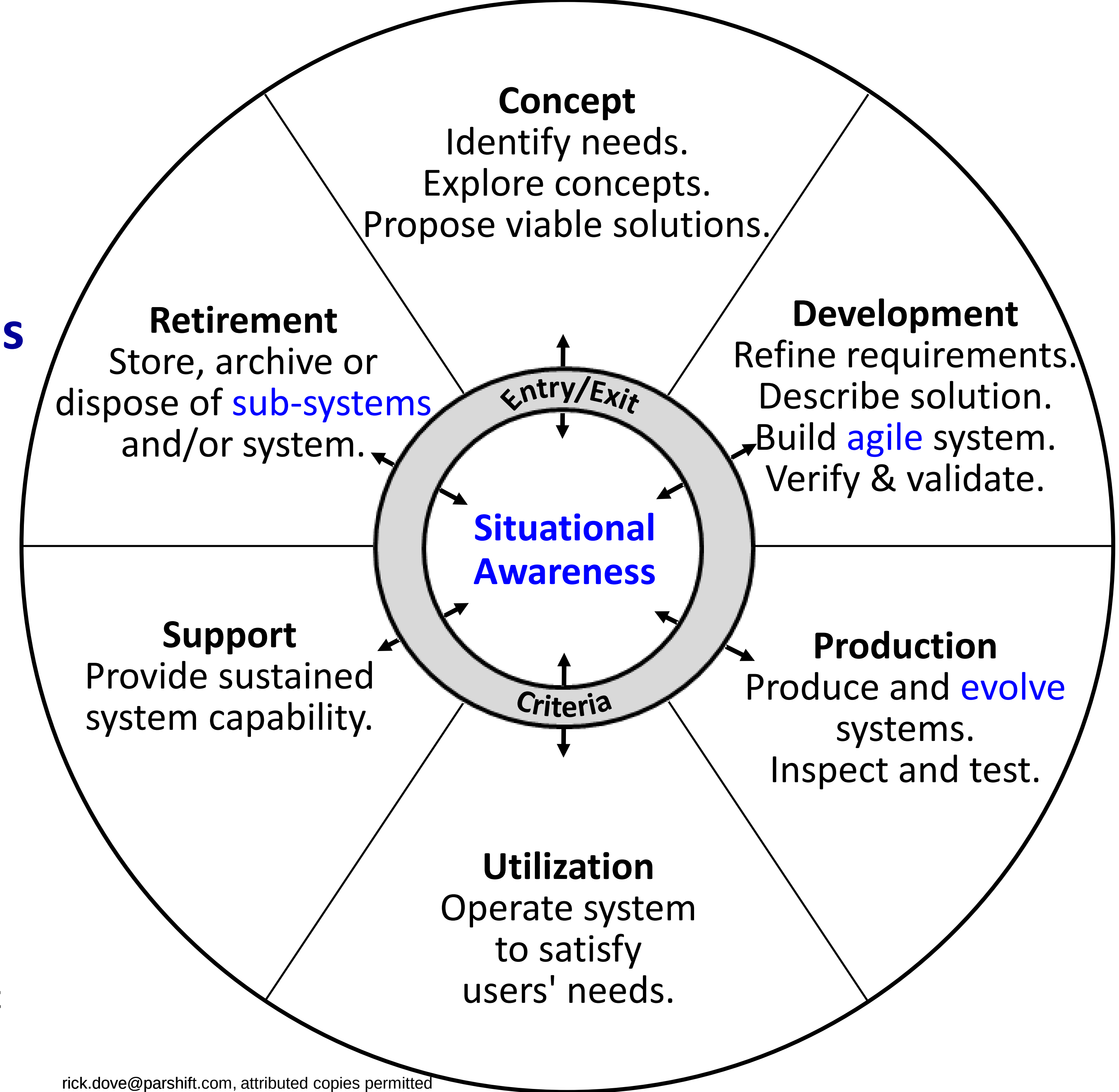
No conjecture, no kinda good idea, no opinion.

Agile SE Life Cycle Model (ASELCM)

Situational Awareness Engages Other Stages and Tasks Asynchronously Concurrently Experimentally Incrementally Iteratively

Compatible with ISO/IEC/IEEE 24748-1:2018

Systems and software engineering — Life cycle management — Part 1: Guidelines for life cycle management



Clarifications

Life Cycle Model [ISO/IEC/IEEE 15288:2015]

“framework of processes and activities concerned with the life cycle that may be organized into stages, which also acts as a common reference for communication and understanding.”

Life cycle models are about a systems entire life, from birth to death.
They differentiate stages of different activity during life,
to demarcate decision points and criteria for stage entries and exits.

ISO/IEC/IEEE standards call out six “common” generic stages:
concept – development – production – utilization – support – retirement

SEBoK on the Vee Model

"A Primarily Pre-specified and Sequential Process Model: The Vee Model"

"…Its core involves a sequential progression of plans, specifications, and products that are baselined and put under configuration management. … The Vee Model encompasses the first three life cycle stages listed in the "Generic Life Cycle Stages" table of the INCOSE *Systems Engineering Handbook*: exploratory research, concept, and development (INCOSE 2012)."

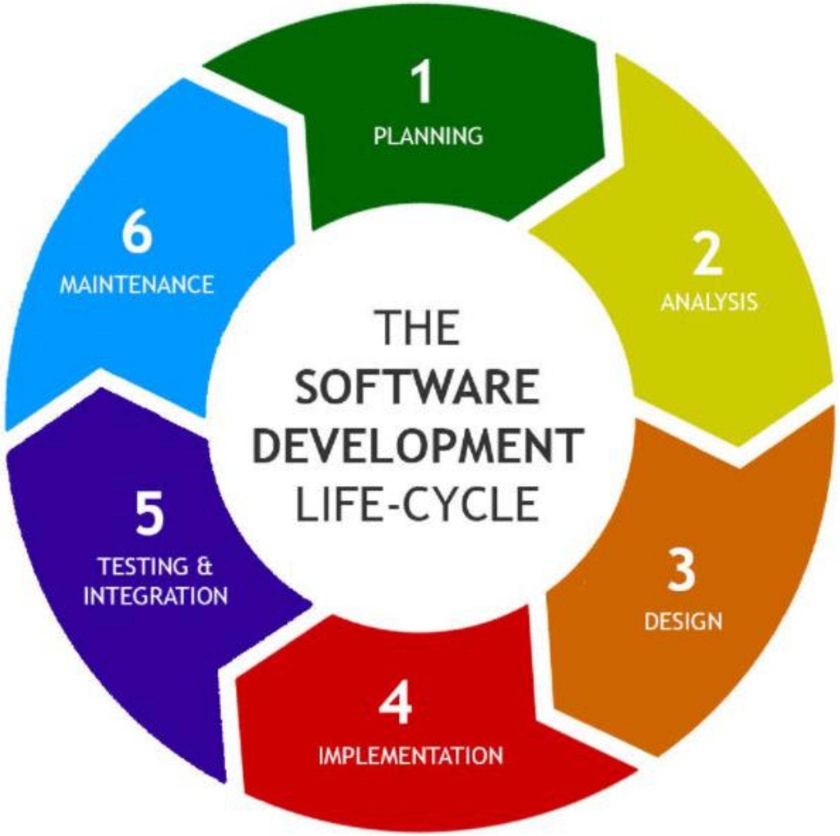
Notes:

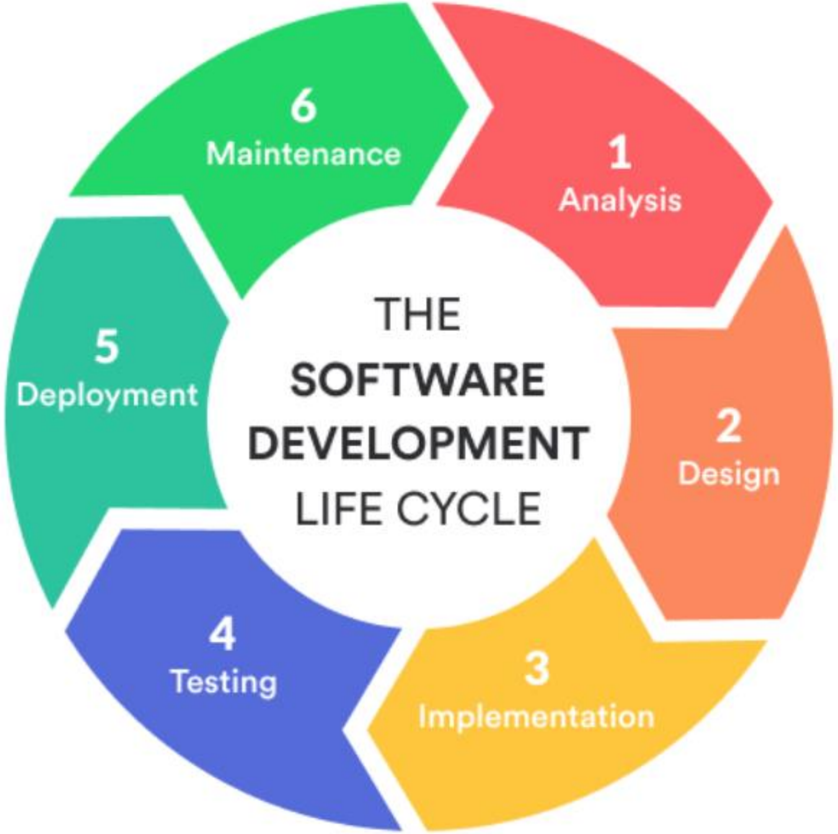
This SEBoK article is outdated, referencing the Handbook v3.2.2 version.
Handbook v4 and 24748-1:2018 combine Research and Concept stages into a single stage.

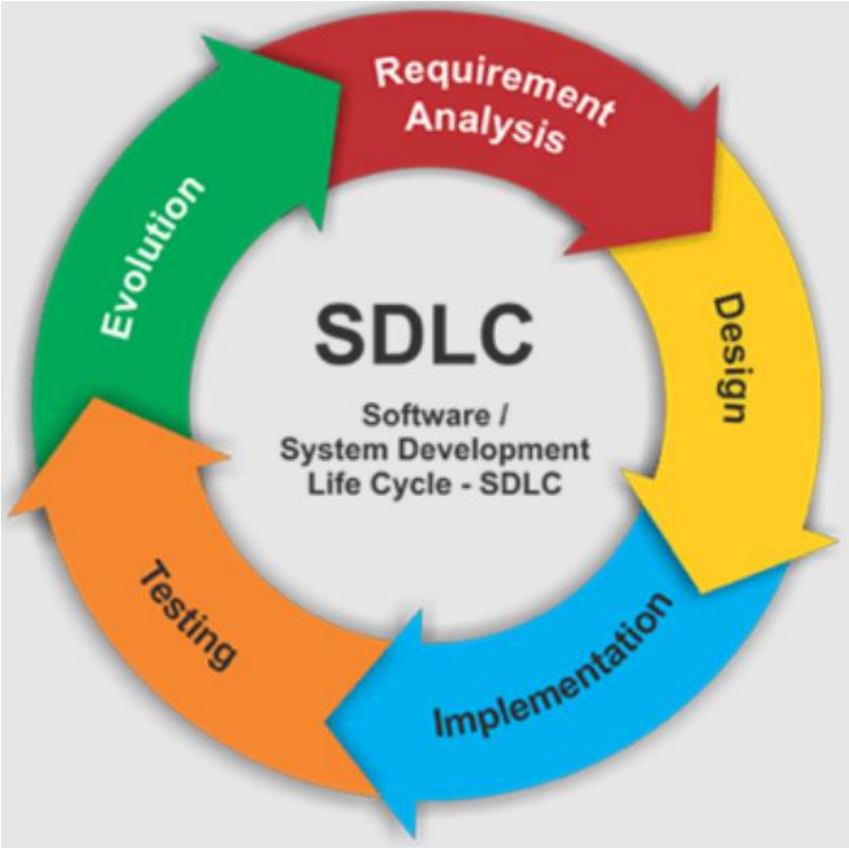
Vee is not a system life cycle model,
it is a useful model of activity relationships within the concept and development stages.

www.sebokwiki.org/wiki/System_Life_Cycle_Process_Models:_Vee

Circular Life Cycles are Not New







Many nuanced variations, but all depict a repeating sequential process, of perpetually evolving software systems

Agile Software Development

Scrum, Kanban, SAFe, LeSS, et al. … are not life cycle models.

They are organizational and work flow patterns for development activity.

The Agile LCM is Familiar Practice

Concurrent Stage Activity

- You're driving a Tesla = utilization stage.
- Simultaneously the factory is downloading an AI upgrade = production stage.
- Simultaneously that upgrade is replacing an older capability = retirement stage.
- Simultaneously engineers are creating next month's upgrade = development stage.
- Simultaneously marketeers are dreaming up next year's upgrade = concept stage.
- Simultaneously maintenance is downloading a controls change to compensate for wear = support stage.

Asynchronous Stage Activity

- You're using a Dell desktop PC in the morning = utilization stage.
- In the afternoon an SSD (Solid State Drive) is installed = production stage.
- Which replaces the Hard Drive = retirement.
- Next day the BIOS are adjusted by hand for optimal SSD performance = support stage.
- Dell is creating a new widescreen monitor you'll purchase when it is available = development stage.
- Dell is always dreaming up product line extensions = concept stage

Behind the Scenes

The Agile SE Life Cycle Model
is not an invention, a good practice, or a recommended procedure,
it is simply an observation of how systems live through time.

We experience it with many systems.
But systems engineering doesn't play it like an instrument.
It is leading us rather than us leading it.

Time to change that.

Six Findings Covered in Prior 3 Webinars

Problem-Space Characterization

CURVE

Caprice: Unknowable situations.
Unanticipated system-environment change.

Uncertainty: Randomness with unknowable probabilities.
Kinetic and potential forces present in the system

Risk: Randomness with knowable probabilities.
Relevance of current system-dynamics understanding.

Variation: Knowable variables and associated variance ranges.
Temporal excursions on existing behavior attractor.

Evolution: Gradual successive developments.
Experimentation and natural selection at work.

Response Strategies

	Domain	Response Strategies
Proactive	Creation	•Threat and opportunity awareness • Acculturated memory •Response actions/options • Decisions to act
Proactive	Improvement	•Awareness/Sensing • Action/option effectiveness •Memory in culture, options, ConOps, SEMP
Proactive	Migration	•New fundamentally-different types of threats and opportunities
Proactive	Modification (Capability)	•Actions appropriate for needs •Personnel appropriate for actions
Reactive	Correction	•Insufficient awareness • Wrong decisions •Ineffective actions/options
Reactive	Variation	•Effectiveness of actions/options •Effectiveness of evaluation
Reactive	Expansion (Capacity)	•Capacity to handle 1-? actions simultaneously
Reactive	Reconfigu-ration	•Elements of an action •Response managers/engineers

Stake Holder Engagement

Developers, Operators, Customers
Subcontractors, Producers, End Users
Security Engineers, Maintainers, Management

Two older forms of stakeholder engagement:

Integrated product team (IPT) is a multidisciplinary group of people who are collectively responsible for delivering a defined product or process. The emphasis of the IPT is on involvement of all stakeholders (users, customers, management, developers, contractors) in a collaborative forum.

Concurrent engineering (CE) is a work methodology emphasizing the parallelization of tasks (i.e. performing tasks concurrently), which is sometimes called simultaneous engineering or integrated product development (IPD) using an integrated product team approach. It refers to an approach used in product development in which functions of design engineering, manufacturing engineering, and other functions are integrated to reduce the time required to bring a new product to market.

Two newer forms of stakeholder engagement

DevOps is a set of software development practices that combine software development (*Dev*) and information-technology operations (*Ops*) to shorten the systems-development life cycle while delivering features, fixes, and updates frequently in close alignment with business objectives.

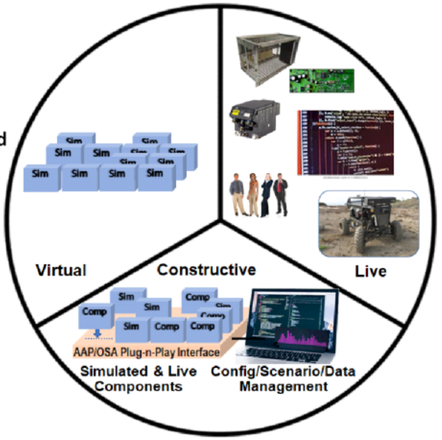
Live-Virtual-Constructive is an early integrated system making use of simulations and other proxies as well as work in process and finished components for early integration testing, demonstration, and review.

Continuous Integration Platform

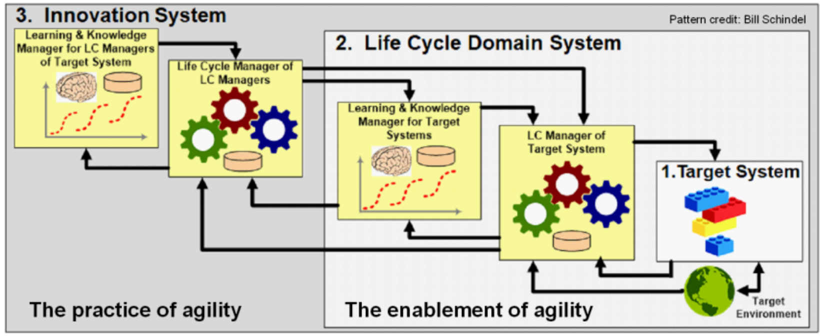
Live components
(people, things)
Virtual components
(component simulations)
Constructive capabilities
(component mix, scenario simulation, and performance monitoring/recording)

L&V components are functional system elements; configured, challenged and monitored by C elements for performance and anomalies.

Demonstration/test/experimental events can occur at any time with the latest instantiation of simulations & components.



Operational Pattern



Operational Principles

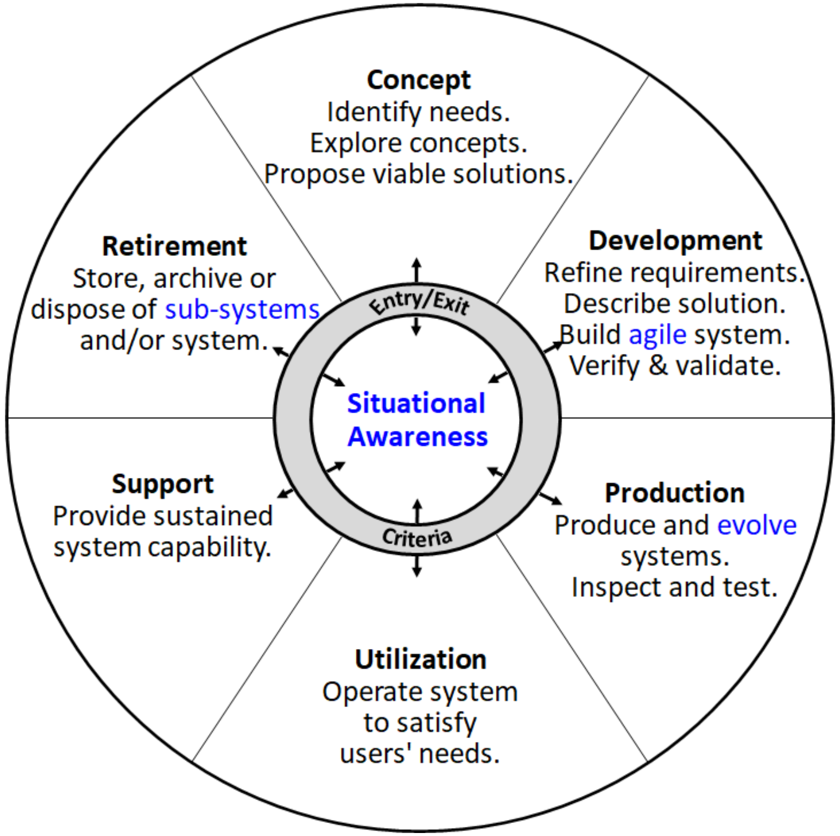
Sensing (observing, orienting)
- External awareness (proactive alertness)
- Internal awareness (proactive alertness)
- Sense making (risk & opportunity analysis, trade space analysis, …)

Responding (deciding, acting)
- Decision making (timely, informed)
- Action making (invoke/configure process activity for the situation)
- Action evaluation (validation & verification)

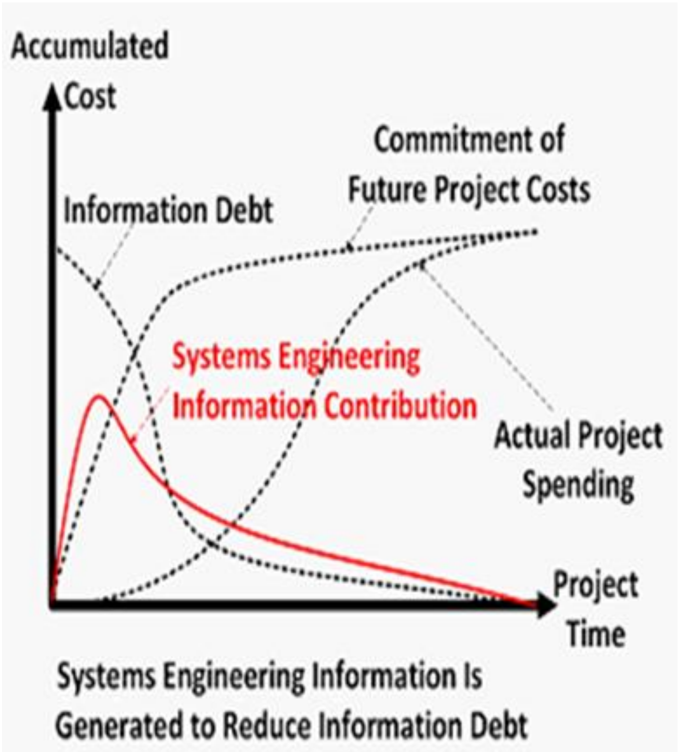
Evolving (improving above with more knowledge and better capability)
- Experimentation (variations on process ConOps)
- Evaluation (internal and external judgement)
- Memory (evolving culture, response capabilities, and ConOps)

Two Additional Findings are Core Enablers

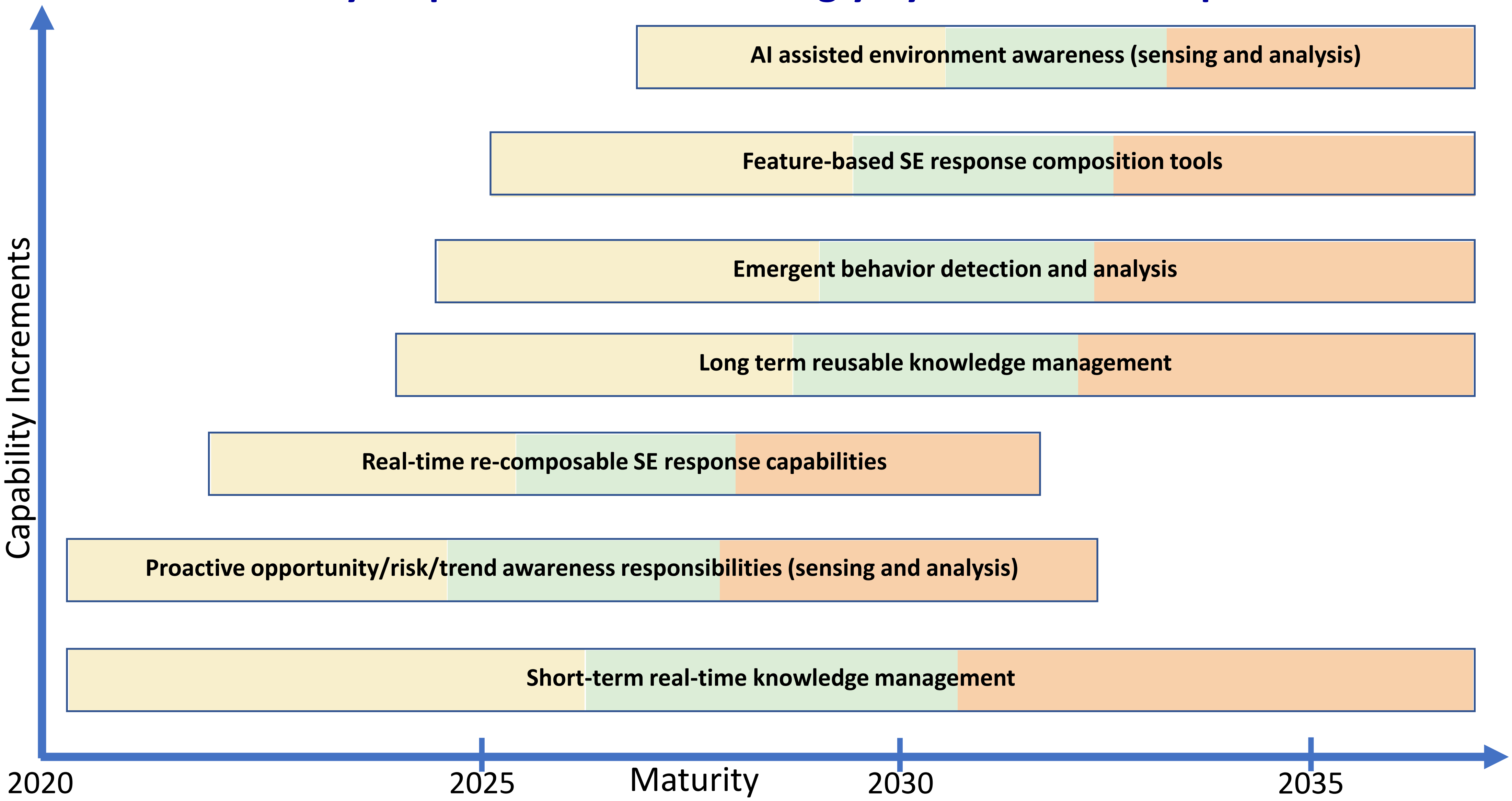
Situational Awareness



Knowledge Management



A Roadmap to Meet an Emerging Need:
SE anticipates and effectively responds to an increasingly dynamic and complex environment



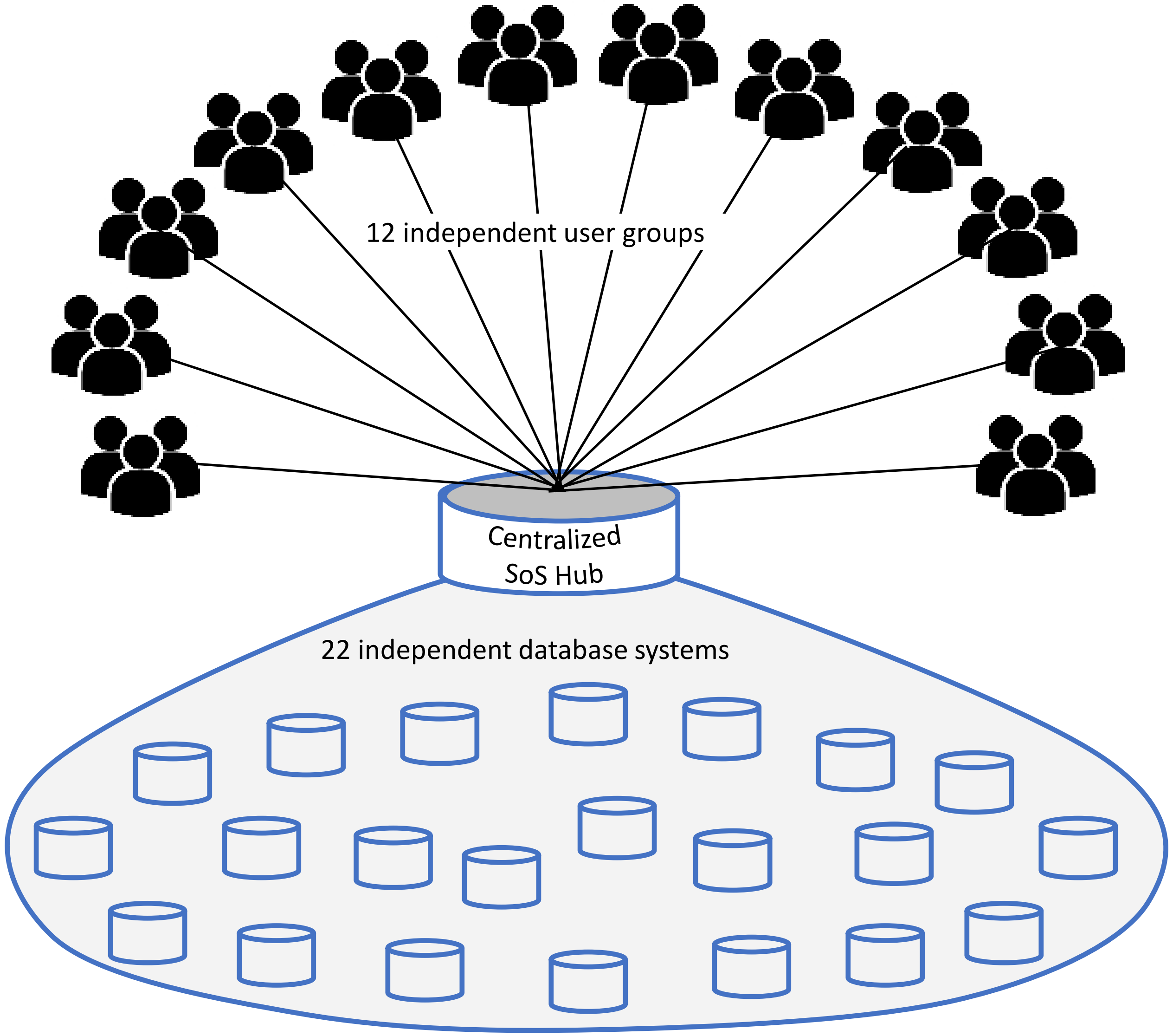
Coping with an Increasingly Complex and Dynamic Environment

What	How
1. Digital engineering platforms that facilitate dynamic cross-discipline interoperable modeling and simulation.	M&S tool vendors embrace interoperability standards.
2. Foundation open architectures that leverage encapsulated modularity for asset reuse, coherent interaction, composable innovation, and adaptable resilience.	Wide adoption of Product Line Engineering methodologies with supporting tools.
3. Collaborative platforms that facilitate cross-team work flow, change management, and dynamic short- and long-term knowledge management.	Team-support tool vendors move broadly into system engineering and knowledge management.
4. Sensors that monitor life cycle environments for potential impairment and opportunity.	Human tasking with increasing technology assistance, especially from AI.

Military Logistics Centralized Systems-of-Systems Web-Hub

Northrop Grumman (case study reference on final slide)

Case Study of
Northrop Grumman's Global Combat Support System – Joint (GCSS-J) group in Herndon, Virginia.



Six years of
effective employment and evolution,
winning praise from GAO and users alike.

CURVE Environment

Northrop Grumman

Caprice

- External data sources change their services at will
- COTS (Common Off The Shelf) software upgrades deprecate existing interfaces

Uncertainty

- Software and/or hardware may go end-of-life at any point

Risk

- May not be able to meet 15-day schedule for delivery of security fixes

Variation

- Number of security vulnerabilities to address varies greatly week-to-week
- Development man-hours available for capability evolution in competition with higher priority patches and security updates

Evolution

- As technology changes, the program must port existing capability to new technology

Some Notable Process Concepts

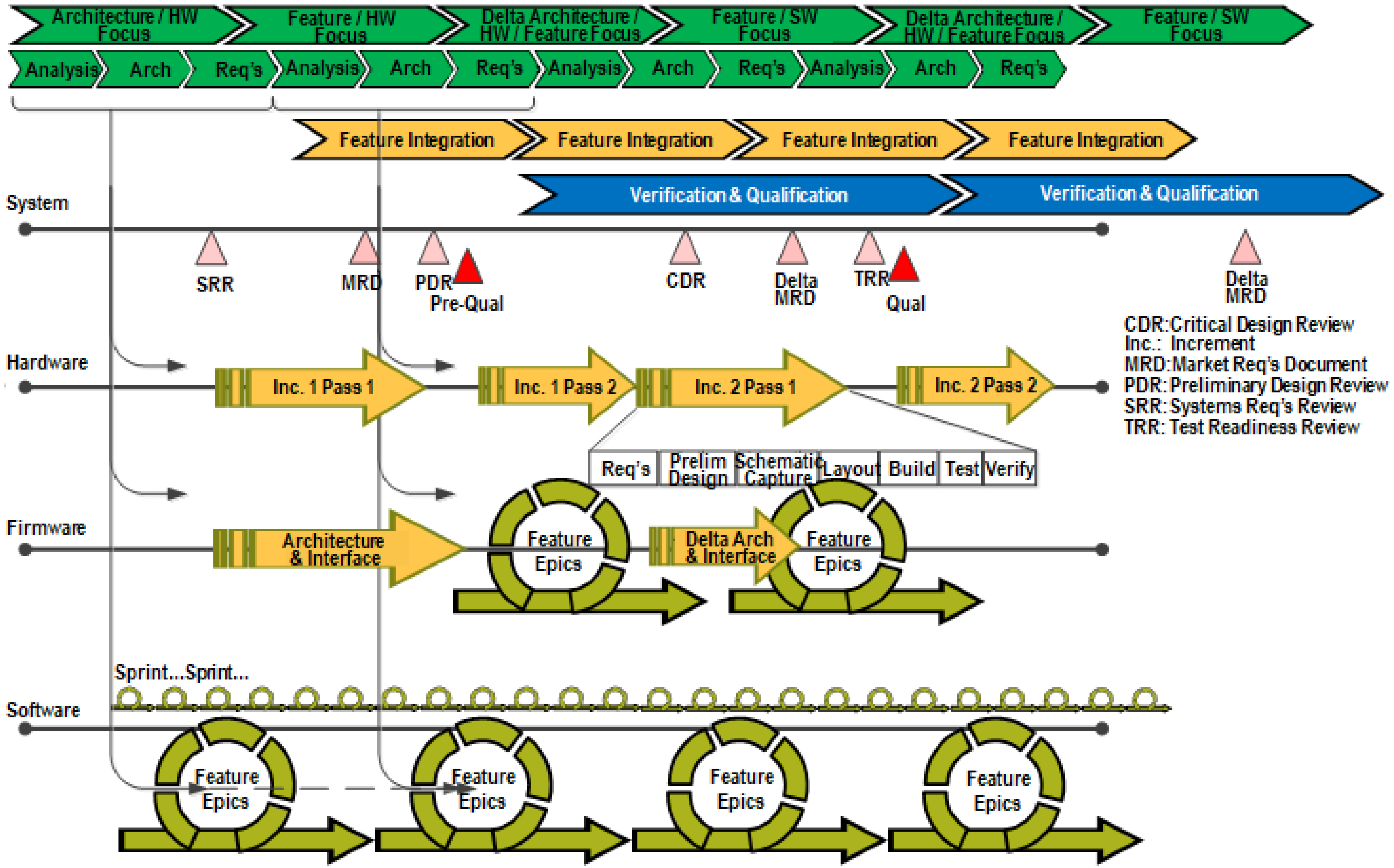
Northrop Grumman

- Intimate stakeholder involvement in the SE process.
- Asynchronous and simultaneous life cycle stage activity,
  in never-ending system evolution.
- Hybrid Scrum/Waterfall/Wave process-model integration,
  in contract conformance.
- CMMI level 5 procedure discipline,
  providing seamless new-release operational stability.
- Awareness and mitigation of external environment evolution.
- Real-time optimal process-control model,
  for re-prioritizing development-increment activity and
  acting on feedback.

Military Radio Product-Line

Rockwell Collins (case study reference on final slide)

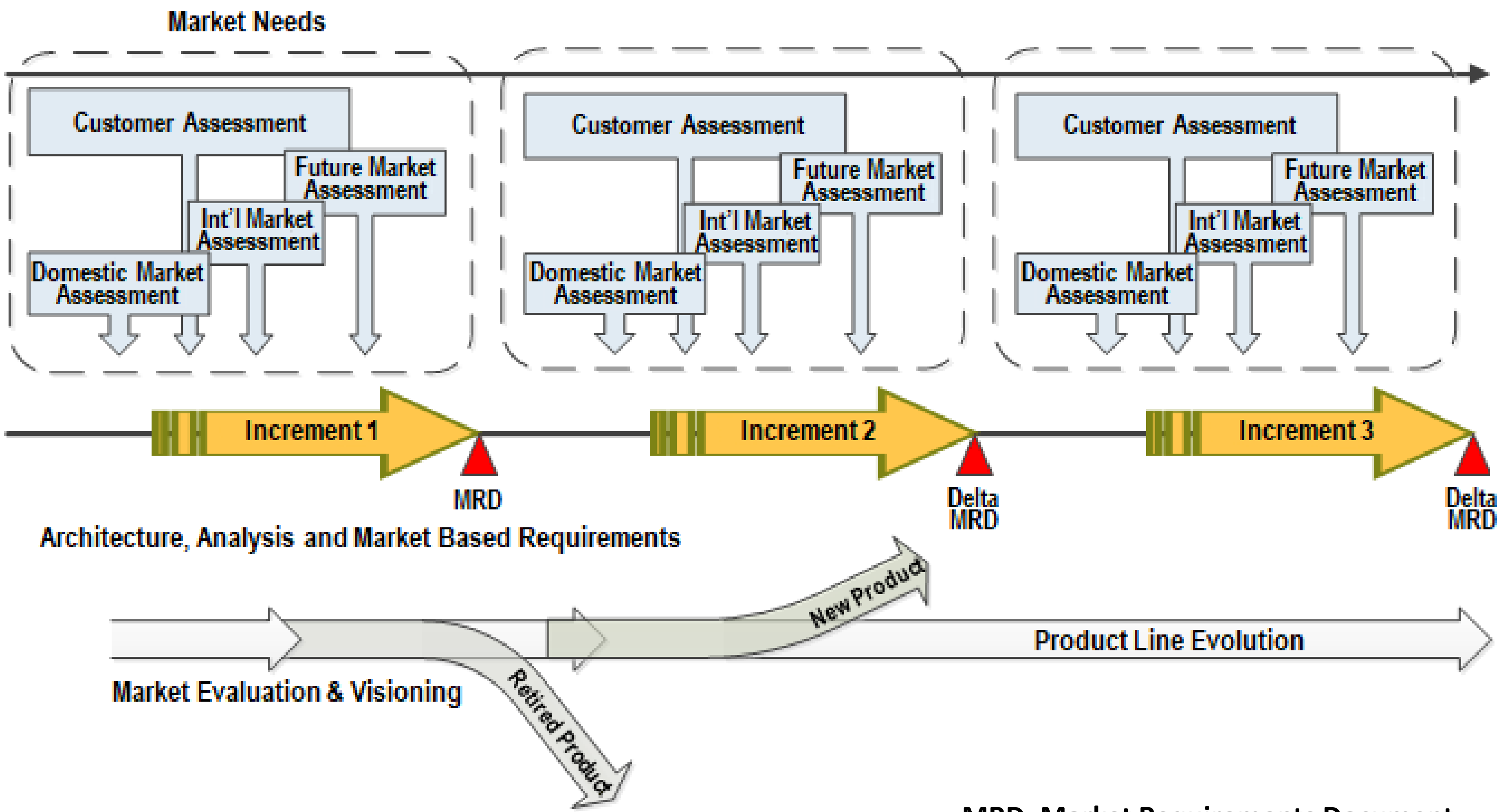
Asynchronously Aligned Discipline Increments



slide credit: Will Hartney

Incremental Awareness Attention

Rockwell Collins



MRD: Market Requirements Document

slide credit: Will Hartney

Warplane Evolution

Lockheed Martin IFG Fort Worth (case study reference on final slide)

In 2015 IFG was in early experimentation with a self-funded Continuous Integration Platform concept, called the Agile Non-Target Environment (ANTE).

ANTE systems consist of simulated components, previously built components, wip components, finished components, low-fidelity COTS proxies, software wip.

Subcontractors are required to provide early device simulations to ANTE specs.

By mid-2017 ANTE was declared a successful experiment, achieving applause in customer feedback that values:

- Early and incremental demonstration of work in process.
- Early exposure to difficulties in need of attention.

IFG: Integrated Fighter Group, F16/F20/F35 upgrades

Process Instrumentation

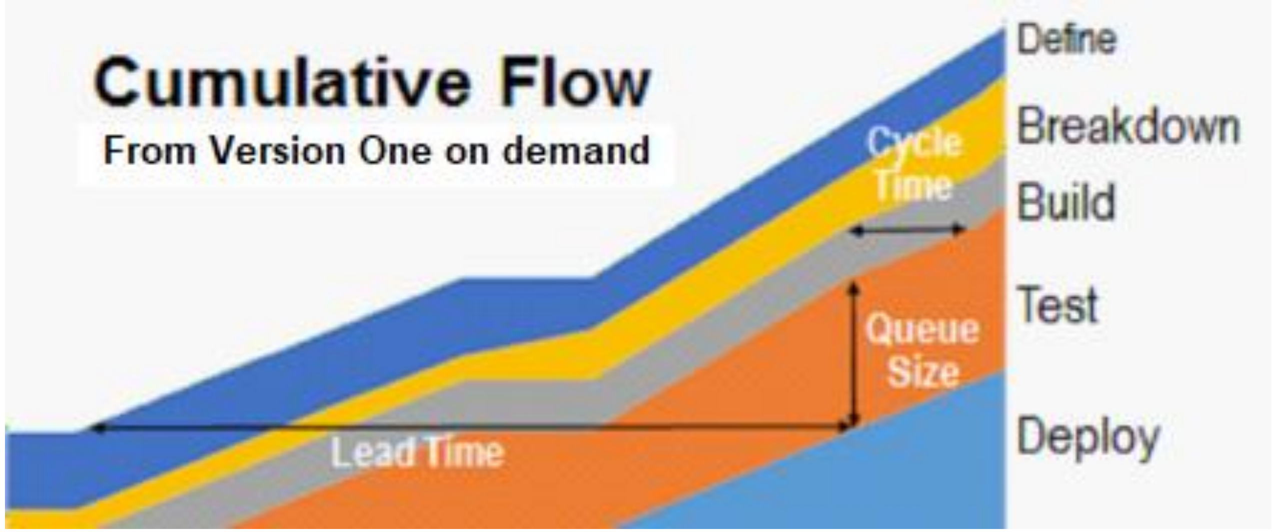
Lockheed Martin IFG

Workflow management critical to avoid schedule-threat bottlenecks.

Instrumentation provides awareness and bottleneck prediction.

Examples:

- **Test facility bottlenecks mitigated by managing queue size.**
- **Team loading bottlenecks mitigated be assigning tasks to less-loaded teams (rather than most-expert teams).**



Automated cumulative process-flow metrics, with queue size predicting cycle time in a test facility.

See Don Reinertsen. 2009. *The Principles of Product Development Flow.*

Continuous Integration Platforms

SpaWar Systems Center Pacific (case study reference on final slide)

RaDER
Reconnaissance and Detection Expendable Rover

Evolving Capability

autonomous off-road vehicle technology

Full system test and demo every 6 months, with next cycle adding new features.

Asynchronous testing of wip within the 6-month cycle frequently.

Platforms are instrumented to detect integration problems early (e.g., a wip device from a subcontractor hogging to much bandwidth or CPU cycles).

SE team evolves the platform architecture every cycle to accommodate new needs.

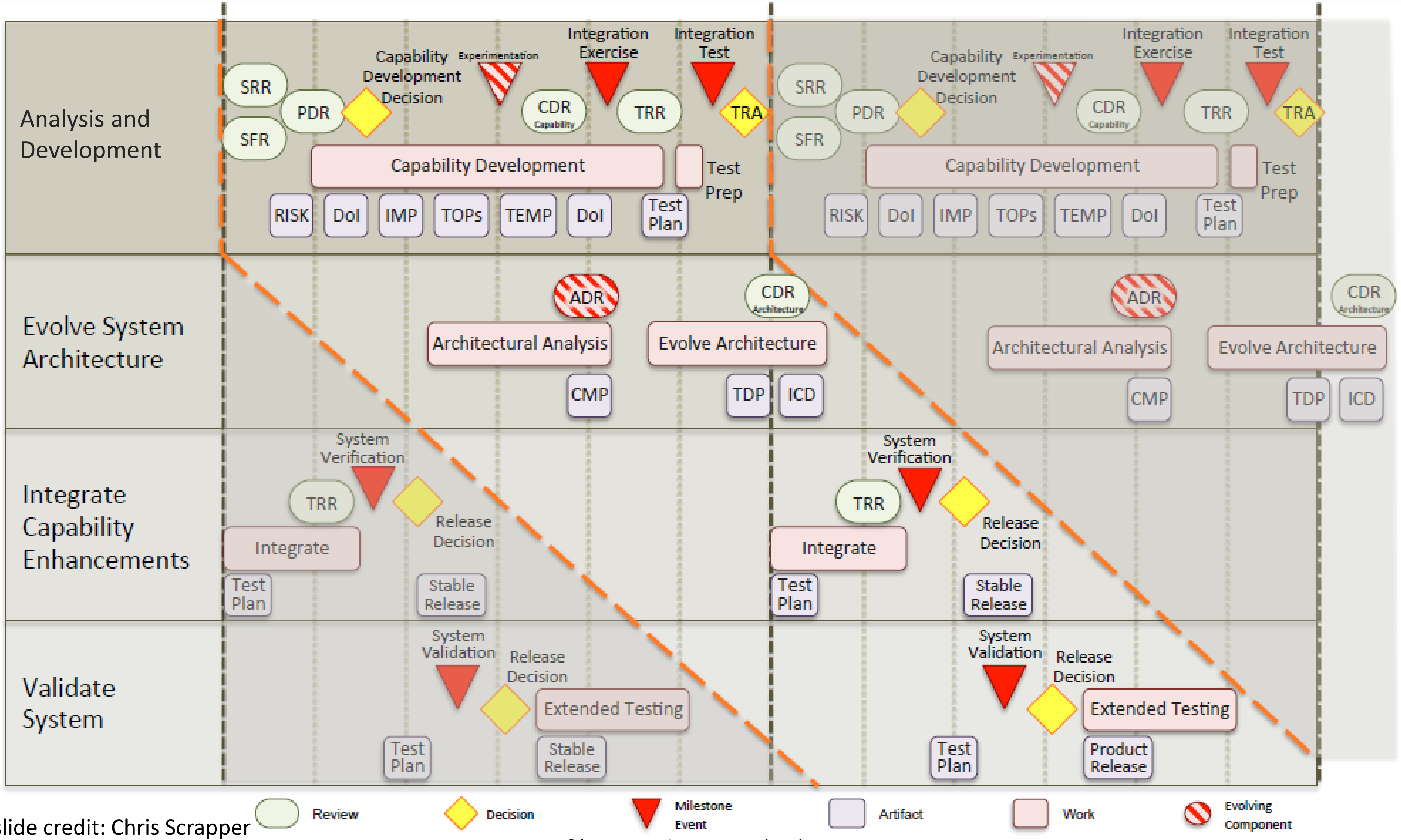
Both warfighters and sponsors witness end-of-cycle tests and demos, and often show up during a cycle for wip demos.

EV1
Expeditionary Vehicle 1

CDR: Critical Design Review
DoI: Declaration of Intent
PDR: Preliminary Design Review
SDR: System Design Review
SFR: System Functional Review
SRR: System Requirements Review
TEMP: Test and Experimentation Master Plan
TOP: Test Operating Procedures
TRR: Test Readiness Review

Integrated Strategy Chart

SpaWar Systems Center Pacific



slide credit: Chris Scrapper

Collective Consciousness

SpaWar Systems Center Pacific

The Continuous Integration Environment (CIE) is a data-driven repository of knowledge, with customized viewing templates for different needs. CIE provides user interfaces that separate internal representations of data (the *model*) from the ways that information is presented to users (the *view*), with custom views for different stakeholders.

This homegrown CIE is structured as a federation of independent capabilities, mostly off the shelf, and is being evolved to provide real-time relevant and comprehensive views of history and current status to all team members.

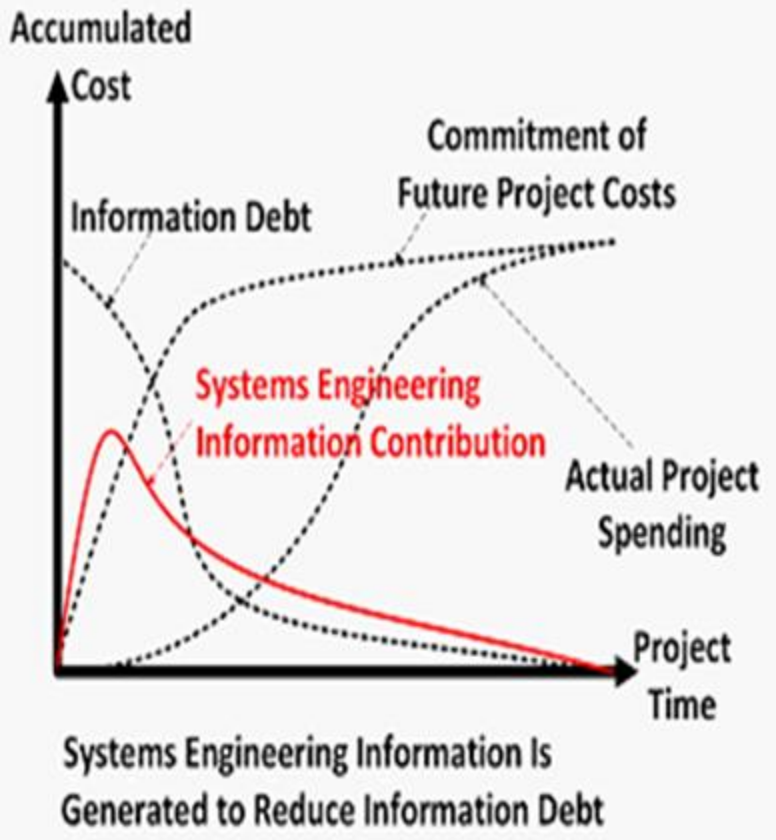
The CIE intent is to facilitate a real-time collective consciousness, where all team members are plugged in to all information associated with full project success, as well as to the information of relevance to their specific responsibilities and tasks.

New data, new decisions, new issues, new test results, ripple through the relevant federation of CIE components and CIE user views immediately.

This collective consciousness manifests for the team much like it does for musicians in a symphony orchestra, where off notes and bad timing are immediately sensed by all.

Knowledge Management

creation, curation, dissemination, expulsion



Knowledge Management for:

- **Situational awareness**
- **Lessons learned (for product and process)**
- **Reusable knowledge for other projects**
- **Team member attrition and replacement**
- **Production and maintenance support**
- **Other-party sustainment**
- **Evolution when original developers are gone**

Incremental and Iterative Methods

affordable knowledge development

Generally increments add capabilities and iterations improve capabilities.

An increment of capability development may encompass a series of iterations intended to experimentally-converge on a satisfying result.

Useful when requirements are unclear from the beginning, or to hold the SoI open to the possibilities of inserting new technology, or when a minimally viable SoI is desired quickly with new or improved capabilities that can follow.

Learning Cycles

Cycle times for increments and iterations.

- **Increment cycles are beneficially timed to accommodate coordinated events such as integrated testing and evaluation, capability deployment, experimental deployment, or release to production.**
- **Iteration cycles are beneficially timed to minimize rework cost as a project learns experimentally and empirically.**

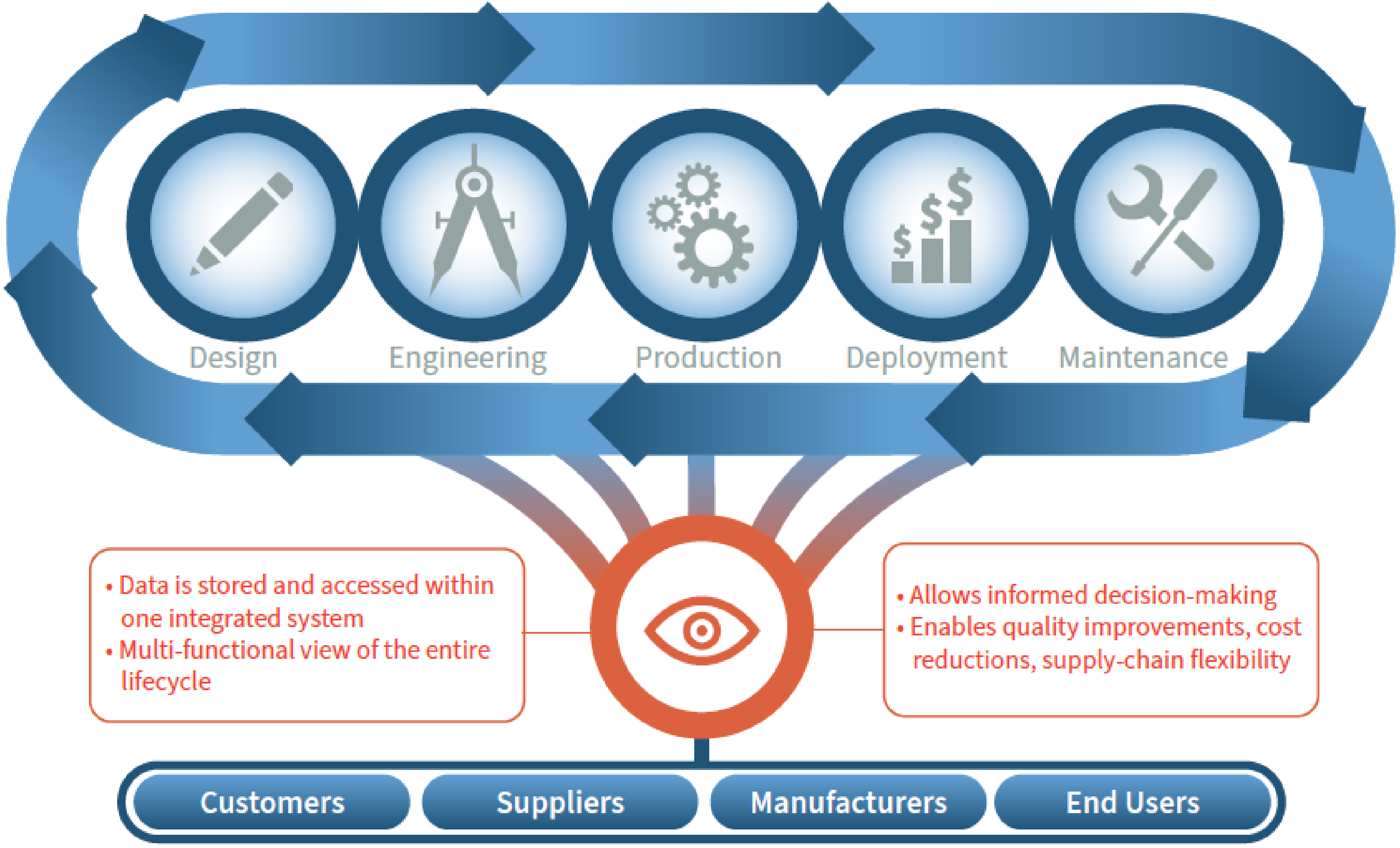
In evolutionary innovation, cycle times may have a constant cadence beneficially.

Appropriate and different for each engineering domain.

In revolutionary innovation, cycle times may vary beneficially based on risk perception.

SpaceX did 5 iterations to achieve reusable vehicle capability,
iterations ended when likelihood of success was greater than 50%

Product Lifecycle Management (PLM) – Knowledge Management Support



Product lifecycle management enables quality improvements, cost reductions, supply-chain flexibility, and more informed decision-making throughout a product's lifespan.

Figure from: Evolving Strategies for Initiating Product Lifecycle Management Processes. National Center for Manufacturing Sciences. 2021
https://www.ncms.org/evolving-strategies-for-initiating-product-lifecycle-management-processes/

Emerging Fundamental Principles

All case studies enabled and facilitated (with different methods):

- **Project situational sensing and response.**
- **Team-members' engagement sensing and response.**
- **Development-issue sensing and response.**
- **Integration-issue sensing and response.**
- **Assimilated shared-culture and evolution.**
- **Process and procedure evolution.**
- **Product evolution.**

Three Categories of Fundamental Principles Emerge:

- **Sense/Monitor – awareness is the driver of agility**
- **Respond/Mitigate – action is the expression of agility**
- **Evolve – applied learning is the sustainer of agility**

Awareness is Central

The situational awareness stage is responsible for

- **monitoring & generating knowledge of the environments**
- **triggering entry into other stages based on that knowledge**
- **passing relevant knowledge to other stages**

Questions begging for answers by the circular depiction

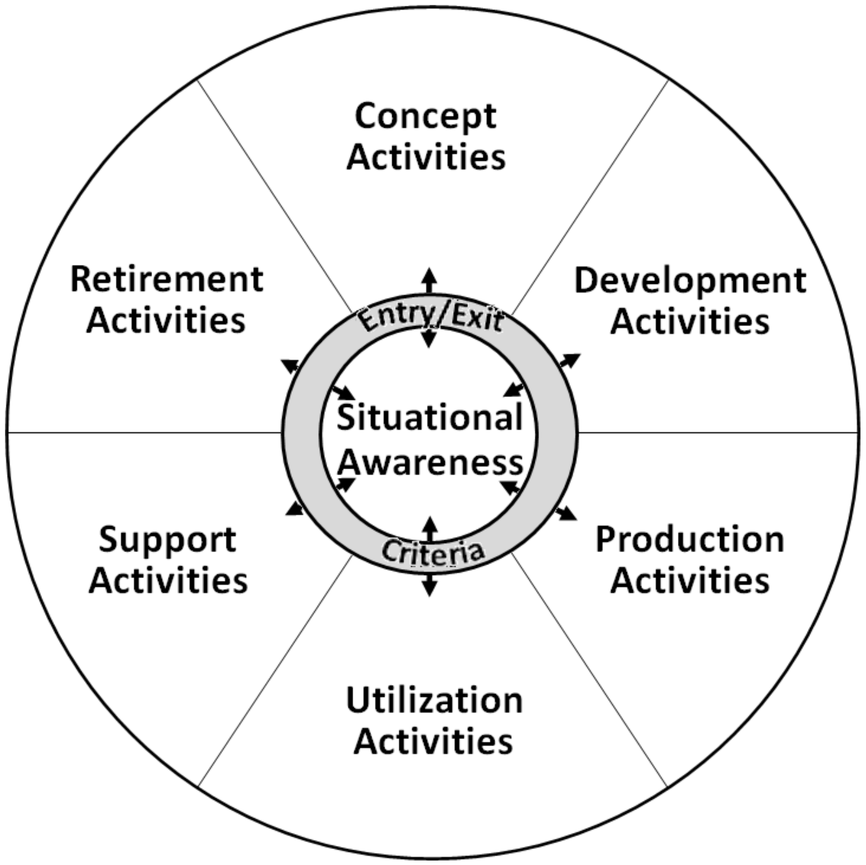
- **What should awareness activities look for?**
- **How fast should knowledge cross stage boundaries?**
- **What are appropriate entry and exit criteria?**
- **How should the above be implemented?**

Value

- **What you see is what you get**
- **A clear mind-set reflection**

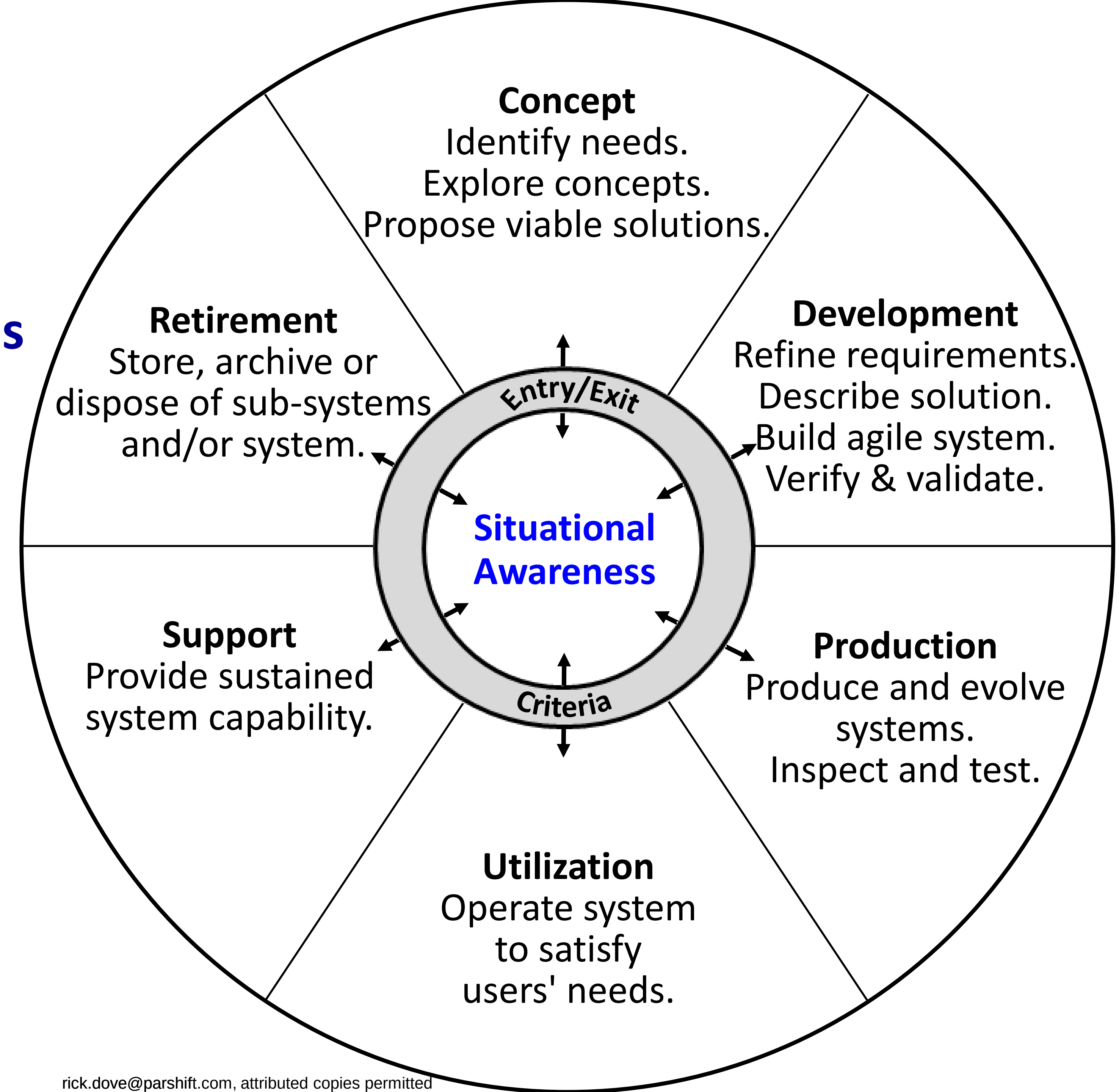
Challenge

- **Text to accompany the depiction as a poster**



Agile SE
Life Cycle Model

Situational Awareness Engages Other Stages and Tasks Asynchronously Concurrently Experimentally Incrementally Iteratively



References and Additional Info

Knowledge Management, Response Ability, and the Agile Enterprise. Dove, R. 1999. Journal of Knowledge Management, Vol 3 No.1, pp. 18-35. March 1. www.parshift.com/s/990301JKM-KnowledgeManagement-RA-AE.pdf

Introduction to the Agile Systems Engineering Life Cycle MBSE Pattern. Schindel, W., R. Dove. 2016. Proceedings International Symposium. International Council on Systems Engineering. Edinburgh, Scotland, July 18-21. www.parshift.com/s/160718IS16-IntroToTheAgileSystemsEngineeringLifeCycleMBSEPattern.pdf

Innovation, Risk, Agility, and Learning, Viewed as Optimal Control & Estimation. Schindel, W. 2017. Proceedings International Symposium. International Council on Systems Engineering. Adelaide, Australia, July 17-20.

Agile Systems Engineering Life Cycle Model for Mixed Discipline Engineering. Dove, R., W. Schindel. 2019. Proceedings International Symposium. International Council on Systems Engineering. Orlando, FL, July 20-25. www.parshift.com/s/ASELCM-05Findings.pdf

The Best of Both Worlds: Agile Development Meets Product Line Engineering at Lockheed Martin. Gregg. S., R. Sharadin, P.C. Clements. 2019. International Council on Systems Engineering. Insight Vol 22, No 2. August.

Systems Engineering the Conditions of the Possibility (Towards Systems Engineering v2.0). Willett, K.D. 2020. Proceedings International Symposium. International Council on Systems Engineering. July 20-22. www.researchgate.net/publication/343306942_Systems_Engineering_the_Conditions_of_the_Possibility_Towards_Systems_Engineering_v20

[Case Study:] Agile systems engineering process features collective culture, consciousness, and conscience at SSC Pacific Unmanned Systems Group. Dove, R., W. Schindel, C. Scrapper. 2016. Proceedings International Symposium. International Council on Systems Engineering. Edinburgh, Scotland, July 18-21. www.parshift.com/s/ASELCM-01SSCPac.pdf.

Case Study: Agile Hardware/Firmware/Software Product Line Engineering at Rockwell Collins. Dove, R., W. Schindel, R. Hartney. 2017. Proceedings 11th Annual IEEE International Systems Conference. Montreal, Quebec, Canada, April 24-27. www.parshift.com/s/ASELCM-02RC.pdf

Case study: agile SE process for centralized SoS sustainment at Northrop Grumman. Dove, R, W. Schindel, M. Kenney. 2017. Proceedings International Symposium. International Council on Systems Engineering. Adelaide, Australia, July 17-20. www.parshift.com/s/ASELCM-03NGC.pdf.

Case Study: Agile Systems Engineering at Lockheed Martin Aeronautics Integrated Fighter Group. Dove, R., W. Schindel, K. Garlington. 2018. International Council on Systems Engineering, International Symposium, Washington, DC, July 7-12. www.parshift.com/s/ASELCM-04LMC.pdf

Full Series

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:
https://connect.incose.org/Library/Webinars/Pages/INCOSE-Webinars.aspx

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