Webinar

Agile SE Processes 202: Mixed Discipline Continuous Integration

18-Sep-2019

Rick Dove

Anthem, AZ, dove@parshift.com, 575-770-7101
CEO/CTO, Paradigm Shift International
Adjunct Professor, Stevens Institute of Technology
Instructor, California Institute of Technology
Chair: INCOSE WG for Agile Systems & Systems Engineering

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

At core agile systems engineering is about learning and application of that learning continuously to work-in-process – with the objective of minimizing rework.

Agile software development relies on short-cycle development iterations to accomplish this learn-as-you-go affordably.

Agile mixed-discipline development can't keep pace with software's short cycle iterations, has cross-discipline integration issues that are typically late in discovery, and attempts to synchronize multiple suppliers over long-cycle integration and test events.

This webinar discusses the needs, benefits, and nature of mixed-discipline, mixed-supplier Continuous Integration Platforms (CIPs); DevOps collaboration; set-based design as potentially CIP-enabled; and incremental considerations for creating project-specific CIPs affordably.

Prelude:
Eight Findings (so far) from the INCOSE ASELCM Project

(ASELCM: Agile Systems Engineering Life Cycle Model)

1. **Agile SE Life Cycle Framework**
2. **ASELCM Operational Pattern**
3. **Problem-Space Characterization**
4. **Operational Principles**
5. **Concept of Information Debt**
6. **Response Requirements**
7. **Stakeholder Engagement**
8. **Continuous Integration Platform**

Prelude:
Eight Findings (so far) from the INCOSE ASELCM Project

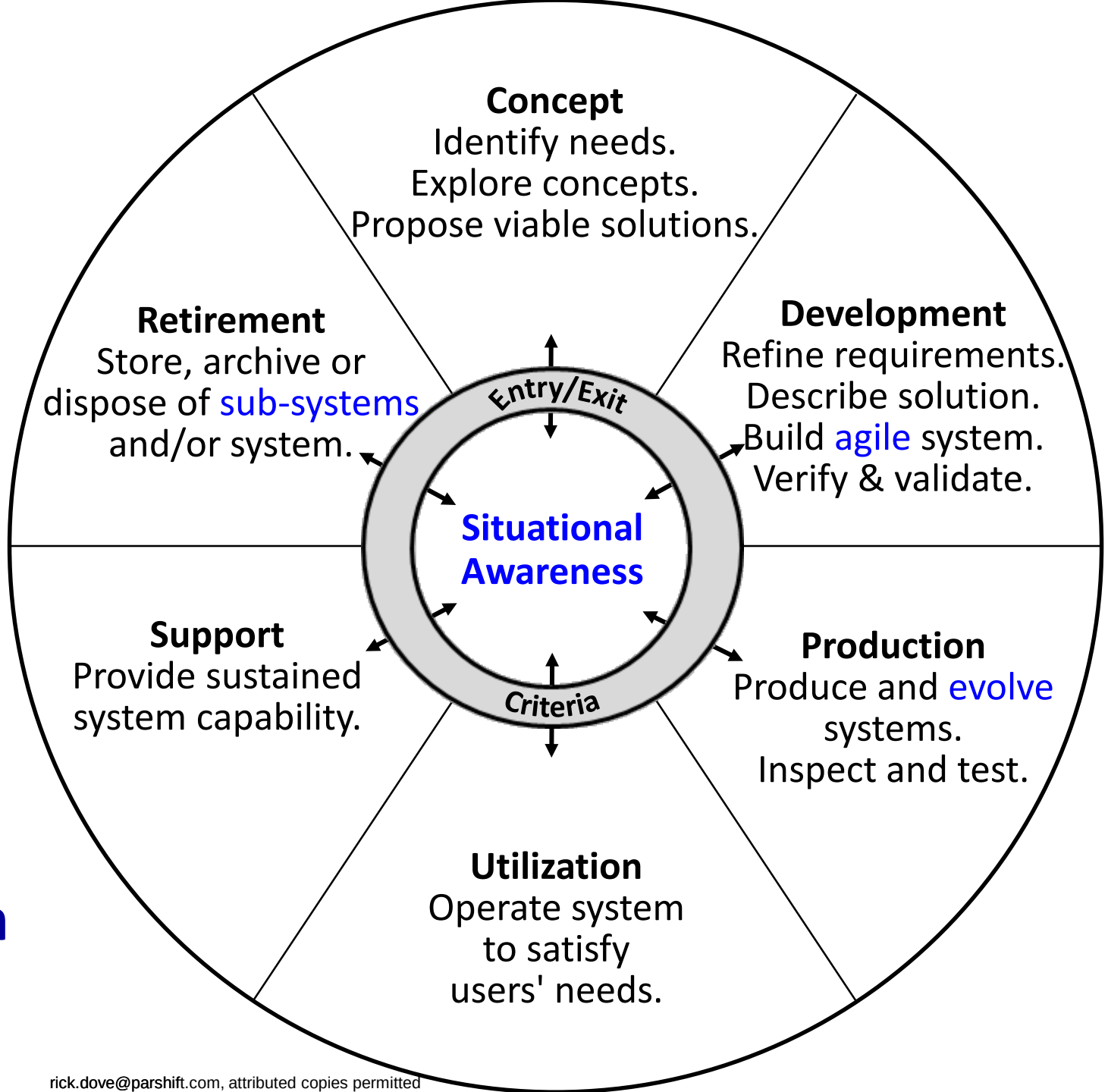
(ASELCM: Agile Systems Engineering Life Cycle Model)

1. Agile SE Life Cycle Framework
2. ASELCM Operational Pattern
3. Problem-Space Characterization
4. Operational Principles
5. Concept of Information Debt
6. Response Requirements
7. Stakeholder Engagement
8. Continuous Integration Platform

1. Agile SE Life Cycle Framework

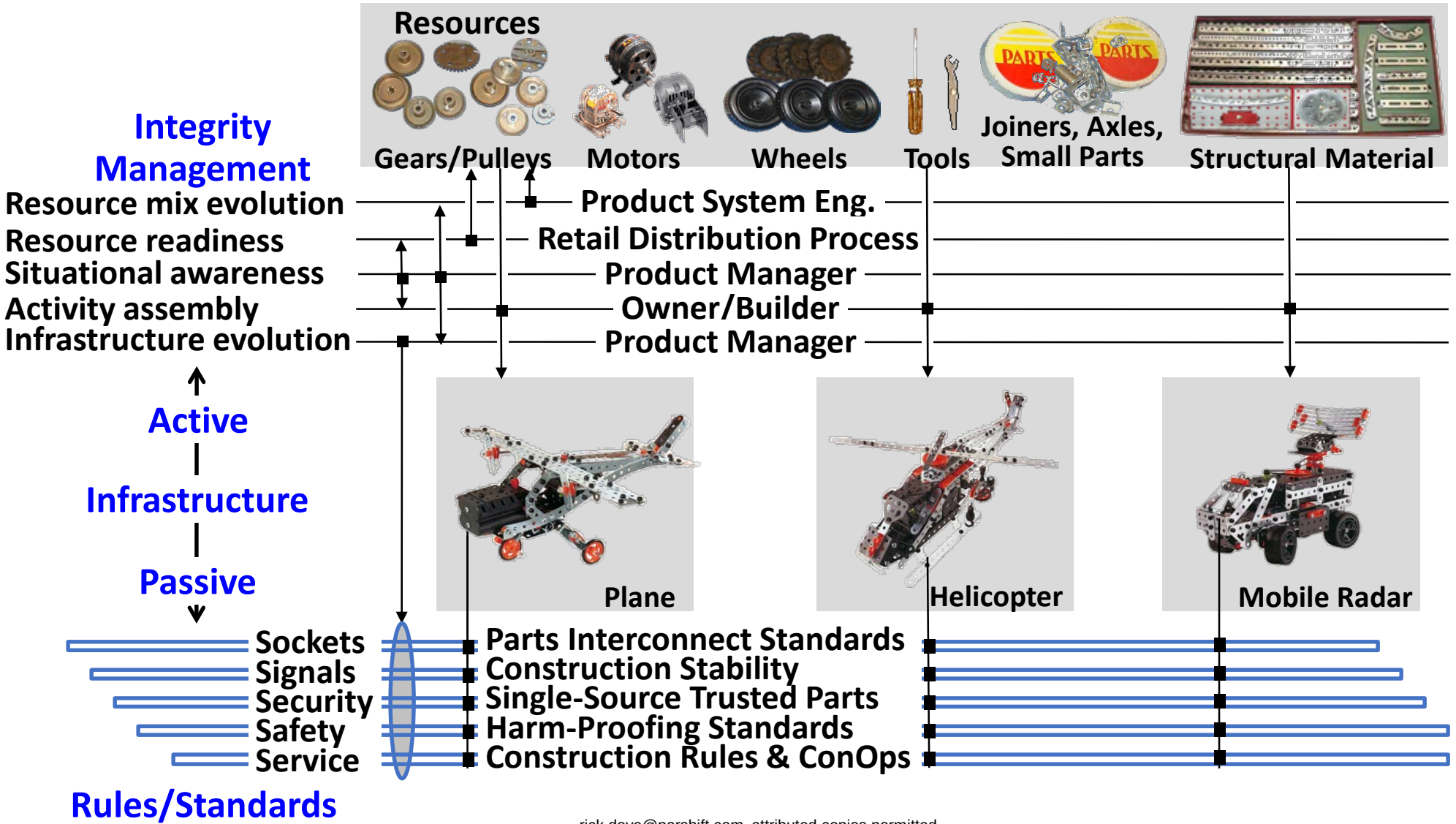
Asynchronous/Concurrent Stages.
Consistent with
ISO/IEC/IEEE 24748-1:2018

Situational Awareness Engages System Evolution Stages and Tasks

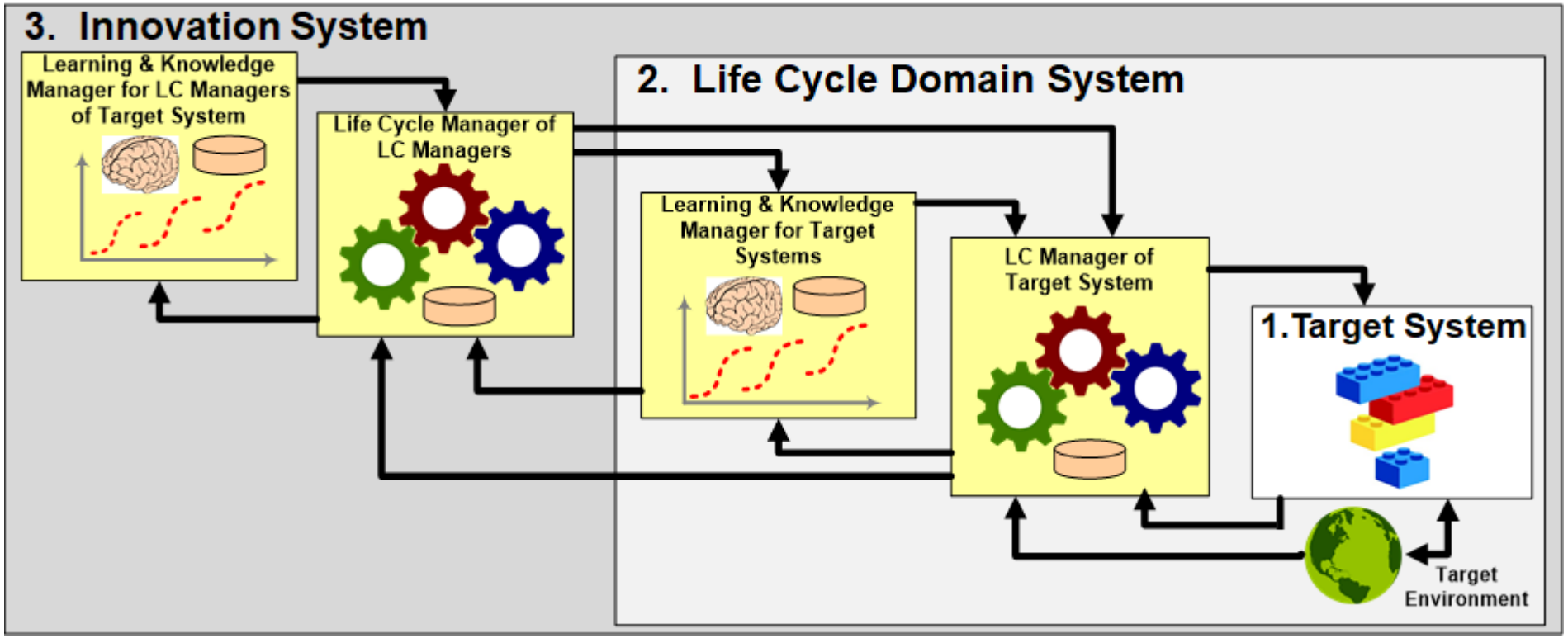


Agile Architecture Pattern (AAP) – Enables Agility
Notional Concept: System Construction/Augmentation Kit

Details in www.parshift.com/s/140630IS14-AgileSystemsEngineering-Part1&2.pdf



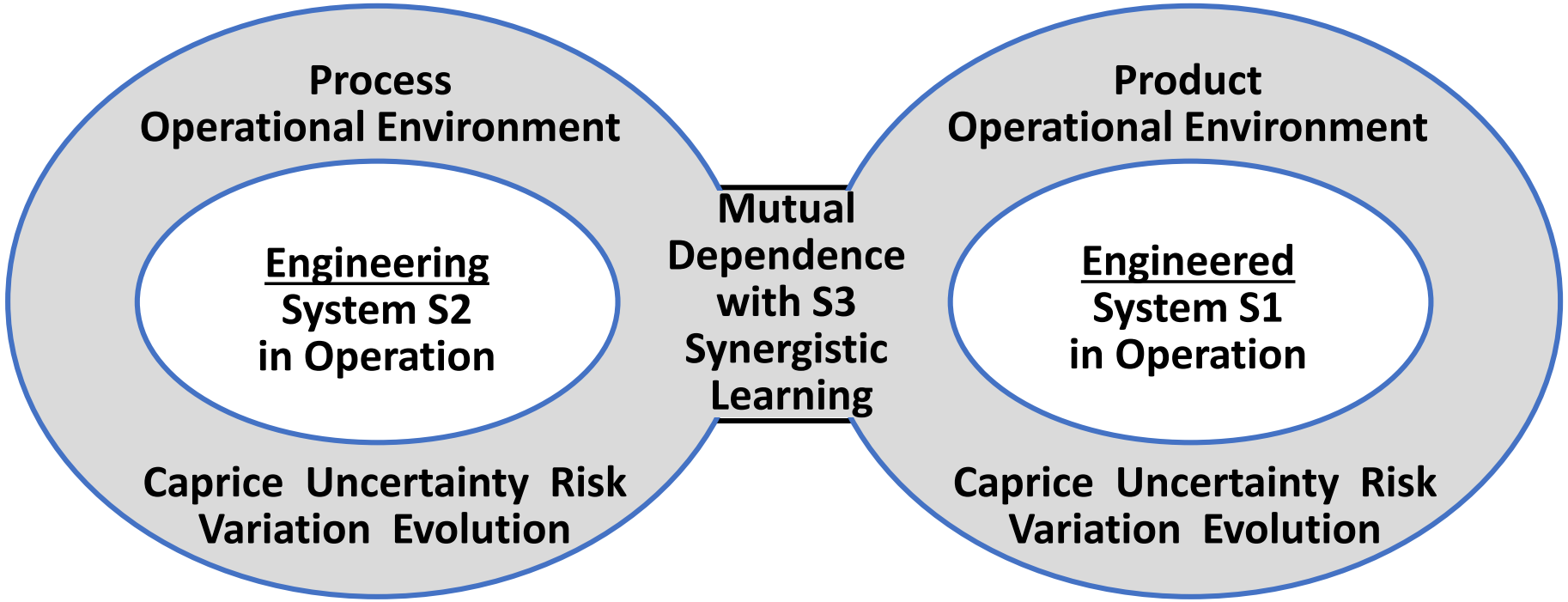
2. ASELCM Operational Pattern – Three Concurrent Systems



- System-1 is the target system under development.
- System-2 includes the basic systems engineering development and maintenance processes, and their operational domain that produces System-1.
- System-3 is the process improvement system, called the system of innovation that learns, configures, and matures System-2.

The Innovation System is responsible for situational awareness and evolution, the provider of operational agility. Intent is continuous, not episodic, info flow.

S1 and S2
with synergistic dependencies



You can't have
an agile engineering process
if it doesn't engineer an agile product
(and vice versa)

3. Problem Space Characterization

Internal and external environmental forces that impact process and product as systems

Caprice: unanticipated system-environment change
(randomness among unknowable possibilities)

Uncertainty: kinetic and potential forces present in the system
(randomness among known possibilities with unknowable probabilities)

Risk: relevance of current system-dynamics understanding
(randomness among known possibilities with knowable probabilities)

Variation: temporal excursions on existing behavior attractor
(randomness among knowable variables and knowable variance ranges)

Evolution: experimentation and natural selection at work
(relatively gradual successive developments)

The goal of agile systems engineering is S2 and S1 compatibility with their CURVEd environments.

The general CURVE shown here is applicable to both S2 and S1.

S2 and S1 have cyber-physical-social dimensions.

General SE CURVE
Caprice
• Survivability (i.e., current order compatibility) • Occurrence and nature of emergent behavior • Game-changing technologies • Availability of symbiotic social relationships
Uncertainty
• Relevance (i.e, appropriate to current desires) • Cohesion in the greater SoSs (multiple) • Integrity and symbiosis of social relationships.
Risk
• Viability (i.e., capable of working successfully) • Cohesion among constituent parts
Variation
• Operational environments • Social compatibility
Evolution
• Toward more operating environment complexity • Toward more SoI complexity • Toward shorter SoI static viability • Toward new technology options • Toward new malevolent threats to viability • Toward greater social involvement.

4. Operational Principles

Sensing (observe, orient)

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

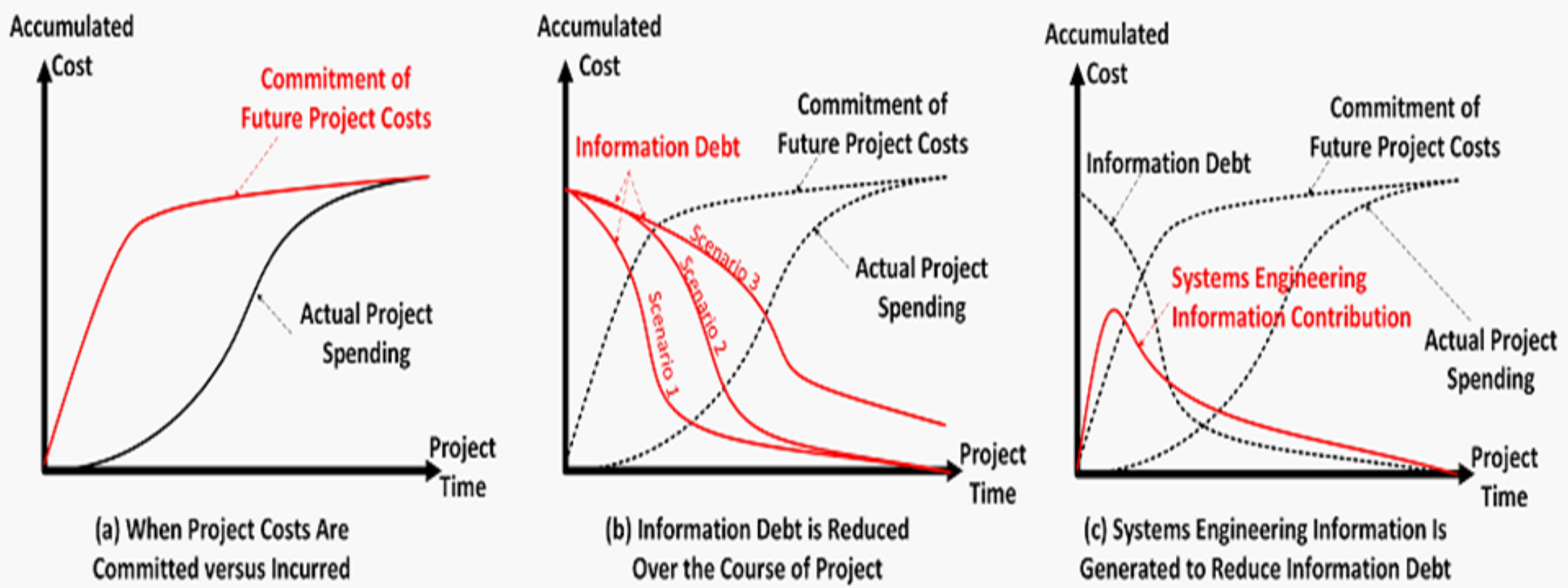
Responding (decide, act)

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

Evolving (improve above with more knowledge and better capability)

- Experimentation (variations on process ConOps/OpsCon)
- Evaluation (internal and external judgement)
- Memory (evolving culture, response capabilities, process ConOps/OpsCon)

5. Concept of Information Debt



Future costs of a project become committed early by SE decisions. One of the traditional arguments for early stage SE investment.

Will development end with outstanding information debt – a "working system" with impaired sustainment and evolution agility?

SE information must be generated (e.g., reqs, architectures, risk assessments, etc.) early enough in the project.

6. Response Requirements

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

Here We Focus on Two Findings

7. **Stake Holder Engagement**
8. **Continuous Integration Platform**

7. Stake Holder Engagement

Developers	Subcontractors	Security Engineers
Operators	Producers	Maintainers
Customers	End Users	Management

Three typical 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. (Wikipedia)

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. (Wikipedia)

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. (Wikipedia)

(Wikipedia access 11-Sep-2019)

Engagement can’t be forced.
Engagement should not be perceived as a time-eating task.
Engagement must provide experiential and take-away value to all involved.

DevOps concepts offer a relevant discussion path.

What is DevOps?

Excerpts from The 7 Principles of DevOps and Cloud Applications. 2015. Gerardo Dade at SolarWinds
www.slideshare.net/SolarWinds/the-7-principles-of-devops-and-cloud-applications

"DevOps is [was born as] a software development practice where development and operation teams work together, taking the intelligence of how an application runs to inform and improve how the application is being built, in a rapid iterative process.

There is no official definition of DevOps, there are many.

DevOps allows to respond faster to customers, fix things faster, produce with more quality. Quality includes performance, security and less bugs.

However, you don't 'do' DevOps, it's not a process, it is a business practice, an approach."

Thus – a look at underlying principles will establish a foundation for an approach.

Seven Principles of DevOps

(slightly modified from Dade's software-focused text)

DevOps is a development practice where development and operation teams work together, taking the intelligence of how a product runs to inform and improve how the product is being built, in a rapid iterative process.

1: End User Focus – It's all about the total end user experience.

2: Collaboration – Develop, Test, and Demo/Run – an integrated process.

3: Performance Orientation – Performance is measured all the time, everywhere.

4: Development Speed – Short, asynchronuous iterative processes accelerate innovation and learning.

5: Service Orientation – Producers and maintainers operate a service to the user.

6: Automation and Repitition – Software configured/controlled regression testing.

7: Monitor Everything – Measure and display what matters.

How do you put these principles into effective practice for mixed discipline systems engineering?

8. Continuous Integration Platforms

Agile SE processes deal with changing knowledge and environment

- **They learn and employ that learning during SE process operation**
- **They modify/augment product-development work-in-process, enabled by an SoI Agile Architecture Pattern (AAP)**

Agile SW development relies on AAP for SoI structure – commercially available

- **Program code development employs an object-oriented development platform (e.g., C++, Java, Eclipse)**
- **Web code development employs a loosely-coupled modularity inherent with hyperlinked web-pages**

Agile HW development doesn't have off-the-shelf integration platforms

- **Proprietary Product-Line-Engineering employs AAP**
- **Proprietary Open System Architecture (OSA) employs AAP**
- **Proprietary Live-Virtual-Constructive employs AAP**

Agile Systems Engineering Goals

produce an innovative result,

produce a “success-assured” result,

produce a sustainable result,

rapidly.

Rework is the bane of Rapid.

Continuous Integration Platform
for mixed-discipline, mixed-supplier projects

Need: Minimize rework.

Intent: An agile Continuous Integration Platform (CIP),
that enables and facilitates…

- An asynchronous continuous test capability (less rework).
- Early detection of integration issues (less rework).
- WIP feedback demos to users/customers/management/suppliers (less rework).
- DevOps/DevSecOps/IPT/CE collaborative development interaction (less rework).
- Alternative/prototype experimentation (less rework).
- A set-based knowledge-development test stand (less rework).

Less rework is a value common to all engineering disciplines.

Boeing's F-22 team accelerates avionics modernization with new approach

The F-22 Agile Integration Lab interlinks the 757 Flying Test Bed with a ground-based test and evaluation facility. This provides the capability to test several avionics software versions dynamically during a single six-hour flight.

"It's not a stretch to say we can accomplish in a day what used to take a month."

They upload multiple software versions into the flying test bed's workstations, test them during a single six-hour flight, land and park the airplane beside the ground-based lab, reconnect the umbilical, and certify the software updates in concert with F-22 systems that don't need to fly on the test bed: e.g. weapons, engines and flight controls.

They use existing assets to provide more robust test and evaluation capability at lower cost and shorter delivery time.

Cantwell, Doug. 2008. F-22 Team Accelerates Avionics Modernization with New Approach. Boeing Frontiers, Goin' Agile.
www.boeing.com/news/frontiers/archive/2008/july/i_ids05.pdf

SpaWar Continuous Integration Platforms

RaDER
Reconnaissance and Detection Expendable Rover

Two of the Integration Platforms

(for 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

Rockwell Collins Continuous Integration Platform

Cedar Rapids military radio projects

The focus is on the evolution of firmware-containing circuit cards needed by software development for incremental testing during sprint iterations, and especially at three-month increment testing events.

Four elements are evolved, to accommodate this:

- **Integrated computing platform (ICP): a Rockwell-built scalable circuit card rack with supporting power and cabling that can accommodate multiple circuit cards, and interface with external devices and computers.**
- **Commercially available system-on-chip prototype boards.**
- **Product Line component inventory: Rockwell-built circuit cards are readily available as either actual end-product reusable cards or sufficiently similar to act as proxies for early software interface testing.**
- **Hardware chassis are either drawn from the product line inventory or developed new, with employment of an inventoried LRU favored for early physical form.**

ICP hardware evolves continuously and asynchronously with software sprints.

Lockheed IFG Continuous Integration Platform

In 2015 IFG was in early experimentation with a CIP concept, called the Agile Non-Target Environment (ANTE).

ANTE systems consist of simulated components, previous re-usable components, wip components, finished components, low-fidelity COTS proxies, IFG software work-in-process, and operators.

Of note: ANTE employs lower-fidelity open-market proxy devices with similar capability but lower performance than what is eventually expected.

Subcontractors are required to provide device simulations to ANTE specs.

ANTE concept was self funded for values they expected and realized.

By mid-2017 ANTE was declared a successful experiment, and had achieved eventual applause in customer feedback that values:

- **Early and incremental demonstration of working concepts.**
- **Early exposure to difficulties in need of attention.**

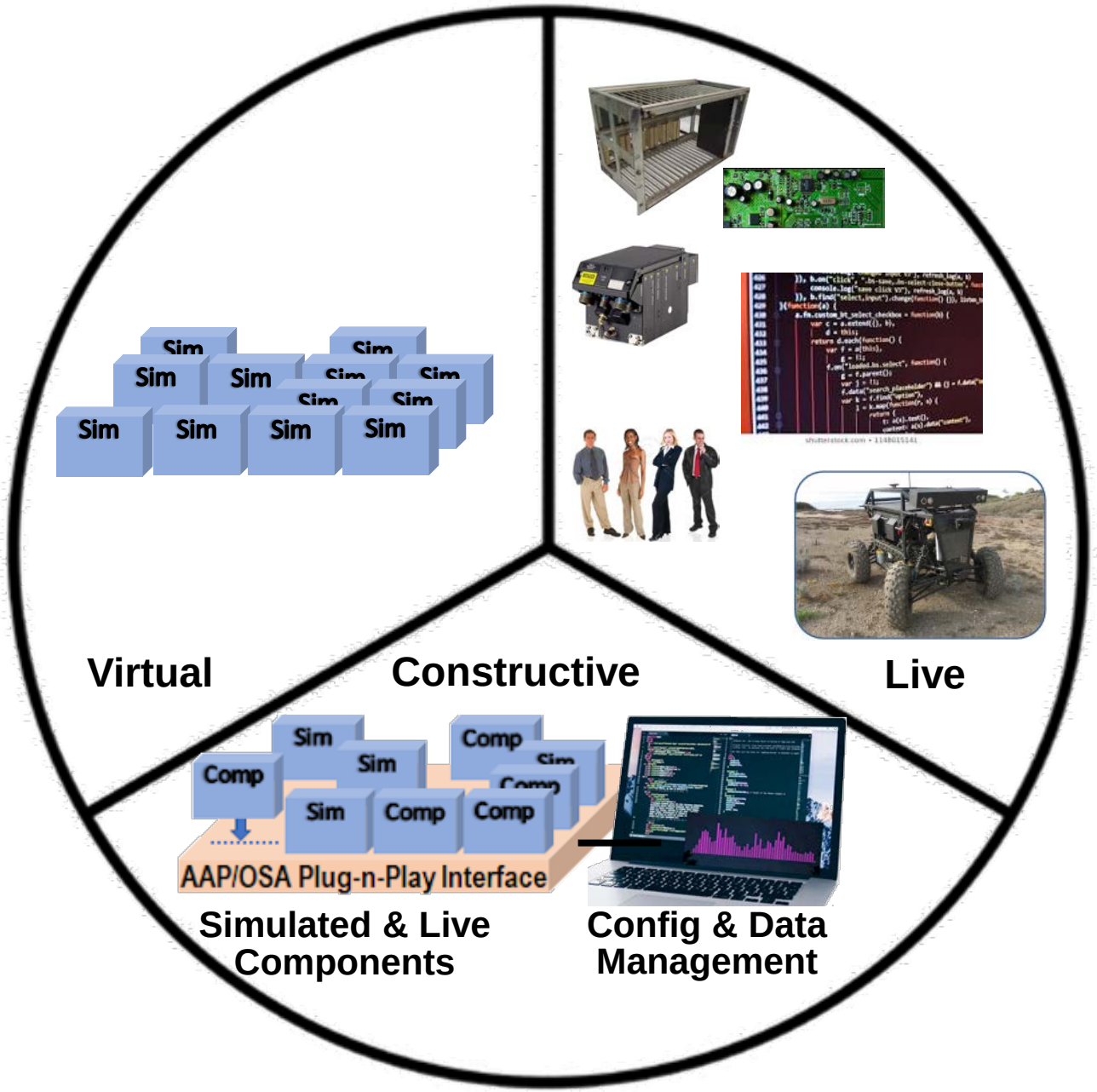
Live Virtual Constructive CIP

Live components
(people, things)
Virtual components
(simulations)
Constructive capabilities
(configuration and data management)

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

An LVC/CIP… demonstration/test/experimental events can occur at any time with the latest instantiation of simulations & components.

Caveat: LVC internet search is dominated by military training applications.



AAP: Agile Architecture Pattern
OSA: Open Systems Architecture

You Can't Buy It – You Have to Design/Build/Evolve It

Start Affordably and Evolve Incrementally

Establish preliminary needs and intents (purposes and goals)

Publish preliminary Agile Architectural Pattern: define/evolve plug-and-play 5s infrastructure interface specs

- **Sockets: physical interconnect**
- **Signals: data/stuff interconnect**
- **Security: trust interconnect**
- **Safety: environment interconnect (relative to internal and external safe operation)**
- **Service: user interconnect (ConOps and OpsCon)**

Consider preliminary simulation stubs (accept requests & data, reply with temporary pro-forma responses)

Consider CIP-operation configuration management (for tests and demos)

Consider performance instrumentation (conflict detection, operational results)

Consider visual stakeholder interface (for collaborative DevOps and wip Demos)

Consider re-use of available SIL equipment, previously developed components, COTS temporary proxies

Consider automated regression testing (retest everything tested before and add new tests cumulatively)

Consider physical as well as cyber interfaces

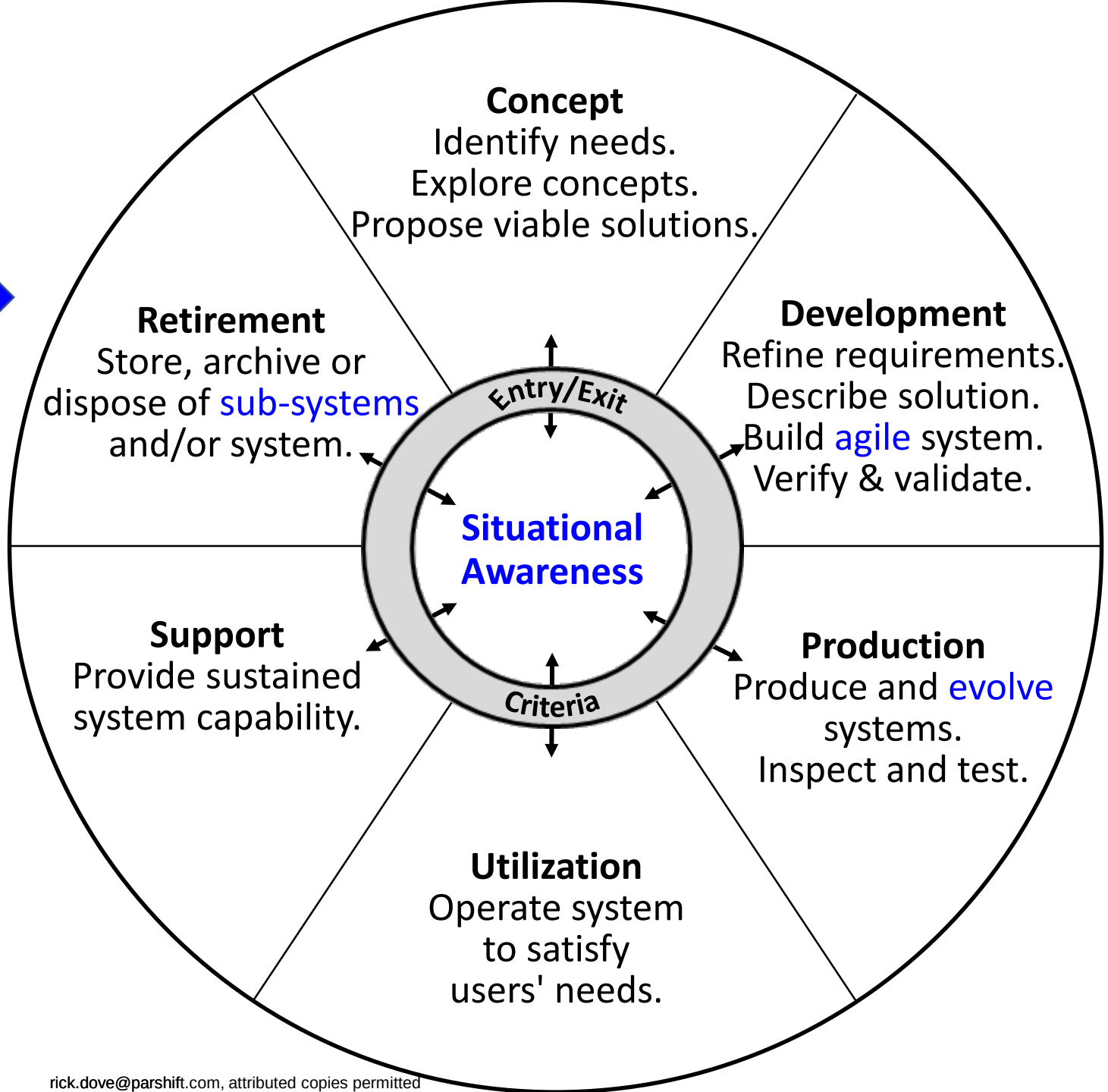
Consider set-based data accumulation over time

Consider ghosting – comparing outputs of one component with a potential replacement

Early & frequent wip stakeholder feedback should illuminate rework cost/time avoidance/reduction.
Build some before and after similar-project comparisons to justify additional CIP investment.

Wrap Up

1. Agile SE Life Cycle Framework
2. ASELCM Operational Pattern
3. Problem-Space Characterization
4. Operational Principles
5. Concept of Information Debt
6. Response Requirements
7. Stakeholder Engagement
8. Continuous Integration Platform



References – LVC for Rapid Prototyping and Testing

Live, Virtual & Constructive Simulation for Real Time Rapid Prototyping, Experimentation and Testing using Network Centric Operations. William J. Bezde, Joel Maleport, Robert Z Olshan. AIAA Modeling and Simulation Technologies Conference, Honolulu, Hawaii, 18-21 Aug 2008. www.incose.org/docs/default-source/midwest-gateway/events/incose-midwest_2009-05-20_bezdek_presentation.pdf

The use of live platforms, real time virtual simulators and constructive entities have been used to provide improved systems engineering requirements and to allow customers to be involved during the entire development and test process.

By applying the techniques described for the hardware, simulation and computer assets, an integrated Live-Virtual-Constructive (LVC) environment was created to provide the ability to "test concepts like as we fight", which has been a multi-service goal for nearly a decade.

Evolution of A Distributed Live, Virtual, Constructive Environment for Human in the Loop Unmanned Aircraft Testing. James R. Murphy, Neil M. Otto. Modelling and Simulation in Air Traffic Management; 14-15 Nov. 2017; London; United Kingdom. https://ntrs.nasa.gov/search.jsp?R=20170011121

Excerpt: As with any development effort, compromises in the underlying system architecture and design were made to allow for the rapid prototyping and open-ended nature of the research. A distributed test environment was developed incorporating Live, Virtual, Constructive, (LVC) concepts. This LVC infrastructure enables efficient testing by leveraging the use of existing assets distributed across multiple NASA Centers.

References and Additional Info

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

[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 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 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