Agile SE Process Features
Collective Culture, Consciousness, and Conscience
at SpaWar Systems Center Pacific Unmanned Systems Group

IS16 Paper: www.parshift.com/s/ASELCM-01SSCPac.pdf

21-October-2016

MITRE Technical Exchange Meeting

Agile Transformation Challenge –Adapting to Rapid Change

Rick Dove

INCOSE Working Group Chair: Agile Systems & Systems Engineering

INCOSE Working Group Chair: Systems Security Engineering

Stevens Institute of Technology and Paradigm Shift International

Context

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

We are now analyzing real-world <u>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.

(INCOSE ASELCM project details at: <u>www.parshift.com/ASELCM/Home.html</u>)

Why Agility Matters

CURVE

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

Capriciousness: unanticipated system-environment change

Uncertainty: kinetic and potential forces present in the system

Risk: relevance of current system-dynamics understanding

Variation: temporal excursions on existing behavior attractor

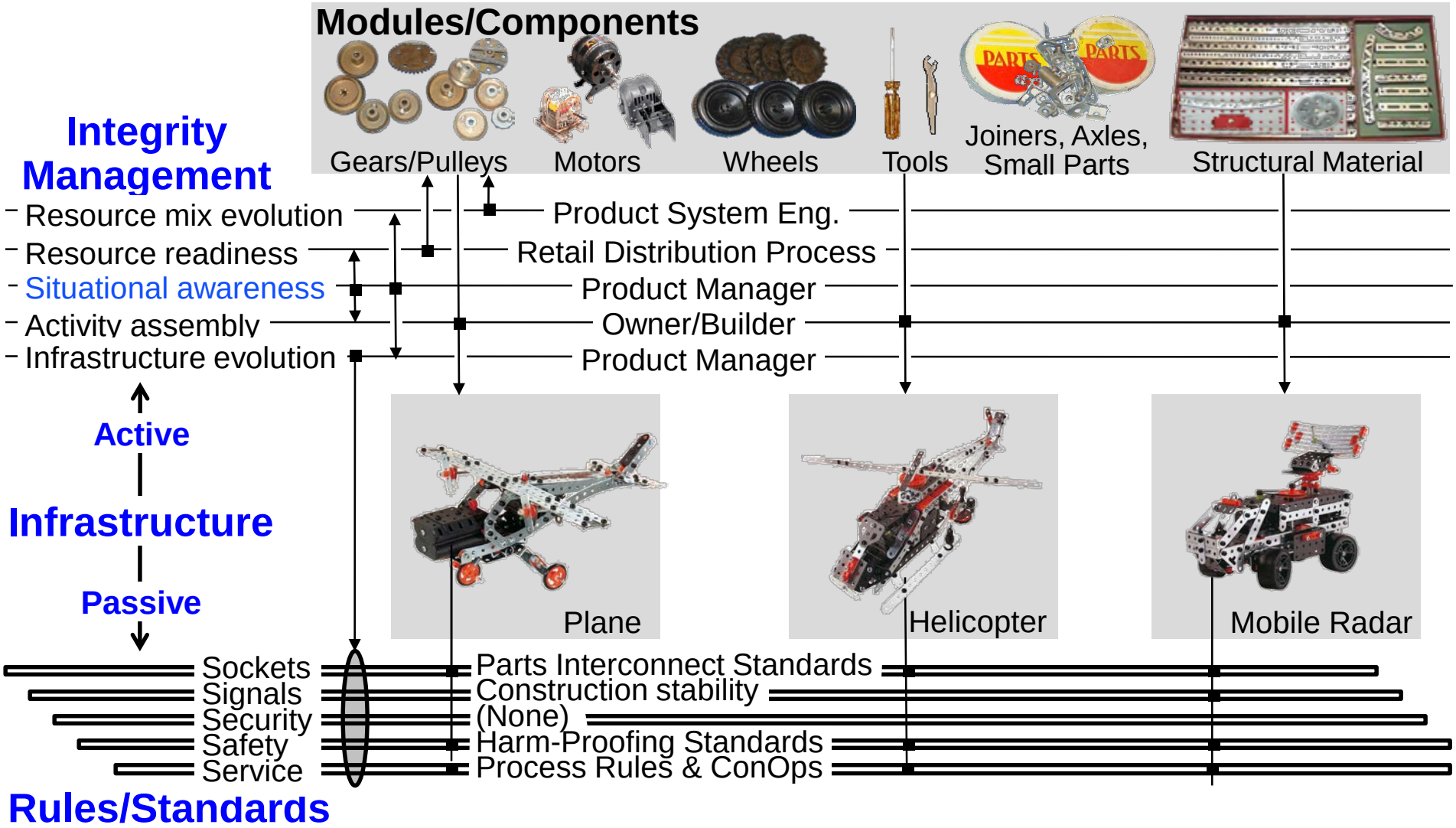
Evolution: experimentation and natural selection at work

(CURVE: formerly known as UURVE, Capriciousness = Unpredictability)

Agile-System Iconic Architecture Pattern (AAP)

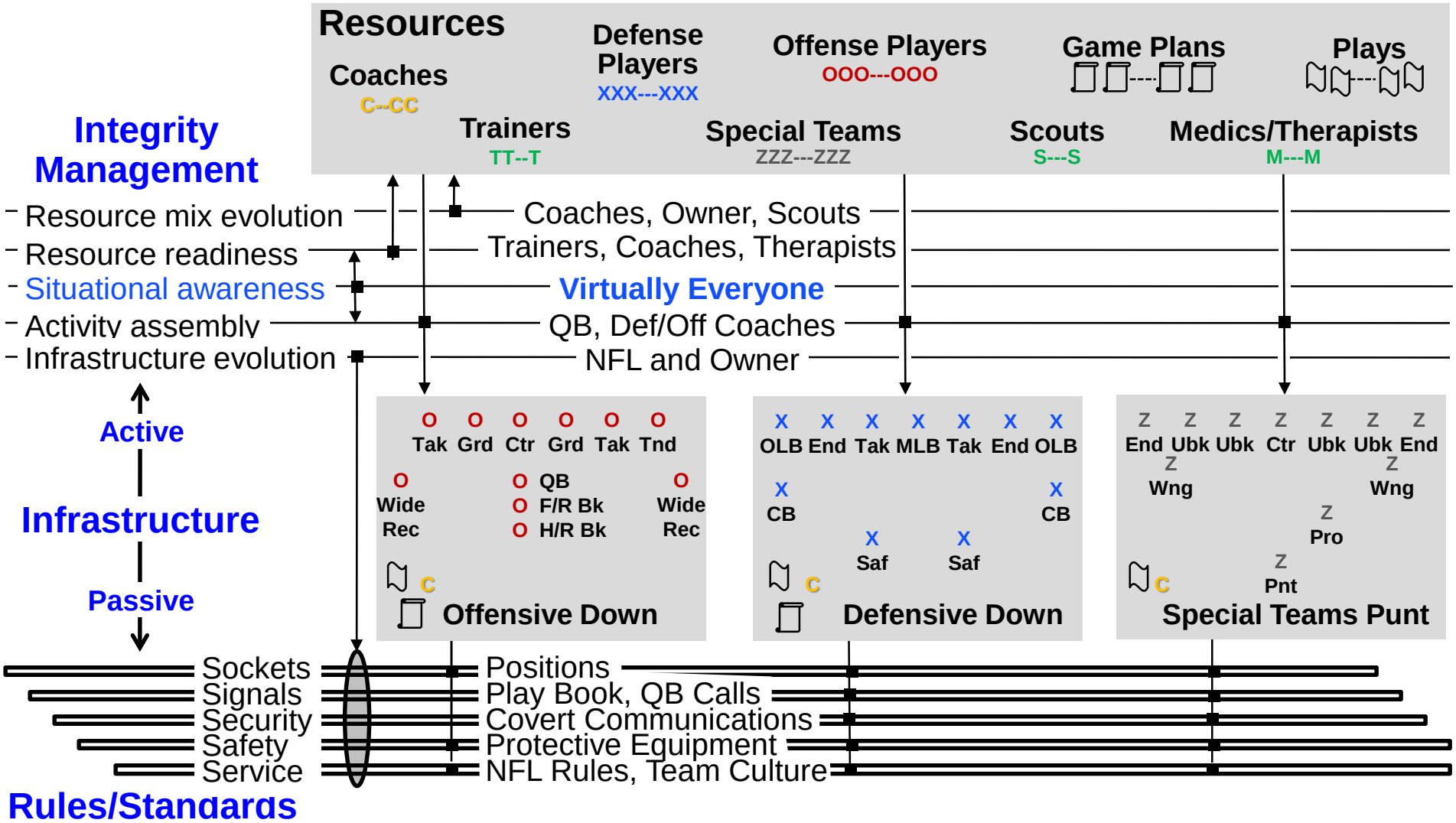
System Response-Construction Kit

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



Agile-Process AAP for USA Football

Drag-and-drop resources in a plug-and-play infrastructure



(a concept example, not exhaustive)

Sustaining Agility Requires …

- Proactive awareness of situations needing responses
- Effective options appropriate for responses
- Assembly of timely responses

Five Agility-Sustaining Responsibilities:

1. Resource Mix Evolution – Who (or what process) is responsible for capabilities of resources appropriate for needs?
2. Resource Readiness – Who (or what process) is responsible for conditions of resources deployable rapidly?
3. Situational Awareness: Who (or what process) is responsible for monitoring, evaluating, and anticipating the operational environment?
4. Activity Assembly – Who (or what process) is responsible for assembling new response configurations as situations require?
5. Infrastructure Evolution – Who (or what process) is responsible for evolving the passive and active infrastructures?

Agility-Enabling Design Principles (RRS)

Prior '90s Work: see INCOSE Webinar, www.parshift.com/s/AgileSystems-103.pdf

Reusable

- **Encapsulated modules**
- **Facilitated interfacing**
- **Facilitated re-use**

Reconfigurable

- **Peer-peer interaction**
- **Deferred commitment**
- **Distributed control & information**
- **Self organization**

Scalable

- **Evolving infrastructure standards**
- **Redundancy and diversity**
- **Elastic capacity**

Agility-Facilitating Behavior Principles (MME)

Current Work: 2015 Discoveries of the INCOSE ASELCM Project (WIP)

Monitoring (Observing, Orienting)

- **External awareness**
- **Internal awareness**
- **Sense making**

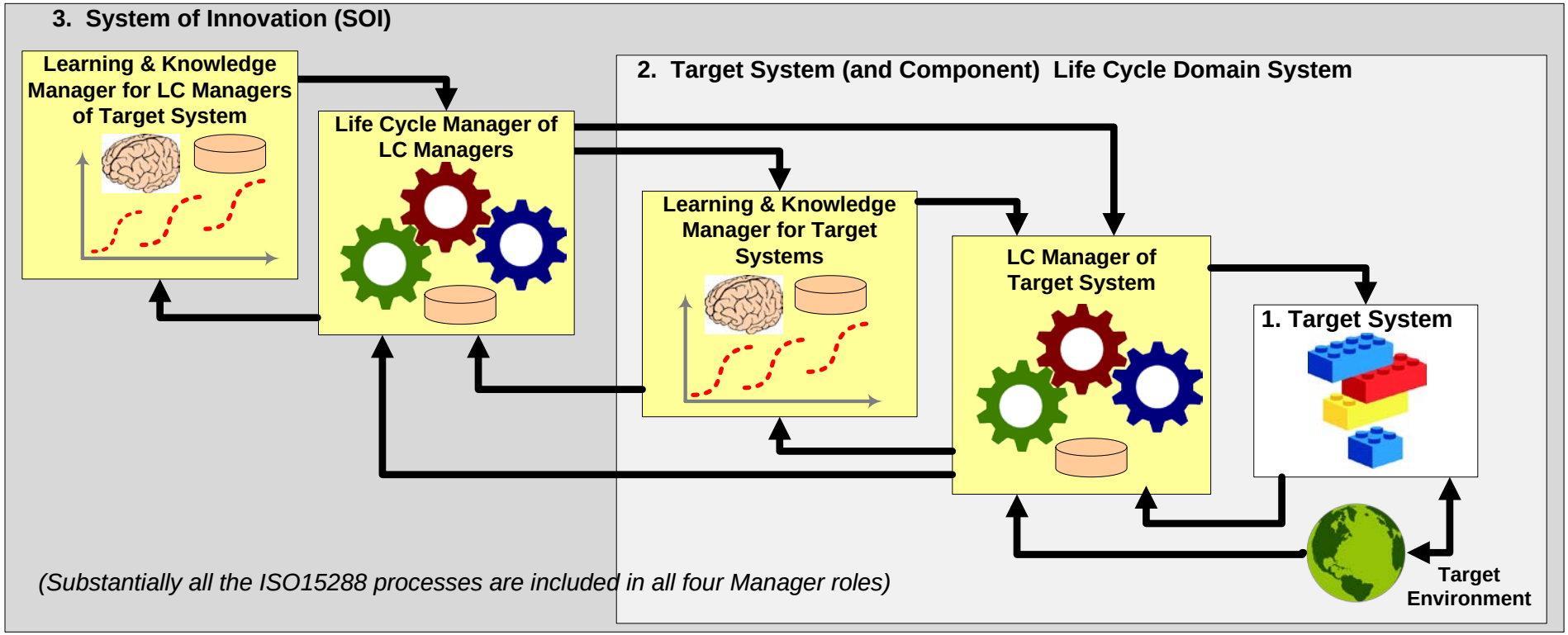
Mitigating (Deciding, Acting)

- **Decision making**
- **Action making**
- **Action evaluation**

Evolving (Improving Above)

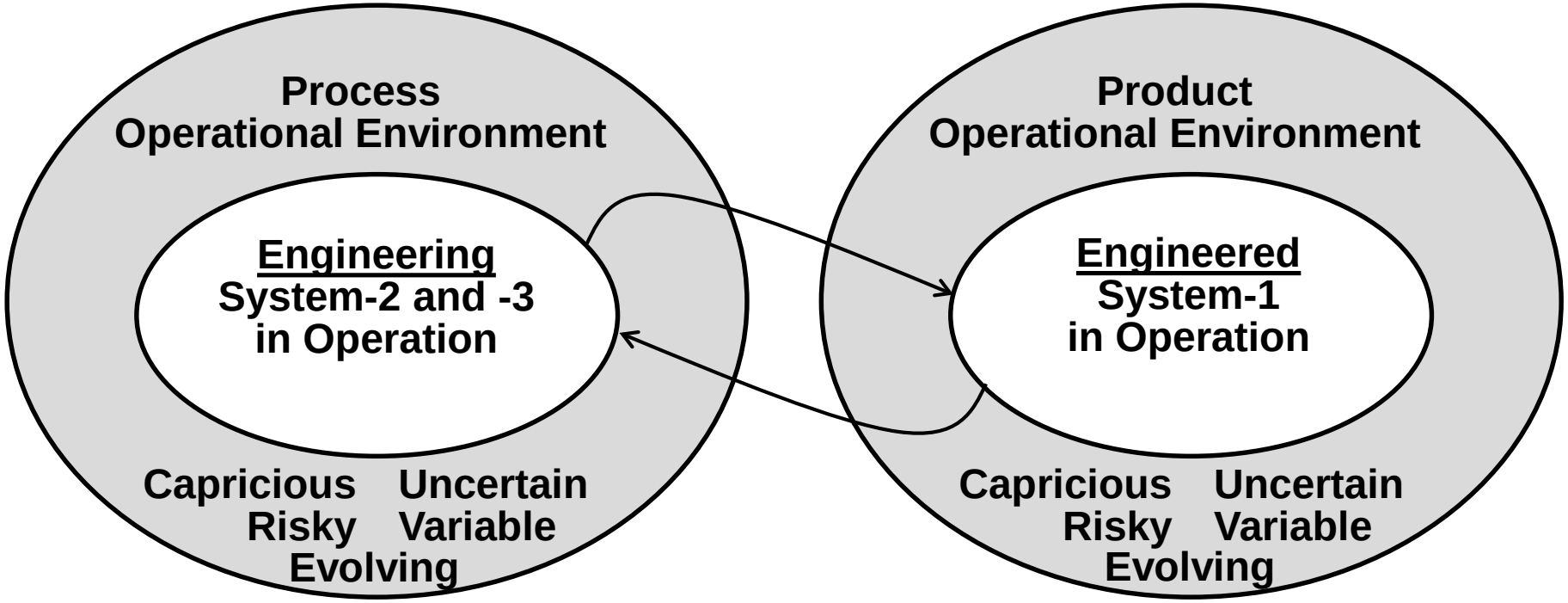
- **Experimentation**
- **Evaluation**
- **Memory**

Agile Systems Engineering Life Cycle Pattern
Encompassing Systems 1, 2, and 3



- **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.**

Two different operational environments defining necessary agile counterpoint for the systems they encompass



It is counterproductive to have an agile development process if you don't have an agile product architecture

SSC-Pac Case Study

IS16 paper: www.parshift.com/s/ASELCM-01SSCPac.pdf

This case study reveals concepts with broad application across domains.

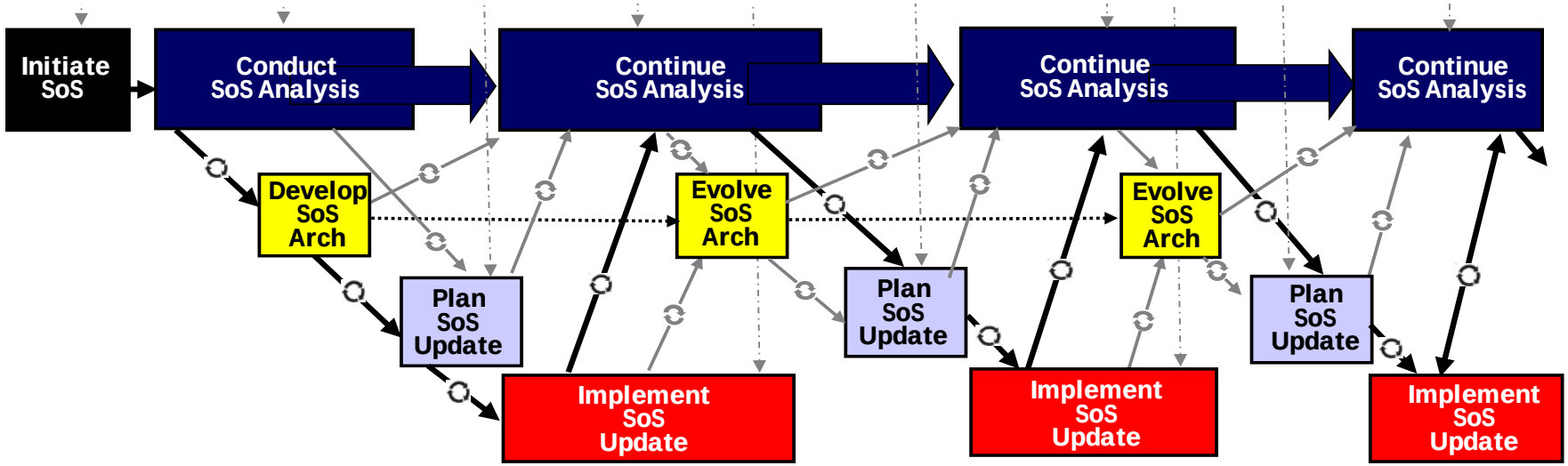
An agile-SE process with 6-month, 4-phase, overlapping "waves":

1. System component development
2. System architecture evolution
3. Capability integration
4. Validation testing

The process capability supports a portfolio of projects, with three years of respected and effective results.

Classic Wave Model, subsequently tailored for the analyzed process

(Scrapper and Dahmann, 2016)



The Process is Successful

…replaced a waterfall process plagued by cost overruns, missed schedules, inadequate development achievement, uncooperative teaming, and poor status visibility.

…orchestrates the interaction of 60-some engineers and managers on the project, plus six external organizations of 4-5 engineers each working on development of functional capabilities.

… encompasses research, development, integration, test, and evaluation of deployable system and component technologies with new capabilities.

… demonstrated effectiveness over three years in lower and predictable costs, on-time capability deliveries, and continual advancements on the overall performance of the systems under development.

… expectations to migrate the process to other programs.

Curve Environment

- **Capriciousness: Unknowable Situations**
  - **Strategic realignment by sponsor**
  - **Engagement and/or availability of personnel & contractors**
- **Uncertainty: Randomness With Unknowable Probabilities**
  - **Feasibility of technical approach and initial designs**
  - **Contracting issues, funding gaps, and budget short falls**
- **Risk: Randomness With Knowable Probabilities**
  - **Failure to meet technical performance measures**
  - **Maturation and integration of required component technologies**
- **Variation: Knowable Variables And Associated Ranges**
  - **Availability of test ranges and test support, and obtaining approvals**
  - **RAM* of vehicle test-beds (vehicle, sensors, computing HW, cables…)**
- **Evolution: Gradual Successive Development**
  - **Technical landscape and insertion of emerging technology**
  - **Programmatic objectives and stakeholder's scope creep**

***RAM: Reliability, Availability, Maintainability**

On Choosing the Agile Wave Model Approach

Scrum learns in 2-4 week sequential development increments, with retrospective analyses of outcomes and process-behavior.

Spiral includes more than software development, necessitating longer learning cycles, with risk reduction as a central cycle-driving theme.

Wave has overlapping learning cycles, decoupling the development effort from the subsequent integration, test, and evaluation efforts.

Decoupling enables back-to-back development increments that don't have to wait for integration, test, and evaluation to start next increment.

Key Take Away:

- **Let an understanding of the problem pull an agile solution that fits.**
- **Don't push a favored agile process … just because.**

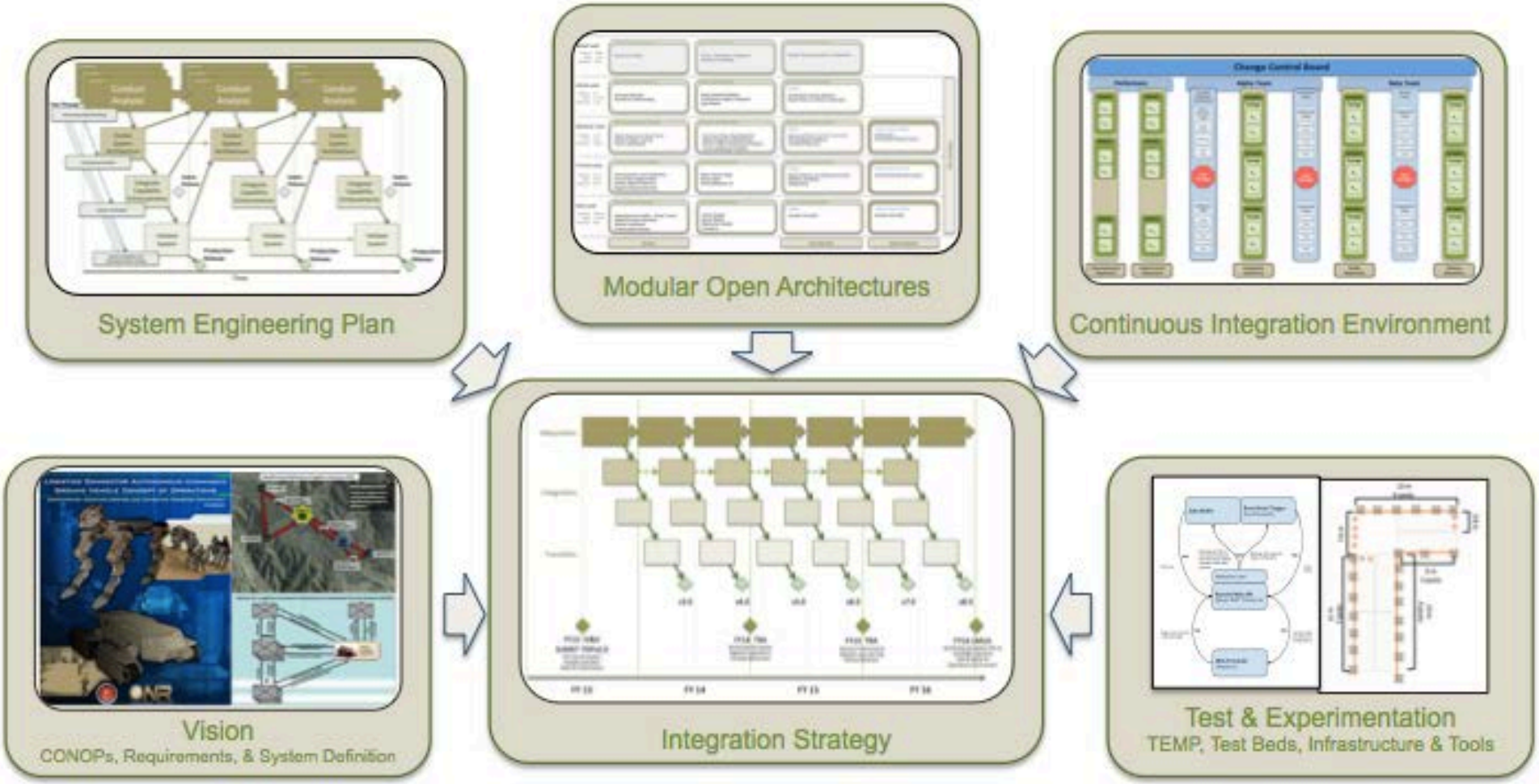
Wave Benefits to this Program

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

There is nothing about the Wave Model that precludes a Scrum approach in the software-development activity, if software developers wish.

The Wave Model approach accommodates tailoring based on size of project, funding levels, and overall project goals.

System-2 Wave, using a modular-component System-1 architecture, lowers costs to all sponsors with re-usable modules across projects.

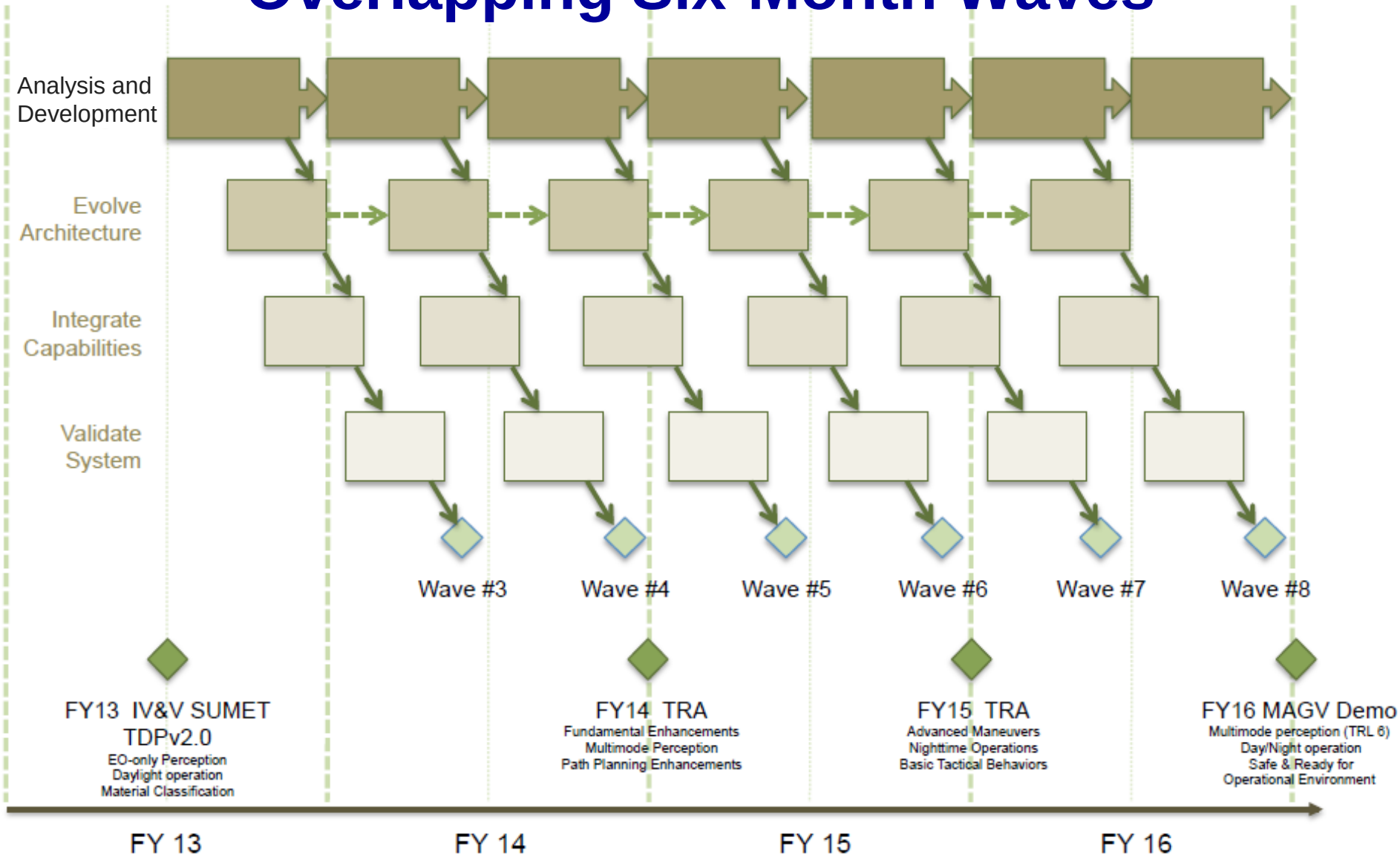


Five elements of the Integration Strategy

- Vision
- Systems Engineering Plan
- Modular Open Product-System Architecture
- Integration Test and Experimentation Master Plan
- Continuous Integration Environment

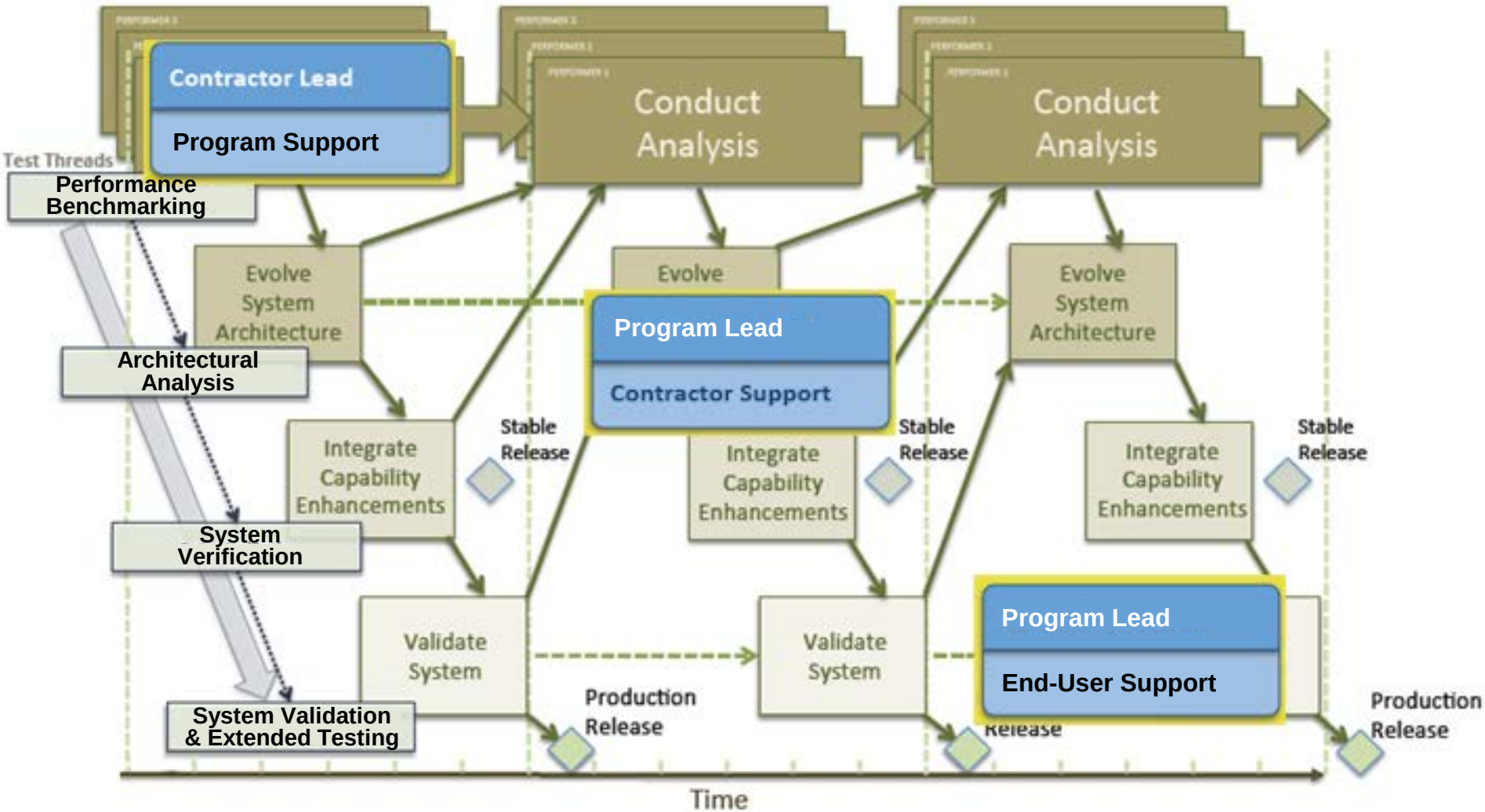
slide credit: Chris Scrapper

Integration Strategy
Overlapping Six-Month Waves



slide credit: Chris Scrapper

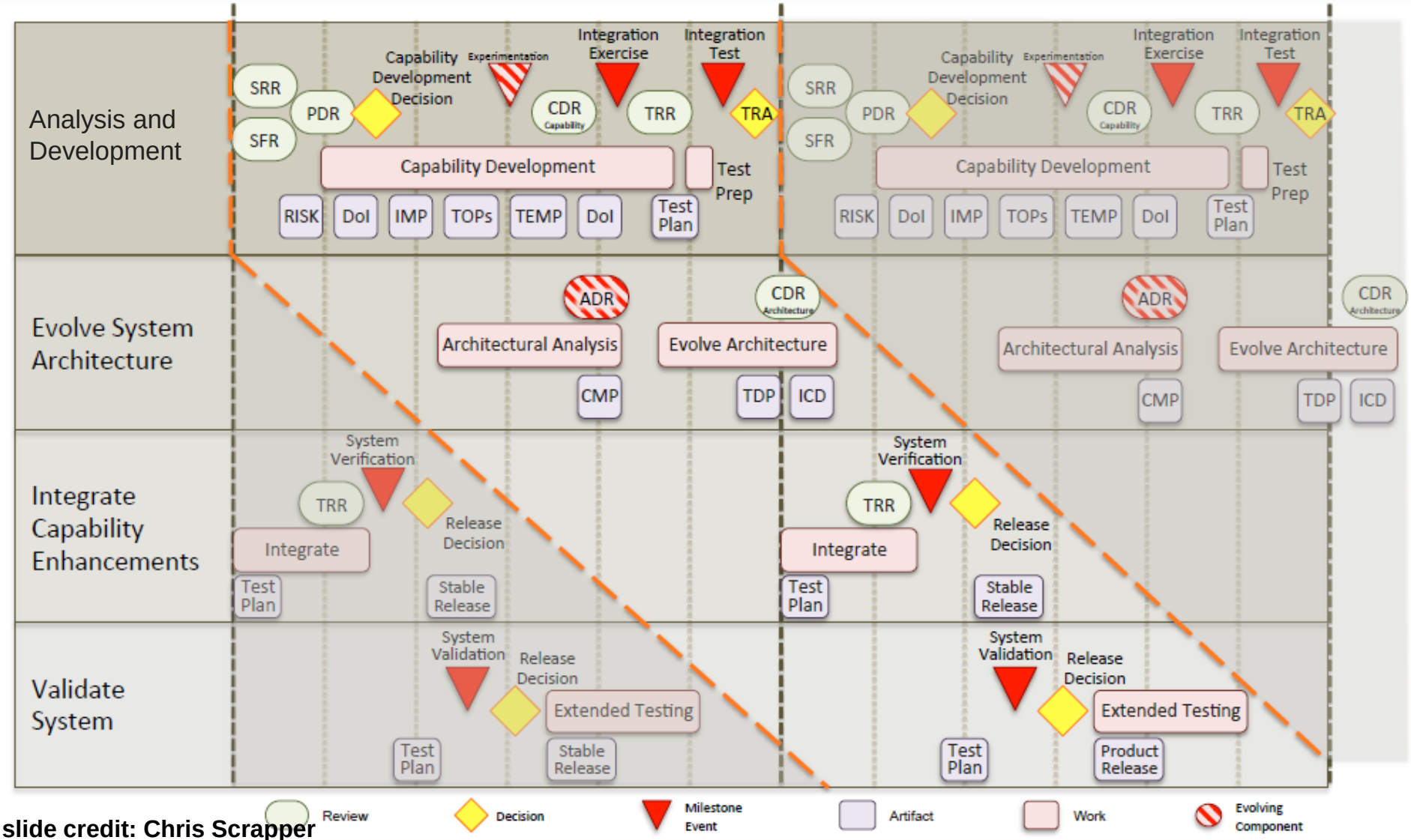
Engaged Integrated Team: Alternate Leads and End-Users



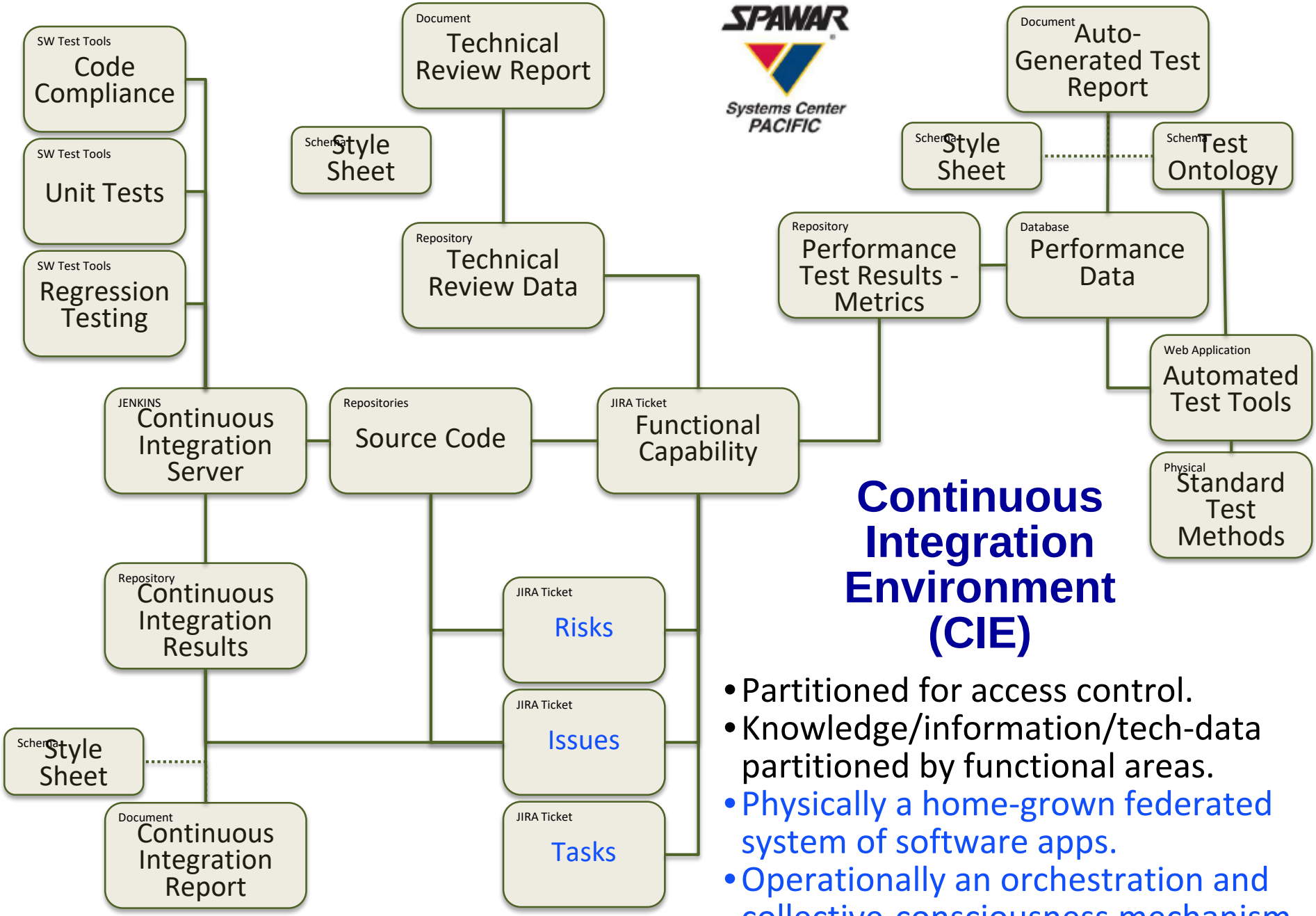
slide credit: Chris Scrapper

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



slide credit: Chris Scrapper



Continuous Integration Environment (CIE)

- Partitioned for access control.
- Knowledge/information/tech-data partitioned by functional areas.
- Physically a home-grown federated system of software apps.
- Operationally an orchestration and collective-consciousness mechanism.

Internal Awareness

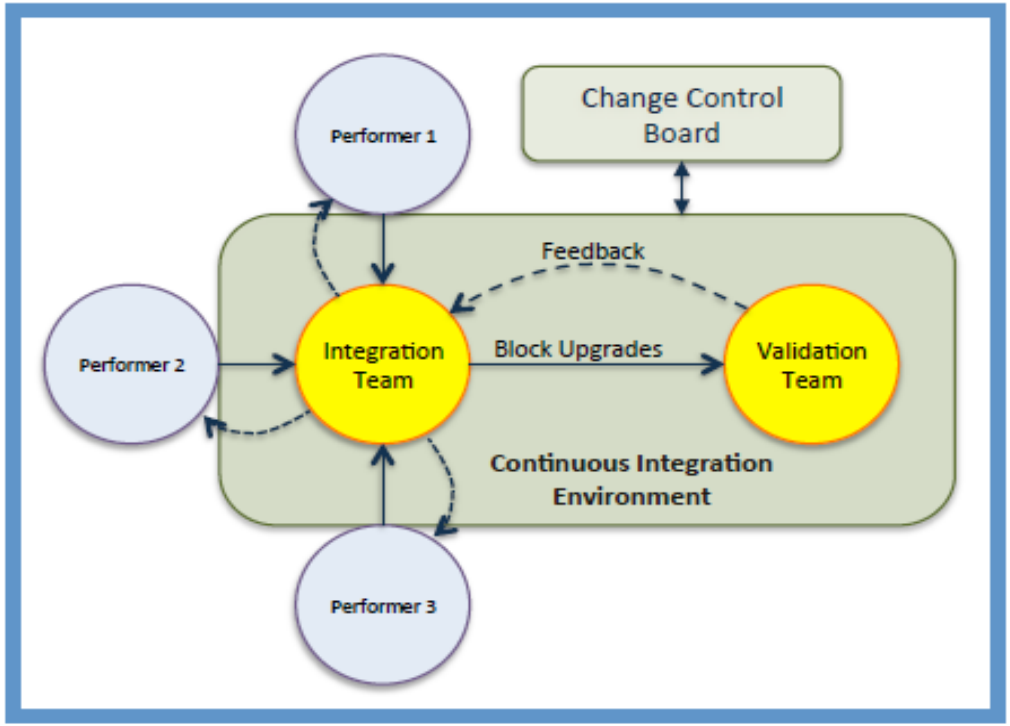
SPAWAR
Systems Center PACIFIC

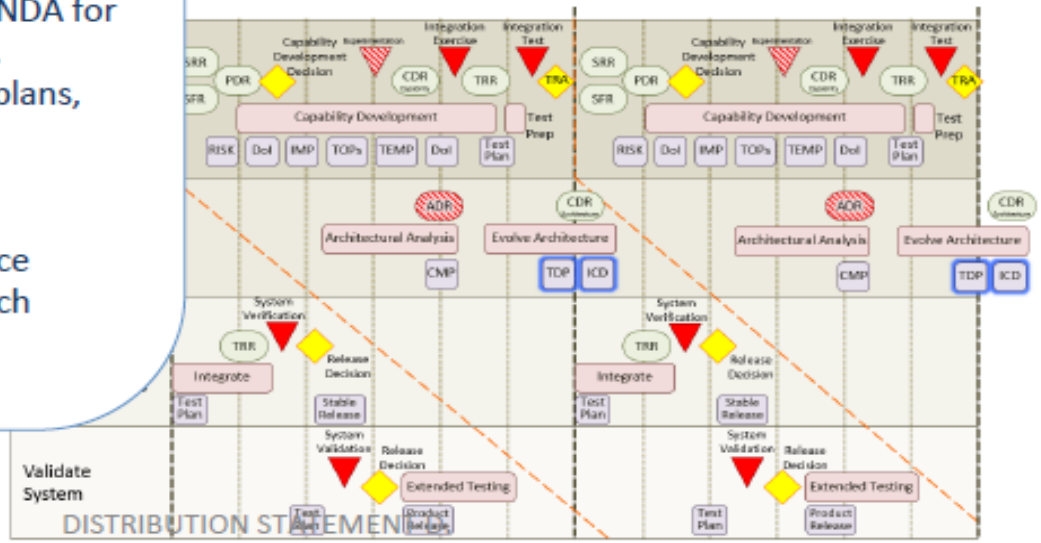
Project Processes:
Information Management Process

Implemented by the Continuous Integration Environment (CIE) that facilitates the continuous integration and feedback between capability developers, integration team, and validation team.

- Best-of-breed collaboration tools (e.g., source code management, issue tracking, and build testing).
- Basis for communication and knowledge transfer between stakeholders.
- Access control and user agreements/NDA for management of Intellectual Property.
- Project documentation (e.g., project plans, bug fixes, change requests, risks, and maintenance reports).

Technical Data Package (TDP) and Interface Control Documentation (ICD) updated each wave.

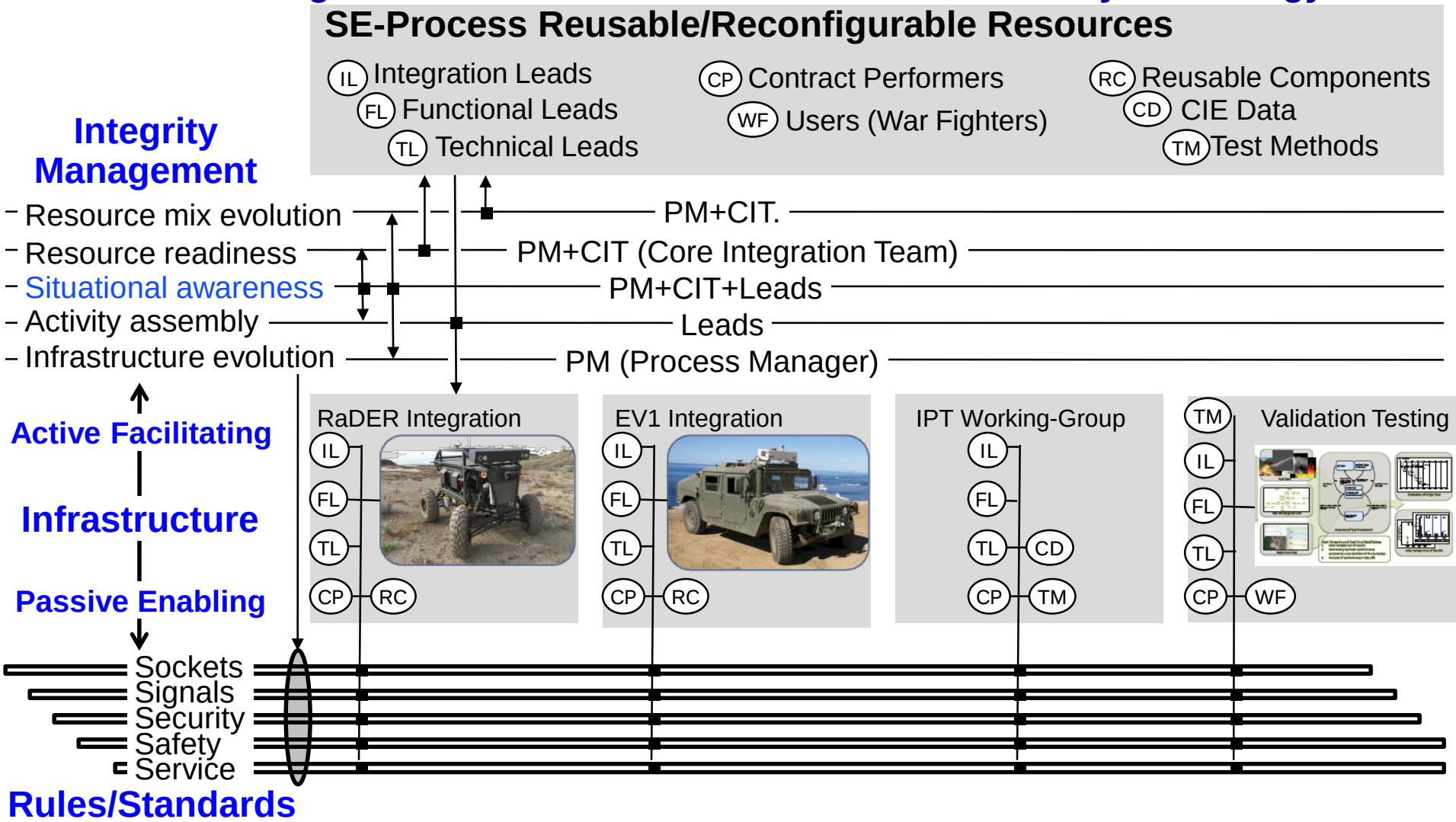




Content: Chris Scrapper, SSC-Pac

Agile SE Process AAP

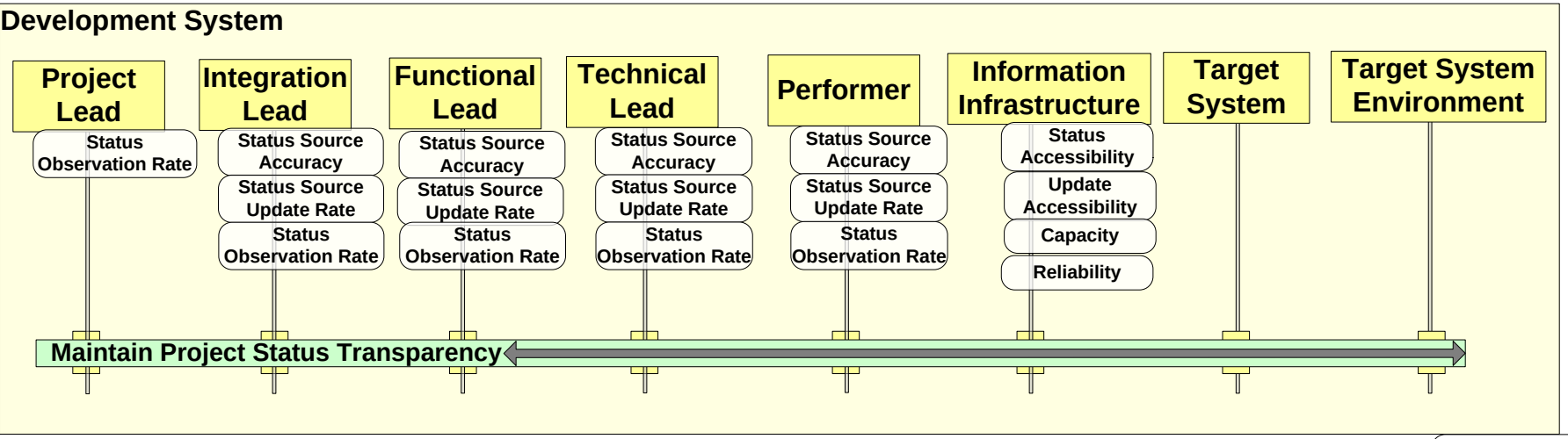
for evolving autonomous off-road-vehicle robotic military technology



Rules/Standards

Sockets: CIE, System-1 modular architecture, roles, culture, test threads
Signals: Vision, Declarations of Intent, Config Mgmnt Plan, Integration Strategy, CIE data, decisions, engaged team feedback
Security: User agreement/NDA, Config Mgmnt Plan, CIE access controls
Safety: Open-process visibility, open communication, protected communication
Service (SE ConOps): Vision, Culture, Consciousness(CIE), Conscience, Wave, Integration Strategy/TEMP, Sys-1 and Sys-2 AAP

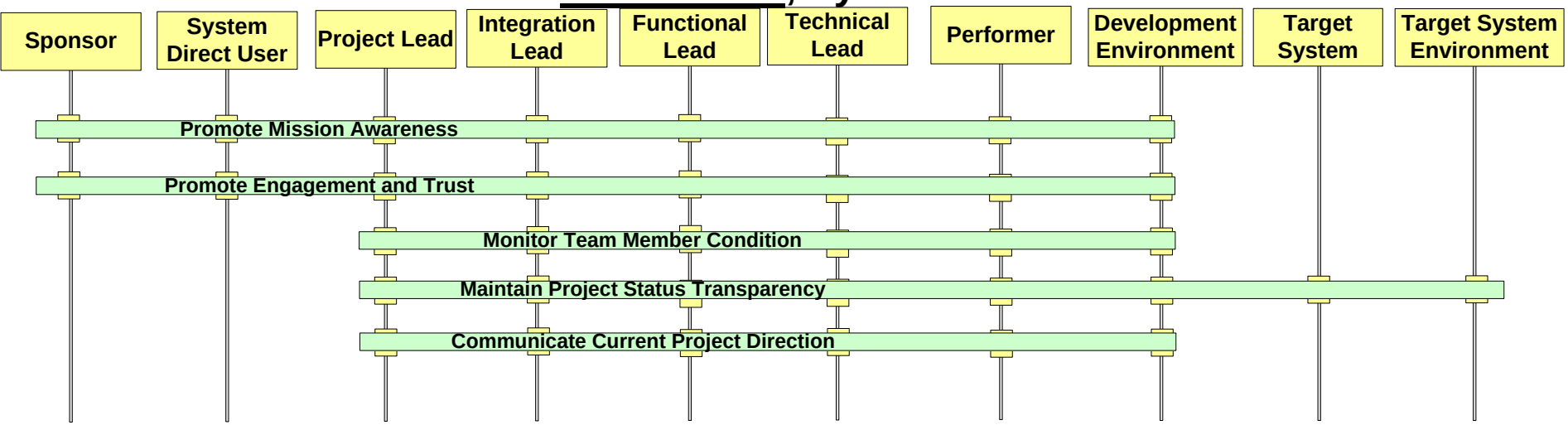
Pattern Modeling Examples from SSC-Pac Case Study



Attributes of Individual Component Roles, and Emergent Systemic Attributes

- Team Status Awareness
- Team Status Accuracy
- Team Status Currency

Selected Subset of ASELCM Interactions, System-2



Collective Culture of Engagement

Most pronounced during the analysis activity was the pervasive nature of the culture, its thoughtful development, and its continual reinforcement. Done with a combination of soft skills and supporting infrastructure.

Culture is a shared set of expectations for behavior, and an environment that enforces that behavior.

Here culture isn't written like a mission statement, but is rather practiced by leadership, shaped by consistent reinforcement, and enforced by dealing openly with infractions detrimental to the team and at odds with a pervasive collective agreement to work together toward total success.

Full and active engagement with the SE process intent and the SE project objectives is the expectation. All team members are on a shared mission, and all team members need to support and be supported by all other team members, at all times.

The nature of the SE process, its leadership, and the transparency of comprehensive real-time project status provide team-engagement sensitivity. If the culture doesn't fit an individual, that individual will either move on, or adjust. The culture does not tolerate inaction.

Collective Consciousness

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.

Collective Conscience

Meeting openings remind everyone that the customers are taxpayers and warfighters. These reminders don't stop with a simple statement. They are rooted in image and story that elevates them to personified walking needs with faces.

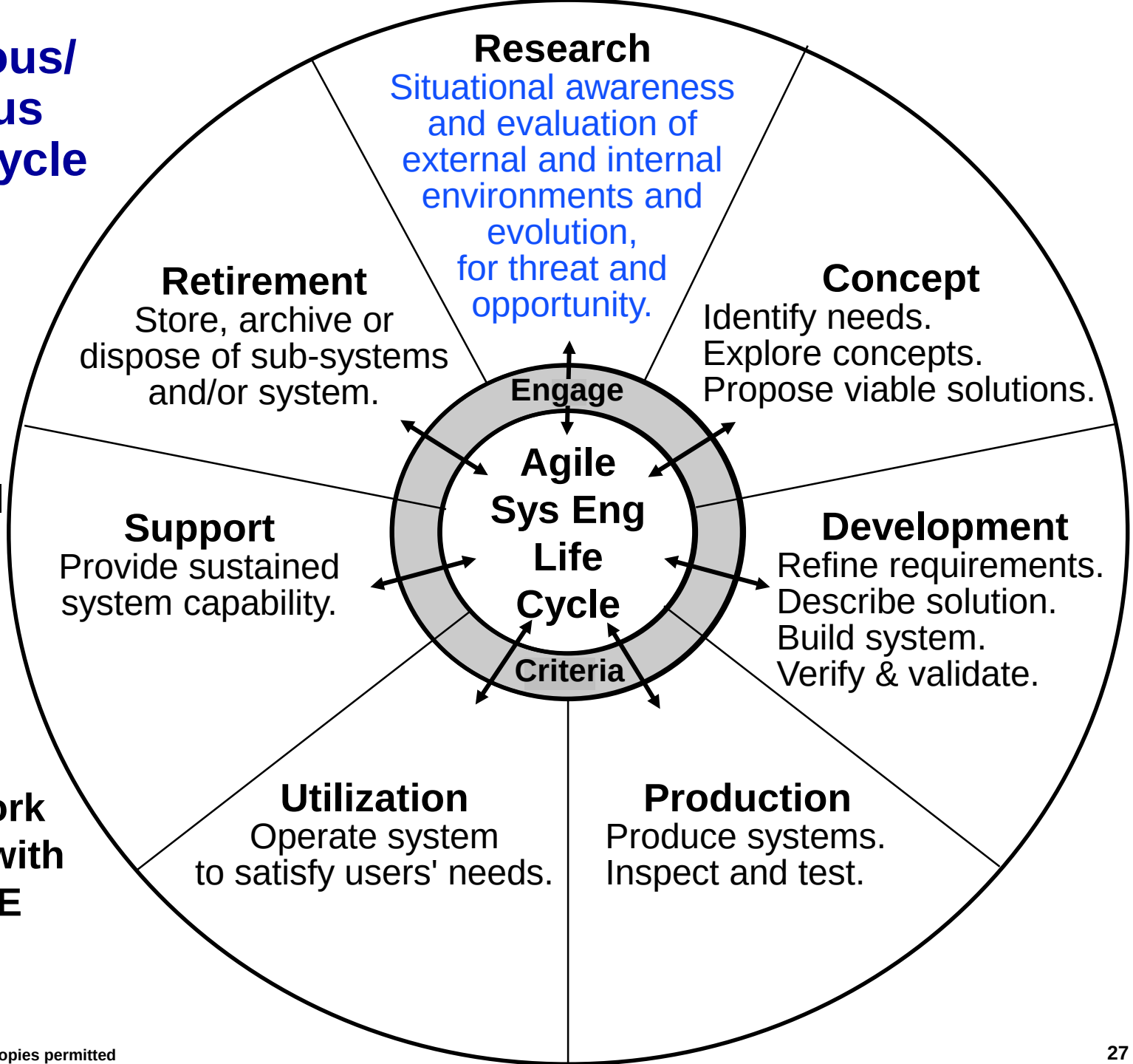
The warfighter needs tools that are effective, timely, and affordable for mission achievement and self preservation. Warfighter reality is obtained with their critical presence at testing events, and with structured workshops between waves.

The tax payer needs tools that are effective, timely, and affordable for national/homeland security – capability that is affordably deployable, not costly technology that limits production quantities and threatens sustainable programs.

In these contexts (warfighter and taxpayer) the team accepts responsibility, and evaluates decisions with that critical internal customer voice.

The team develops and maintains a collective conscience to do what is responsibly right. This breaks the inertia of building upon favorite and comfortable technical approaches, to consider technologies that address the fundamental needs.

Asynchronous/ Simultaneous Agile Life-Cycle Framework



Observed in all workshops to date

This framework is consistent with ISO/IEC/IEEE standards

ASELCM Project – First Year

Accomplished

- 4 analysis workshops: SSC-Pac, NGC, Rockwell-Collins, Lockheed
- Case studies: 1 finished (IS16), 2 in draft and 1 in start (IS17 targeted)

Learned relative to project objectives

- Working hypothesis developed for 9 fundamental principles
- Asynchronous/Simultaneous life cycle model framework fits practice
- Awareness (research) life-cycle stage is necessary addition
- 15288 process activities are distributed/integrated throughout stages
- Agile SE producing agile systems
  enables/facilitates system evolution and life-extension

Next

- Secure 4-5 second-round hosts
- Vet 9-principle hypothesis: confirm/deny/augment
- <u>Explicitly</u> capture examples of the employment of the principles

References

Dove, R. and W. Schindel. 2016. Agile Systems Engineering Process Features Collective Culture, Consciousness, and Conscience at SSC Pacific Unmanned Systems Group. INCOSE International Symposium (IS 2016), Edinburgh, Scotland, UK, July 18-21. www.parshift.com/s/ASELCM-01SSCPac.pdf

Schindel W. and R. Dove. 2016. Introduction to the Agile Systems Engineering Life Cycle MBSE Pattern. INCOSE International Symposium (IS 2016), Edinburgh, Scotland, UK, July 18-21. www.parshift.com/s/160718IS16-IntroductionToTheAgileSystemsEngineeringLifeCycleMBSEPattern.pdf

Scrapper, C., R. Halterman, and J. Dahmann. 2016. An Implementers View of the Evolutionary Systems Engineering for Autonomous Unmanned Systems. IEEE Systems Conference, Orlando FL, 18-21 April.

INCOSE Webinars:

Agile 101: Architecture Pattern,
www.parshift.com/s/AgileSystems-101.pdf, www.parshift.com/s/AgileSystems-101.wmv

Agile 102: Design Requirements,
www.parshift.com/s/AgileSystems-102.pdf, www.parshift.com/s/AgileSystems-102.swf

Agile 103: Design Principles,
www.parshift.com/s/AgileSystems-103.pdf, www.parshift.com/s/AgileSystems-103.mp4

Agile 104: Engagement Quality,
www.parshift.com/s/AgileSystems-104.pdf, www.parshift.com/s/AgileSystems-104.mp4

Agile 105: Operational Awareness,
www.parshift.com/s/AgileSystems-105.pdf, www.parshift.com/s/AgileSystems-105.mp4

ASELCM project and workshop Host information/details:
www.parshift.com/ASELCM/Home.html