A Few Words First

Courtesy – Please mute your phone (*6 toggle).

Upcoming Chapter Meetings:

- **Sep 13, Beyond Biomimicry to Systems Mimicry**
  Len Troncale, Professor Emeritus California State Polytechnic University
- **Oct 06-07, 2017 Socorro Systems Summit at NM Tech.**
- **Oct 11, 2017, Why is Human-Model Interactivity Important to the Future of Model-Centric Systems Engineering?**
  Dr. Donna Rhodes, Massachusetts Institute of Technology
- **Nov 9, Architecting Cyber Physical Systems**
  Dr. Cihan Dagli, Missouri University of Science & Technology
- **Dec 8, Holiday Social at a place TBD**
  Mary Compton, Event Producer

CSEP Courses by *Certification Training International:*
Course details | Course brochure
Course Schedule (close by, but many more locations and dates):
2017 Oct 30-Nov 3 | Las Vegas, NV
2018 Feb 26-Mar 2 | Las Vegas, NV
2018 Apr 02-Apr 5 | Denver

First slide, not recorded but retained in pdf presentation.

And Now - Introductions

Enchantment Chapter Monthly Meeting

9 August 2017 – 4:45-6:00 pm:

Agile Systems and Processes 106 – Risk Management and Mitigation

Rick Dove, Paradigm Shift International, dove@parshift.com

Abstract: To be effective, projects/processes/products (all viewed as systems) have to mate well with their operational environments. Operational environments are not static, they react to disturbances and evolve with opportunity and whimsy. Inserting a system into an environment is a disturbance. Sustaining a system in an environment entails compatible evolution. The environment is the problem space the system will occupy. Understanding the requirements for a compatible-to-the-space solution is best done before system functional requirements get too far ahead and shape an incompatible path. Given enough understanding about the problem, effective solution requirements and features becomes (almost) obvious. The problem shapes and constrains effective solution. But how do we characterize the environment as a dynamic problem space and develop solution-response requirements; and then, how do we structure a solution for risk-mitigating agility? This webinar introduces methods for dynamic problem-space characterization, and reviews methods for risk-mitigating solution-space agility.

Download slides today-only from GlobalMeetSeven file library or anytime from the Library at www.incose.org/enchantment

NOTE: This meeting will be recorded

Today's Presentation

Things to Think About

How can this be applied in your work environment?

What did you hear that will influence your thinking?

What is your take away from this presentation?

Speaker Bio

Rick Dove is a leading researcher, practitioner, and educator of fundamental principles for agile enterprise, agile systems, and agile development processes. In 1991 he initiated the global interest in agility as co-PI on the seminal 21st Century Manufacturing Enterprise Strategy project at Lehigh University. Subsequently he organized and led collaborative research at the DARPA-funded Agility Forum, involving 250 organizations and 1000 participants in workshop discovery of fundamental enabling principles for agile systems and processes.

He is CEO of Paradigm Shift International, specializing in agile systems research, engineering, and education; and is an adjunct professor at Stevens Institute of Technology teaching graduate courses in agile and self-organizing systems.

He chairs the INCOSE working groups for Agile Systems and Systems Engineering, and for Systems Security Engineering, and is the leader of the current INCOSE Agile Systems Engineering Life Cycle Model Discovery Project.

He is an INCOSE Fellow, and the author of *Response Ability – the Language, Structure, and Culture of the Agile Enterprise*.

This webinar is the final installment of a six-part series.

It includes material from prior webinars and adds some new material.

It is not a tutorial,
but rather a comprehensive overview of tools for design.

It reflects the content of a 40-hour course provided by
Stevens Institute of Technology, ES 678.

Context

Technology, knowledge, expectations, competitors, and adversaries are changing fast and faster.

Q: How is system relevancy and viability sustained in this reality?

A: By using and evolving available response options.

Q: How do needed response options become available?

A: By analyzing the problem-space for response requirements and designing mitigating response capability.

Value Proposition for Agility

Faster, lower cost system development?

An appealing argument, but only a side effect (at best).

The value proposition for agility is Risk Management.

Sustainability of innovation/process/product at risk.

Content

This webinar is about Systems Engineering Agility,
(not Agile Software Practices – but what is exposed applies)

- **Context for this Discussion**
- **SE Operational Point of View for Risk Management**
  - **Concurrent Stages Life Cycle Framework**
  - **Sense-Respond-Evolve Operational Principles**
  - **Nested Logical-Systems Pattern Boundaries**
- **Problem Space Characterization Tools**
  - **CURVEd Operating Environment Characterization**
  - **Reality Factors**
  - **Response Situation Analysis**
- **Solution Space Structure and Design Tools**
  - **Establishing Goals and Strategy**
  - **Agile Architecture Pattern**
  - **Reusable/Reconfigurable/Scalable Design Principles**
  - **Design Closure and Traceability**
- **Wrap Up**

Operational POV for SE Agility

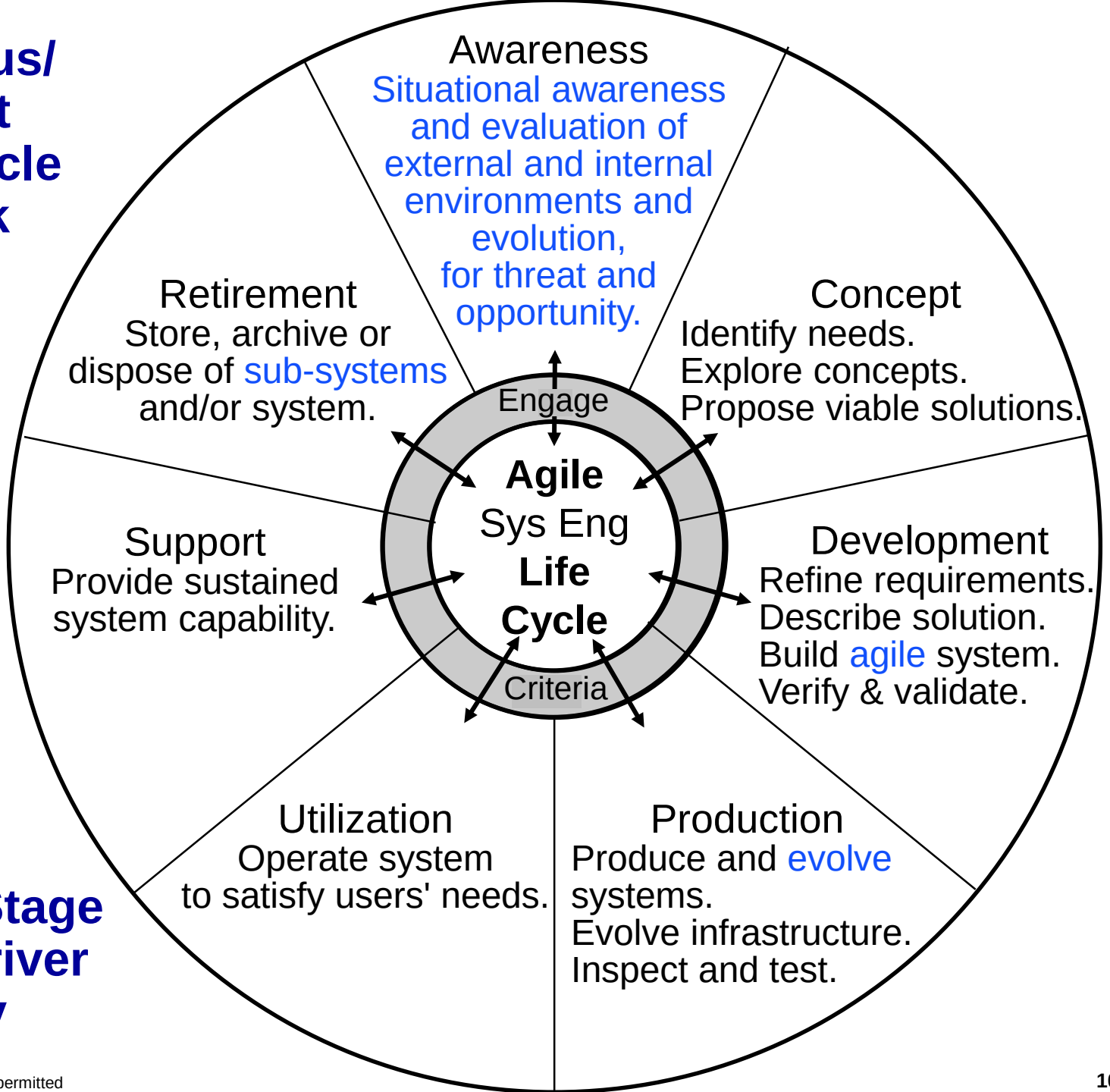
The INCOSE ASELCM Project is discovering
Agile Systems Engineering Life Cycle Model fundamentals

These manifest as core SE-agility necessities, not best practices

Three outcomes are relevant at this point in the discussion:

- **Concurrent-Stage Life Cycle Framework**
- **Sense-Respond-Evolve Operational Principles**
- **Nested Innovation Pattern Boundaries**

Asynchronous/ Concurrent Agile Life-Cycle Framework



New Awareness Stage is Critical Driver of Agility

What's Different – What's Not?

What's Different …

The addition of an Awareness seventh stage breaths life into the agile SE process, taking it beyond a repetitive execution of development increments fulfilling a pre-determined backlog of planned features.

The life cycle model framework does not have fixed starting and ending points. It implies and accommodates perpetual evolution beyond initial delivery. And requires that the product produced by the process is agile.

The *Retirement* stage recognizes that subsystems and older system versions are retired frequently, as the "current" system evolves. This has implications for maintenance, disposal, and reversion processes.

What's Not …

ISO/IEC 2010, page 32, clearly accommodates asynchronous and concurrent activity in any and all stages with this clarification statement:

"…one can jump from a stage to one that does not immediately follow it, or revert to a prior stage or stages that do not immediately precede it. … one applies, at any stage, the appropriate life cycle processes, in whatever sequence is appropriate to the project, and repeatedly or recursively as appropriate."

Agility-Facilitating Operational Design Principles (SRE)

Discoveries of the INCOSE ASELCM Project

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)
- Evaluation (internal and external judgement)
- Memory (evolving culture, response capabilities, and ConOps)

SRE Operational Principles

Fundamentally the purpose of agility is survival in an uncooperative dynamic environment.

Sensing and Responding mirror John Boyd's OODA loop.

- **Observe and Orient are encompassed by Sensing.**
- **Decide and Act are encompassed by Responding.**

The Evolving category wasn't overlooked by Boyd.

- **It is the ultimate purpose of his OODA loop.**
- **He valued the necessity to learn and improve the practice of OODA looping, each time through the loop.**

There is nothing new here. It is the natural way we navigate through life.

It is, however, a new way of appreciating where the cul-de-sac of artificial, seemingly logical, SE approaches have taken us.

Deeper Appreciation

In Boyd's application of the OODA loop to fighter pilot dog-fight engagement, he recognized a human cognitive activity, and expects an increasing learned intuition.

Putting this roughly in neuroscience terms, the brain is a pattern learner and recognizer that drives motor action.

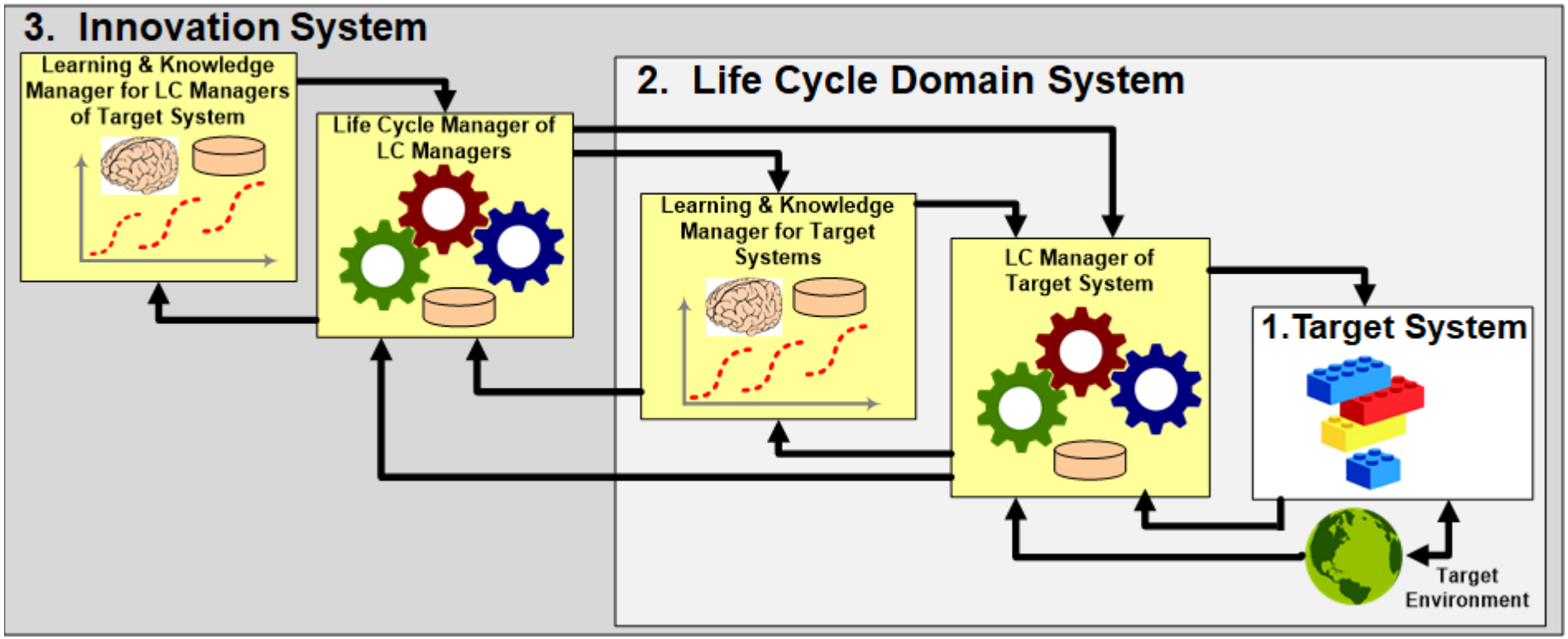
- **Increased learning experience drives these functions away from reasoning and closer to direct and immediate motor control.**
- **Action becomes systemically autonomic – the ultimate objective of Evolution in the SRE operational framework.**

Note the difference between Plan-Do-Check-Act and OODA. PDCA has a sequential procedural feel, OODA has an in-the-moment dynamic-engagement feel. OODA is focused on awareness-driven re-evaluation of the changing problem space, rather than marching to a plan.

The accomplished OODA loop practitioner is not running through a sequence of four activities in incremental repetition, but is rather engaged in all four activities simultaneously.

Agile Systems Engineering Life Cycle Pattern

Systems 1, 2, and 3 Logical/Behavioral (not physical) Boundaries



- 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.

This pattern depiction is the work of Bill Schindel,
a member of the ASELCM leadership team.

Slide credit: Bill Schindel

The Innovation Behavior System

The system of innovation – is alert to risks and opportunities and has means to mitigate risk and take advantage of opportunity.

Responsible for situational awareness and appropriate evolution.

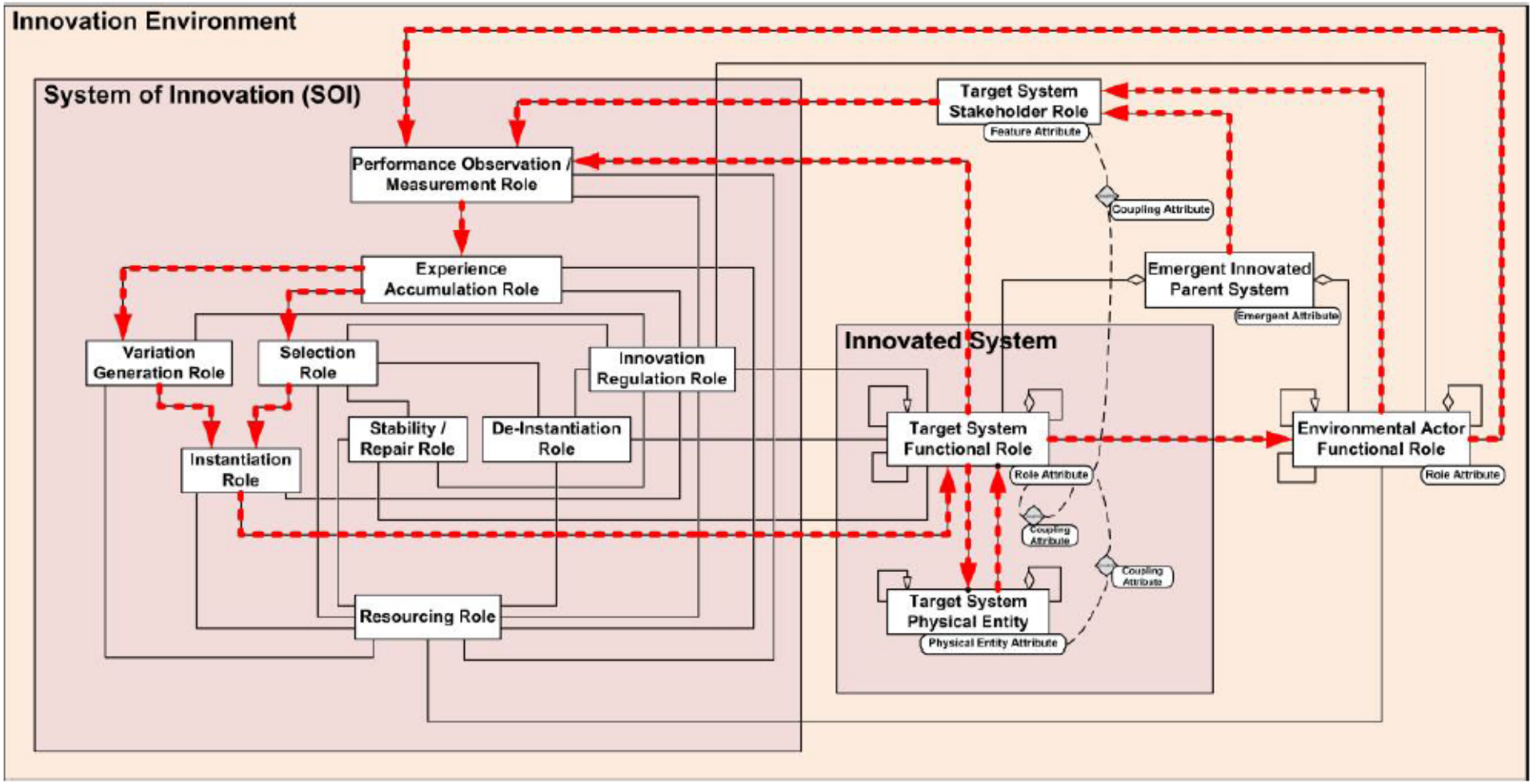
Schindel: "Innovation" is defined here as the realization of significantly enhanced stakeholder benefit. This distinguishes innovation from invention, novelty, ideation, creativity, or similar concepts that become parts of innovation in at least some cases."

Thus, innovation in our sense here is the effective management of risk & opportunity.

Risk arises from external and internal sources – some can be anticipated before system design occurs, more will emerge subsequently during system operation.

Innovation Behavior Pattern, With Roles

Illustrative signaling paths in innovation.
Systems of Innovation, combined with the Target Systems that they innovate, form complex adaptive systems. (Beihoff and Schindel, 2012).



Tools for Problem-Space Analysis

The INCOSE ASELCM Project employed these tools to analyze what various agile SE processes were dealing with.

But these tools were developed for identifying the response requirements for a to-be agile SE process design.

Three tools are relevant at this point in the discussion:

- **CURVE characterization of the problem space**
- **Reality Factors in the problem space**
- **Response Situation Analysis for design requirements**

Characterizing the Problem-Space

CURVE Tool

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

Capriciousness: 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.

Example: Agile SE Process Environment CURVE
From an ASELCM Project Case Analysis

Capriciousness (Unpredictability): Unknowable Situations
- ❑ Urgent pre-emptive customer needs
- ❑ Depot maintenance responsibility and capability

Uncertainty: Randomness With Unknowable Probabilities
- ❑ Initial process framework applicability and nature of tailoring needed
- ❑ Regression impacts – the effects of integrating new development with prior development
- ❑ Contract compatibility
- ❑ Management agile-process engagement commitment
- ❑ Documentation requirements compatibility
- ❑ Feature vs. capability reconciliation (amount of feature-requirements freedom)
- ❑ Employee SE-process engagement

Risk: Randomness With Knowable Probabilities
- ❑ Cultural incompatibility
- ❑ Ability to keep and attract talent
- ❑ Systems of Systems requirements changes
- ❑ External stakeholder schedule timelines (e.g. certification)

Variation: Knowable Variables And Associated Ranges
- ❑ Multiple projects competing for bottlenecks (e.g. test facilities)
- ❑ System Of Systems integration
- ❑ Subcontractors development-process compatibility

Evolution: Gradual Successive Development
- ❑ Planned modernization/sustainment increments
- ❑ Open Mission Systems and OSA evolution
- ❑ SE-process tailoring evolution
- ❑ Depot maintenance and upgrade responsibility
- ❑ Contract SE-process accommodation

Reality Factors Tool

Requirements often assume a relatively benign environment, and tend to focus on the capability and feature needs. This framework tool analyzes the external environment.

Human Behavior – Human error, whimsy, expediency, arrogance...

Organizational Behavior – Survival rules rule, nobody's in control...

Technology Pace – Accelerating vulnerability-introductions...

System Complexity – Incomprehensible, unintended consequences...

Globalization – Partners with different ethics, values, infrastructures...

Partially Agile Fads – Outsourcing, web services, cots policies & effects...

Agile Adversaries/Competitors/Customers – Distributed, collaborative, self organizing, proactive, impatient, innovative…

Example from an ASELCM Project Case Analysis

Reality Factors
Human (Including Customer) Behavior – Human error, whimsy, expediency, arrogance... • Leadership wants to please the customer without knowing the technology or organization. • Availability/quantity of subject matter experts unmatched to needs.
Organizational Behavior – Survival rules rule, nobody's in absolute control... • Every program is considered Most-Important. • Redirected team resources (cherry picking best resources for other needs).
Technology Pace – Accelerating technology and security-vulnerability introductions,... • Customers demanding cutting edge technology.
System Complexity – Incomprehensible, unintended consequences, emergence... • Numerous simultaneous projects with numerous stakeholders per project. • Multi-project resource contention.
Globalization – Different ethics, values, infrastructures, cultural assumptions... • Local certification and accreditation authorities. • Cultural differences in global marketplace.
Partially-Agile Enterprise Faddish Practices – Outsourcing, COTS policies/affects... • COTS supply/supplier affects. • Agile software-practice thinking dominance. • Different degrees of agility across the different disciplines (HW, FW, SW, Systems).
Agile Customers/Competitors/Adversaries – Distributed, collaborative, impatient, … • Large, complex programs with accelerating market-need dates.

Response Situation Analysis Tool

Response Domain		
Proactive	Creation (and Elimination)	Proactive responses are generally triggered internally by the application of new knowledge to generate new value. They are still proactive responses even if the values generated are not positive and even if the knowledge applied is not new – self initiation is the distinguishing feature here. A proactive response is usually one that has effect rather than mere potential; thus, it is an application of knowledge rather than the invention or possession of unapplied knowledge. Proactive response proficiency is the wellspring of leadership and innovation in system capability.
	Improvement	
	Migration	
	Modification (of Capability)	
Reactive	Correction	Reactive responses are generally triggered by events which demand a response: problems that must be attended to or fixed, opportunities that must be addressed. The distinguishing feature is little choice in the matter – a reaction is required. Reactive responses often address threatening competitive or environmental dynamics, new customer demands, agility deterioration/failure, legal and regulatory disasters, product failures, market restructuring, and other non-competitor generated events. Reactive response proficiency is the foundation of resilience and sustainability in system capability.
	Variation	
	Expansion (of Capacity)	
	Reconfiguration	

Example: Response Situation Analysis

Core Issue Amalgamation from ASELCM Project Case Analyses

	Domain	Response Issue
Proactive	Creation (and Elimination)	What must the system be creating or eliminating in the course of its operational activity? • Risk and opportunity awareness/knowledge • Acculturated memory • Mitigation actions/options • Decisions to act
	Improvement	What performance will the system be expected to improve during operational life cycle? • Awareness • Effectiveness of mitigation actions/options • Memory in acculturation, inventoried response options, and ConOps
	Migration	What major events coming down the road will require a change in the system infrastructure? • New fundamentally-different types of risks and opportunities
	Modification (Add/Sub Capability)	What modifications in resources-employed might need made as the system is used? • Mitigation response action appropriate for specific response need • Personnel appropriate and available for a response action
Reactive	Correction	What can go wrong that will need an automatic systemic detection and response? • Insufficient/inadequate awareness • Wrong decisions • Ineffective mitigation actions/options
	Variation	What process variables will range across what values and need accommodation? • Effectiveness of mitigation actions/options • Effectiveness of mitigation evaluation
	Expansion (and Contraction of Capacity)	What are “quantity-based” elastic-capacity range needs on resources/output/activity/other? • Capacity to handle 1-? critical mitigation actions simultaneously
	Reconfigu-ration	What types of resource relationship configurations will need changed during operation? • Elements of a mitigation action • Mitigation managers/engineers

Getting it Right

Requirements *shall statements* define

exactly what must be accomplished.

If you miss even one you could have a dysfunctional result.

For Response Situation Analysis…

you do not need to develop a *comprehensive* list of shall statements, but rather a *sufficient and evolvable* list of *response needs* –

which if accomplished,

will stretch the envelope of agile response capability

to encompass all necessary response needs,

even if they are not on the list.

Tools for Solution-Space Synthesis

The INCOSE ASELCM Project employed the AAP tool to analyze architecture and structure of the various agile SE processes.

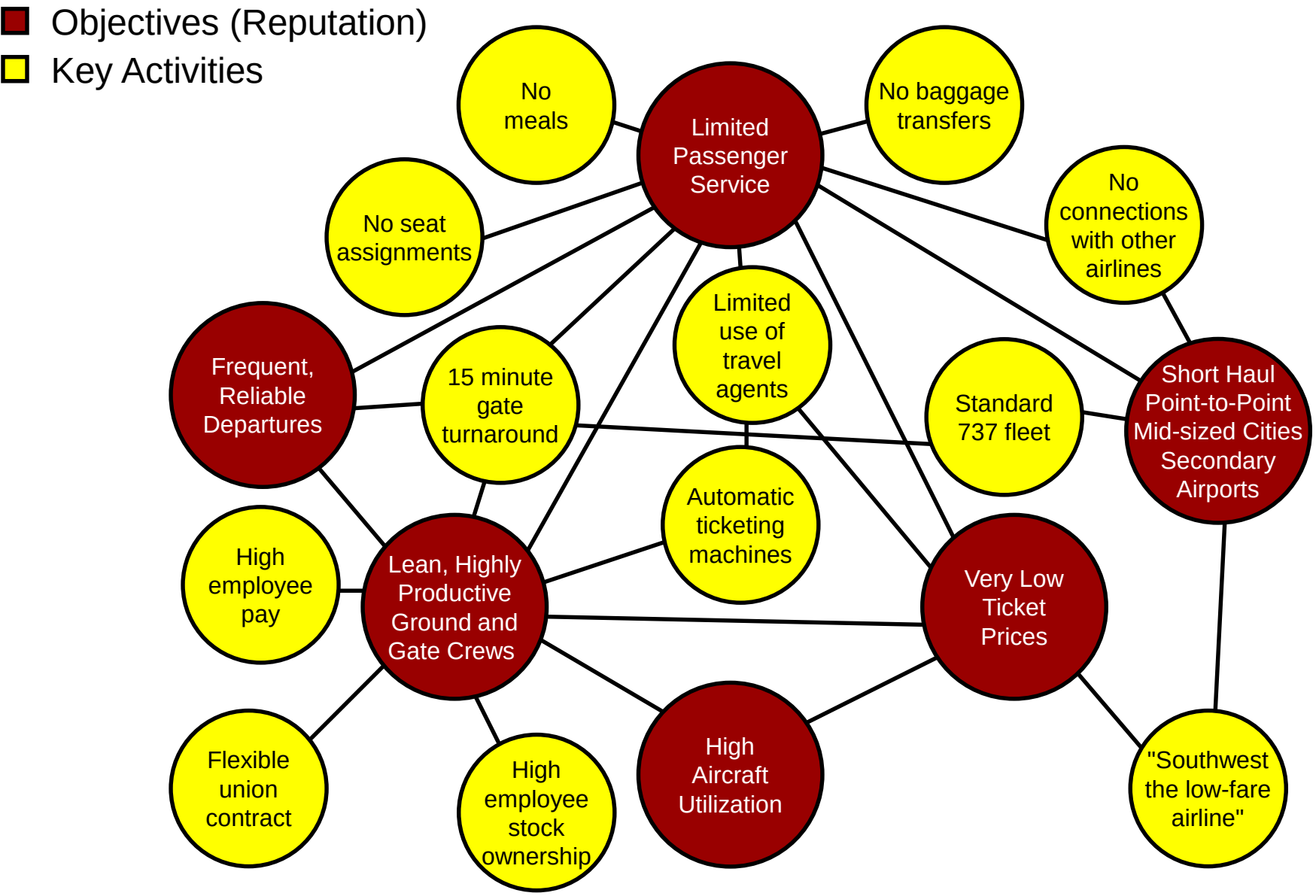
But these tools were developed for creating the agile SE structural and operational conceptual design.

Four tools are relevant at this point in the discussion:

- **ConOps Strategic Activity Web**
- **ConOps Agile Architecture Pattern**
- **RRS design principles**
- **Closure Matrix for design enrichment and traceability**

Southwest Airlines

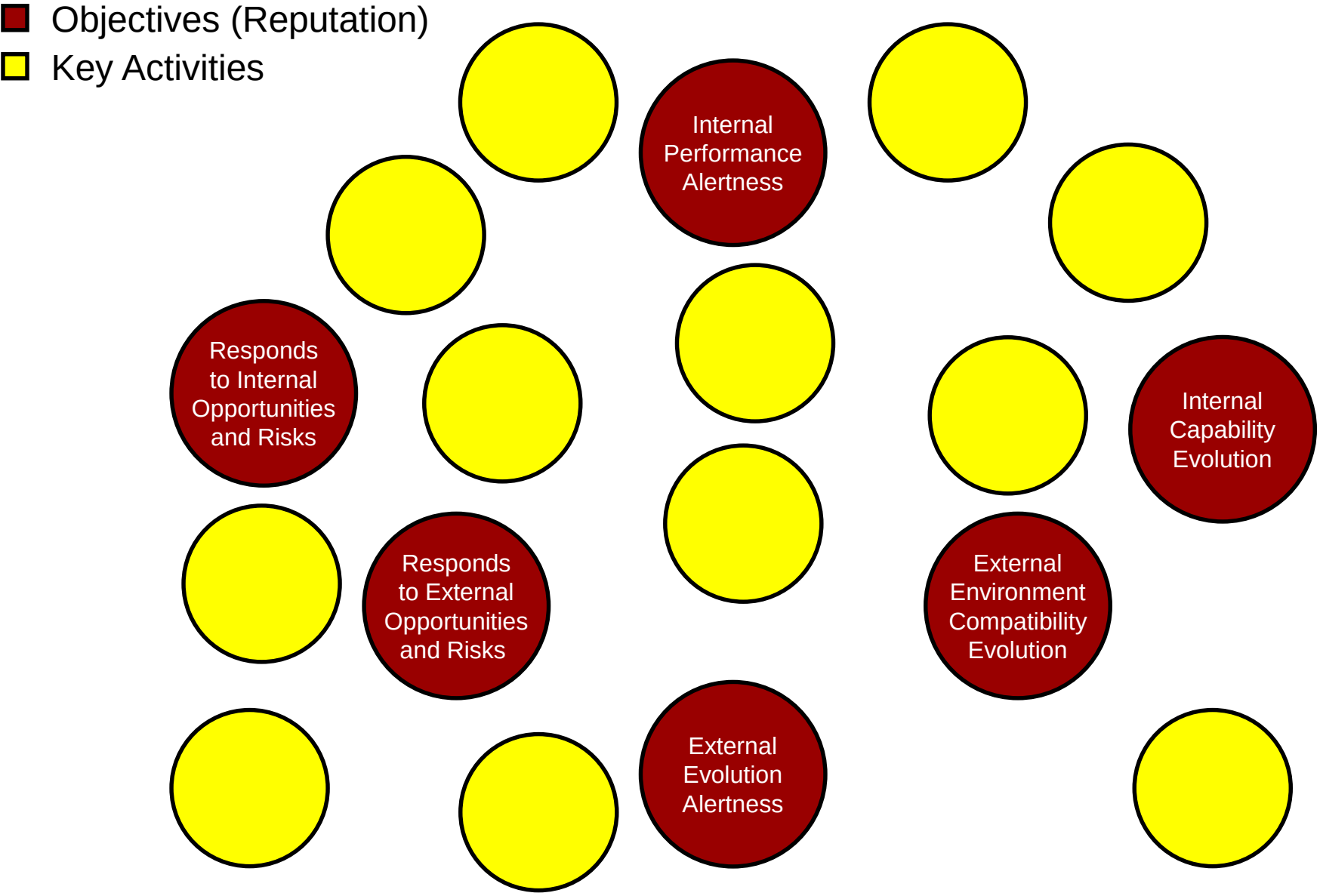
Strategic Activity Web



"What is Strategy?", Michael Porter, Harvard Business Review, Nov-Dec 1996

Innovation Behavior System

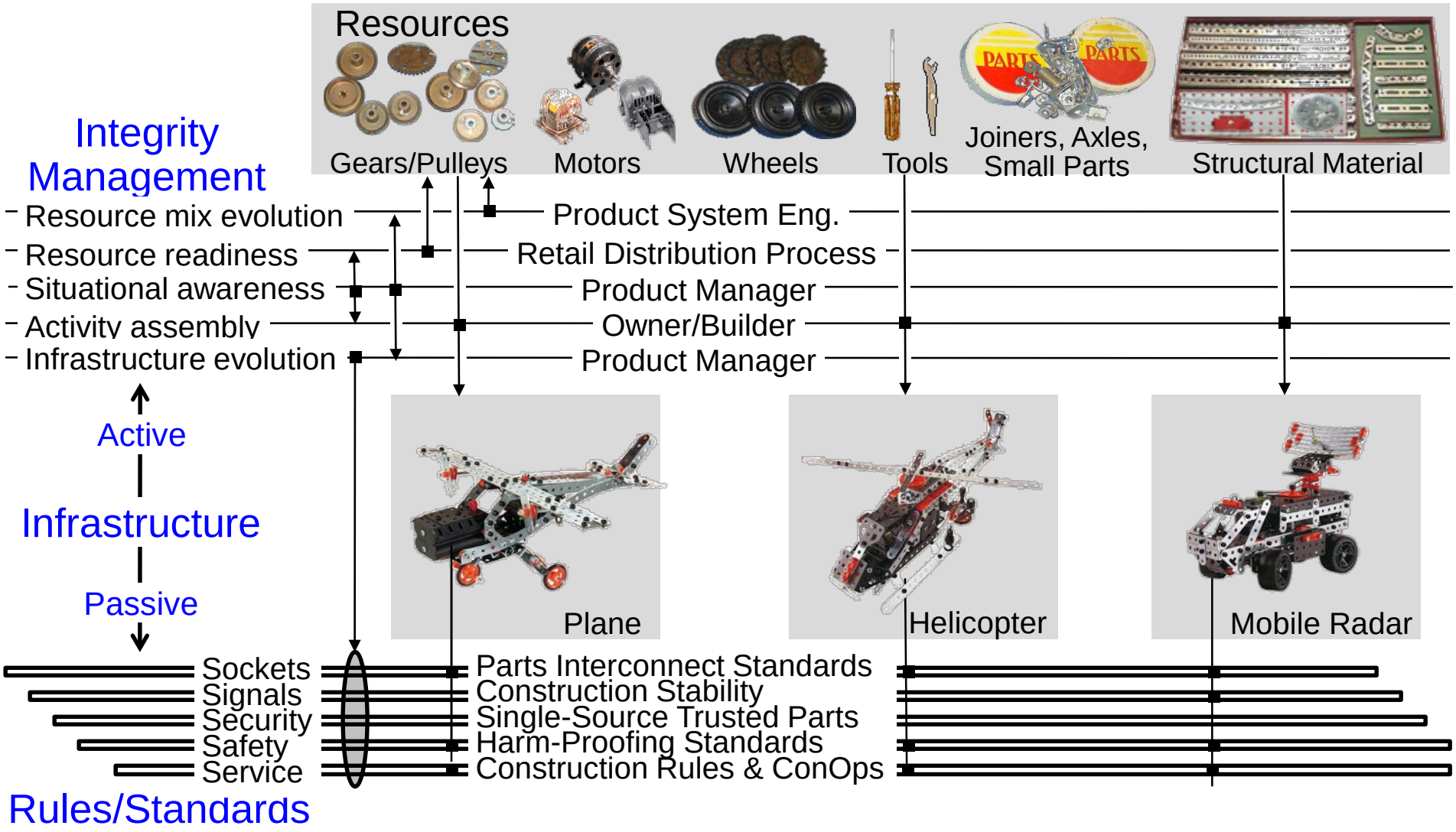
Strategic Activity Web Tool



General Objectives, Activities and Connections are Specific-System Context Dependent

Agile Architecture Pattern (AAP) Tool
Notional Concept: System Response-Construction Kit

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



Sustaining Agility Requires …

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

Five Necessary 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. **System 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?**

Architectural ConOps Graphic

An Agile Architecture Pattern for systems and processes is useful for its succinct descriptive effect.

AAP displays the principle architectural structure and design components that depict what enables and facilitates agility.

It is a framework
for customer and management communication,
for training new team members,
for capturing lessons learned, and
for maintaining a current central understanding of
key operational concepts as they evolve.

It serves well as a single-graphic road map for the operational concept.

www.parshift.com/s/ASELCM-01SSCPac.pdf

Multi-Project SE Process

for evolving autonomous off-road-vehicle robotic military technology

SPAWAR Systems Center PACIFIC

SE-Process Reusable/Reconfigurable Resources

- IL Integration Leads
- FL Functional Leads
- TL Technical Leads
- CP Contract Performers
- WF Users (War Fighters)
- RC Reusable Components
- CD CIE Data
- TM Test Methods

Integrity Management

- Resource mix evolution — PM+CIT.
- Resource readiness — PM+CIT (Core Integration Team)
- Situational awareness — PM+CIT+Leads
- Activity assembly — Leads
- Infrastructure evolution — PM (Process Manager)

Active Facilitating

Infrastructure

Passive Enabling

RaDER Integration

IL, FL, TL, CP, RC

EV1 Integration

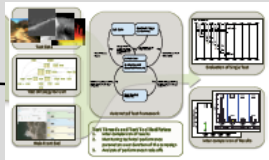
IL, FL, TL, CP, RC

IPT Working-Group

IL, FL, TL, CD, CP, TM

Validation Testing

TM, IL, FL, TL, CP, WF



- Sockets
- Signals
- Security
- Safety
- Service

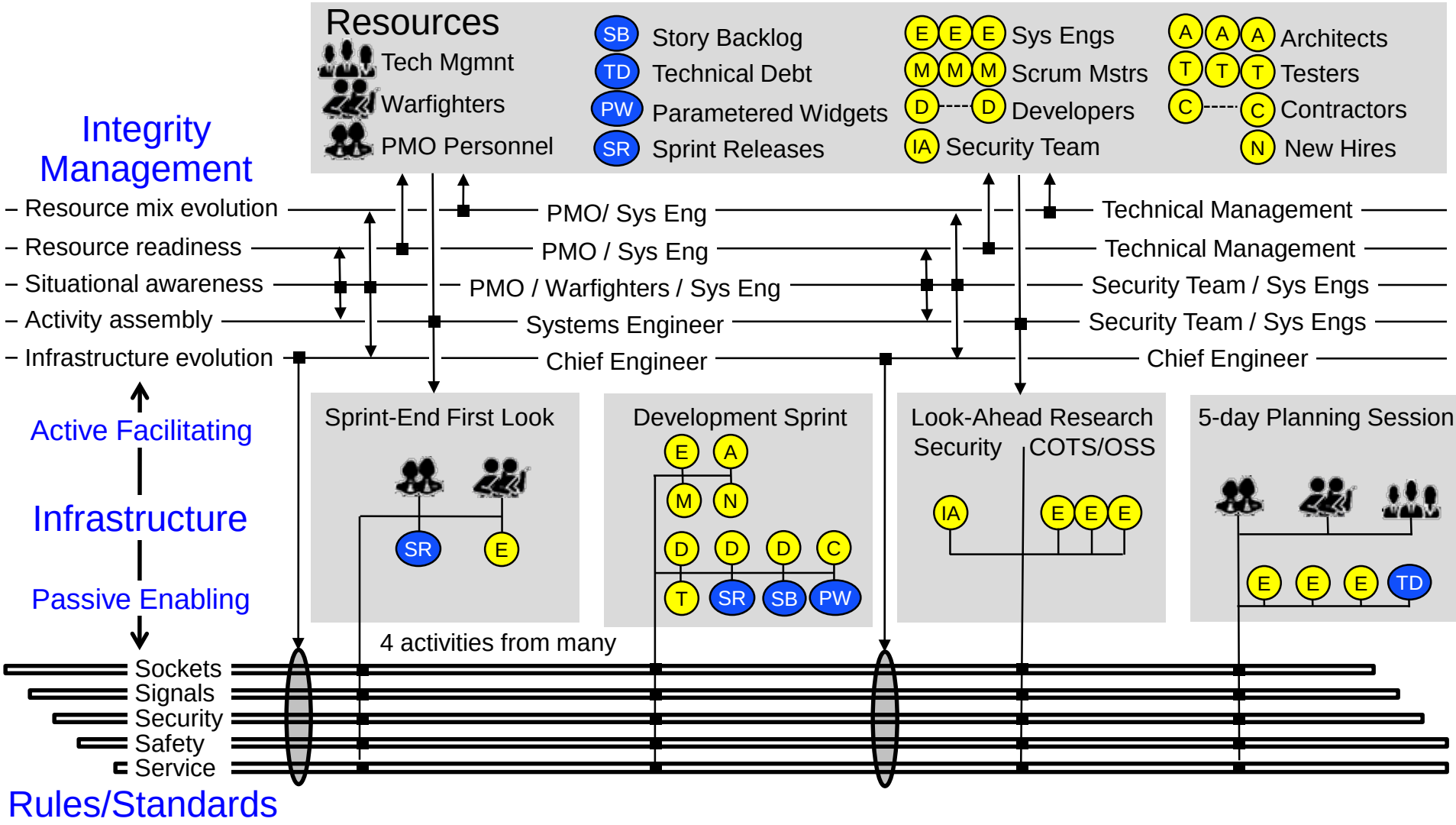
Rules/Standards

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

www.parshift.com/s/ASELCM-03NGC.pdf

SoS Web-Portal Evolution Process

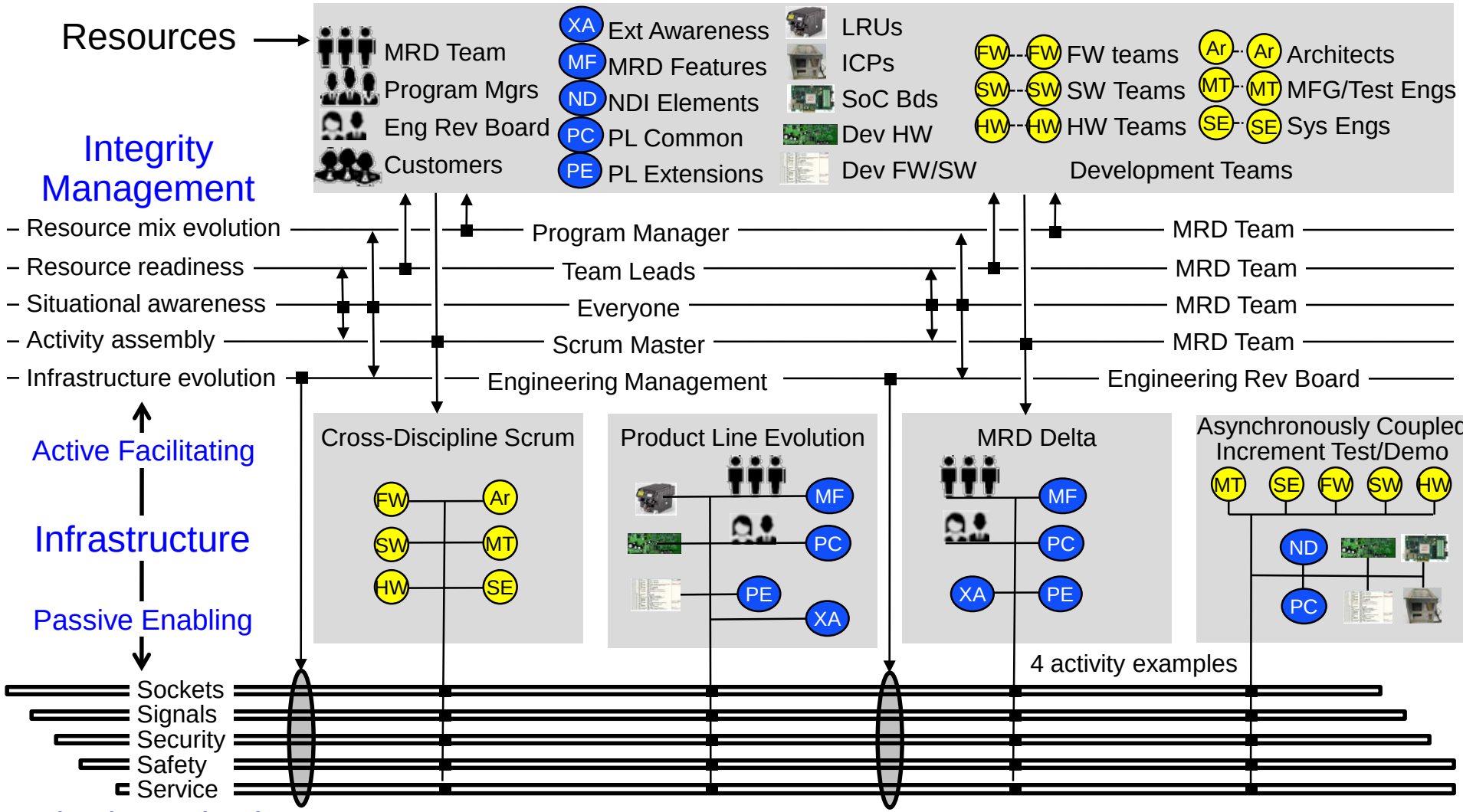
NORTHROP GRUMMAN



Rules/Standards

Sockets: Meeting formats, Sys-1 modular architecture, Automated build environment, User story acceptance criteria, Roles, Culture
Signals: Vision/Intent, Release themes, Spikes, User stories, Wireframes, Code, SCR, Process status/metrics, Deliverables, Behavior
Security: Governance, Leadership, Cultural oversight, QA, Metrics, CMMI level 5 oversight, Configuration management
Safety: Open-process visibility, Open no-penalty communication, On-boarding, Team user-story estimation, 40-hour work load
Service Documented accessible ConOps, Embedded environment awareness, Continuous DevOps integration, AAP for Systems 1&2

Rockwell Collins System 2 Product-Line AAP



Rules/Standards

Sockets: PL component-interface standards, Scrums, Collaboration space
Signals: MRD, Epics, Stories, Specifications, Requirements, IMS, JIRA issues, Confluence data
Security: Program reviews, Retrospectives, Scrum ceremonies
Safety: Training, Scrum Ceremonies
Service: RC Agile process ConOps, Market requirements document, Confluence, HW development platforms

Agile-Transition System 3

Lockheed IFG Avionics

Process-Innovation Resources

- Process Mgmnt Team
- Chief Engineers Office
- Customers

- SE SAFe elements
- TE Tailored elements
- EE Experimental elements

- E E E Sys Engs
- S S S Scrum Teams
- C C Coaches

- A A A Architects
- T T T Testers
- O O Outsources

Development Teams

Integrity Management

- Resource mix evolution
- Resource readiness
- Situational awareness
- Activity assembly
- Infrastructure evolution

Process Framework Team
Process Mgmnet Team
Process Mgment Team
Process Framework Team
Process Framework Team

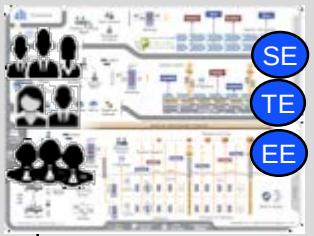
Technical Management
Coaches
Virtually Everyone
Process Execution Team
Process Framework Team

Active Facilitating

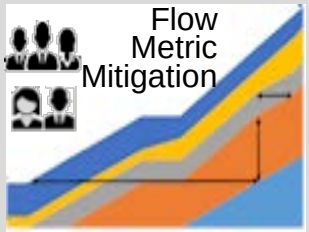
Infrastructure

Passive Enabling

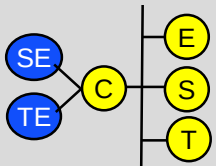
Process Framework



Process Instrumentation

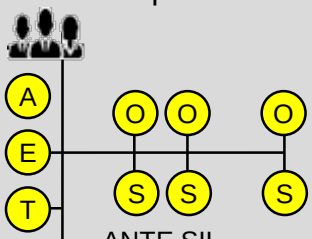


Process Conformance



Training/Coaching/Therapy

Process Experimentation



ANTE SIL

4 activity examples

- Sockets
- Signals
- Security
- Safety
- Service

Rules/Standards

Sockets: Process framework, Roles, Teams, Meeting formats, ANTE/Simulation frameworks
Signals: Flow, Info debt, Process conformance, Experiment results, Contract performance
Security: Executive commitment, Governance, Cultural consistency
Safety: Information radiators, No-penalty measurement, Flow monitoring/mitigation, Real-time status information, 2-3 PI look-ahead
Service (ConOps): Process framework, Cadence, Customer/User involvement, Optimal-process control, System 1-2-3 AAP

Agility-Enabling Structural Design Principles (RRS)

see INCOSE Webinar Agile 103

Reusable

- Encapsulated resources
- 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

Response Able System Principles – RRS Tool

Reconfigurable, Reusable, Scalable (Think: Plug-and-Play, Drag-and-drop)

Encapsulated Resources Modules are encapsulated independent units loosely coupled through the passive infrastructure.	Reusable	Scalable	**Evolving Infrastructure** ConOps and Resource interface and interaction standards and rules that evolve slowly.
Facilitated Interfacing (Pluggable) Resources & infrastructure have features facilitating easy module insertion/removal.			**Redundancy and Diversity** Duplicate Resources provide fail-soft & capacity options; diversity provides functional options.
Facilitated Reuse Resources are reusable and/or replicable; with supporting facilitation for finding and employing resources.			**Elastic Capacity** Resource populations & functional capacity may be increased and decreased within existing infrastructure.
Reconfigurable			
Peer-Peer Interaction Resources communicate directly on a peer-to-peer relationship; parallel rather than sequential relationships are favored.			**Distributed Control & Information** Decisions made at point of maximum knowledge; information accessible globally but kept locally.
Deferred Commitment Resource relationships are transient when possible; decisions & fixed bindings are postponed until necessary.			**Self-Organization** Resource relationships are self-determined; and component interaction is self-adjusting or negotiated.

Example RRS Principles – Agile ERP SE Process

Dove, R. 2005. Fundamental Principles for Agile Systems Engineering.

Encapsulated Resources Resources are encapsulated independent units loosely coupled through the passive infrastructure. Bus vendor, ERP app vendors, database vendor, app requirements developers, infrastructure requirements developers, infrastructure implementers.	Reusable	**Evolving Infrastructure** ConOps and resource interface and interaction standards that evolve slowly. 3-phase implementation, 90-day phases max, no spec/requirement changes once phase begins, internal total infrastructure design responsibility, vendor total application responsibility (HW/SW)
Facilitated Interfacing (Pluggable) Resources & infrastructure have features facilitating easy module insertion/removal. Vendor interface rules clear, agreed in advance, & managed.		**Redundancy and Diversity** Duplicate resources provide fail-soft & capacity options; diversity provides functional options. Cross-trained BSA dept responsibilities, mixed outsource/insource resources and expertise.
Facilitated Reuse Resources are reusable and/or replicable; with supporting facilitation for finding and employing appropriate resources. BSA group, business process development system.		**Elastic Capacity** Resource populations & functional capacity may be increased and decreased widely within the existing infrastructure. Outsource implementers managed by small internal group.
Reconfigurable		
Peer-Peer Interaction Resources communicate directly on a peer-to-peer relationship; parallel rather than sequential relationships are favored. All vendors are peers, BSAs have direct access to everyone.		**Distributed Control & Information** Decisions made at point of maximum knowledge; information accessible globally but kept locally. BSA business rule development autonomy, SSA infrastructure rules/design autonomy, vendor implementation autonomy.
Deferred Commitment Resource relationships are transient when possible; decisions & fixed bindings are postponed until necessary. Implementation doesn't begin until requirements are firm.		**Self-Organization** Resource relationships are self-determined; and component interaction is self-adjusting or negotiated. BSA team relationships and assignments.

Closure Matrix Tool – Where Deep Design Begins

Details: *Response Ability*, Chapter 7 section headed “Principle-Based design”

(Case: An Insight Development System)

Activities (Functions)		RRS Principles									
		Encapsulated Resources	Facilitated Interfacing	Facilitated Re-Use	Peer-Peer Interaction	Deferred Commitment	Distributed Control & Info	Self Organization	Elastic Capacity	Redundancy & Diversity	Evolving Infrastructure
Establish personal values	1										
Analyze external case for ideas	2										
Analyze local case for principles	3										
Design a business practice	4										
Package as *response ability models*	5										
Rotate student / mentor roles	6										
Review and select for quality	7										
RSA Issues (Requirements)		**Principle-Based Activities, and Issues Served**									
Proactive: Capturing hidden tacit knowledge	3567	35	356	57	3	37	6	3		3	37
Proactive: Creating student interest and value	124	1	1	1	12	124		124	1	1	
Proactive: Improving knowledge accuracy	367		6		3	37	6	3		3	7
Proactive: Improving knowledge effectiveness	1245	45	245	45	1			12	5	2	
Proactive: Migrating the knowledge focus	247	27	4	2		4	7	247		4	47
Proactive: Accommodating different student types	(all)	25	6			347	2	12345	1	17	2
Proactive: Injecting fresh outside knowledge	26	26	26		2		6	2			
Reactive: Finding and fixing incorrect knowledge	367	7		7	3	3	6	3		3	7
Reactive: Excising poor value knowledge	2357	7		7	3	3	2	23		35	257
Reactive: Allowing flexible student schedules	34	34			34			34			
Reactive: Accommodating any size group	2345	2345	234				2	25	34	234	
Reactive: Reinterpret rules for new applications	23457	27		5		2	357				23457

Closure Matrix Tool – Where Deep Design Begins

Details: *Response Ability*, Chapter 7 section headed "Principle-Based design"

(Case: An Insight Development System)

Activities (Functions)		RRS Principles: Encapsulated Resources	Facilitated Interfacing	Facilitated Re-Use	Peer-Peer Interaction	Deferred Commitment	Distributed Control & Info	Self Organization	Elastic Capacity	Redundancy & Diversity	Evolving Infrastructure
Establish personal values	1										
Analyze external case for ideas	2										
Analyze local case for principles	3										
Design a business practice	4										
Package as *response ability models*	5										
Rotate student / mentor roles	6										
Review and select for quality	7										
RSA Issues (Requirements)		**Principle-Based Activities, and Issues Served**									
Proactive: Capturing hidden tacit knowledge	3567	35	356	57	3	37	6	3		3	37
Proactive: Creating student interest and value	124	1	1	1	12	124		124	1	1	
Proactive: Improving knowledge accuracy	367		6		3	37	6	3		3	7
Proactive: Improving knowledge effectiveness	1245	45	245	45	1			12	5	2	
Proactive: Migrating the knowledge focus	247	27	4	2		4	7	247		4	47
Proactive: Accommodating different student types	(all)	25	6			347	2	12345	1	17	2
Proactive: Injecting fresh outside knowledge	26	26	26		2		6	2			
Reactive: Finding and fixing incorrect knowledge	367	7		7	3	3	6	3		3	7
Reactive: Excising poor value knowledge	2357	7		7	3	3	2	23		35	257
Reactive: Allowing flexible student schedules	34	34			34			34			
Reactive: Accommodating any size group	2345	2345	234				2	25	34	234	
Reactive: Reinterpret rules for new applications	23457	27		5		2	357				23457

Risk Management & Mitigation Tools

"When I am working on a problem,

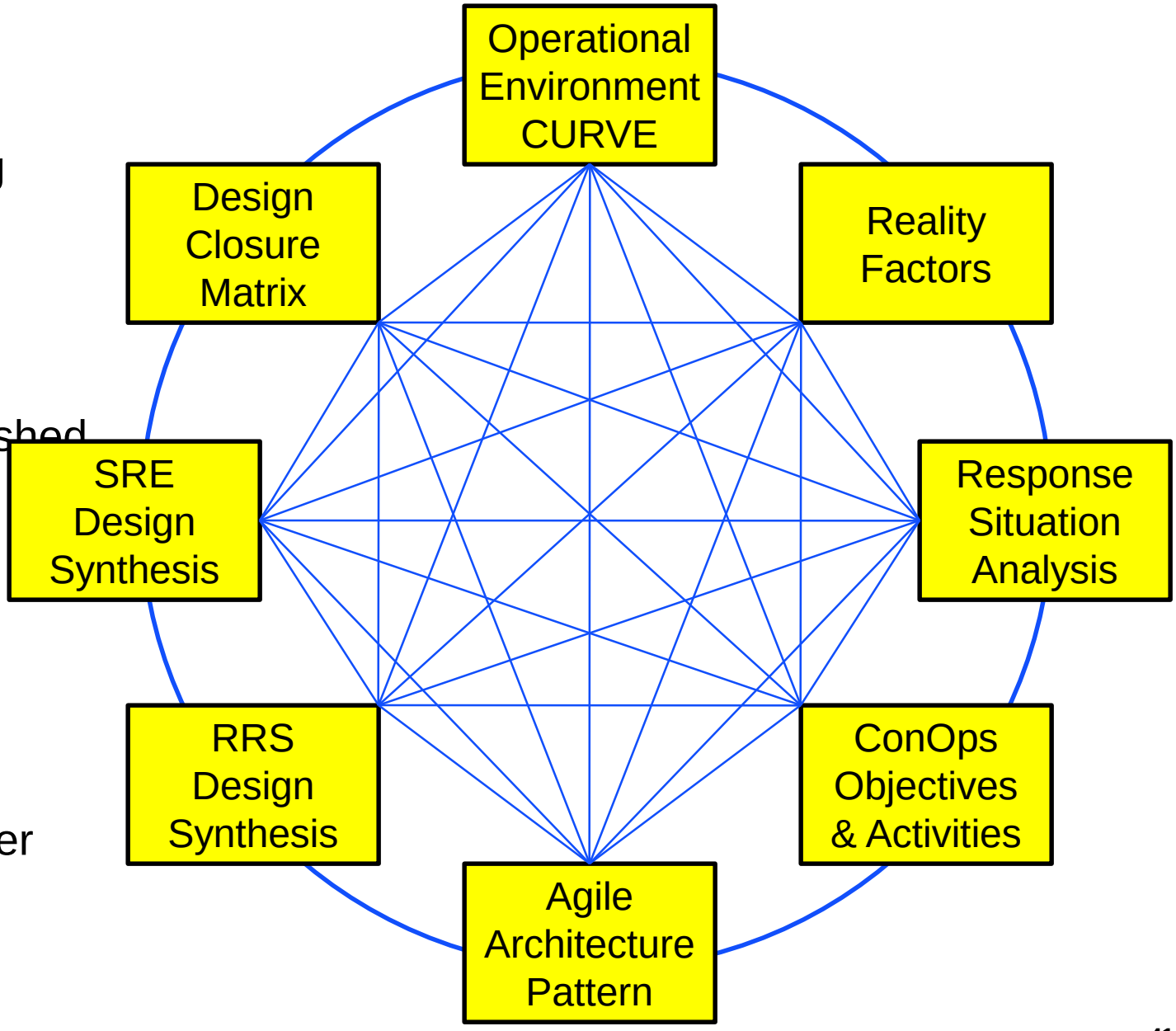
I never think about beauty,

but when I have finished

if the solution is not beautiful,

I know it is wrong."

R. Buckminster Fuller



Full Series

Download 106 webinar slides: Agile System/Process Risk Management & Mitigation
Download 105 webinar slides: Agile System/Process Operational Awareness
Download 104 webinar slides: Agile System/Process Engagement Quality
Download 103 webinar slides: Agile System/Process Design Principles
Download 102 webinar slides: Agile System/Process Design Requirements
Download 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 103 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

General References

Beihoff, Bruce C. and William D. Schindel. 2012. Systems of Innovation I: Summary Models of SOI Health and Pathologies. Proceedings INCOSE International Symposium, Rome, Italy, July 9–12, 2012. http://onlinelibrary.wiley.com/doi/10.1002/j.2334-5837.2012.tb01325.x/abstract

Dove, R. 2001. *Response Ability – The Language, Structure, and Culture of the Agile Enterprise*. Wiley.

Dove, R. 2005. Fundamental Principles for Agile Systems Engineering. Conference on Systems Engineering Research (CSER), Stevens Institute of Technology, Hoboken, NJ, 24 March. www.parshift.com/s/050324Cser-FundamentalsAndSilterra.pdf

Dove, R., R. LaBarge. 2014. Fundamentals of Agile Systems Engineering – Part 1 and Part 2. International Council on Systems Engineering, International Symposium, Las Vegas, NV, 30Jun-3Jul. www.parshift.com/s/140630IS14-AgileSystemsEngineering-Part1&2.pdf

ISO/IEC. 2010. Systems and Software Engineering — Life Cycle Management — Part 1: Guide for Life Cycle Management. ISO/IEC TR 24748-1:2010(E) first edition. Switzerland.

Schindel, William D. 2013. Systems of Innovation II: The Emergence of Purpose. Proceedings INCOSE International Symposium, Philadelphia PA, USA, June 24–27, 2013. http://onlinelibrary.wiley.com/doi/10.1002/j.2334-5837.2013.tb03069.x/abstract

Wikipedia. nd. [John Boyd's] OODA Loop. https://en.wikipedia.org/wiki/OODA_loop

ASELCM Project References

Dove, R. n.d. Agile Systems Engineering Life Cycle Fundamentals Project, Documents at https://connect.incose.org/ProgramsProjects/aselcm/Pages/Home.aspx, alternatively at www.parshift.com/ASELCM/Home.html

Dove, R., W. Schindel, and C. Scrapper. 2016. Agile Systems Engineering Process Features Collective Culture, Consciousness, and Conscience at SSC Pacific Unmanned Systems Group. Proceedings International Symposium. International Council on Systems Engineering. Edinburgh, Scotland, 18-21 July. www.parshift.com/s/ASELCM-01SSCPac.pdf

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

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

Dove, R., W. Schindel. 2017. Case Study: Transition to Scaled-Agile Systems Engineering at Lockheed Martin's Integrated Fighter Group. Working paper. www.parshift.com/s/ASELCM-04LMC.pdf

Dove, R. 2017. Agility in Systems Engineering – Findings from Recent Studies. Working paper, 15-April. www.parshift.com/s/ASELCM170415-AgilityInSE-Findings.pdf

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

Today's Presentation

Things to Think About

How can this be applied in your work environment?

What did you hear that will influence your thinking?

What is your take away from this presentation?

Please

The link for the online survey for this meeting is

www.surveymonkey.com/r/2017_08_MeetingEval

www.surveymonkey.com/r/2017_08_MeetingEval

Look in GlobalMeet chat box for cut & paste link.

Slide presentation can be downloaded now/anytime from:

The library page at: www.incose.org/enchantment.

Recording will be there in the library tomorrow.