A Few Words First

Audio Connection – Please mute phone (*6 toggle) – or your GM left-side name

All phone connections may be muted during the presentation. If so, save your comments and questions for Q&A and/or put them in the GlobalMeet chat window.

Upcoming Meetings:

- **Sep 20-22: Western States Regional Conference, Ogden, Utah**
  Website: https://incose-wsrc.eventbrite.com, Presentation call open all of March
- **Oct 10: Using Enterprise Architecture for Analysis of a Complex Adaptive Organization's Risk Inducing Characteristics**
  Lura Salguero, Sandia National Labs, R&D Systems Engineer
- **Oct 26: Tutorial – Design Structure Matrix Methods and Applications**
  Tyson Browning, Professor of Operations Management, Texas Christian Univ.
- **Nov 14: An SE Approach to Providing PV in Ghana**
  Marlene Brown, Sandia National Labs, Systems Engineer

CSEP Courses by *Certification Training International:*
Course details (with more locations and dates)
Upcoming Course Schedule (somewhat nearby):
2019 Feb 11-15 | San Francisco, CA
2019 Aug 12-16 | Austin, TX
Chapter SEP mentors: Ann Hodges alhodge@sandia.gov, Heidi Hahn hahn@lanl.gov

First slide, not recorded but retained in pdf presentation.

And Now - Introductions

Enchantment Chapter Monthly Meeting

12 September 2018 – 16:45-18:00 MT

Agile Operations 201 – Problem Space Derived Solution Requirements

Rick Dove, Paradigm Shift International, CEO/CTO
dove@parshift.com

Abstract: The definition of agile systems engineering is rooted in what it does, not how it does it. What it does is respond effectively in a life cycle environment that is capricious, uncertain, risky, variable, and evolving. How it does that is a product of analyzing response requirements dictated by the nature of the life cycle environment. The design and evolution of an operationally effective agile systems engineering process is itself a systems engineering activity, one that requires an attentive emphasis on problem space characterization and ongoing evolution. This webinar will cover methods for developing and maintaining problem space characterization, and identifying and tracing the life cycle response requirements dictated by that characterization. If you don't know where you are going, any road will do. Process examples analyzed in the INCOSE Agile Systems Engineering Life Cycle Model project will demonstrate the application of these methods.

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

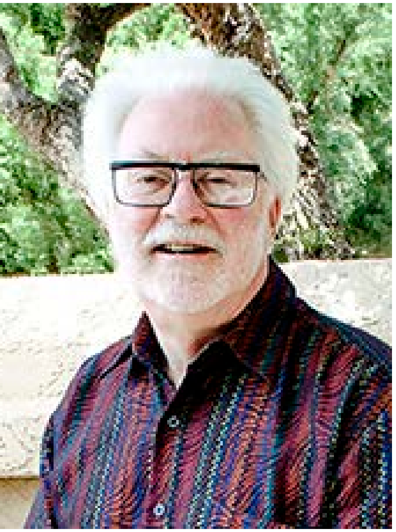
Things to Think About

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.

Agile SE Processes 201:
Problem Space Derived Solution Requirements

Enchantment Chapter

September 12, 2018

Rick Dove

Abstract

The definition of agile systems engineering is rooted in what it does, not how it does it.

What it does is respond effectively in a life cycle environment that is capricious, uncertain, risky, variable, and evolving.

How it does that is a product of analyzing response requirements dictated by the nature of the life cycle environment.

The design and evolution of an operationally effective agile systems engineering process is itself a systems engineering activity, one that requires an attentive emphasis on problem space characterization and ongoing evolution.

This webinar will cover methods for developing and maintaining problem space characterization, and identifying and tracing the life cycle response requirements dictated by that characterization.

If you don't know where you are going, any road will do.

Process examples analyzed in the INCOSE Agile Systems Engineering Life Cycle Model (ASELCM) project will demonstrate the application of these methods.

Agility is …

effective response **to opportunity and problem, within mission ... always.**

An *effective response* is one that is:

- **timely (fast enough to deliver value),**
- **affordable (can be repeated as often as necessary),**
- **predictable (can be counted on to meet expectations),**
- **comprehensive (everything within mission boundary).**

Not fast,
…just fast enough

Agility is the ability to survive and thrive in an unpredictable and uncertain environment

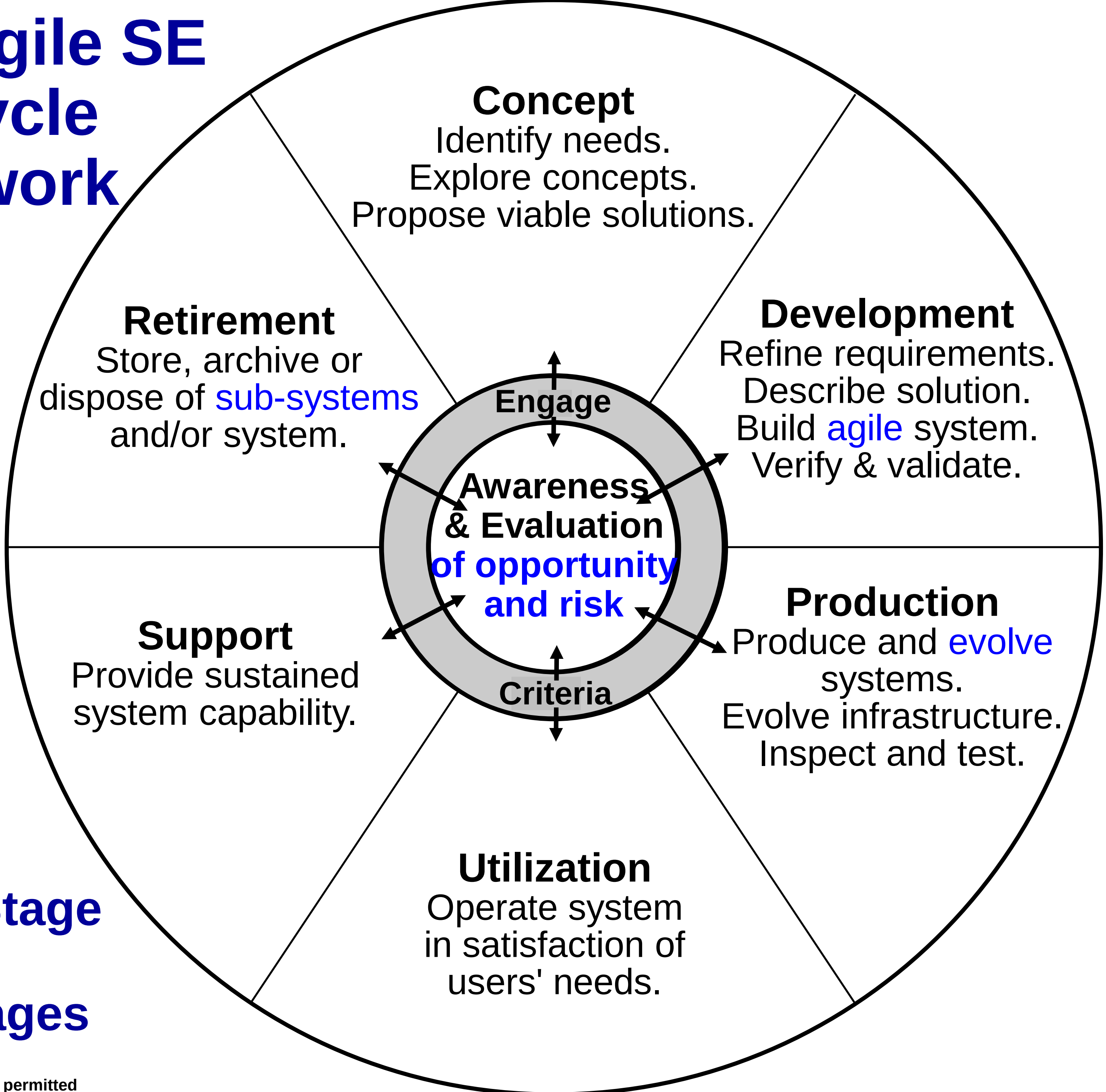
Agility is Risk Management: decreasing vulnerability and risk by increasing response options and predictability

Prelude:
Six Findings from the ASELCM Project
(Agile Systems Engineering Life Cycle Model)

- General Agile SE Life Cycle Framework
- General Operational Pattern
- General Operational Principles
- General Concept of Information Debt
- General Problem-Space Characterization
- General Response Requirements

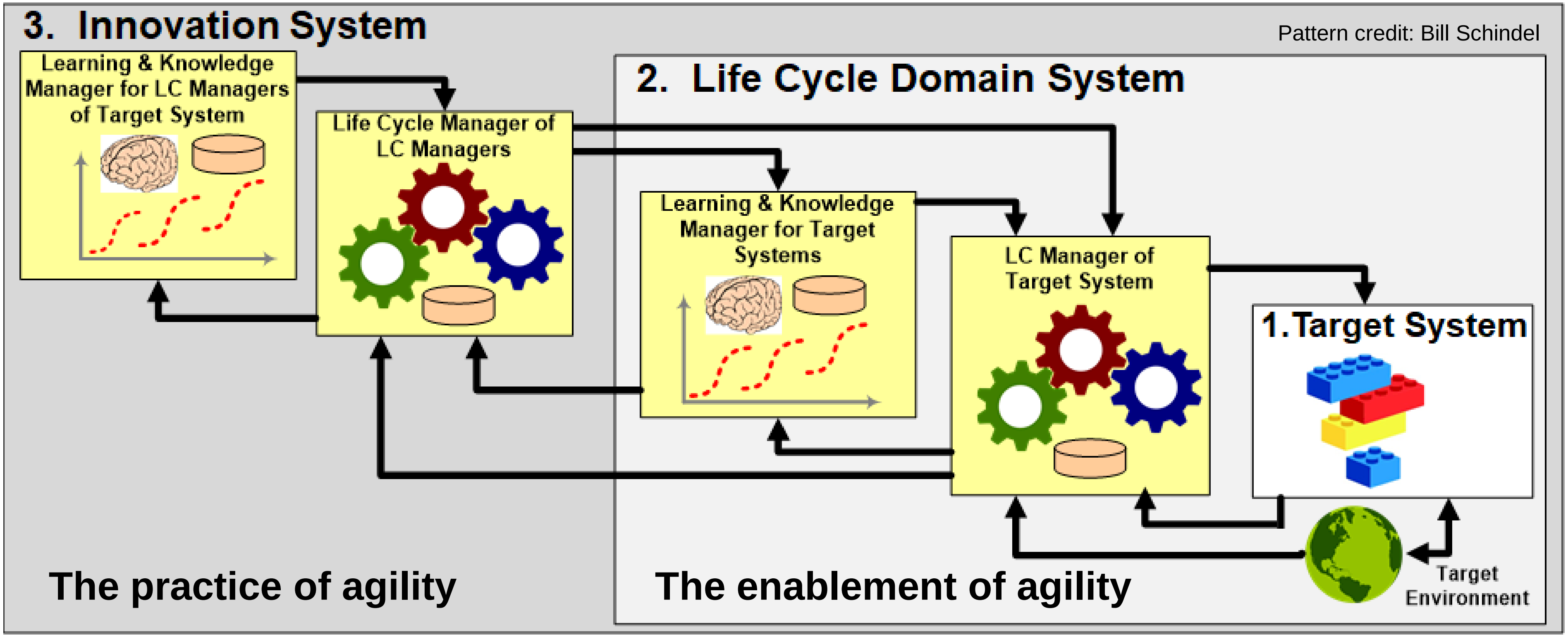
General Agile SE Life Cycle Framework

Asynchronous/Concurrent. Consistent with Systems and Software Engineering — Life Cycle Management — Part 1: Guidelines for Life Cycle Management. ISO/IEC TS 24748-1:2016



Central Awareness Stage Engages All Other Stages

General Operational Pattern
Systems 1, 2, 3 Logical/Behavioral 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.

General 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)**

General Concept of Information Debt

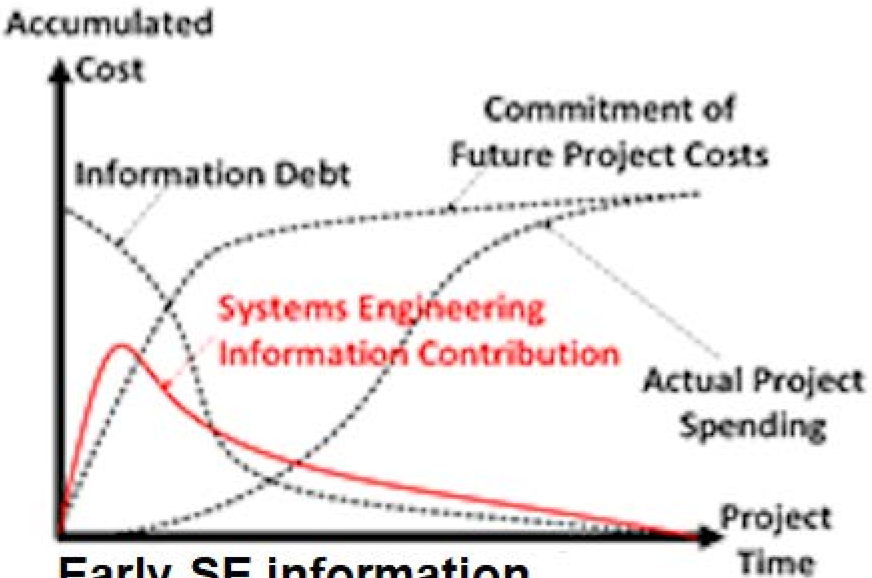
Information Debt expresses the difference between the information currently available and the information immediately needed to support the life cycle.

As an explicit concept this helps us address the perceived tension between Agile Software Development methods and traditional Systems Engineering methods.

Does the Agile Manifesto mean that the project will end with remaining information debt, leaving us with a "working system" but a shortage of needed information?

Thoughtful early stage systems engineering reduces information debt and can be generated without an equivalent surge in systems engineering expense; and can reduce SE expense when thoughtful deferred commitments reduce rework.

Scenario 3 illustrates the worrisome case



Early SE information reduces information debt

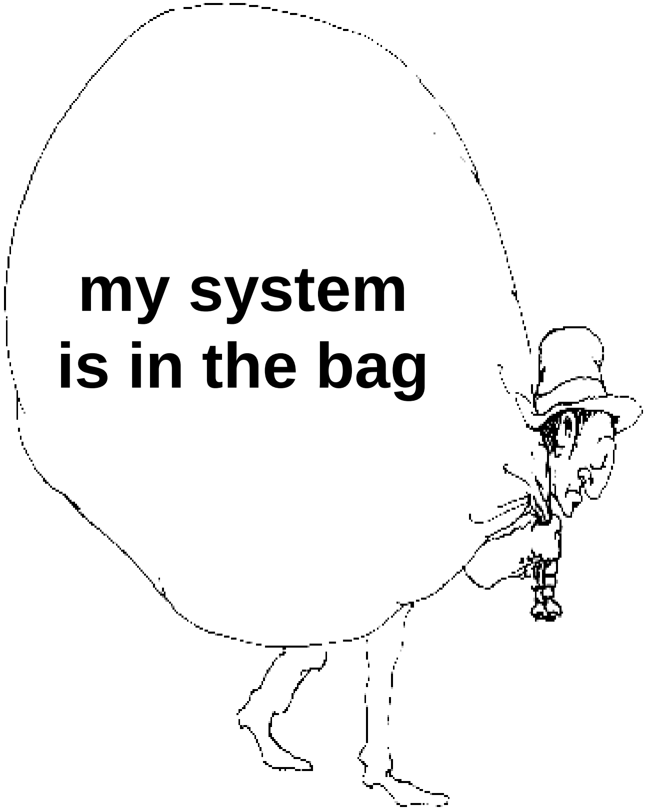
Art Credit: Bill Schindel

Here We Focus on Two Findings:

- ❑ **General Problem-Space Characterization**
- ❑ **General Response Requirements**

other findings will be covered in subsequent webinars

Functional Focus



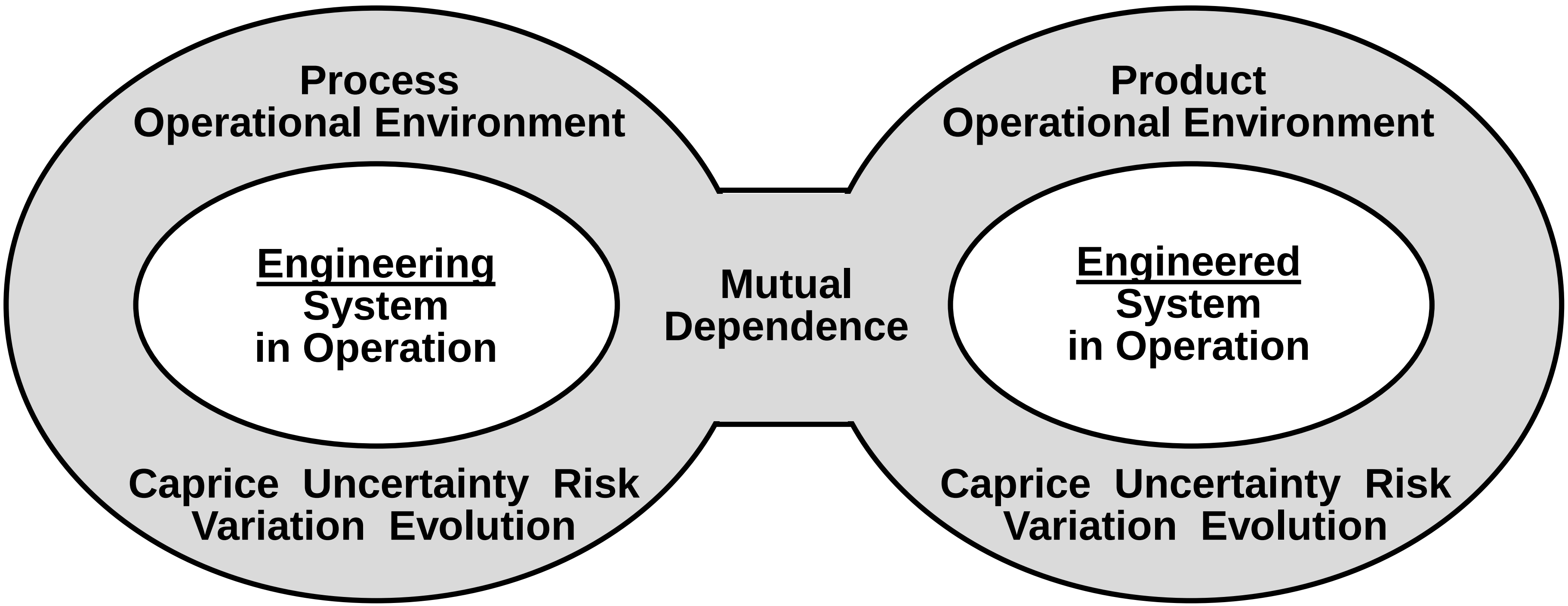
Art by Shel Silverstein

(your system is a disturbance that will cause a reaction)

Art by Shel Silverstein

Two different operational environments requiring agile counterpoint for the systems they encompass

(a first principle)



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

General Problem

Describing the shape and dynamics of the environment, and the requirements necessary for effective dynamic interface.

Well structured problems:

- **Can be described in numerical variables.**
- **Goals can be specified in well defined equations.**
- **Algorithms can find numerical solutions.**

Ill-structured problems:

- **Essential variables are not numeric.**
- **Goals are vague and not quantitative.**
- **Computational algorithms are not available.**

A heuristic technique
is any approach to problem solving, learning, or discovery
that employs a practical method,
not guaranteed to be optimal or perfect,
but sufficient for the immediate goals.

A Little Background

We are all wired to rush ahead with a solution space focus when thinking about system requirements – and devote relatively little effort to understanding the problem space as well.

The literature and process-approaches are heavily slanted this way.

From nature's point of view that works well – as a quick reactive capability to avoid the immediate threatening problem.

Neuroscience and psychology literature has investigated our natural abilities to deal with problems of the past and present, rather than the future – and concludes that we are not wired for that, and may even be counter-wired somewhat.

Thinking about the future of the problem-space is an un-natural act.

Environments

Project-Environment Shaping Elements:

Customers, users, other stakeholders, internal and external project adversaries, funding, resources, technology, competition, evolution, …

A project takes place within an organization, within the larger existing-systems "community" (e.g., competition), and within acquisition reality (e.g., possible termination).

Process-Environment Shaping Elements:

Customers, users, competencies, culture, resources, mixed engineering disciplines, procedures, standards, regulation, certification, evolution, …

Product-Environment Shaping Elements:

Users, owners, maintainers, updaters, operators, product adversaries, competition, evolution, …

General Problem-Space Characterization

CURVE

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

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.

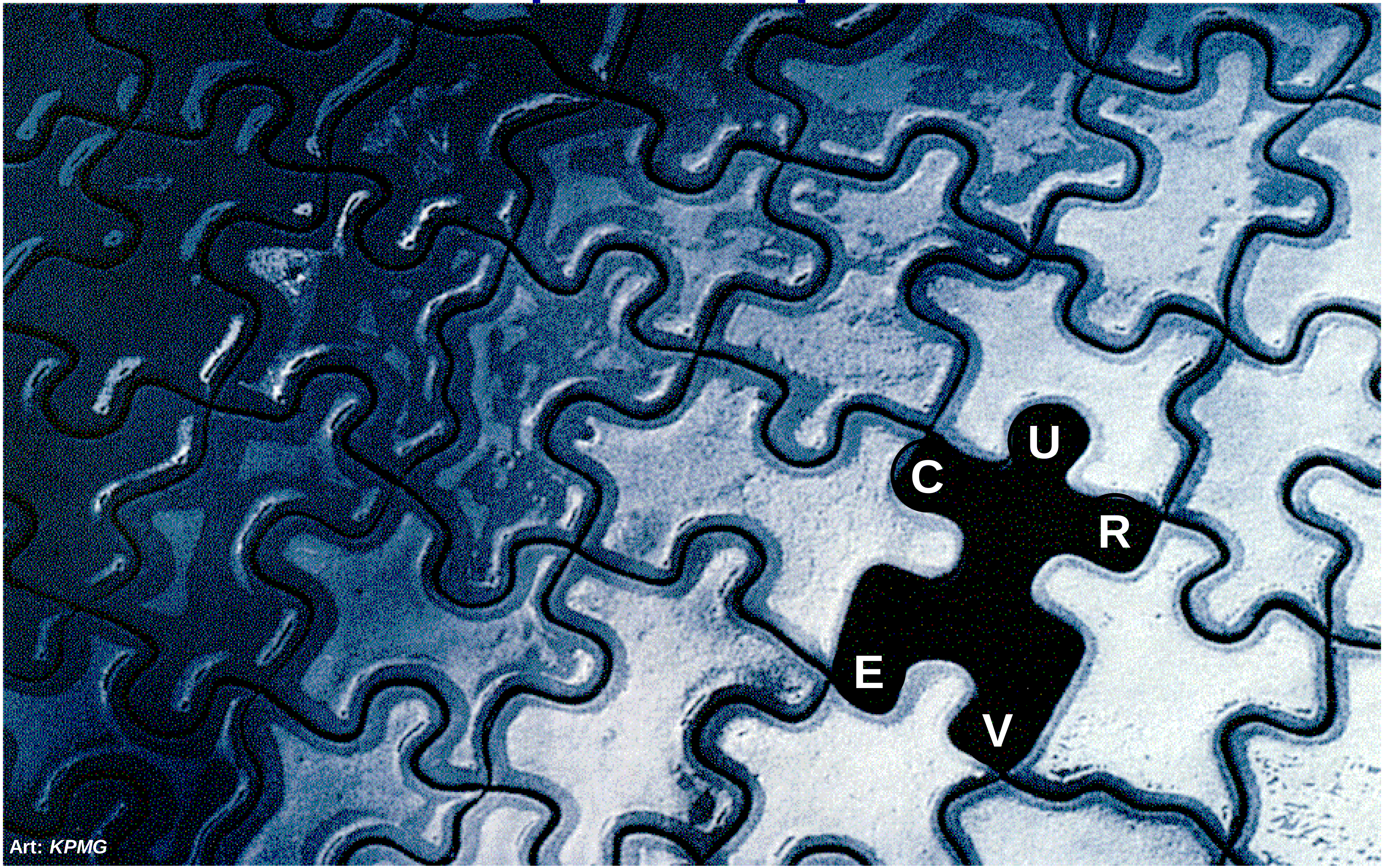
Variation: Knowable variables and associated variance ranges.
Temporal excursions on existing behavior attractor.

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

CURVE High-Level Response-Needs of Concern
Simple Example for System Engineering Process

- Caprice: unknowable situations
  - Obsolescence of solution approach before completion
  - Requirements additions and changes
- Uncertainty: randomness with unknowable probabilities
  - Feasibility of solution design
  - Continuous political and funding support
- Risk: randomness with knowable probabilities
  - Unacceptable cost increases
  - Failure to meet necessary schedule
- Variation: knowable variables and variance range
  - Critical test facility availability
  - Multiple COTS-source performance differences
- Evolution: gradual successive developments
  - Continuous incremental change in targeted operating environment
  - Alternative technology improvement curves (Moore’s law effect)

Your Characterization (Model) of the Environment Defines the Required Shape of the Solution



But all pieces are shape shifters – the puzzle is dynamic – there really isn't a void

Lockheed Martin IFG-TS Example

(www.parshift.com/s/ASELCM-04LMC.pdf)

The Lockheed Martin Aeronautics Integrated Fighter Group (IFG), in Fort Worth, Texas, was motivated to move to an agile system engineering (SE) development methodology by the need to meet urgent defense needs for faster-changing threat situations.

IFG has and is tailoring a baseline Scaled Agile Framework (SAFe®) systems engineering process for a portfolio of mixed hardware/software aircraft weapon system extensions, involving some 1,200 people in the process from executives, through managers, to developers.

The process is referred to here as IFG Tailored SAFe (IFG-TS)

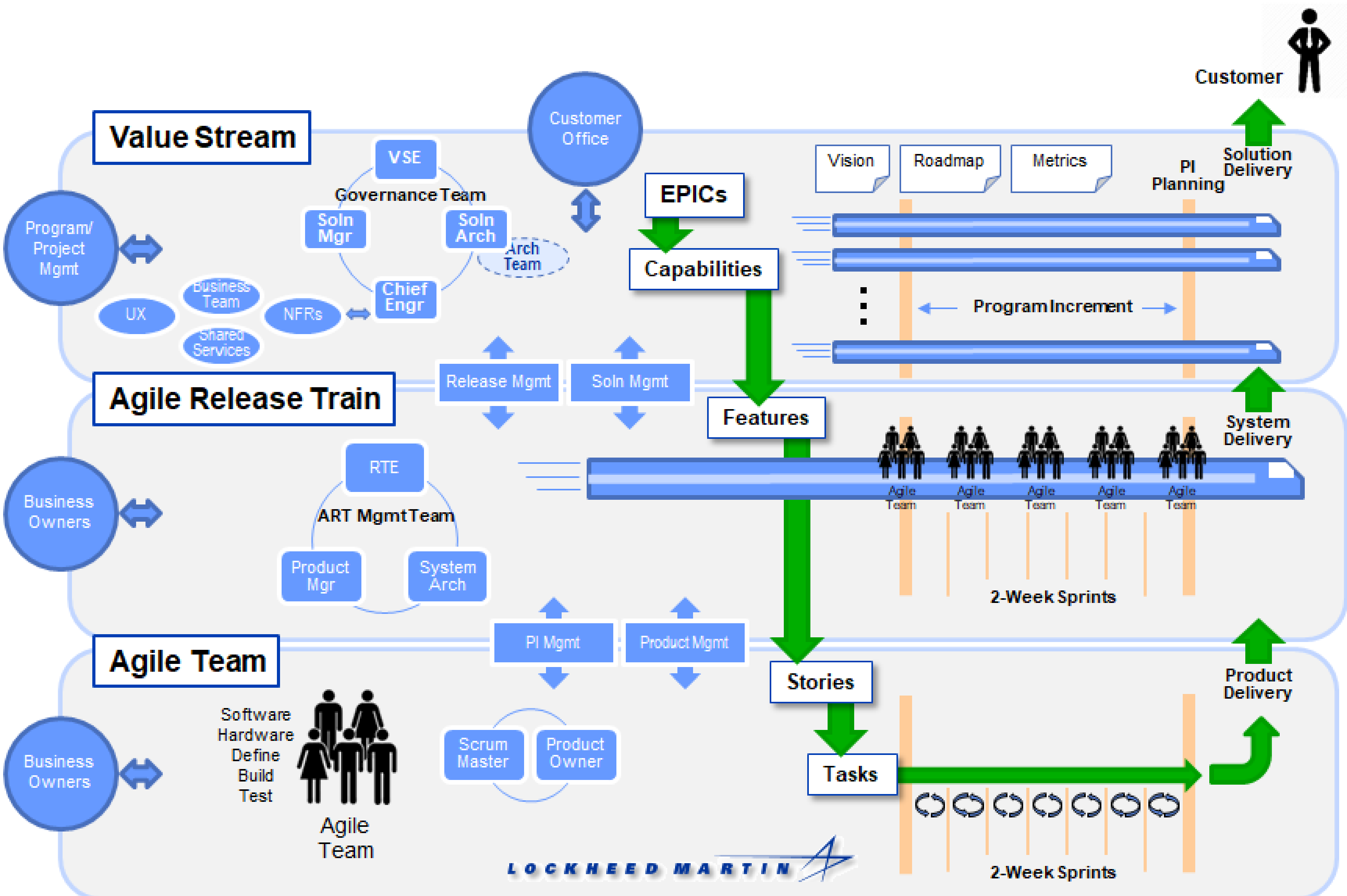
Notably, the SE process is facilitated by a transformation to an Open System Architecture aircraft-system infrastructure, enabling reusable cross platform component technologies and facilitating faster response to new system needs.

The process synchronizes internal tempo-based development intervals with an external mixture of agile/waterfall subcontractor development processes.

This example is as-presented by IFG,
it is instructive but not necessarily comprehensive.

SAFe and Scaled Agile Framework are registered trademarks of Scaled Agile, Inc

IFG-TS Process Operational Model



IFG-TS CURVE Characterization

Caprice: Unknowable situations
CC1: Urgent pre-emptive customer needs, sometimes called Quick Reaction Notice events
CC2: Changes in business environment, e.g., congressional funding commitments or legal requirements
CC3: Project scope change

Uncertainty: Randomness with unknowable probabilities
CU1: Effectiveness of process tailoring
CU2: Contract/customer compatibility with agile approach
CU3: Management support/engagement in agile approach
CU4: Team-member engagement with agile approach

(Tag for requirements tracing)

Risk: Randomness with knowable probabilities
CR1: Cultural incompatibility
CR2: Ability to keep and attract talent
CR3: External stakeholder schedules (e.g. certification)
CR4: Systems of Systems requirements changes

Variation: Knowable variables and ranges
CV1: Multiple-project resource conflicts (e.g. test facilities, key people)
CV2: Subcontractor development compatibility
CV3: System of Systems integration integrity
CV4: Requirements of differing importance levels

Evolution: Gradual Successive Development
CE1: OSA/OMS emphasis
CE2: Customer mission needs
CE3: New compelling technology availability

Reality Factors

Reality Factors

Requirements often assume a relatively benign environment, and tend to focus on the functional response situations. 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…

Reality Factors – TSA Screening Example

Human (Including Customer) Behavior Reality – Human error, whimsy, expediency, arrogance...
- Hangover isn't paying attention, routine produces boredom, fatigue, care for the job.
- Overreaction and stereotyping, subjective standards, training exercises that test only expected procedures.

Organizational Behavior Reality – Survival rules rule, nobody's in absolute control...
- Performance metrics, knee jerk open ended reaction.
- Counterproductive incentives, airline circumvention.

Technology Pace Reality – Accelerating technology and security vulnerability...
- Scanning machines.
- Sniffing machines.

System Complexity Reality – Incomprehensible, networked, unintended consequences, emergence...
- Training for all (un)reasonably possible threats.

Partners/customers/employees with different ethics, values, infrastructures, culture...
Ethical and cultural differences, passengers of all kinds of background.
- Reliance on flight origin for certain standards.

Partially-Agile Enterprise Reality (Faddish Practices) – Outsourcing, COTS policies...
- Insufficient cross-training.
- Outsourced Non-TSA security personnel for night shift.

Agile Customers/Competitors/Adversaries – Distributed, collaborative, impatient, innovative...
- Bad guys watch and find weaknesses in repetitive patterns, share info on Internet.

Other?
- ?

Real (no-attribution) Example

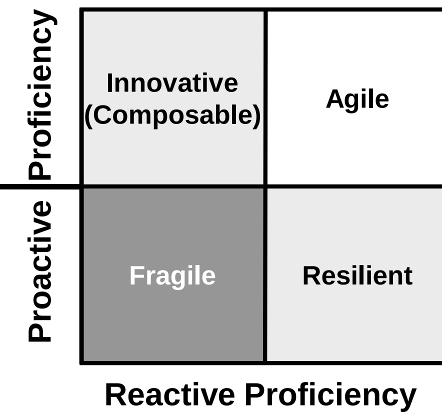
Reality Factors
Human (Including Customer) Behavior – Human error, whimsy, expediency, arrogance... • Hero mentality, “just a fad”, “not the way we do things here”, programmatic and project commitments have priority over what’s best for the system (but the focus is on “my piece”). • Culture shift of customer and leadership external to the project.
Organizational Behavior – Survival rules rule, nobody's in absolute control... • Commitment pipeline that constrain process/behavior change (don’t add risk to current contract), business measures don’t align with technical measures. • Decentralization of control and decisions.
Technology Pace – Accelerating technology and security-vulnerability introductions,... • Technology forecasting critical to architectural runway viability. • Cyber threat handling methods/resilience. • We are being asked to accept software from external vendors and plug and play.
System Complexity – Incomprehensible, unintended consequences, emergence... • Simultaneously partners and competitors. • Collaboration overly complex due to closed environment. • Integrated vs. federated avionics.
Partners/customers/employees with different ethics, values, infrastructures, culture... • Work-share with in country resources. • Process change disruptive to some foreign partners. • Foreign offset requirements.
Partially-Agile Enterprise Faddish Practices – Outsourcing, COTS policies/affects... • Interaction with traditional Systems Engineering models like waterfall. • Fighting the faux-Agile myths. • Claim to follow Agile but do not.
Agile Customers/Competitors/Adversaries – Distributed, collaborative, impatient, ... • Customer’s mission is changing rapidly. • Our customers are moving to agile methods. • Some business models rely on late discovery for additional revenue to make late changes.

Response Situation Analysis

8 Domains of Response Requirements

for Response Situation Analysis (RSA)

Response Domain		General Characteristic
Proactive	Creation (and Elimination)	**Proactive** — Innovative/Composable; Creates Opportunity; Takes Preemptive Initiative
	Improvement	
	Migration	
	Modification (of Capability)	
Reactive	Correction	**Reactive** — Resilient; Seizes Opportunity; Copes with Adverse Events
	Variation	
	Expansion (of Capacity)	
	Reconfiguration	



Response Requirements

**Response requirements are
system operation-time requirements,
not system design-time requirements.**

**They should be stated as operational needs,
independent of possible solution strategies
which will evolve with time.**

Response requirements are generally timeless.

Proactive Response Domains

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	

Creation/Elimination

What artifacts/resources must the system create during operation, and eliminate as situational evolution causes obsolesce of artifacts/resources?

The distinguishing feature is the creation of something new or reincarnated that is not currently present during operation.

Situations to identify are those that require response activity configuration during operation, and those that require new resources for employment in response activities.

Improvement

What improvements in system response performance will be expected over the system's operational life?

The distinguishing feature is performance of existing response capability, not the addition of new response capability.

Situations to identify are generally those involving competencies and performance factors, and are often the focus of continual, open-ended improvement campaigns.

Migration

What evolving technologies and opportunities might require future changes to the infrastructure?

The distinguishing feature is a need to change the nature of the plug-and-play resource-interconnection infrastructure.

Situations to identify are generally those that enable the transition to possible and potential next generation capabilities.

Modification (of capability)

What evolving technologies and opportunities might require modification of the available resources and resource pools?

The distinguishing feature is a necessary change in available resource capabilities.

Situations are generally those that require something unlike anything already present, or the upgrade or change of something that does exist.

IFG-TS Proactive Response Requirements

What must the process be creating or eliminating during its operational life?

RC1: A safe environment for people to take prudent risks (CR2)<-Traced to CURVE
RC2: Risk identification and mitigation plans at project and functional level (CC2/3, CU4)
RC3: Loading plans with spare capacity for unknowns/inaccurate planning (CV1)
RC4: Architectural development horizon to accommodate variation (CC3, CV4, CE2)
RC5: Experience accumulation (CU1)

What performance will the process be improving during its operational life?

RI1: System level optimization vs. local/functional optimization (CU1/4, CR1)
RI2: Responsiveness to customer needs (CC1)
RI3: Stakeholder, developer, and supplier alignment (CU2/3, CR1/3, CV2)
RI4: Customer acceptance rate from acceptance testing events (CC1)
RI5: Agility of existing integrated system (CU1, CE1)
RI6: Awareness of evolving process effectiveness (CU1)
RI7: Effectiveness of distributed knowledge exchange (CU1, CR2, CV2)

What anticipated events will require a change in process infrastructure?

RM1: Evolution of customer missions (CE2)
RM2: Cybersecurity and related standards (CC3, CU2, CR3)
RM3: DoD Open Missions approach (CE1)

↑

(Tag for feature tracing)

What modifications might need made during operational life?

RA1: Personnel that make up a team (CV1, CR2, CV4)
RA2: Test infrastructure to maintain throughput (CV1)
RA3: Modification in project-specific details of the operational model (CU1)
RA4: Addition of subcontractor with new technology and/or process expertise (CE3)
RA5: Reallocation of work between prime contractor and other entities (CC1, CV1)

Reactive Response Domains

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, equipment malfunctions, 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	

Correction

What types of response activities might fail in operation and need correction?

The distinguishing feature is a dysfunction or inadequacy during attempted response.

Situations to identify are those that require a recovery from response malfunction, recovery from unacceptable side effects of a response, and inability to assemble an effective response.

Variation

What aspects of operational conditions and resources vary over what range when response activities must be assembled?

The distinguishing feature is predictable but uncertain variance.

Situations to identify are those that manifest as variances in resource availability, resource performance, and resource interactions.

Expansion/Contraction

What are the upper and lower bounds of response activity needs?

The distinguishing feature is capacity scalability.

Situations to identify are those that can be satisfied with planned capacity bounds, as well as those that have indeterminate and unbounded capacity needs.

Reconfiguration

What types of situations will require reconfiguration of response activities in order to respond effectively?

The distinguishing feature is the configuration and employment of available resources for new or reincarnated response needs.

Situations to identify are those that are within the system mission boundaries, and that may require a reconfiguration of an existing response activity, perhaps augmented with removal of resources or addition of available resources.

IFG-TS Reactive Response Requirements

What can go wrong that will need a systemic detection and response?

RW1: Leadership and stakeholder churn that change vision and expectations (CC2, CC3, CU3)

RW2: Non detection of variances (CU4. CV1, CV3)

RW3: Insufficient identification and management of opportunities and risks (CR1, CR4)

What process variables will need accommodation?

RV1: Tailored process self-improvement and policing (CU1, CU4)

RV2: Alignment and coordination of PI Planning (CC1, CC3, CU1, CV4)

RV3: Organizational acceptance and adoption of tailored process (CU3, CU4, CR1)

What elastic-capacity will be needed on resources/output/activity/other?

RE1: System test capacity (CV1)

RE2: Development capacity band to avoid disruption when work is more than expected in volume or difficulty (CC1, CC3, CV3, CV4)

What types of resource relationship configurations will need changed during operation?

RR1: Team-personnel assignments among multiple weapon systems (CC1, CR2, CV1)

RR2: Work reassignments to match team capacities (CU1, CR2, CV1)

RR3: Priorities for requirements (CC3, CV1, CV4)

RR4: Acquisition procedures/policies/contract for situational and objectives reality (CC1, CU2, CE2, CE3)

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* list of *response needs* –

which if accomplished,

will stretch the envelope of agile response capability

to encompass <u>all</u> necessary response needs,

even if they were not on the list.

General Response Requirements

Domain		Response Requirements
Proactive	Creation (and Elimination)	What will the process be creating or eliminating in the course of its operational activity? • Opportunity and risk awareness/knowledge • Response options • Acculturated memory • Decisions to respond • Life cycle sustainment documentation
	Improvement	What performance will the process be expected to improve during operational life cycle? • Awareness/Sensing • Effectiveness of response actions/options • Memory in acculturation, inventoried response options, and ConOps
	Migration	What major events coming down the road will require a change in the process infrastructure? • New fundamentally-different types of opportunities and risks
	Modification (Add/Sub Capability)	What modifications in resources-employed might need made as the system is used? • Response action appropriate for specific response need • Personnel appropriate and available for a response action
Reactive	Correction	What can go wrong that will need systemic detection and response? • Insufficient/inadequate awareness • Ineffective response actions/options • Wrong decisions • Information debt
	Variation	What process variables will need accommodation? • Effectiveness of response actions/options • Effectiveness of response evaluation
	Expansion (and Contraction) of Capacity	What elastic-capacity ranges will be needed on resources/output/activity/other? • Capacity to handle 1-? critical response actions simultaneously
	Reconfiguration	What types of resource relationship configurations will need changed during operation? • Elements of response actions/options • Response managers/engineers

Wrap Up

Three Tools for Developing
Problem-Space Response Requirements

- ❑ **CURVE**
- ❑ **Reality Factors**
- ❑ **Response Situation Analysis**

An INCOSE Fellows Discussion
Derek Hitchens – May 2017

“Suppose we, corporately, were to accept that systems engineering was the whole thing, from problem space to solution, and that INCOSE – *so far* – had only touched upon the final, post requirements phase for one ‘kind’ of solution, then we, INCOSE could go about reconstituting the complete pre-requirements phases, leading from the problem space to the systems design and could also conceive and design one, or more, INCOSE SE Methodologies, along with tools and methods.

Oh! and how about *an open system is a complex, organized whole that exchanges energy, information and substance with its environment, and adapts to the exchange.* (For the science-minded, that is either *Newton’s Third*, or *Le Chatelier’s Principle*)”

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.

Agility in Systems Engineering – Findings from Recent Studies. Dove, R., W. Schindel. 2018. Working paper, April 14. www.parshift.com/s/ASELCM-AgilityInSE-RecentFindings.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

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/2018_09_MeetingEval

www.surveymonkey.com/r/2018_09_MeetingEval

Look in GlobalMeet chat box for cut & paste link.

Or use cell phone to capture this QR Code:

Slide presentation can be downloaded now/anytime from:

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

Recording will be there in the library tomorrow.