Domain Independent
Agile Systems Engineering Life Cycle Model Fundamentals
Discovery Project

NDIA Systems Engineering Conference

27-30 Oct 2014, Springfield VA

Rick Dove
575-586-1536, rick.dove@parshift.com

CEO/CTO, Paradigm Shift International, Taos County, New Mexico
Chair, INCOSE Agile Systems & Systems Engineering Working Group
Chair, INCOSE Systems Security Engineering Working Group
Adjunct Professor, Stevens Institute of Technology

Presentation download at:
www.parshift.com/s/ASELCM-DevelopmentProject.pdf
Asynchronously updated, last update 2-Nov-2014

INCOSE Corporate Advisory Board (CAB)

Top Five CAB Priorities:

1) SE Professional development
2) Agile/Expedited methods
3) Effective Trade Studies
4) Product lines, re-use
5) Better Value proposal for INCOSE and SE

CAB workshop 27-Jun-2014 to clarify bullet 2:

- Bechtel
- Ford
- Honeywell
- Ministry of Defence (UK)
- Pacific Northwest Nat'l Lab
- Raytheon
- Virginia Tech

Clarifying the Issues of CAB Agile-SE Priority

What the CAB workshop clarified on Agile Expedited Methods priority:

1. **Clarity/consistency on what agile means independent of the software practice.**
2. **Guidance on when/where to use an agile approach.**
3. **Integrating agile approach concepts with planned approach concepts.**
4. **Systems as works in process after deployment**
5. **How to pivot a project effectively when feedback dictates a path change.**
6. **Short cycle constant evolution – e.g., counter-IED "systems"**
7. **Long cycle constant evolution – e.g., 20-year design/build for complex plants.**
8. **Meaningful WIP measures when an agile approach is employed.**
9. **Dealing with hardware design/build timeframes and sunk costs.**
10. **Case studies.**

NOTES:

- **Universal dissatisfaction among this group with the Agile SW Manifesto as a guide for agile SE.**
- **Conclusion: all needs are being addressed by the Agile Sys & SE WG, or will be in the Agile SE Life Cycle Model project.**

What is an SE Life Cycle Model?

Systems and software engineering — Life cycle management — Part 1: Guide for life cycle management, ISO/IEC TR 24748-1:2010(E)

3.2.1 System life cycle model

Every system, whatever the kind or size, inherently follows some life cycle, evolving from its initial conceptualization through its eventual retirement…

A life cycle model, then, is a decision-linked conceptual segmentation of the definition of the need for the system, its realization as a product or service, and its utilization, evolution and disposal.

A system life cycle model is typically segmented by stages to facilitate planning, provisioning, operating and supporting the system-of-interest.

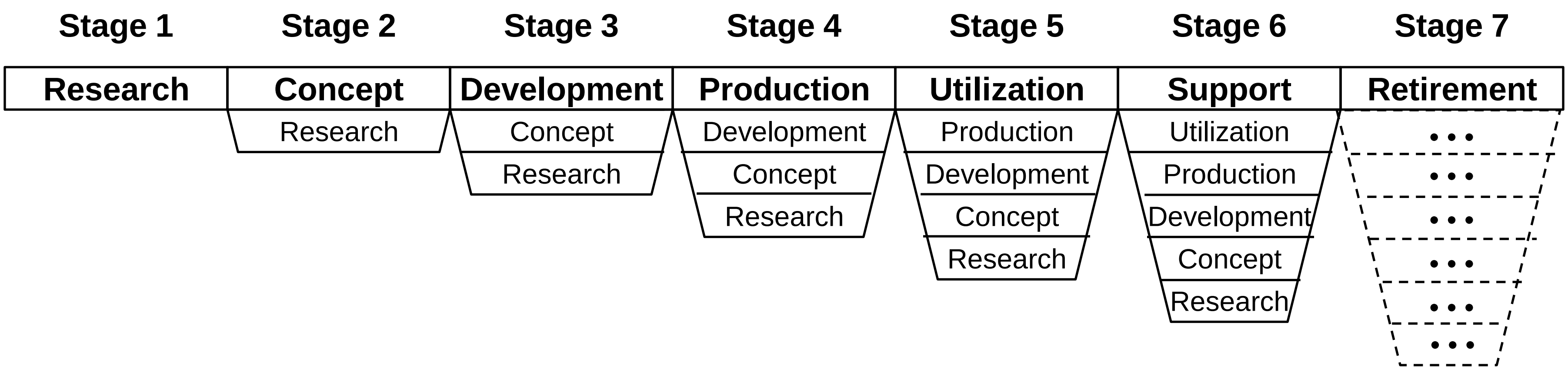
These segments provide an orderly progression of a system through established decision-making gates to reduce risk and to ensure satisfactory progress.

As stated before, it is the need to make a decision to specific criteria before a system can progress to the next stage that is the most important reason for using a life cycle model.

Notes:

- Implies, but does not say, an SOI is in one and only one stage at any time.
- An Agile SE Life Cycle Model is distinguished from waterfall by allowing non-sequential stage progression and multiple-stage activities simultaneously.
- Key is the decision criteria that permits/demands any stage's process activity.

Linear Framework of Agile SE Life Cycle Model



Depicts constant evolution of all prior ISO/IEC 15288 life cycle stages as the life cycle progresses through maturity

Retirement in this framework depiction is the terminal stage for the entire system – it likely has prior-stage process "concepts," but they are unrelated to the creation & utilization of the deployed system. However, subsystem retirement happens constantly, as portions of the system are refreshed, and needs special consideration in an Agile SE LCM, not depicted here.

Diagram of Asynchronous-Stage Agile SE-LCM

Systems and software engineering — Life cycle management — Part 1: Guide for life cycle management   ISO/IEC TR 24748-1:2010(E)

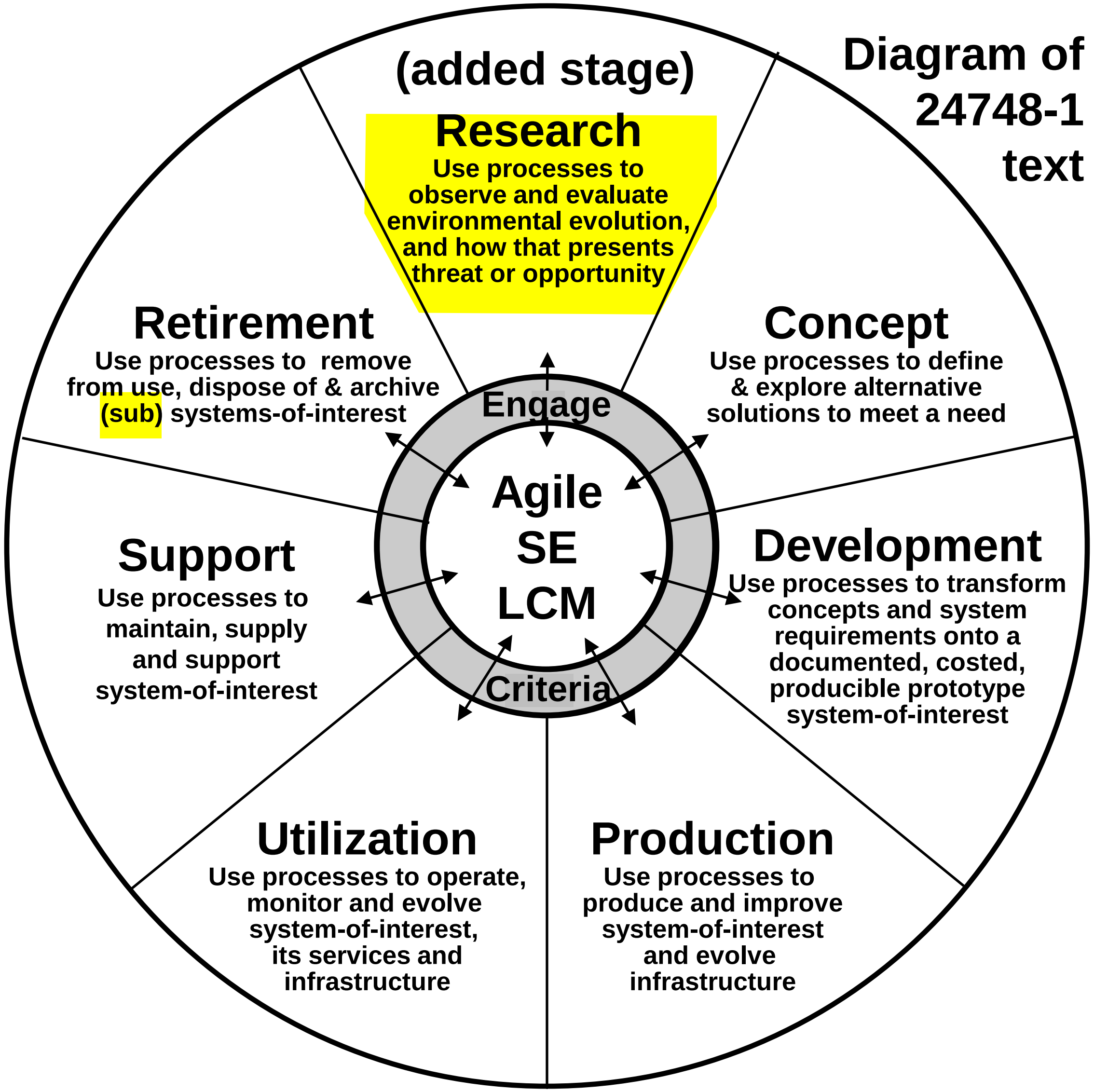
Section 5.5.5 (p. 32):

"… to convey the idea that 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."

"Further, the text in the model indicates that one applies, at any stage, the appropriate life cycle processes, in whatever sequence is appropriate to the project, and repeatedly or recursively if appropriate."

"While this may seem to be a total lack of structure, indeed it is not."

"Rather, the structure has well defined parts that can be juxtaposed as needed to get the job done, flexibly but still in a disciplined manner, just as a real structure would be created."



Seven asynchronously-invoked stages can be engaged repetitively and simultaneously to achieve benefit when engagement criteria are met

Project: Agile SE Life Cycle Model (ASELCM) Fundamentals

Objectives:

A. Discover generic principle-based life-cycle stages/processes/activities
   that can be embraced and applied intuitively,
   rather than compromised by situational reality factors,
   for dealing with uncertain, unpredictable, evolving SE environments.

B. Cover four or five types of SE projects:
   1. discovery (verifying requirements and solution feasibility),
   2. programmatic (Systems and SoS from proven components),
   3. approach (e.g., ICSM methodology and product line architecture),
   4. quick reaction (rapid development and fielding), and
   5. evolving (continuous change of system operational viability and opportunity, rapid sequential generations).

C. Recognize that ASELCM process activities within multiple life cycle stages may be occurring simultaneously, particularly after initial deployment.

System life cycle processes

ISO/IEC 15288

Agreement processes
- Acquisition
- Supply

Enterprise processes
- Enterprise environment management
- Investment management
- System life cycle processes management
- Resource management
- Quality management

Project processes
- Project planning
- Project assessment
- Project control
- Decision-making
- Risk management
- Configuration management
- Information management

Technical processes
- Stakeholder requirements definition
- Requirements analysis
- Architectural design
- Implementation
- Integration
- Verification
- Transition
- Validation
- Operation
- Maintenance
- Disposal

Special process
- Tailoring

18 Processes of Interest

Source: ISO/IEC JTCI/SC7/WG7 presentation on ISO/IEC 15288.

Project Artifacts (Products)

1. An instructive technical report describing a generic Agile SE Life Cycle Model with supporting exemplar case studies. The model will support rather than supplant common agile systems-and-software SE processes.

2. The Life Cycle Model descriptive organization will be facilitated and augmented with a PBSE/MBSE three-level hierarchy (next slide).

3. Collateral technical information in briefer form and focus is anticipated as papers targeted for relevant SE journals and conferences.

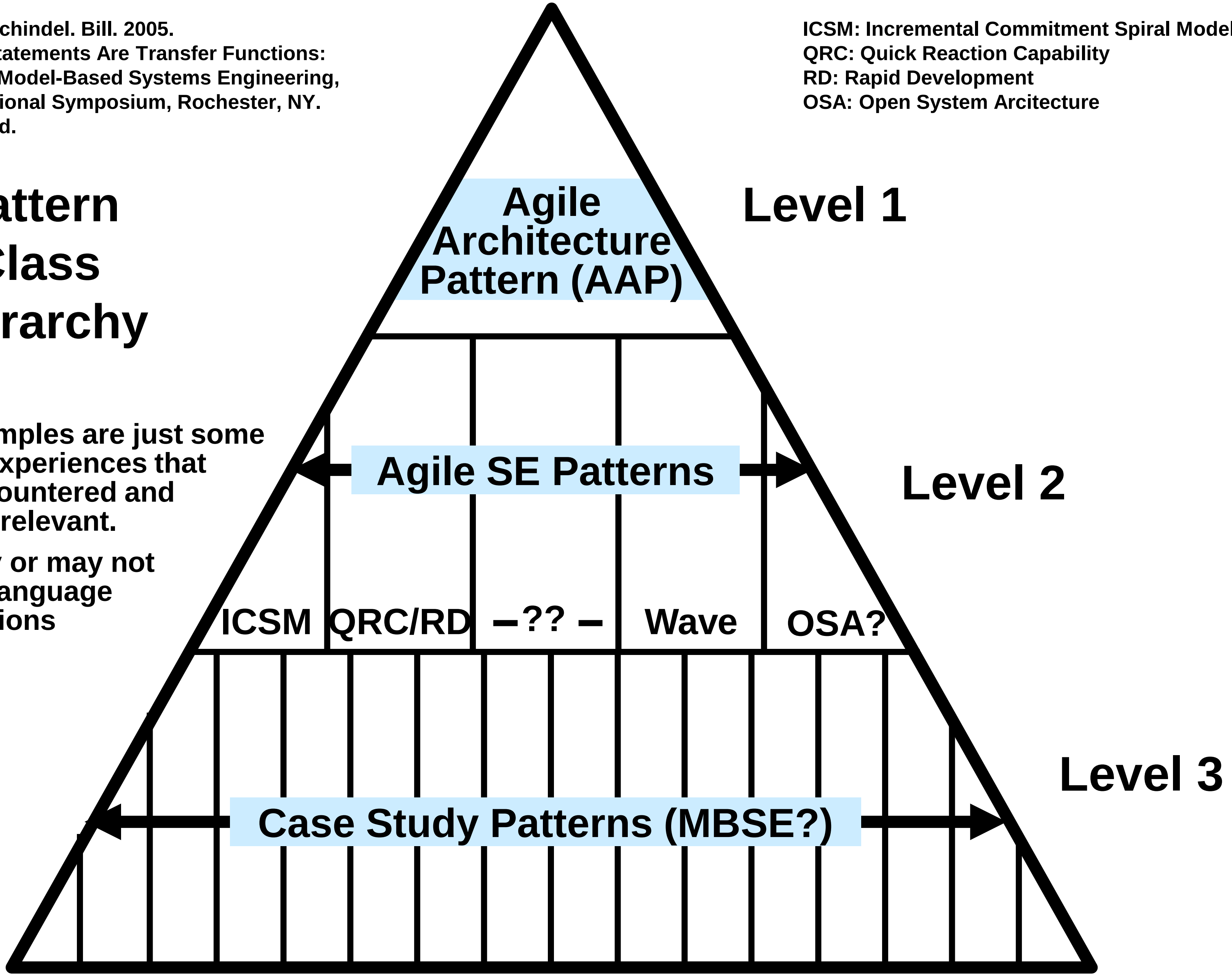
Estimated project report completion is later half of 2016

MBSE Pattern-Based System Engineering (PBSE)

Adapted from: Schindel. Bill. 2005.
Requirements Statements Are Transfer Functions:
An Insight from Model-Based Systems Engineering,
INCOSE International Symposium, Rochester, NY.
Best Paper award.

ICSM: Incremental Commitment Spiral Model
QRC: Quick Reaction Capability
RD: Rapid Development
OSA: Open System Arcitecture

Pattern Class Hierarchy



Notes:

Level 2 examples are just some workshop experiences that may be encountered and considered relevant.

Level 3 may or may not use MBSE language for descriptions

Strategies[1/2]

1. The project will be guided by ISO/IEC TR 24748-1:2010 and recognize six primary continuous and potentially simultaneous stages of process activity: Research, Concept, Development, Production, Utilization, and Support. A seventh terminal stage, Retirement, may be considered if anything unique to agile SE is discovered during the project. Guidance will also be taken from ISO/IEC 15288 to specifically analyze the 18 Project and Technical Processes.

2. Workshop hosts will provide discussion and presentation of one agile-SE experience with a completed project for analysis, and a discussion/presentation of one SE approach in need of more agility to fuel a synthesis exercise based on accumulated learning.

3. Traveling Participants will be open-call volunteers limited to 10 per workshop, with each participant required to attend a minimum of 3 workshops.

Strategies[2/2]

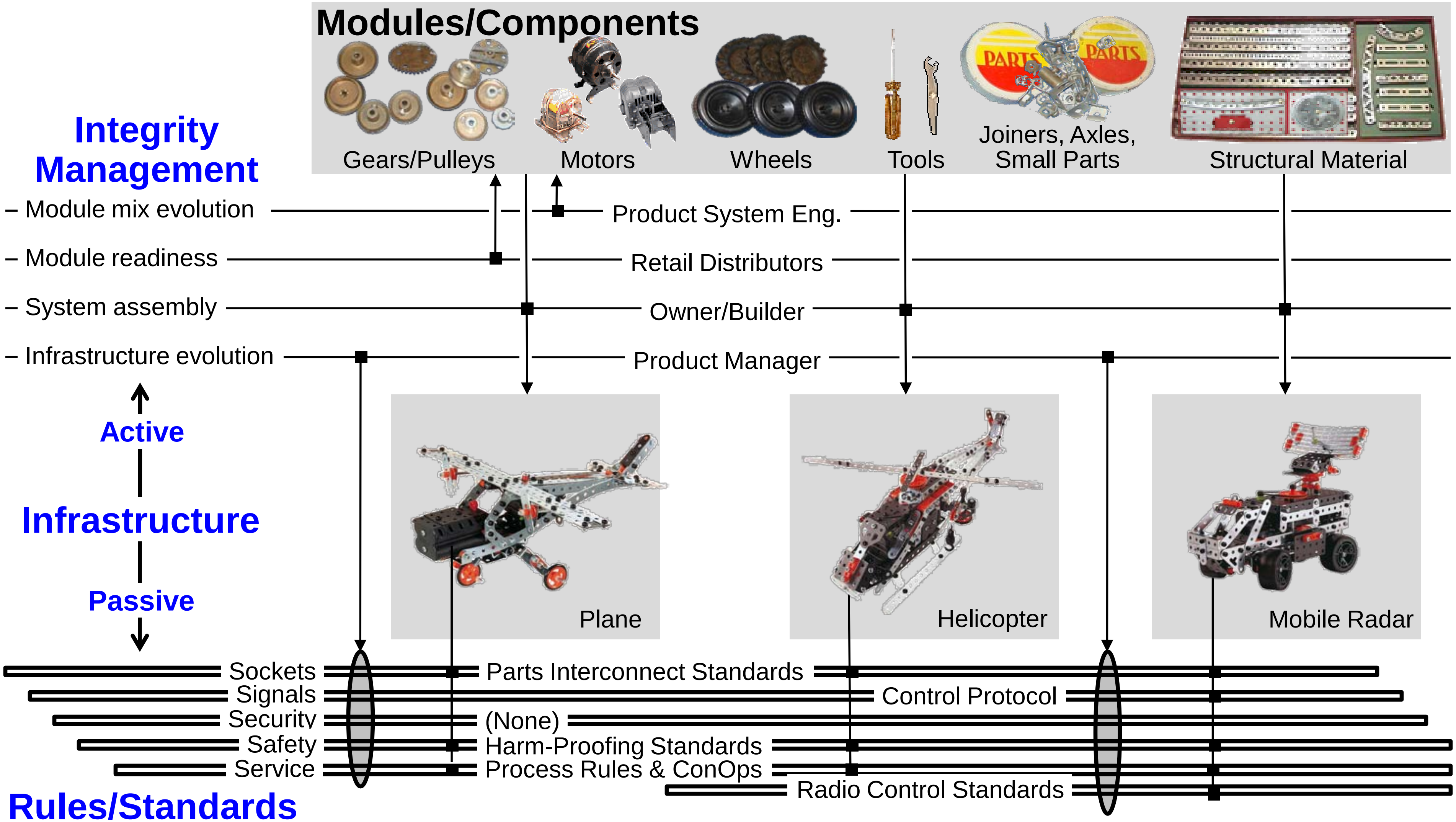
4. **With a structured analysis approach, analyze experience from employed agile SE practices in both defense and commercial SE projects that involve combined aspects of software, hardware, and wetware (management, engineering, operator, maintainer). Management includes supplier and acquirer project management aspects.**
   - **Discover and justify ("why" reasoning) common necessary and sufficient agile SE needs and reality factors, independent of what agile SE practice may be entrenched, favored, under consideration, or subsequently adopted.**
   - **Discover and justify ("why" reasoning) principle-based stages, processes, and activities that satisfy the project objectives.**

5. **With a structured synthesis approach, apply discovery and provide benefit to workshop hosts and participants with an application of accumulated learning to a relevant host opportunity or problem.**

6. **Workshop structure, analysis tools, and synthesis tools will be guided by a prior workshop series (Dove 1998) that discovered fundamental architecture and design principles necessary & sufficient for agile systems & processes.**

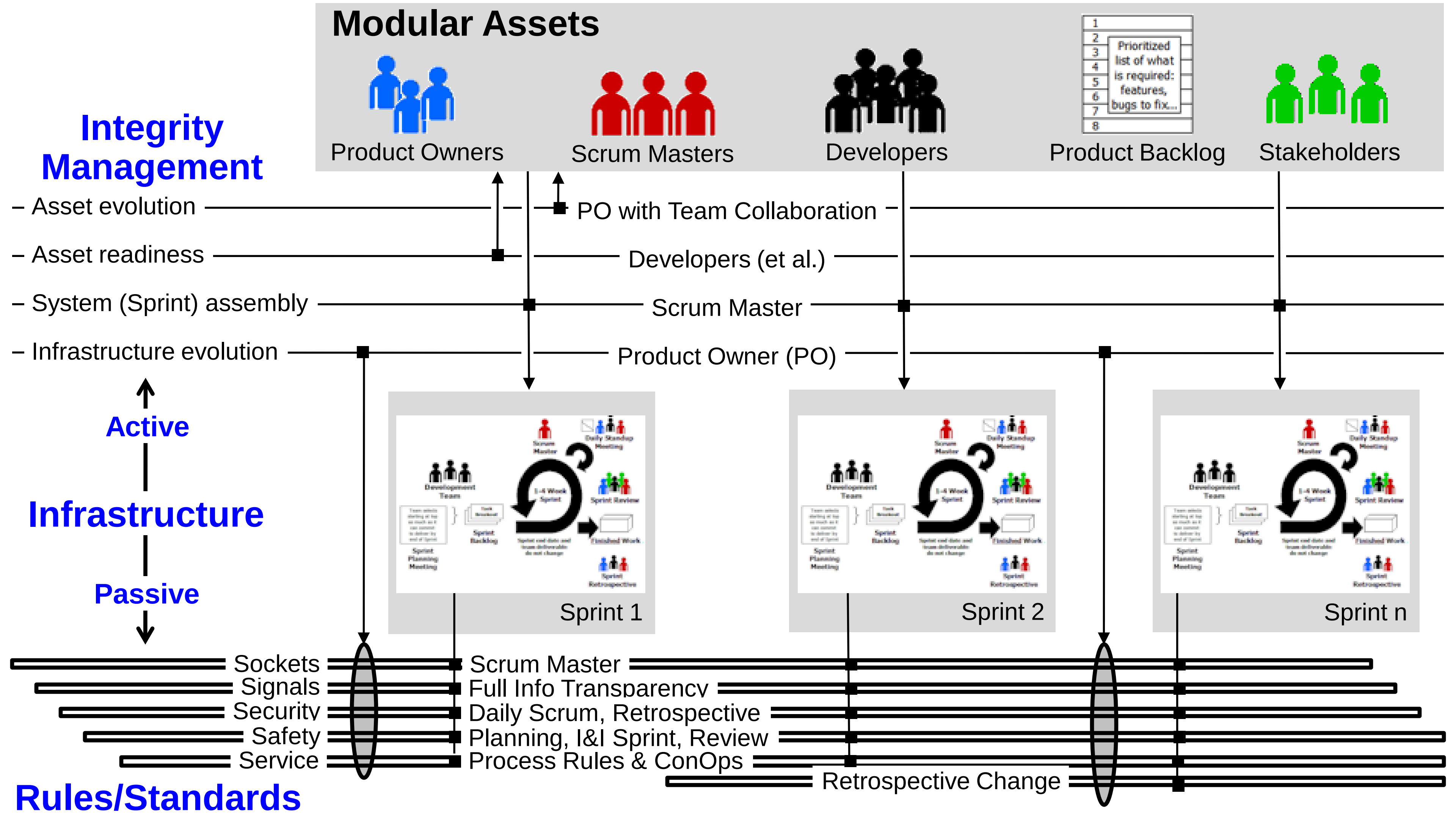
Agile Architecture Pattern (AAP)
Concept: System Response-Construction Kit

Details in (Dove & LaBarge 2014)



Conceptual Example of Agile Architecture Pattern (AAP)

Details in (Dove & LaBarge 2014)



This example was developed for a different purpose, and will not survive the ASELCM workshop process – but it demonstrates the AAP conceptual structure.

Conceptual Example of Response Situation Analysis (RSA)

Details in (Dove & LaBarge 2014)

	Change Domain	
Proactive	Creation (and Elimination)	• management strategy; project team; system requirements • system architecture; system design; activity plans • V&V/test plans; team collective understanding & learning
	Improvement	• activity effort estimating • activity completion to schedule • reducing uncertainty and risk (earlier)
	Migration	• new technology availability • project scope change • lean process principles
	Modification (of Capability)	• new team member unfamiliar/uncomfortable with agile SE • new environmental situation
Reactive	Correction	• wrong requirement • inadequate developer • non-compliant supplier • failed V&V/test
	Variation	• expertise and skill levels among team members • allowable deliverable performance range • availability, interaction, customer involvement expertise
	Expansion (of Capacity)	• 2x (or half x) project scope change • x to y engineers distributed across n to m locations
	Reconfiguration	• unanticipated expertise requirement • development activity-sequence priority change

Conceptual Example of Environmental Reality Factors

RSA exercises often assume a reasonably behaved and supportive environment, and tend to focus on the system's internal functional response situations. This framework tool moves the analysis into the external environment.

Reality Factors
Human Behavior: **Non-team behavior, error, expediency, uncommitted customer rep, …**
Organizational Behavior: **Change in stakeholders, corp priorities, resource access, ...**
Technology Pace: **Evolving technology, testing trade-offs, ...**
Complexity: **External interface misunderstandings, emergent interaction affects, ...**
Globalization: **Partners/teams with different ethics, cultures, infrastructures, …**
Partially-Agile Enterprise Concepts: **Outsourcing, COTS affects, ...**
Agile Customers/Competitors/Adversaries: **Continuous evolution, high external collaborative knowledge, environmental innovation, …**

Planned (Roughly) Workshop Agenda

----------- Day 1 – 8 hours of structured work

2.00 – Introductions, objectives, workshop agenda structure, tools and processes, accumulated learning review.
2.00 – Host process presentation and Q&A – with UURV(E) presented (advance guide provided to host outlining the points we need to hear and discuss, with an example).
Lunch(one hour lunch allows informal conversation)
2.00 – Break-out analysis of RSA/RF (two separate teams doing identical analysis on total SE process overview)
2.00 – Brief-out: Analysis results and discussion.
Dinner (host-funded for all participants)

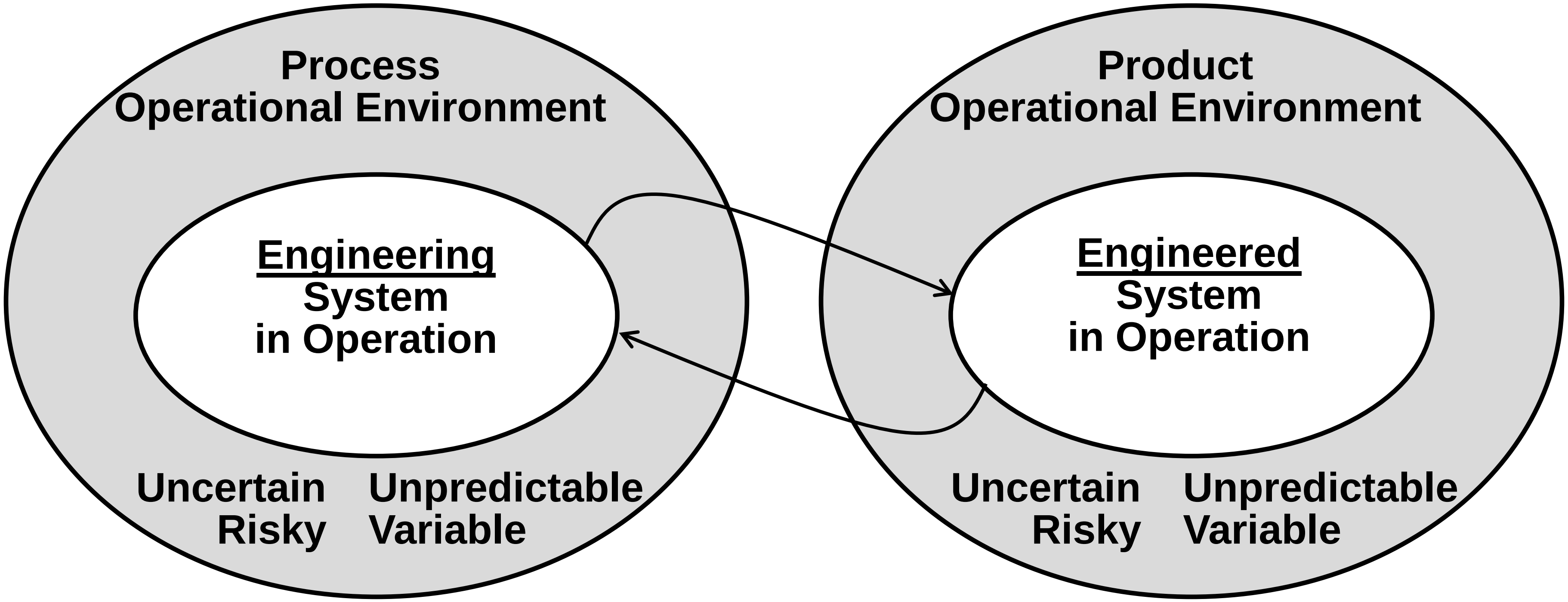
----------- Day 2 – 8 hours of structured work

1.00 – Review of yesterdays salient learning
3.00 – Host presentation and Q&A of 18 processes (guide provided to host outlining the points we need to hear and discuss, with an example) (1 at a time for 6 Hr with 15288 template)
Lunch(one hour lunch allows informal conversation)
2.00 – Break out ties 18 processes to RSA/RF issues, and develops AAP of SE process overall.
2.00 – Brief-out: Analysis results and discussion

----------- Day 3 – 8 hours of structured work

1.00 – Review/discussion of yesterday's salient learning
2.00 – Host presentation and Q&A of problem to solve (in any form they wish), with UURV(E) captured
1.00 – Break out synthesis exercise – Key RSA issues that need process activity solutions, stage engagement criteria, key AAP elements
Lunch(one hour lunch allows informal conversation)
2.00 – Break out cont. - Synthesis exercise at overall process level – Key RSA issues that need process activity solutions, general ACP elements
1:30 – Brief out and wrap up
0:30 – Reflection on the workshop process, tools, learning, and results

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 design an agile product

Action Plan

- **12-15 (TBD) three-day structured workshops will be conducted at host sites in the US and Europe to analyze a variety of different types of agile SE experiences.**
- **Workshops are anticipated to begin Feb/Mar of 2015, approximately one/month.**
- **Traveling participants must participate in at least 3 workshops. Host sites must provide at least two traveling participants that will attend 2 additional workshops.**
- **Host sites will include both defense and commercial organizations.**
- **Workshops will analyze a host life-cycle experience, and then use accumulated learning to synthesize a host-chosen SE approach in need of more agility.**
- **Hosts will be expected to prepare a discussion presentation covering the processes to be analyzed and synthesized.**
- **Workshops will have up to 5-10 traveling participants and 5-10 host participants.**
- **Within 30-days of each workshop: a results-synopsis write-up, an evolving synthesis of accumulated discovery, and a case study write-up.**
- **No system-functional details need be revealed, only SE life-cycle process and activity procedures. Proprietary and classified projects should not be a problem.**

Outcomes and Benefits

Systems Engineering Community:

- **Gains a generic principle-based framework for evaluating, choosing, tailoring, integrating, and evolving agile SE knowledgably.**

Host Sites:

- **Have an analysis of an experienced agile SE process for fundamentals that enable and inhibit effective response to uncertain, unpredictable, evolving SE environments.**
- **Gain a deep understanding of necessary and sufficient fundamental principles and justifications for agile SE life cycle model processes and activities applicable to any type of Agile SE process for any type of project.**
- **Have a synthesis of project-independent analyzed-learnings applied to an insufficiently agile SE situation in need of some organized thought.**
- **Gain an insightful understanding and competency developed among at least a few host participants for knowledgeable internal leadership.**

Traveling Participants:

- **Gain an insightful understanding and competency for knowledgeable internal leadership.**
- **Obtain bench-mark exposure to a variety of agile SE processes combining HW/SW/WW development activity.**

Status

INCOSE Technical Project Plan approved 13-Oct-2014.

Next

Detail planning and preparation.

Workshops will occur approximately one per month.

Identify and secure relevant host sites (yours?).

Identify candidate traveling participants (you?).

Workshops anticipated to begin in Mar of 2015.

Project Lead: Rick Dove, led prior agile-fundamentals workshop series

Co-Lead: Kevin Forsberg, V diagram and INCOSE Handbook involvement

Co-Lead: Jack Ring, participated in prior agile-fundamentals workshop series

Co-Lead: Garry Roedler (pending budget approval), 15288 involvement

Co-Lead: TBD (from Europe)

References

Boehm, Barry, Jo Ann Lane, Supannika Koolmanojwong and Richard Turner. 2014. *The Incremental Commitment Spiral Model: Principles and Practices for Successful Systems and Software*. Addison-Wesley Professional.

Carson, Ron. 2013. Can Systems Engineering be Agile? INCOSE IS13, Philadelphia, PA, 24-27 June.

Dahmann, Judith, Jo Ann Lane, George Rebovich, Jr. and Kristen J. Baldwin. 2011. An Implementers' View of Systems Engineering for Systems of Systems. IEEE International Systems Conference, Montreal, Canada, 4-7 April. www.acq.osd.mil/se/docs/ImplementerViewSE-SoS-Final.pdf.

Dove, Rick. 1998. Realsearch: A Framework for Knowledge Management and Continuing Education. IEEE Aerospace Conference. Aspen, CO. 28 March. www.parshift.com/Files/PsiDocs/RealsearchIEEE.pdf.

Dove, Rick and Ralph LaBarge. 2014. Fundamentals of Agile Systems Engineering – Part 1 and Part 2. INCOSE IS14, Los Angeles, CA, 30 June - 03 July. www.parshift.com/s/140630IS14-AgileSystemsEngineering-Part1.pdf, www.parshift.com/s/140630IS14-AgileSystemsEngineering-Part2.pdf

Fowler, Martin and Jim Highsmith. 2001. The Agile Manifesto. Dr. Dobb's Journal, August. www.drdobbs.com/open-source/the-agile-manifesto/184414755.

Leffingwell, Dean, et al. 2014. Foundations of the Scaled Agile Framework (PowerPoint). Scaled Agile, Inc. http://scaledagileframework.com/?wpdmact=process&did=NzkuaG90bGluaw.

Schindel, Bill. 2005. Requirements Statements Are Transfer Functions: An Insight from Model-Based Systems Engineering. INCOSE International Symposium, Rochester, NY, 10-15 July. Best Paper award. https://sites.google.com/site/incosepbsewgtempaccess/home/INCOSE%20IS-2005.pdf.

Schwaber, Ken and Jeff Sutherland. 2013. The Scrum Guide. www.scrum.org.

US Air Force. 2011. Air Force Instruction 63-114, Quick Reaction Capability Process. 4 Jan. http://static.e-publishing.af.mil/production/1/saf_aq/publication/afi63-114/afi63-114.pdf

US DoD. n.d. Open Systems Architecture. www.acq.osd.mil/se/initiatives/init_osa.html.

Agile SE Life Cycle Model Development Project

Presentation download at:
www.parshift.com/s/ASELCM-DevelopmentProject.pdf
Asynchronously updated, last update 2-Nov-2014

Rick Dove
575-586-1536, rick.dove@parshift.com

Backup

Conceptual Example of Design Principles Analysis (RRS)

Details in (Dove & LaBarge 2014)

Not anticipated as workshop analysis exercise, but may be in final report

Reusable		Scalable	
Encapsulated Modules Product owners, Scrum masters, developers, product backlog, stakeholders, …			**Evolving Infrastructure** Retrospective process-learning evolves basic SE process, …
Facilitated Interfacing (Plug Compatibility) Backlog priorities, time boxed activities, all-hands stand-up meetings, customer involvement, agile SE method training, …			**Redundancy and Diversity** Cross-discipline development teams, part time subject matter experts, …
Facilitated Reuse Team members can be reassigned among sub-systems and tasks facilitated by a common SE method and training			**Elastic Capacity** Scope changes accommodated with augmented or reduced team size from commonly trained resources, …

Reconfigurable

Peer-Peer Interaction All-hands stand-up meetings, customer involvement, …	**Distributed Control & Information** Developers control task design, distributed information shared in daily stand-up meetings, …
Deferred Commitment Incremental requirements development, iterative system development, …	**Self-Organization** Team determines Sprint tasks, …

Note: this is a partial Scrum-process analysis example, for concept only

Abstract

For many, the word Agile, with a capital *A*, is used as a noun, referring to a family of software development processes based on principles published as the Agile Software Development Manifesto (Fowler 2001). To the INCOSE Agile Systems and Systems Engineering (AS&SE) working group, the word agile has a small *a*, and is an adjective referring to a capability for operational adaptability in an uncertain and unpredictable evolving environment. Fundamental concepts of agile capability were developed throughout the nineties in projects led by Lehigh University and funded by DoD. See (Dove 2014) for that history. This discussion will review fundamental architecture and design principles that enable agile capability, relate these fundamentals to domain-independent agile systems engineering, and review an INCOSE traveling-workshop series planned for 2015 that will develop a generic agile systems-engineering life cycle model.

We are all very familiar with architectures that accommodate and facilitate real-time structural change. Think of the construction sets we grew up with: Erector/Meccano sets, Tinker Toy, Lego, and other. Each of these construction sets consists of different types of components, with constraints on how these components can be connected and interact. This basic architectural pattern enables reconfiguration, augmentation, and evolution of both the engineering process and the engineered system.

Systems engineering is a disciplined activity that delivers engineered solutions to problems and opportunities – often involving multiple stakeholders, coordination across multiple engineering disciplines, and complexity in both problem and solution. Unlike other engineering disciplines, systems engineering also deals with the social, political, and technical aspects of managing projects that span multiple disciplines.

There is no *a priori* reason to expect domain specific software development practices to be applicable in domain independent systems engineering. For a simple disconnect example see (Carson 2013). Nevertheless, the ball is in motion toward the goal of a domain-independent agile systems-engineering discipline. Perhaps many different balls are in motion, as the pressure to do systems engineering under accelerating environmental dynamics is not waiting for a common disciplined understanding.

Life cycle, as a systems engineering term, demarcates the progressive maturity of a system through a linear sequence of stages. Here the argument is against the continued notions of non-repeating stages and of single-state existence; instead, a life cycle framework that employs progressively concurrent repeated stages, with diminishing emphasis on the lower stages as maturity through primary stages progresses.

It is time to develop an agile systems-engineering life cycle model. This model, if a single one is sufficient, must take into account four or five types of SE projects: discovery (verifying requirements and solution feasibility), programmatic (Systems and SoS from proven components), approach (e.g., ICSM methodology and product line architecture, Boehm 2014), quick reaction (rapid development and fielding), and evolving (continuous change of system operational viability and opportunity, rapid sequential generations).

The INCOSE AS&SE working group project will develop an agile systems-engineering life cycle model by starting with the described agile life cycle framework, take guidance from the ISO/IEC 15288 Standard for Systems and Software Engineering, and move toward identifying fundamental principle-based activities and processes that provide agility. This model will justify the application of these principles, activities, and processes by identifying common systems-engineering environmental situations in need of agile response capability; and will be supported with case studies across a variety of systems engineering domains and project types.

A method called Realsearch (Dove 1998), so called because it employs real people solving real problems in real time, refined and socialized the original agile systems fundamentals discovered and organized in the nineties. It is a process of traveling, structured, collaborative workshops where participants visit host sights by invitation. The process engages first in a collaborative exercise of situation analysis on local examples of agile process, then engages in collaborative identification of principles employed locally that enable agile capability, and finally engages in an exercise that applies learnings to an open problem in need of an agile process solution. A series of such workshops begins in 2015, designed to converge on a fundamental agile systems-engineering life cycle model applicable to the INCOSE systems engineering community. ---- References on next page.

BIO

Rick Dove was co-PI on the DoD funded project at Lehigh University that gave birth to the global interest in agile systems and enterprises in the early nineties. He subsequently led the broad industry research activity as Director of Strategic for the DARPA/NSF funded Agility Forum in the mid-nineties. His ongoing research focuses on necessary and sufficient fundamental design principles for systems of any kind that would be agile. He has employed these principles in the creation of large, agile, enterprise IT systems; outsourced semiconductor manufacturing; rapid metal-parts manufacturing; and resilient enterprise security systems.

He was president of Flexis Controls, pioneering agile factory control system design tools; president of ProMetal, providing production service and equipment for large 3D printed metal parts; and president of Montclair Winery, where he designed and built the production facility and custom production equipment for one of the first boutique urban wineries.

He is a Fellow of the International Council on Systems Engineering (INCOSE), where he founded and chairs the working groups on Agile Systems and Systems Engineering and on System Security Engineering.

He teaches basic and advanced graduate courses in agile and self-organizing systems at Stevens Institute of Technology. He is CEO/CTO of Paradigm Shift International, an applied research firm specializing in agile systems concepts and education, and leads agile self-organizing system security research and development on DHS and OSD funded technology development projects. He is a partner in Kennen Technologies, and was the PI on the DHS funded projects that showed proof of concept and built prototypes for applying Kennen's patented VLSI pattern processor technology to advanced Bio-inspired problem applications.

He is author of Response Ability – the Language, Structure and Culture of the Agile Enterprise; and Value Propositioning – Perception and Misperception in Decision Making. He holds a BSEE from Carnegie Mellon University.