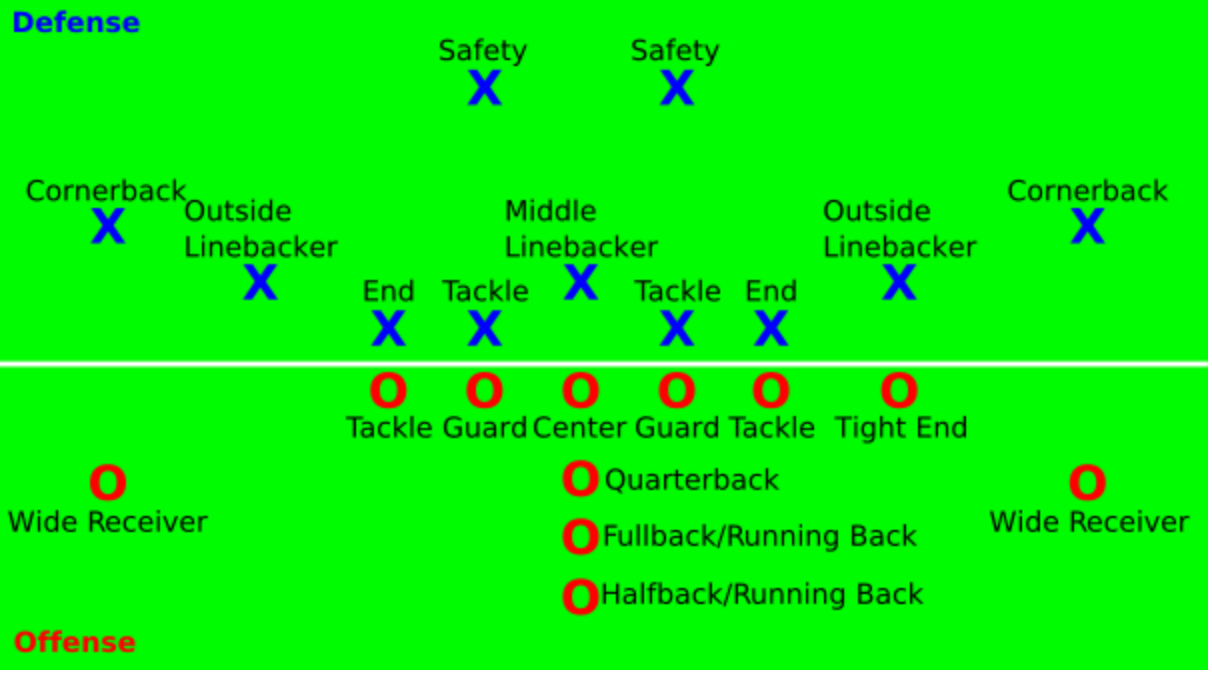
Agile Modeling and Modeling Agile Systems

3 Part MBSE Breakout Session
IW15 – Saturday 24 Jan 2015

Part 1 –

The Agile Architecture Pattern as a model-based system pattern, with implications for agile modeling

Rick Dove



American Football

http://football.about.com/od/footballpositions/Football_Positions.htm

11 players on field per side

Offensive positions:
8 with some pairs
Defensive positions:
6 with many pairs
Special teams positions:
7 with some multiples

Adaptation is an immediate, appropriate, different response in functionality. This can only occur if functional resources can be added, modified, or reconfigured quickly. A good sports team has more players than it fields at any one time, so that the coach can mix and match the players' skill-sets according to the opposition, the situation, and real-time developments.

Reconfiguring a sports team with different players during game time doesn't work, though, if players bring their own rules with them. The players all know the rules of the game and they all know their team's playbook. The coach exercises a drag-and-drop, plug-and-play operational strategy enabled by an actively managed team-system structure. Complex system behaviors arise from the interactions of simple rules. Were this not the case, it would be impossible to sustain complex behavior in the face of increased opportunities for failure.

Today's Agility Interest – Origin & Continuation

- **1991 – SecDef funded project at Lehigh University to identify next manufacturing competitive focus beyond Lean**
  - **13 companies participated full-time in 3-month workshop**
  - **2 vol report: 21st Century Manufacturing Enterprise Strategy**
  - **Problem/opportunity defined (for manufacturing enterprises)**
- **1992 – Agile Manufacturing Enterprise Forum founded at Lehigh, funded by Texas Instruments and General Motors**
  - **Purpose: Identify nature of Agile solution**
  - **Method: Industry collaborative workshop groups**
- **1994 – DARPA/NSF establish $5 Million x 5 year funding**
  - **Name changed to Agility Forum (any kind of enterprise)**
  - **Research steering group and agenda established**
  - **250+ orgs, 1000+ participants in focused workshop groups**
  - **Conferences, papers, reference base, tools, reference model**
- **1998 – Mission accomplished, Agility Forum dissolved**
  - **Agility pursuit by industry and IT vendors entrenched**
- **Since then – Confirmation & employment in various projects**
  - **Many graduate SE student term and masters projects**
  - **Refinement of architectural concepts, no basic changes**

Agile-Systems Research Focus – 1991+

Problem:

- Technology and markets are changing faster than the ability to employ/accommodate
- Life cycle requirements are uncertain and unpredictable
- Flexible system approaches inadequate when requirements change
- New approach needed that could extend usefulness/life of systems

Solution Search:

- Examined 100s of systems of various types
- Looked for systems that responded *effectively*
- Looked for metrics that defined *effectively*
- Looked for categories of response types
- Looked for principles that enabled response

Note: This research took place at the Agility Forum 1992-1996, and in subsequent independent research 1997-1999

Essays chronicle knowledge development at www.parshift.com/library.htm

Agility - Fundamentally

The Ability to Thrive in a Continuously Changing, Unpredictable Environment.

**Agility is *effective response* to opportunity and problem,
within mission ... always … no matter what.**

An *effective response* is one that is:	**Metric**
■ **timely (fast enough to deliver value),**	**time**
■ **affordable (at a cost that leaves room for an ROI),**	**cost**
■ **predictable (can be counted on to meet expectations),**	**predictability**
■ **comprehensive (anything/everything within mission boundary).**	**scope**

You can think of Agility as Requisite Variety.

You can think of Agility as proactive Risk Management.

You can think of Agility as Innovative Response in unpredictable situations.

You can think of Agility as Life Cycle Extension.

**The trick is understanding the nature of agile-enabling fundamentals,
and how they can be applied to any type of system/process.**

Domain Independent

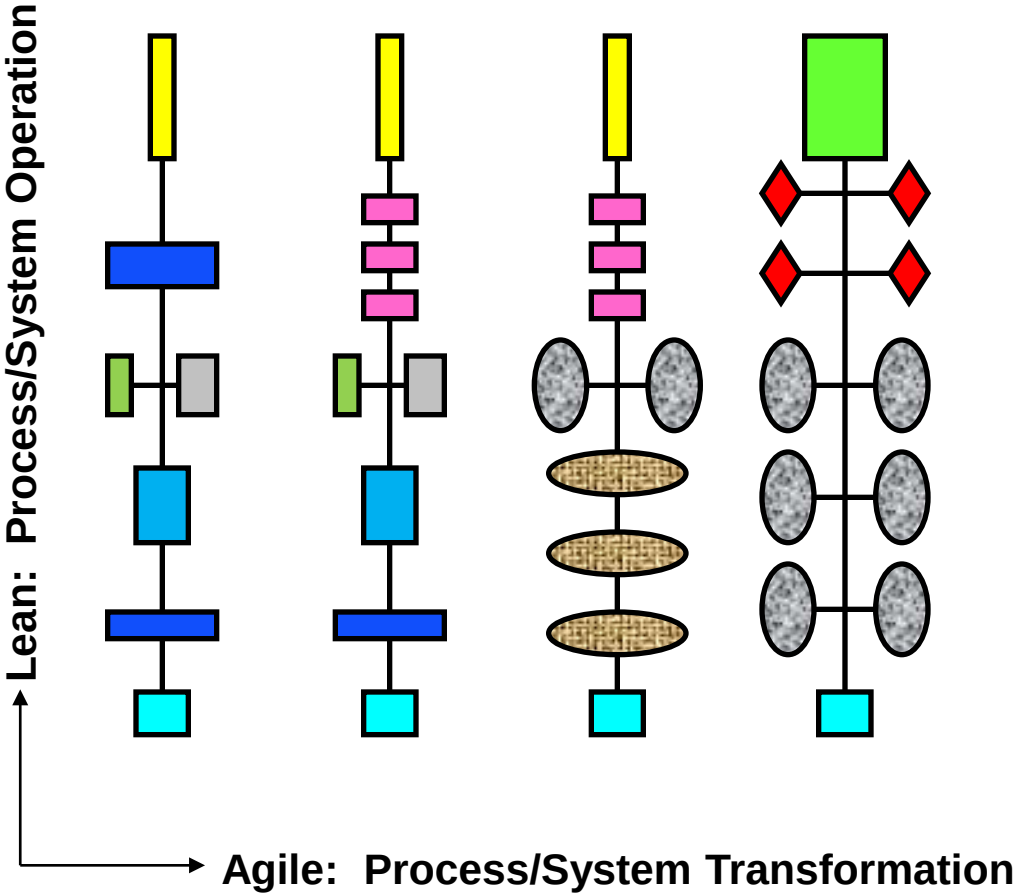
Lean & Agile: Orthogonal Focus

Agility deals with "design-for-transformation".

In a very general interpretation,

Lean values efficiency of operation and achieves this mainly through operational principles;

Agile values effective response ability and achieves this mainly through architectural principles.



Both are concerned with operational effectiveness. Since the two have a different means for achieving different ends they are not necessarily in one-or-the-other conflict – but can be.

The UURVE Environment Drives the Response Need

Agile systems are defined in counterpoint to their operating environments.

Words used to describe the general nature of the target environment often include and combine dynamic, unpredictable, uncertain, risky, variable, and changing, with little attention to clear distinction among them.

To design and develop a system that can deal effectively with changing environments it is useful to articulate the nature of changes that should be considered.

Agile systems have effective situational response options, within mission, under:

- **Unpredictability: randomness among unknowable possibilities.**
- **Uncertainty: randomness among known possibilities with unknowable probabilities.**
- **Risk: randomness among known possibilities with knowable probabilities.**
- **Variation: randomness among knowable variables and knowable variance ranges.**
- **Evolution: gradual (relatively) successive developments.**

The difference between risk and variation in this framework is that risk is viewed as the possible occurrence of a discrete event (a strike keeps all employees away), while variation is viewed as the intensity of a possible event (absenteeism varies with the season).

7 Thought-Guiding Frameworks

(this discussion's focus in yellow)

Response requirements categories (4 reactive and 4 proactive elements):
Reactive: correction, variation, expansion, reconfiguration
Proactive: creation, improvement, migration, modification

Response performance metrics (4 elements):
Response: cost, time, quality, scope

Response-enabling design principles (10 elements):
Encapsulation, Compatibility, Reusability, Redundancy/Diversity, Scalability, Distributed, Loose, Deferred Commitment, Self-Organizing, Evolving Standards

Design quality principles (3 elements):
Requisite Variety, Parsimony, Harmony

An overarching architectural philosophy (3 elements):
Reusable modules Reconfigurable in a Scalable architecture (RRS)

Agility-sustaining responsibilities (4 elements):
Module Mix Evolution, System Assembly/Reconfiguration
Module Readiness, Infrastructure Evolution

An agile architecture pattern:
Drag-and drop modules in a plug-and-play infrastructure

Objective: System X-Ray Vision

http://awespendo.us/animemangacomics/kermit-at-the-doctor/

Case: Home Entertainment Technology Migration

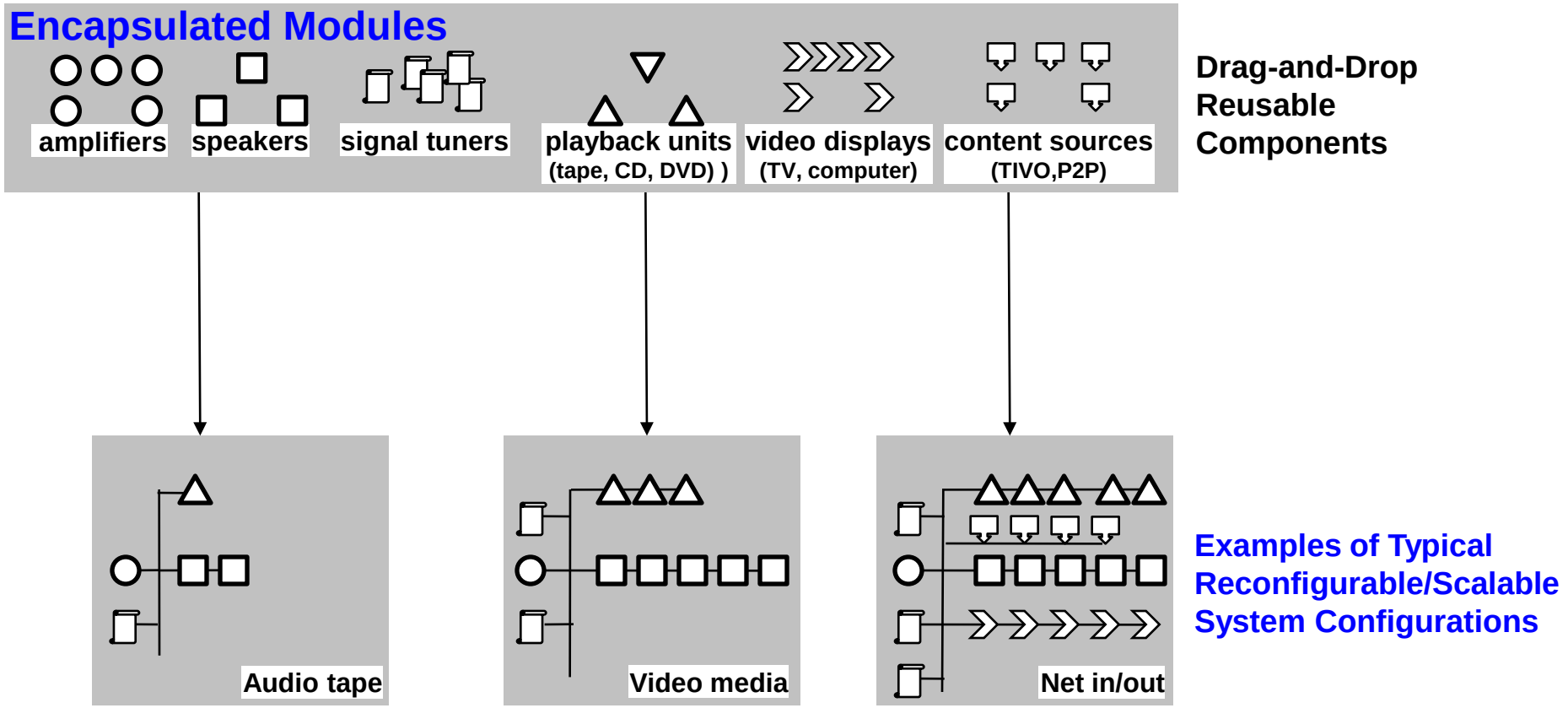
agile architecture pattern: drag-and-drop, plug-and-play

Encapsulated Modules

Drag-and-Drop Reusable Components

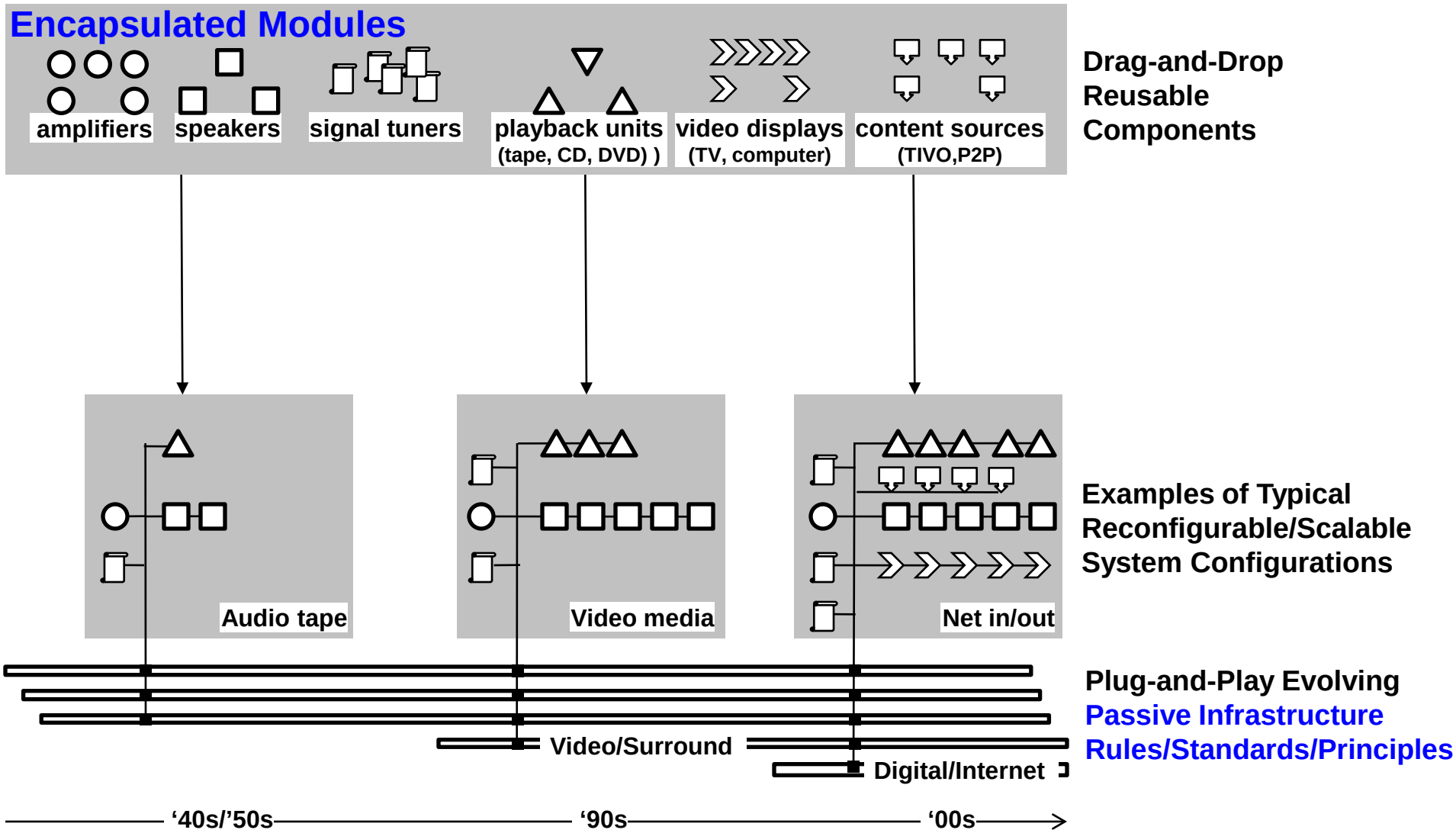
Case: Home Entertainment Technology Migration

agile architecture pattern: drag-and-drop, plug-and-play



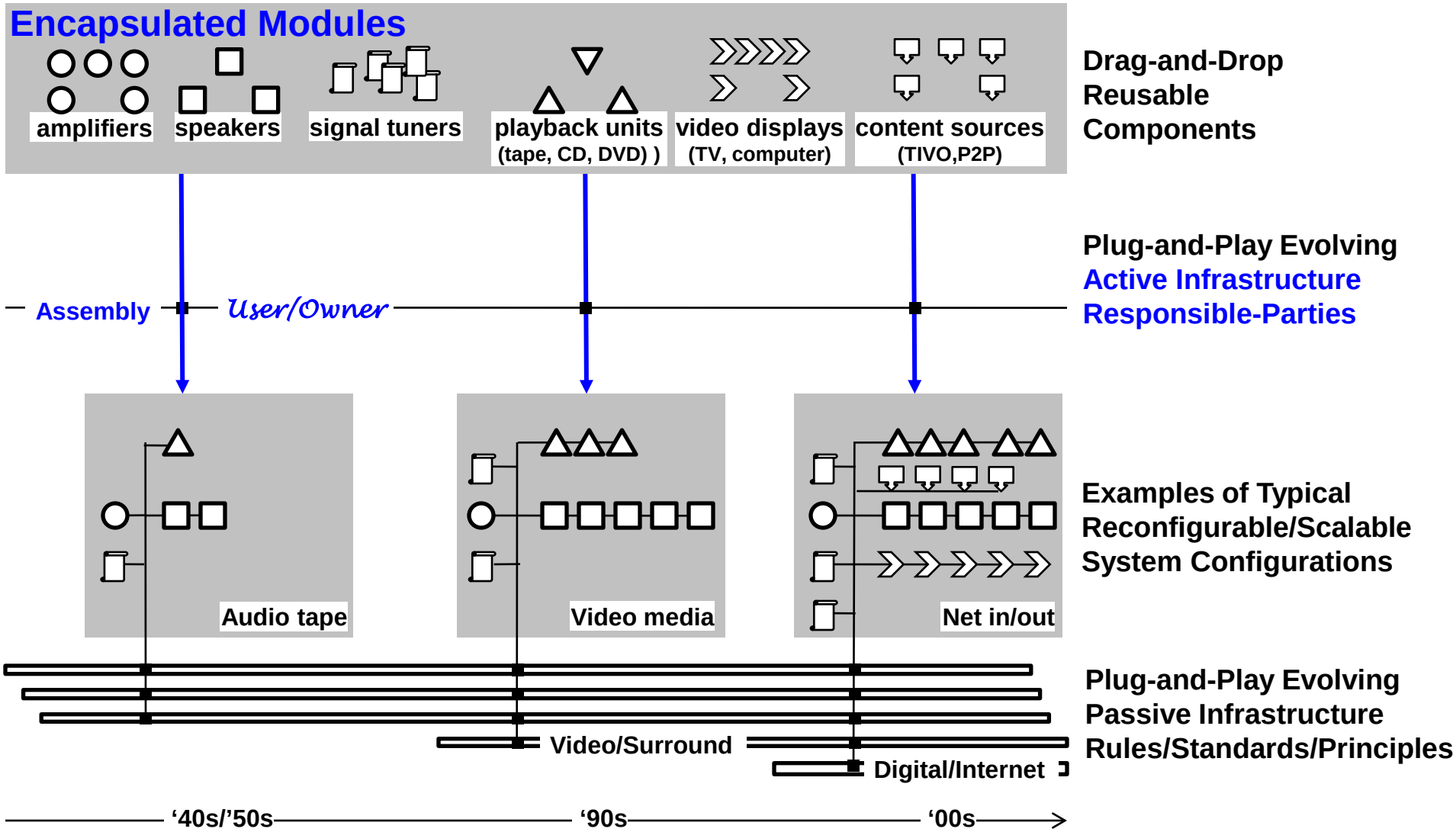
Case: Home Entertainment Technology Migration

agile architecture pattern: drag-and-drop, plug-and-play



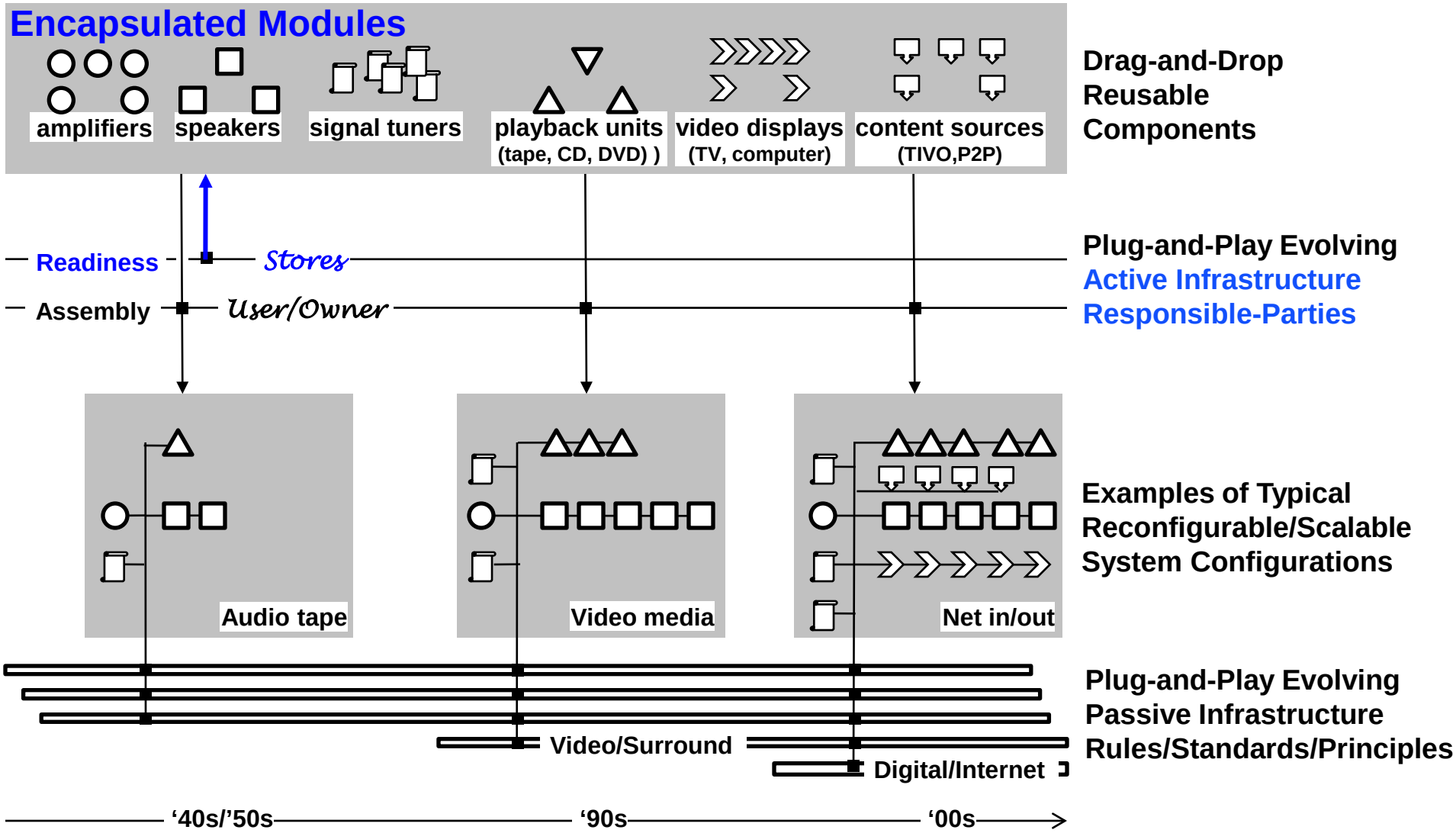
Case: Home Entertainment Technology Migration

agile architecture pattern: drag-and-drop, plug-and-play



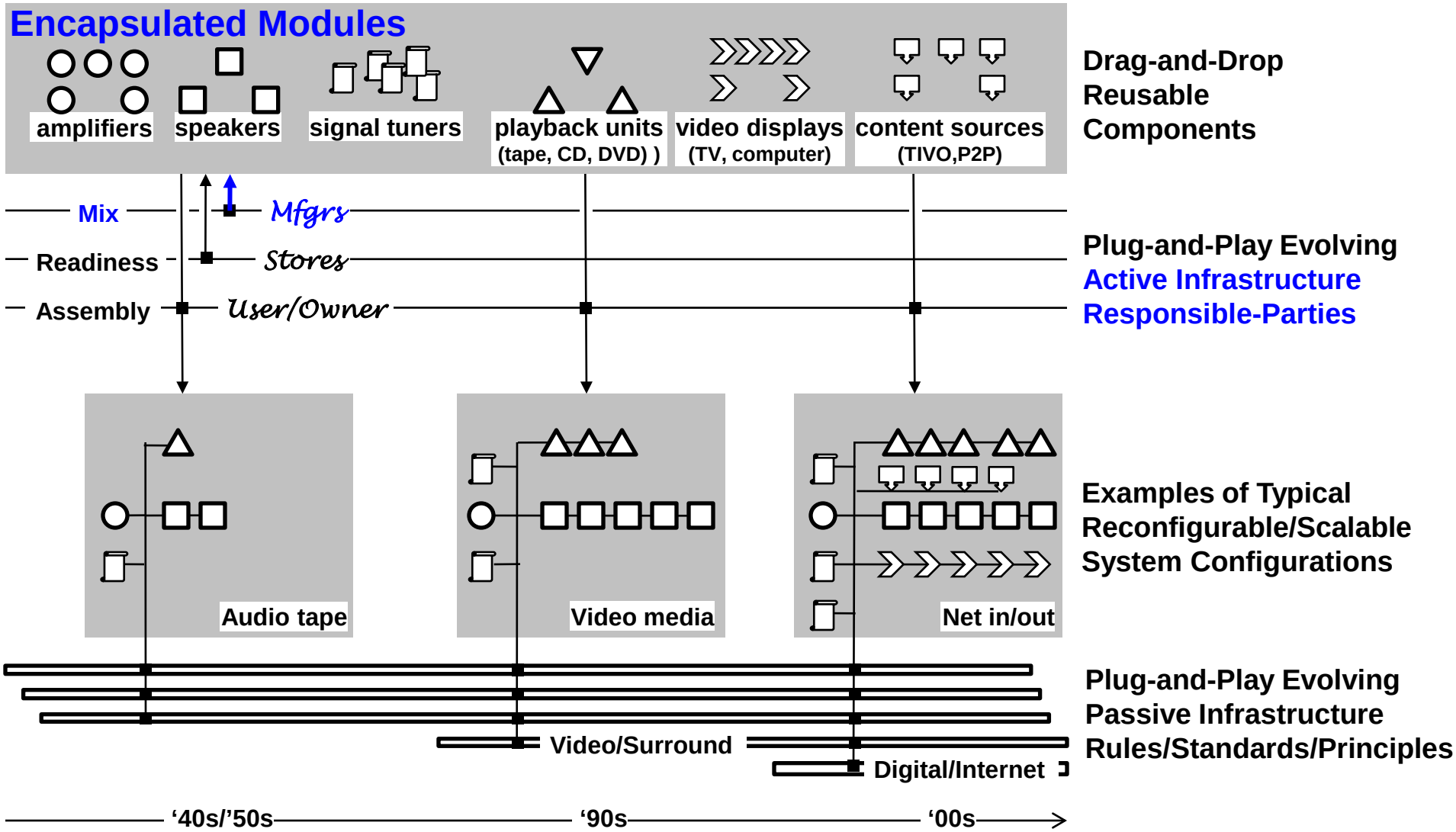
Case: Home Entertainment Technology Migration

agile architecture pattern: drag-and-drop, plug-and-play



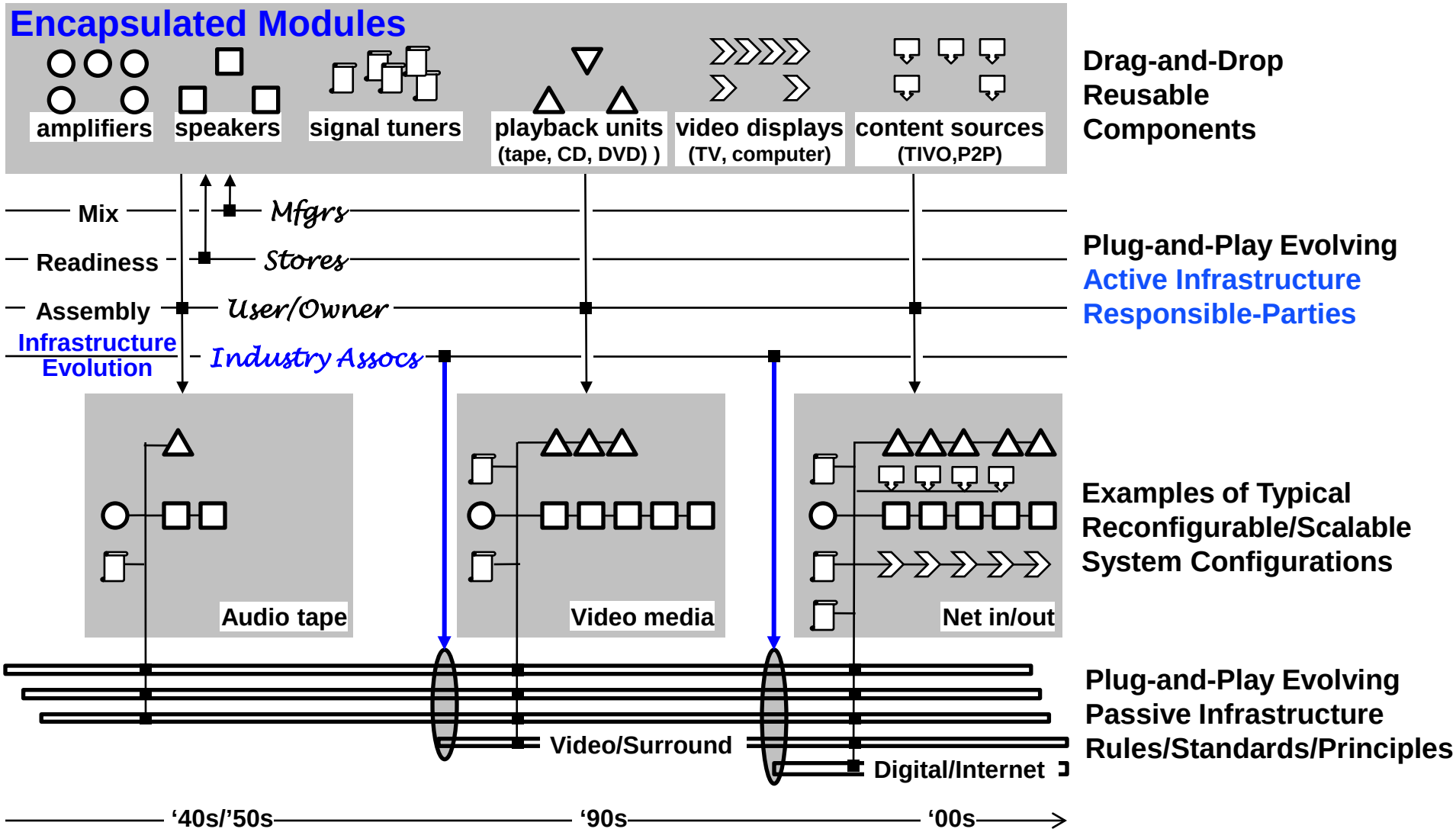
Case: Home Entertainment Technology Migration

agile architecture pattern: drag-and-drop, plug-and-play



Case: Home Entertainment Technology Migration

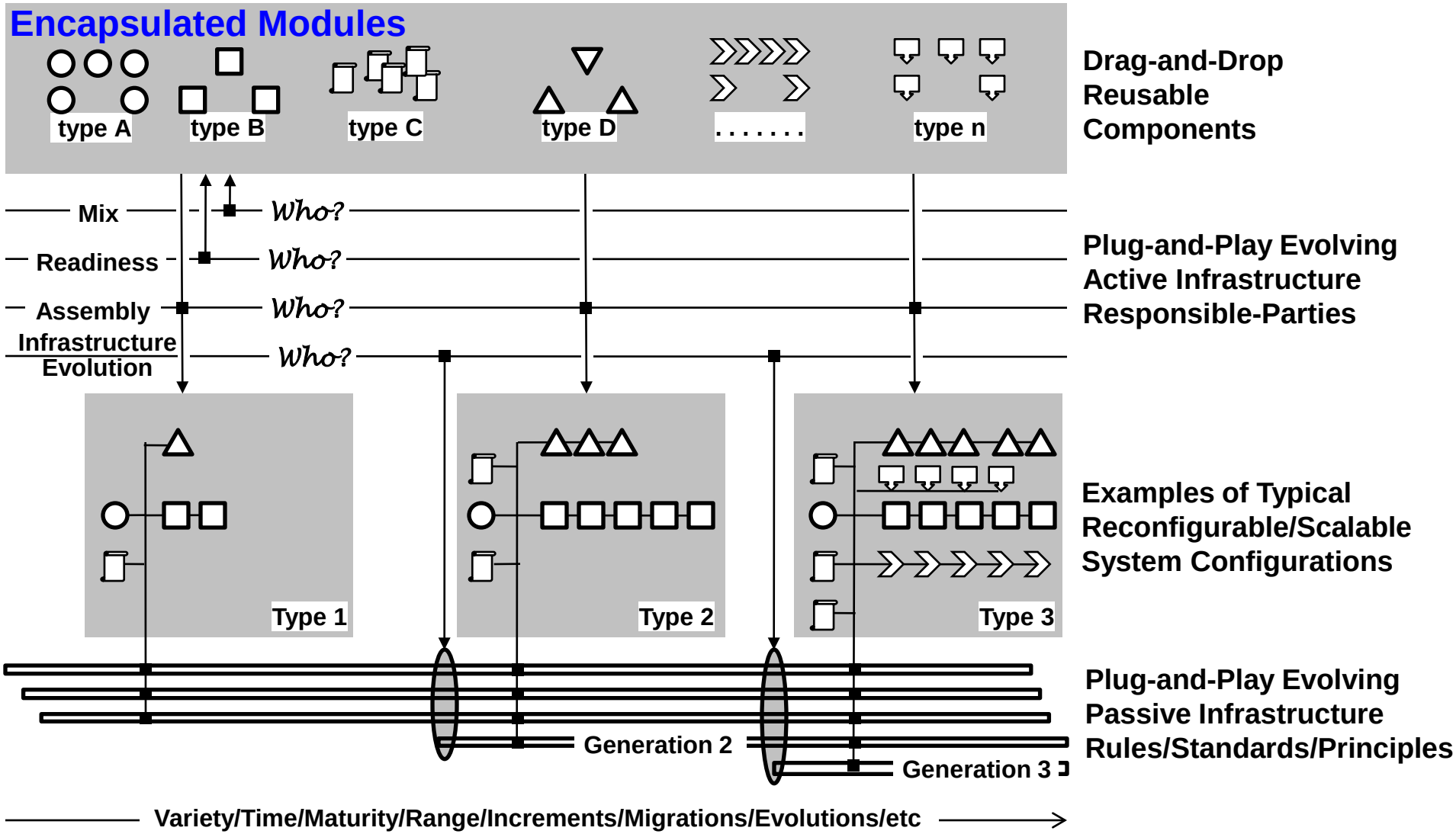
agile architecture pattern: drag-and-drop, plug-and-play



Fundamental Concept

Reusable modules Reconfigurable in a Scalable architecture (RRS)

agile architecture pattern: drag-and-drop, plug-and-play



Who/What is Accountable Sustainability & Effectiveness

The “active” parts of the infrastructure

Module Mix Evolution:

- **Who (or what process) is responsible for ensuring that existing modules are upgraded, new modules are added, and inadequate modules are removed, in time to satisfy response needs?**

Module Readiness :

- **Who (or what process) is responsible for ensuring that sufficient modules are ready for deployment at unpredictable times?**

System Assembly/Reconfiguration:

- **Who (or what process) assembles new system configurations when new situations require something different in capability?**

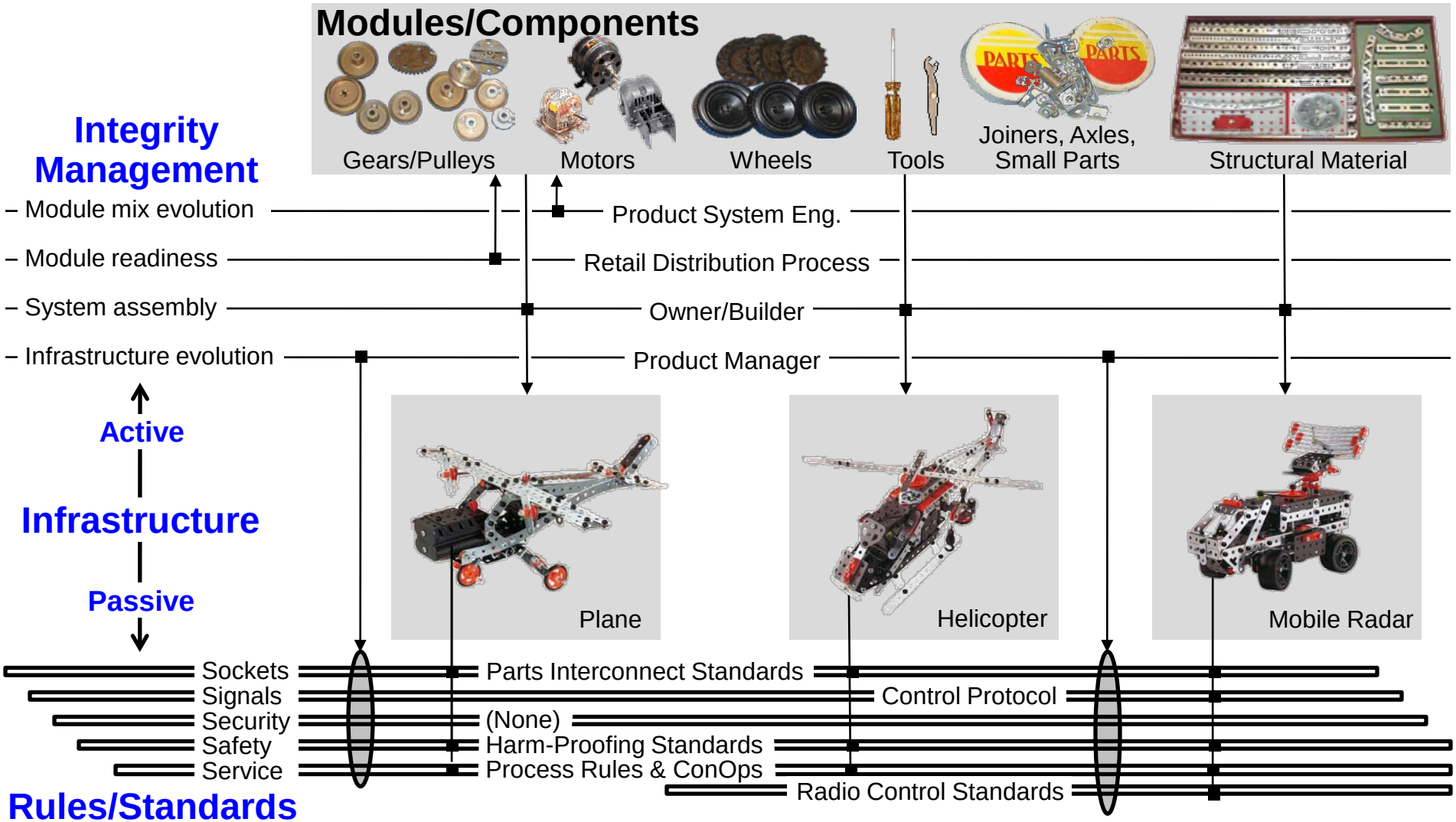
Infrastructure Evolution:

- **Who (or what process) is responsible for evolving the passive and active infrastructures as new rules and standards become appropriate to enable next generation capability.**

The “passive” parts of the infrastructure are the interoperability standards

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

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



Case: Aircraft Refurb QRC

Jason Boss masters project, Agile Aircraft Installation Architecture In a Quick Reaction Capability Environment, INCOSE IS10, Chicago, July 12-15.
www.parshift.com/Files/PsiDocs/Pap100712IS10-AgileAircraftInstallationArchitecture.pdf

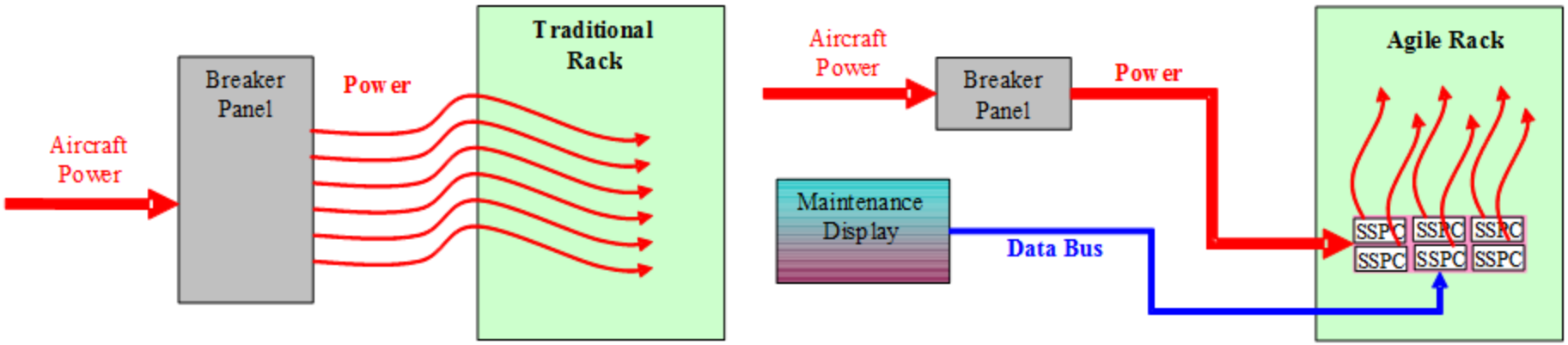
- Mission system installation in military acquisition context.
- Customer's need for the latest technology.
- Technology advances are creating new mission systems at an increasing rate, driving the demand for QRC.
- Goal is to shorten the completion time without compromising quality.
- Mission requirements and "boxes" often change late.
- Army wants QRC for intelligence surveillance reconnaissance (ISR) to be robust, scalable, tailorable.
- Air Force wants QRC challenges continually met, success is measured in rapidly adapted Electronic Warfare.

Example: Agile vs. Traditional Power Distribution

Traditionally a breaker centralized panel distributes power to each box, creating an interface for every box and many wire routing paths. Some aircraft contain over 1000 boxes, and wire routing becomes a large modification effort.

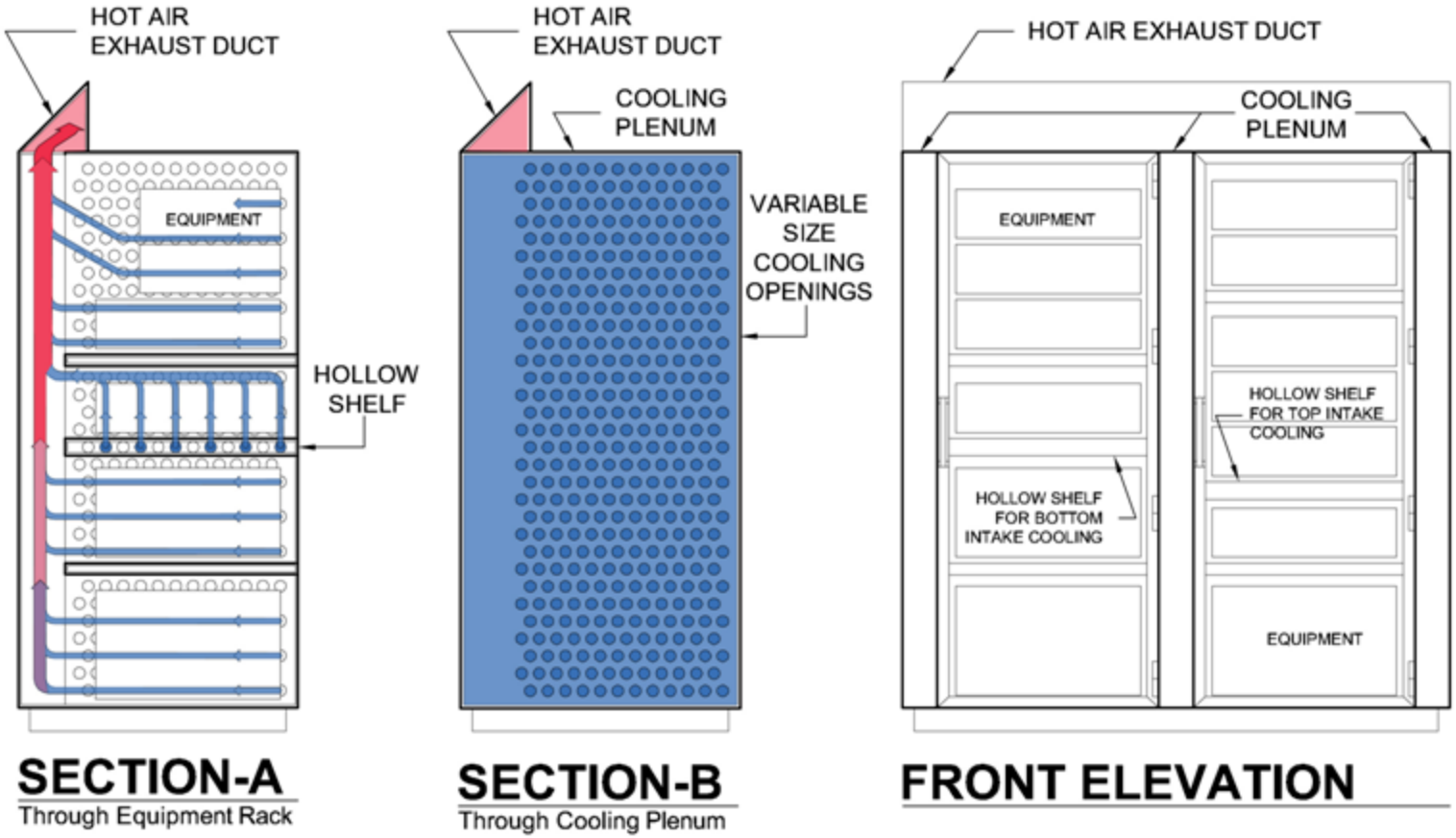
To reduce the number of interfaces, decrease wire routing effort, and allow rack modularity, the power distribution can be moved from the aircraft to within the rack itself.

A single breaker then provides power to the rack, and a secondary breaker panel within the rack would distribute power to each box. Remote controlled solid state power controllers (SSPCs) allows re-programming an SSPC instead of changing a breaker out and routing a new wire between the breaker box and the rack.



Rack becomes an encapsulated module. Power infrastructure is minimal.

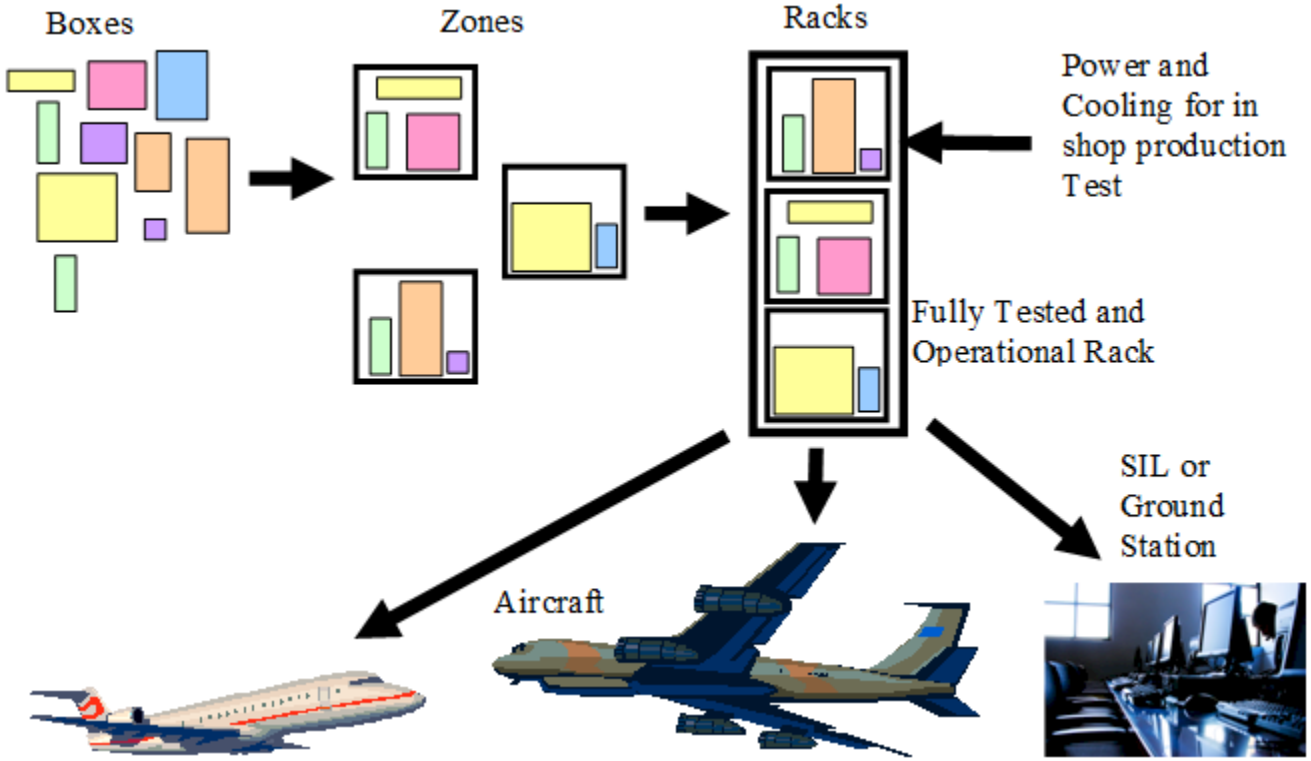
Example: Modular Rack Cooling



Solution mitigates the rerouting effort of existing aircraft ductwork. The proposed cooling architecture is really a combination of a cold air distribution subsystem that gets cold air from the aircraft source to the boxes, and a hot air exhaust subsystem that must dispose of the waste air.

Rack becomes an encapsulated module. Cooling infrastructure is minimal.

Encapsulated Modules, Minimal Infrastructure



Aircraft installation infrastructure is modified… once.

The SIL* has a duplicate infrastructure.

"Everything" is fully integrated and tested in the SIL … before the aircraft arrives.

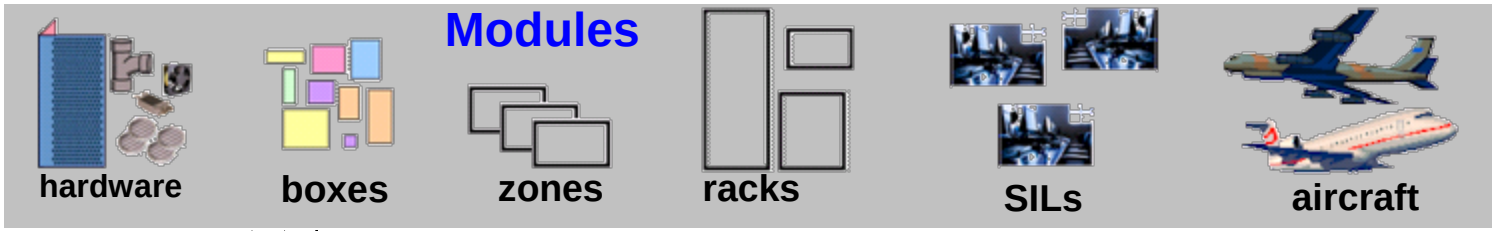
Aircraft installation is a simple relocation of pluggable modules.

Minimizes aircraft downtime and eliminates custom installation work.

Parameter	Nature of Standard *SIL: System Integration Lab
Space	Racks shall be designed in preset widths, depths and heights.
Power	Each rack shall have a maximum kW equipment load rating. Racks with multiple power types (e.g. 115 VAC 400 Hz and 28 VDC) limits should be set on each type.
Weight	Each rack shall have a maximum equipment weight rating.
Cooling	Each rack shall rate the kW cooling capacity at a specified exhaust temperature.
Physical Interfaces	Rack mounting provisions, cooling connections, and electrical connection interfaces shall have standard locations and configurations.

QRC Device/Power/Cooling Installation Architecture

Boss, Jason and Rick Dove. 2010. Agile Aircraft Installation Architecture In a Quick Reaction Capability Environment.
INCOSE International Symposium, Chicago, July 12-15. www.parshift.com/Files/PsiDocs/Pap100712IS10-AgileAircraftInstallationArchitecture.pdf



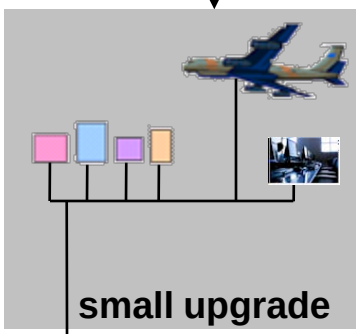
Integrity Management

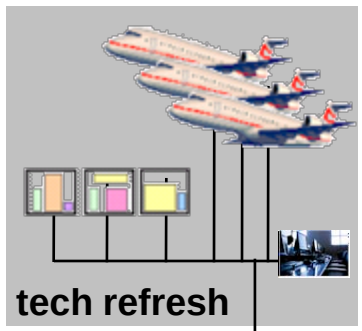
- Module mix evolution — system engineer
- Module readiness — material manager
- Assembly in SIL — production
- Infrastructure evolution — process engineer

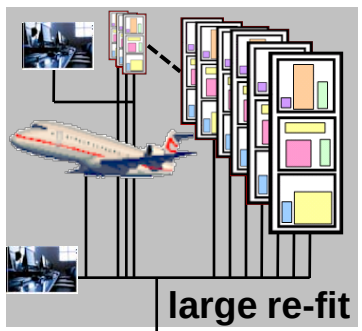
Active

Infrastructure

Passive







Sockets	Physical interconnect standards
Signals	Data/power/cooling transmission
Security	Personnel/Sil/supply-chain/et al.
Safety	Weight/space/installation rules
Service	Agile system/process ConOps

Rules/Standards

A Construction Project Case Study Based on the
"Last Planner System"
by Glenn Ballard

Lean and Agile Project Management
www.parshift.com/AgileSysAndEnt/Cases/Case Last Planner System.pdf

Creating Options
Reconfigurable Task Schedules
Deferred Assignment Commitments
Proactive Expediting

"When environments are dynamic and the production system is uncertain and variable, reliable planning cannot be performed in detail much before the events being planned.

"Consequently, deciding what and how much work is to be done next by a design squad or a construction crew is rarely a matter of simply following a master schedule established at the beginning of the project. **[pages 3-15 and 3-16 of Ballard Thesis]**

Herman Glenn Ballard
Director of Research, Lean Construction Institute, and Lecturer, Construction Engineering and Management Program, Dept. of Civil and Environmental Engineering, University of California at Berkeley, 4536 Fieldbrook Road, Oakland, CA 94619, 510/530-8656, FAX 510/530-2048, ballard@ce.berkeley.edu

Traditional Task Selection from Master Schedule

A key early finding was that only about half of the assignments made to construction crews at the beginning of a week were completed when planned.

Experiments were performed to test the hypothesis that failures were in large part a result of lack of adequate work selection rules (these might also be called work release rules).

Task Selection Method Addressing Schedule Uncertainty

Quality criteria were proposed for assignments regarding definition, sequence, soundness, and size.

In addition, the percentage of assignments completed was tracked (PPC: percent plan complete) and reasons for noncompletion were identified, which amounted to a requirement that learning be incorporated in the control process.

[Ballard Thesis: page 3-16]

Quality Criteria for Work Assignment

Q1: Definition: Are assignments specific enough that the right type and amount of materials can be collected, work can be coordinated with other trades, and it is possible to tell at the end of the week if the assignment was completed?

Q2: Soundness: Are all assignments sound, that is: Are all materials on hand? Is design complete? Is prerequisite work complete? Note: During the plan week, the foreman will have additional tasks to perform in order to make assignments ready to be executed, e.g., coordination with trades working in the same area, movement of materials to the point of installation, etc. However, the intent is to do whatever can be done to get the work ready before the week in which it is to be done.

Q3: Sequence: Are assignments selected from those that are sound in the constructability order needed by the production unit itself and in the order needed by customer processes? Are additional, lower priority assignments identified as workable backlog, i.e., additional quality tasks available in case assignments fail or productivity exceeds expectations?

Q4: Size: Are assignments sized to the productive capability of each crew or subcrew, while still being achievable within the plan period? Does the assignment produce work for the next production unit in the size and format required?

Q5: Learning: Are assignments that are not completed within the week tracked and reasons identified?

As a result of applying these criteria, plan reliability (the percentage of assignments completed) increased, and with it, crew productivity also increased (Ballard and Howell, 1997)[16].

[16] On the whole, improvements tended to be from PPC (percent plan complete) levels around 50% to the 65-70% level, with a corresponding increase of 30% in productivity. Productivity improvement has ranged from 10% to 40%+.

[Ballard Thesis: pages 3-16 and 3-17]

Rules (R1-2-3) and Objectives Established

A set of rules was proposed for allowing scheduled activities to remain or enter into each of the three primary hierarchical levels of the scheduling system:

R1: Allow scheduled activities to remain in the master schedule unless positive knowledge exists that the activity should not or cannot be executed when scheduled.

R2: Allow scheduled activities to remain in the lookahead window only if the planner is confident that the activity can be made ready for execution when scheduled.

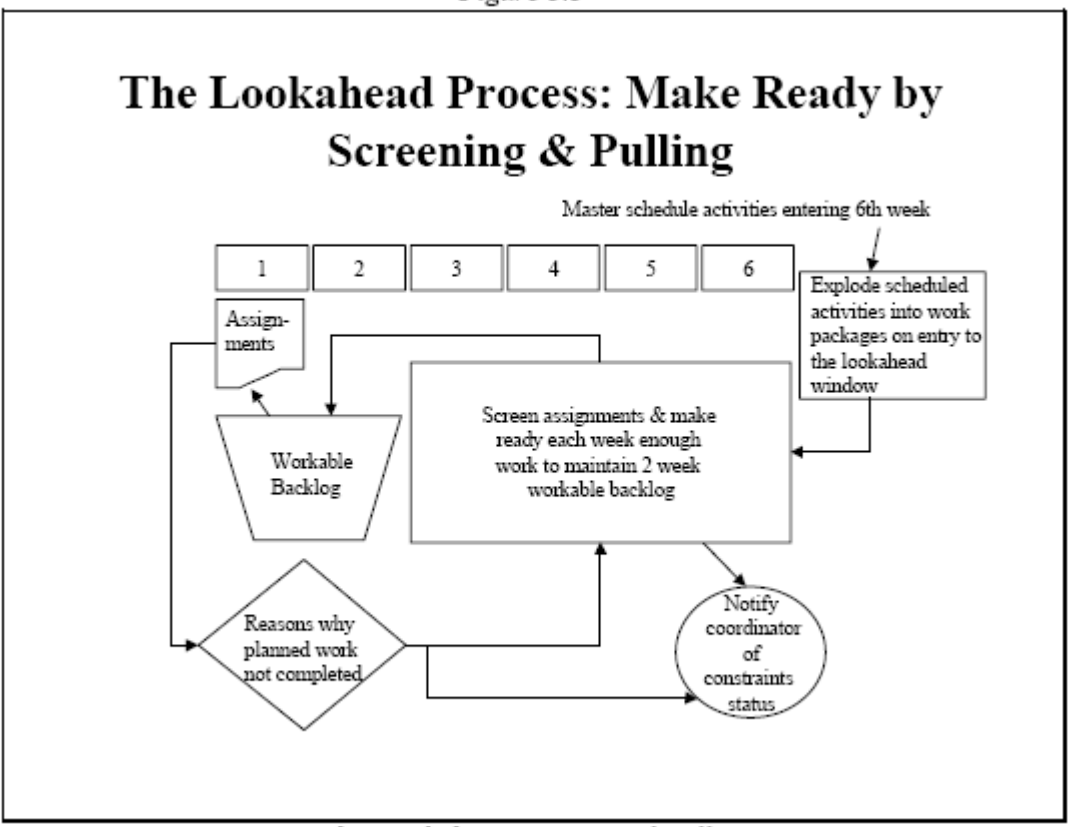
R3: Allow scheduled activities to be released for selection into weekly work plans only if all constraints have been removed; i.e., only if the activity has in fact been made ready.

In addition, a set of objectives was proposed for the lookahead process:

- **Shape work flow sequence and rate**
- **Match work flow and capacity**
- **Decompose master schedule activities into work packages and operations**
- **Develop detailed methods for executing work**
- **Maintain a backlog of ready work**

[Ballard Thesis: pages 3-17 and 3-18]

Figure 3.3



Make Ready by Screening and Pulling

Figure 3.3 is a schematic of the lookahead process, showing work flowing through time, right to left.

Potential assignments enter the lookahead window 6 weeks ahead of scheduled execution, then move forward a week each week until they are allowed to enter into workable backlog, indicating that all constraints have been removed and that they are in the proper sequence for execution.

If the planner were to discover a constraint … that could not be removed in time, the assignment would not move forward.

The objective is to maintain a backlog of sound work, ready to be performed, with assurance that everything in workable backlog is indeed workable.[13] Weekly work plans are then formed from workable backlog, thus improving the productivity of those who receive the assignments and increasing the reliability of work flow to the next production unit.

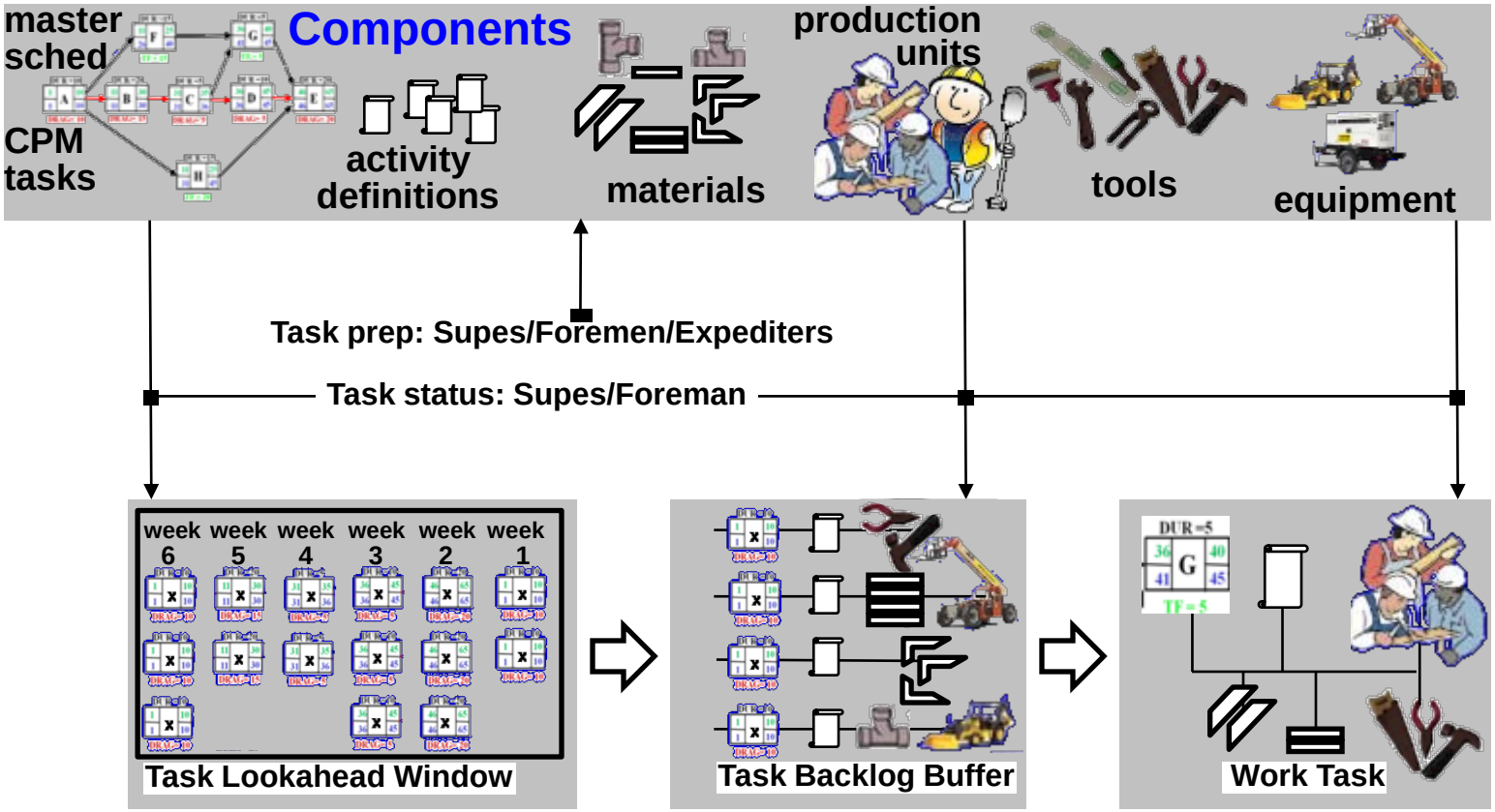
[13] **Deliberately building inventories, inventories of ready work in this case, may seem contradictory to the goals of just-in-time. To clarify, inventories of all sort are to be minimized, but as long as there is variability in the flow of materials and information, buffers will be needed to absorb that variability. Reducing variability allows reduction of buffer inventories.**

[Ballard Thesis: pages 3-7, 3-8 and 3-10]

Last Planner Work Flow Management

www.parshift.com/s/130624Last Planner.pdf

Active management of the anticipated schedule and work flow to ensure there is always a buffer of "quality" jobs ready to work on and matched with resources.



Tasks enter lookahead window 6 weeks in advance of execution schedule, advancing according to readiness, with action on prep for execution.

Tasks enter backlog whenever all necessary elements are ready for execution.

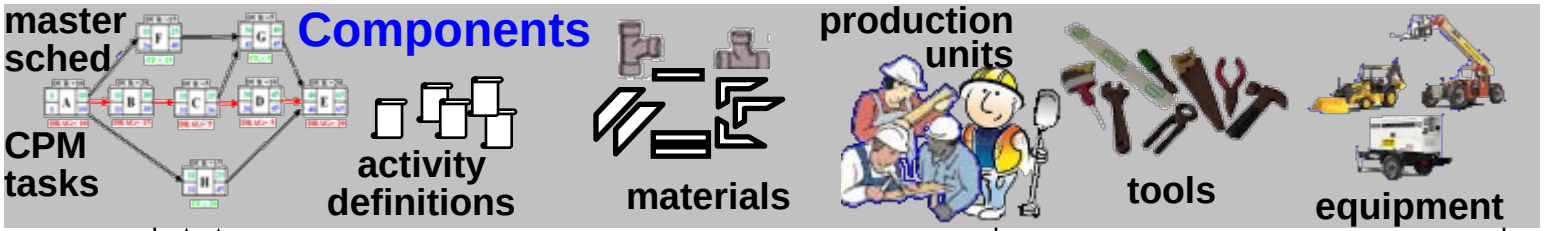
Weekly work tasks are drawn from readiness backlog, keeping crews fully employed.

www.parshift.com/s/130624Last Planner.pdf

Last Planner Agile Project Management

www.parshift.com/s/130624Last Planner.pdf

Active management of the anticipated schedule and work flow to ensure there is always a buffer of "quality" jobs ready to work on and matched with resources.



Integrity Management

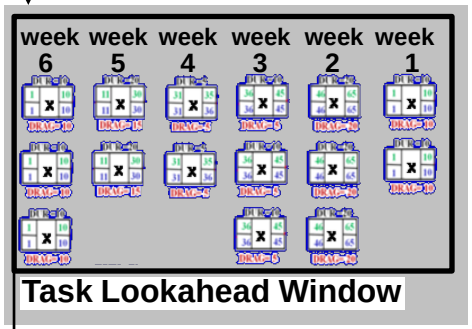
- Task elements: — Project Manager
- Task readiness: — Supes/Foremen/Expediters
- Task assembly: — Supes/Foreman
- Infrastructure evolution: — Last Planner Process Manager

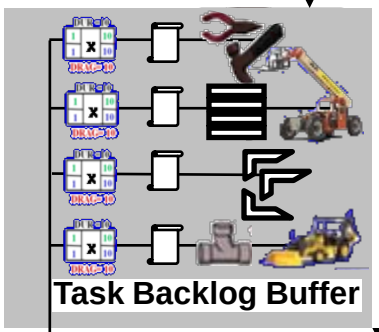
Key Practices:
Rules 1-2-3 and
- **Lookahead**
- **Make ready**
- **Learn & Correct**

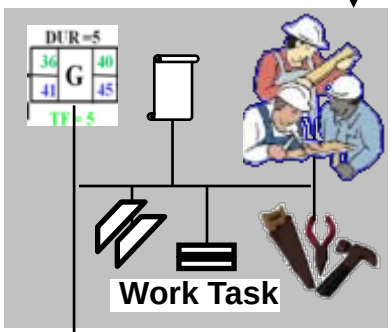
Active

Infrastructure

Passive







Standards

- Sockets — Task Soundness/Sequence/Size
- Signals — Task Definitions
- Security — Physical Site Security
- Safety — Construction Safety Standards/Regs
- Service — Master Sched, Learning, R1-2-3
- MS Learning Change

Agile architecture Pattern based on:

(Ballard 1997) Lookahead Planning: the Missing Link in Production Control
(Ballard 1998) Shielding Production: an Essential Step in Production Control
(Ballard 1999) Improving Work Flow Reliability
(Ballard 2000) The Last Planner System of Production Control-PhD Thesis

RRS Principles – two are necessary the other eight are amplifiers

Encapsulated Modules

- 1:1 physical/functional packaging
- Black box to other modules
- Functional methods can change, but interface protocols cannot

Evolving Infrastructure Standards

- Defines module-interface protocols/standards (and operating rules)
- Enables and constrains agility
- Delicate balance of requisite variety and parsimony

Encapsulated Modules	Reusable	Scalable	**Evolving Infrastructure Standards**
Facilitated Interfacing (Plug Compatibility)			**Redundancy and Diversity**
Facilitated Reuse			**Elastic Capacity**
Reconfigurable			
Peer-Peer Interaction			**Distributed Control and Information**
Deferred Commitment			**Self-Organization**

Case: Silterra: Malaysian Semiconductor Foundry

Rick Dove. 2005. Fundamental Principles for Agile Systems Engineering. Conference on Systems Engineering Research (CSER), Stevens Institute of Technology, Hoboken, NJ, March. www.parshift.com/Files/PsiDocs/Rkd05032.pdf

October 1999 (dot.com bubbling, semiconductor slump ending).

Silterra is a start-up semiconductor foundry in Malaysia, with interim USA top management and ex-pat process experts.

Funded mainly by government designated sources.

Mixed Cultures: 60% Malay, 30% Chinese, 10% Indian.

Few employees have built or run such a company, and have little idea about what they will need or want in business processes.

CEO has a vision for a preemptive modern-day competitor...
Goal: Build a uniquely superior foundry business.
Strategy: Best practices + Agile IT infrastructure.

CIO (interim exec) is writing book on systems agility...
Goal: Meet CEO's goals with Agile Systems design principles.
Strategy: Design a differentiation strategy and apply principles.

Opportunity

New company:

No operating culture, performance metrics, or infrastructure legacy.

+

New technology:

Internet. Broadband. PDAs. XML. Enterprise IT. eBusiness.

+

New environment:

More uncertain, connected, knowledgeable. Faster. Always changing.

+

New customer expectations:

Personal attention. Immediate response. Self service.
Lots of information.

= New Opportunity

to design a company IT support system

fit to the new and changing environment,

and focused on new values

Objectives

Supporting strategy with best-fit tools
is enabled rather than inhibited

Switching/upgrading to new technology and applications
is enabled rather than inhibited.

Accommodating custom electronic "partner" relationships
is enabled rather than inhibited.

Integrating new plants, facilities, mergers, and acquisitions
is enabled rather than inhibited.

All information is accessible electronically
to those authorized to see it.

Electronic "dashboards" will provide real-time vision and monitoring
of operational and strategic activities.

Provide competitive advantage through
enterprise visibility, adaptability, and latest technology

General Strategy

Business System Analyst (BSA) Group:

- Assigned to IT-assist dept managers (cross dept responsibilities)
- Business Process IT application configuration/evolution
- IT tool selection/acquisition

Strategic System Analyst (SSA) Group:

- Evolution of infrastructure framework
- Enforcing infrastructure usage rules

User Collaboration:

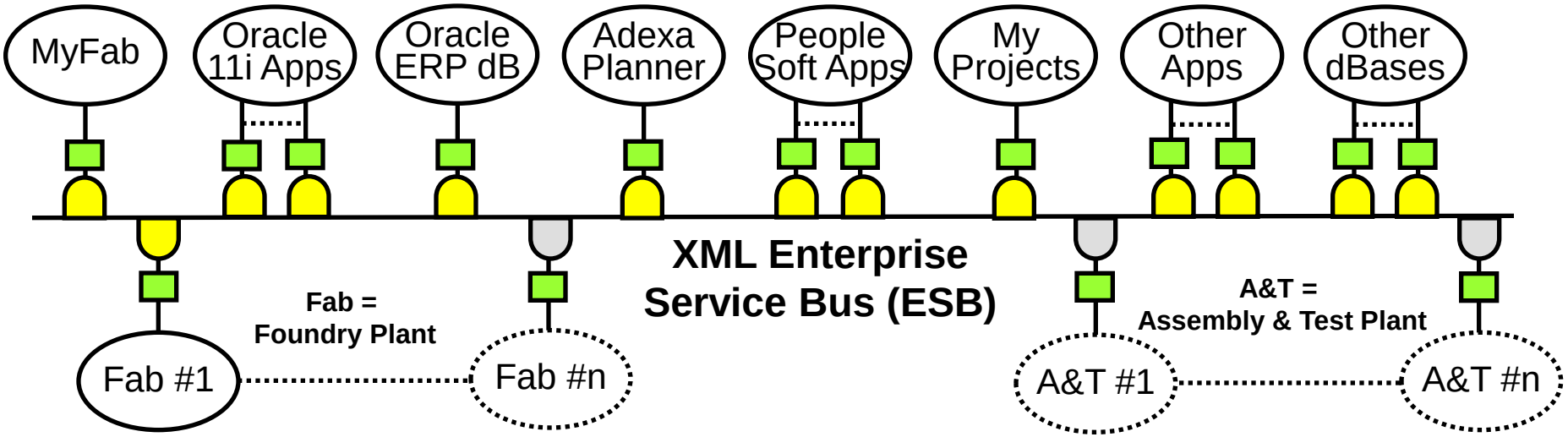
- Mandatory Response Situation Analysis (agility-tool)

COTS Applications: No customization of purchased software

IT Internal Responsibilities – not to be outsourced:

- Infrastructure architecture design and evolution
- Management of installation/integration projects
- Configuration of applications

Enterprise IT-Infrastructure Architecture/ConOps



- = ESB Interface Module (BIM)
- = Extract/Transfer/Load (ETL) Interface Modules
- MyProjects = Web-accessible strategic-project portfolio manager
- MyFab = Web-accessible operations transparency

www.parshift.com/Files/PsiDocs/Rkd050324CserPaper.pdf

Project Development ConOps – Strategy/Rules

- Vendor is responsible for total solution: HW and SW

- Requirements will not change during implementation

- No expedient customization allowed

- Three Phase Implementation Sequence:

  P1: Out-of-box best-practice from vendor – supporting the company
  Vendors configure the applications

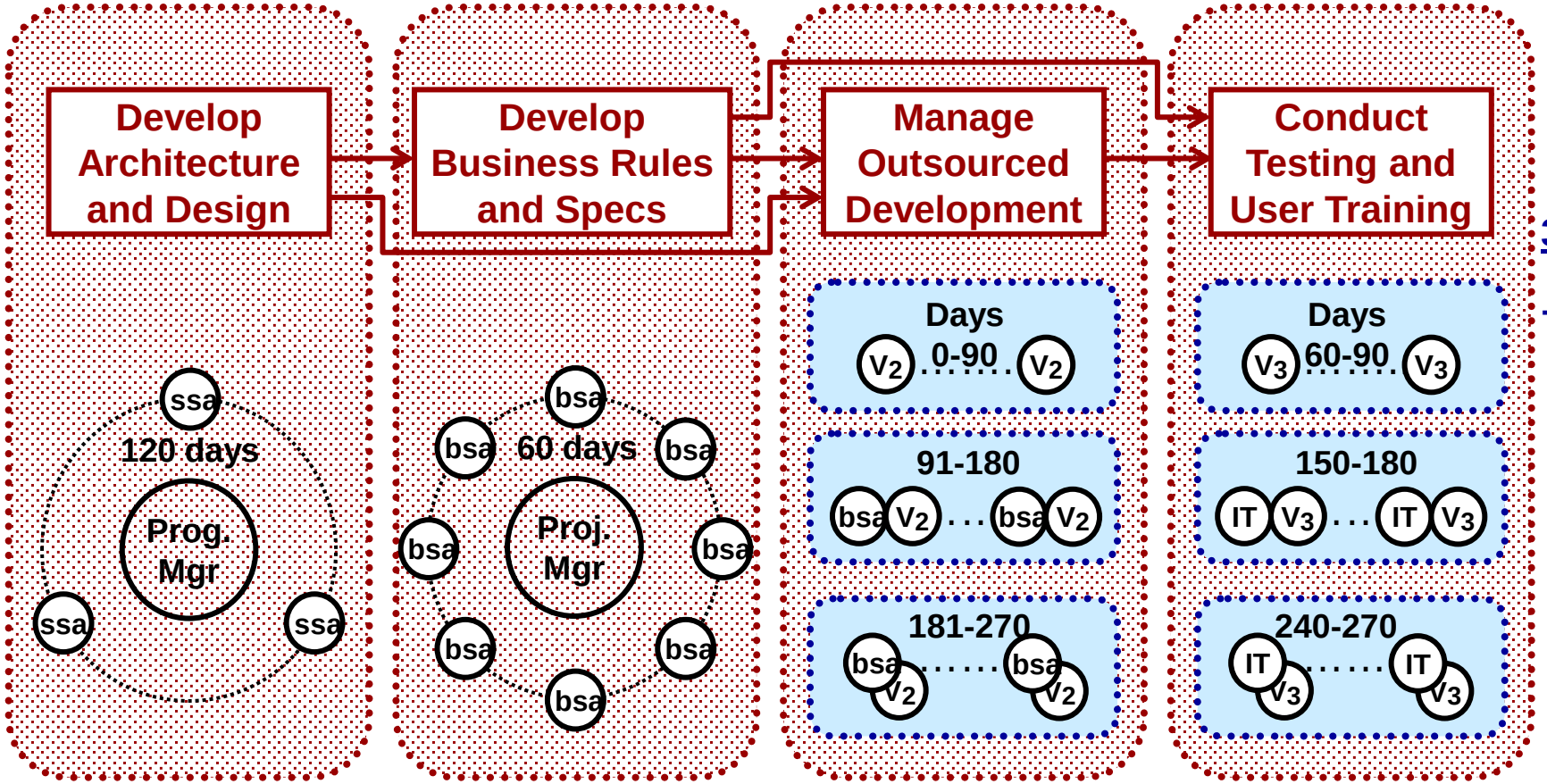
  P2: BSA-developed business process rules
  Vendors + BSAs configure the applications

  P3: Refined (learned) business processes
  BSAs configure the applications

- No violation of infrastructure rules (repeatedly invoked)

- Don't say it can't be done, tell what is needed to do it (repeatedly invoked)

Incremental/Iterative SE Life Cycle with Encapsulated Modules



- Designed to Accommodate Requirements Evolution -

www.parshift.com/Files/PsiDocs/Rkd050324CserPaper.pdf

Effective Response Under Changing Conditions

ERP on time, below budget, on spec

- **3 months functional ERP "best practice" (Phase 1)**
- **3 months later preferred business processes (Phase 2)**
- **3 months later refined business processes (Phase 3)**

HRM modularized and
added below time, on budget, on spec

Adexa planner
added on time/budget/spec

Existing Time and Attendance system
modularized and integrated on time/budget/spec

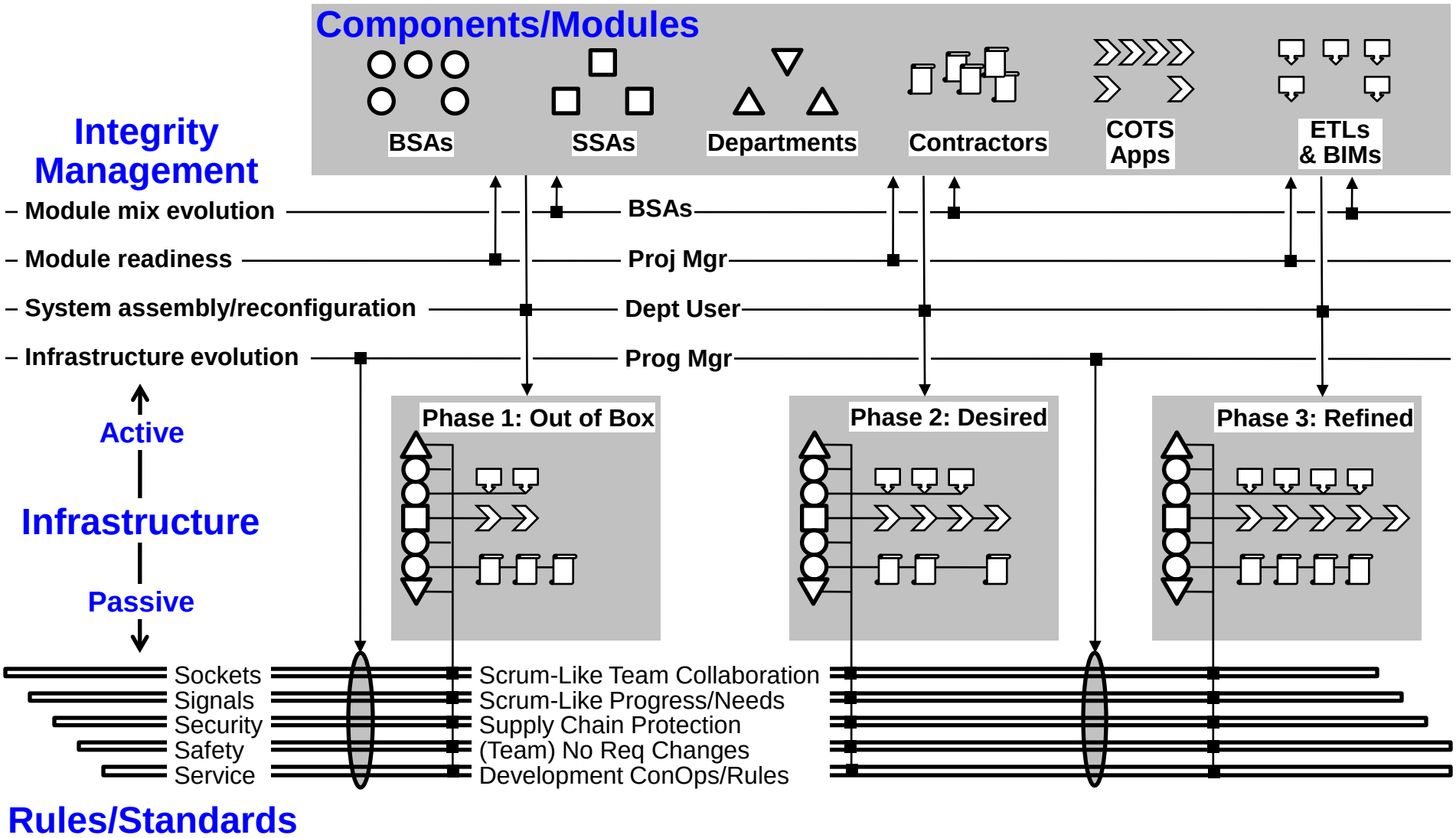
Wish	Typical Imp	Actual Imp
ERP in 12 mos total	24-36 mos	12[1,2]
75% of license budget	200-300%	75%
$10 Million (5 + 5)	$15-25 Million	$9 Million
HRM in 6 mos	12-18 mos	5 mos

HOW??

- Principle-based installation/integration methodology and management
- Adherence to methodology (ie, effective management)
- BSAs utilizing MBW tool to develop and capture business processes
- BSAs taking responsibility for integrating ERP with users
- Bus architecture connecting ERP with HRM
- Experienced outsource to help integrate ERP/CIM[2,3] (did it before)
- Expertise in agile system design and implementation

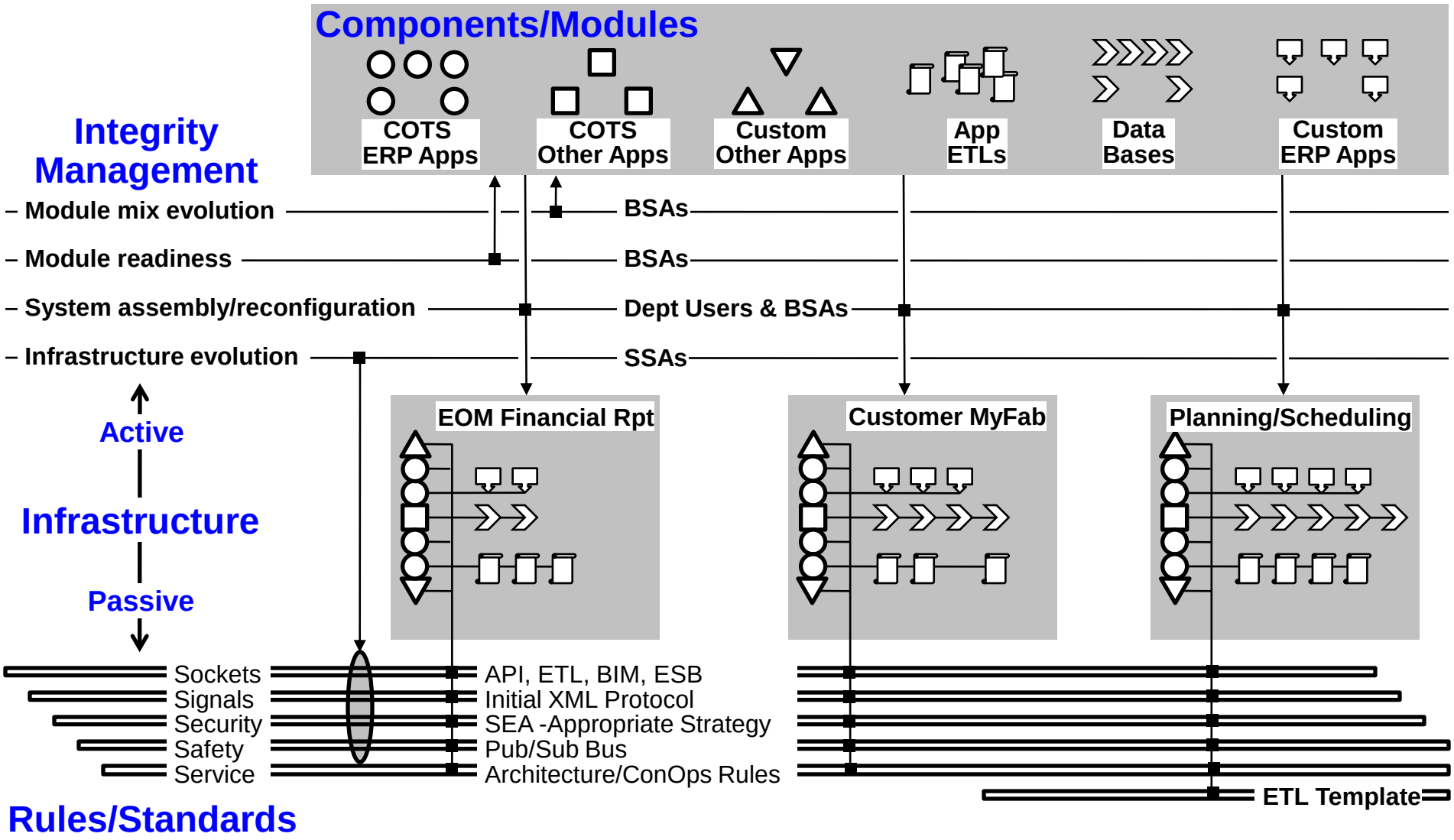
Notes: 1) 12 months = 3 mo concept design and vendor selection + 9 mo implementation,
time included infrastructure bus/ETL/BMI implementation, but not shop floor (CIM) integration (+6)
2) New Oracle 11i ERP with typical bugs and lack of documentation of new systems
3) Additional 6 mos due to independent CIM system shake out

Silterra Agile ERP – Development System



Silterra Agile ERP – Developed System

System examples are SOA-like instances of departmental needs



Re: Agile Software Development

Be aware of the difference between:

- **Agile (a branded software development process) and**
- **agile (a dictionary defined capability/property)**

Agile System-Engineering

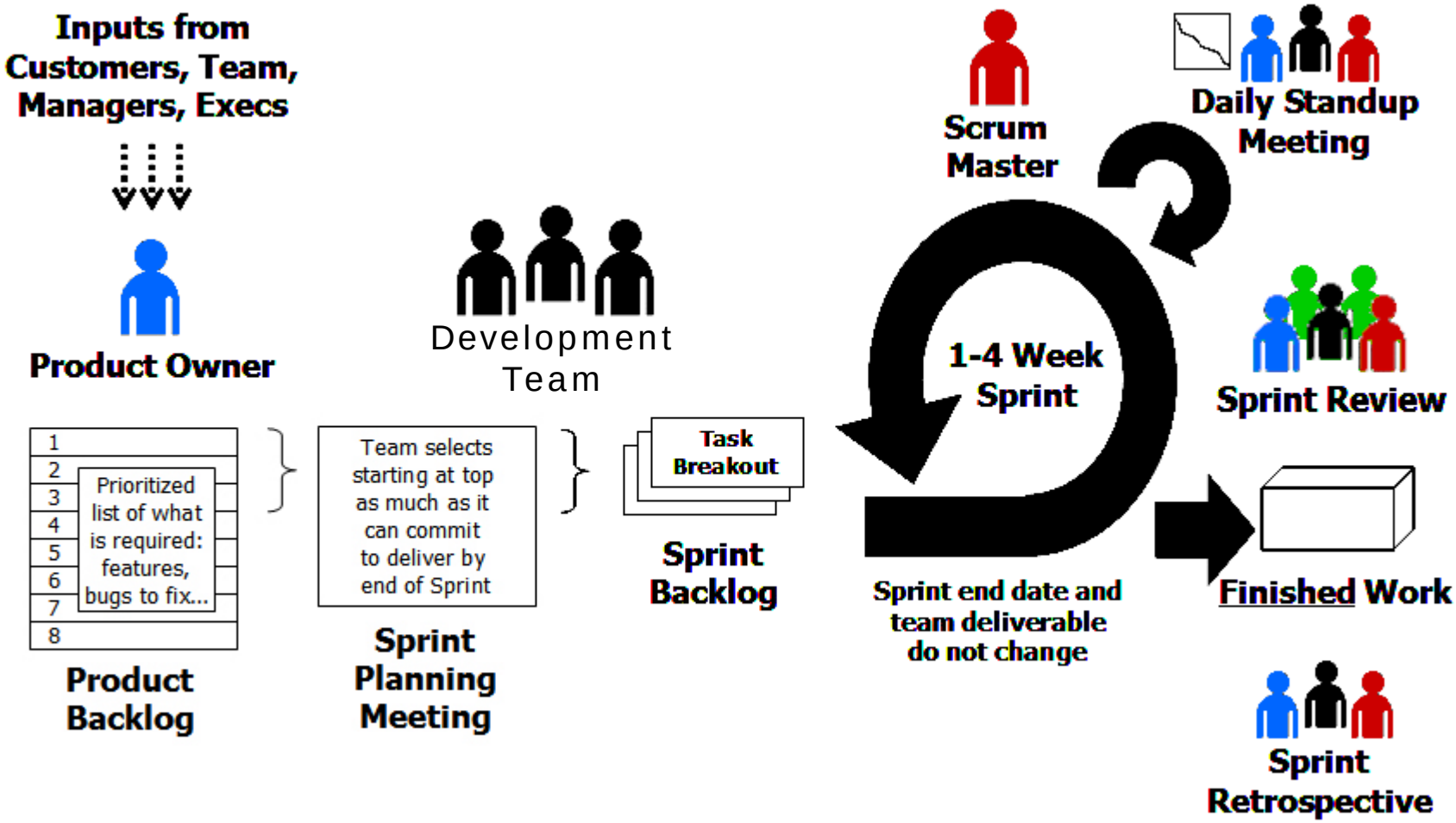
is an instance of

Agile-System Engineering

“Classic” Scrum

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

Jeff Sutherland, Ken Schwaber. 2007. The Scrum Papers: Nuts, Bolts and Origins of an Agile Process. Scrum Foundation. http://scrumfoundation.com



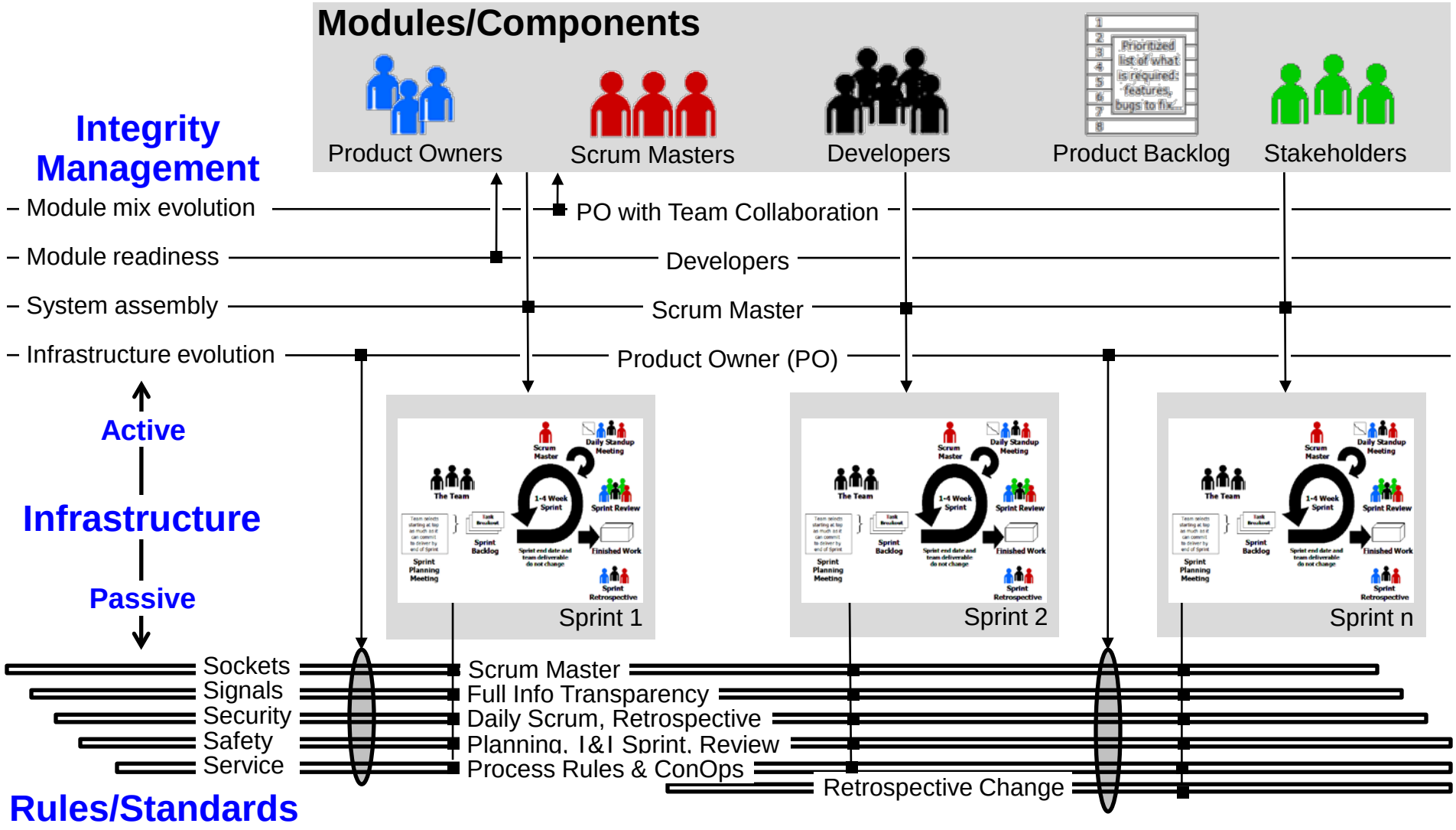
“Scrum’s roles, artifacts, events, and rules are immutable, and although implementing only parts of Scrum is possible, the result is not Scrum.

Scrum exists only in its entirety, and functions well as a container for other techniques, methodologies, and practices.” (Schwaber and Sutherland 2013)

Classic Scrum: an Agile Architecture Pattern (AAP) *Structure*

suitable for agile SW development, but not for agile systems-engineering …

Dove, Rick and Ralph LaBarge. 2014. Agile Systems Engineering – Part 2. International Council on Systems Engineering IS14 Conference, Los Angeles, CA, 30-Jun-03Jul. www.parshift.com/s/140630IS14-AgileSystemsEngineering-Part2.pdf



Conceptual Example of Design Principles Analysis (RRS)

Details in (Dove & LaBarge 2014)

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

Encapsulated Modules Product owners, Scrum masters, developers, product backlog, stakeholders, …	Reusable	Scalable	**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

Wrapping it Up

On Passive infrastructure

…protocols (infrastructure) are far more important … than are modules

Marie E. Csete and John C. Doyle. 2002. Reverse Engineering of Biological Complexity. Vol 295 SCIENCE, 1 March.
www.cds.caltech.edu/~doyle/CmplxNets/CseteDoyle.pdf

Consider the ubiquitous Lego toy system*. The signature feature of* Lego is the patented snap connection for easy but stable assembly of components. The snap is the basic Lego protocol, and Lego bricks are its basic modules.

We claim that protocols are far more important to biologic complexity than are modules. They are complementary and intertwined but are important to distinguish. In everyday usage, protocols are rules designed to manage relationships and processes smoothly and effectively.

If modules are ingredients, parts, components, subsystems, and players, then protocols describe the corresponding recipes, architectures, rules, interfaces, etiquettes, and codes of conduct.

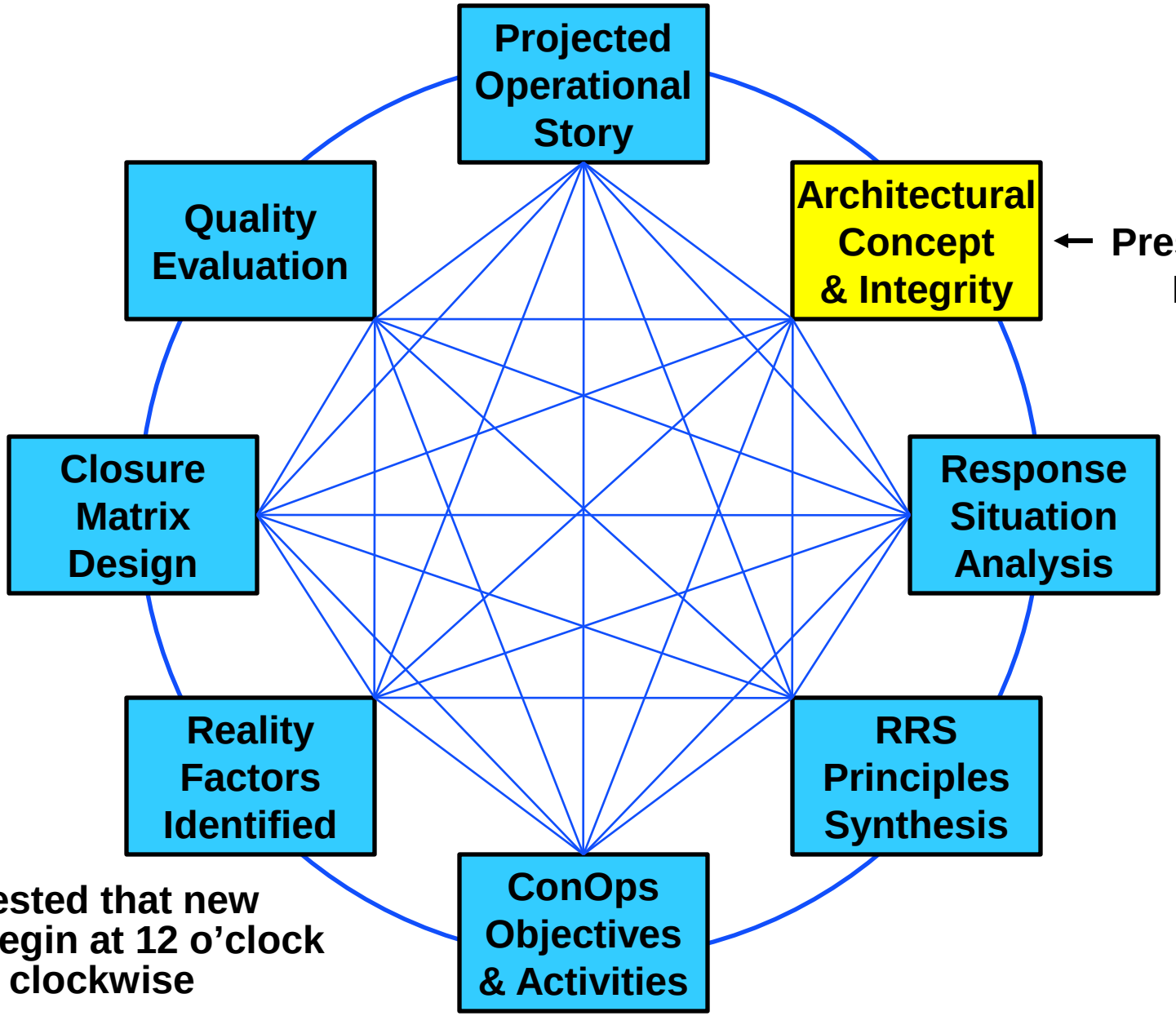
Protocols here are rules that prescribe allowed interfaces between modules, permitting system functions that could not be achieved by isolated modules.

Protocols also facilitate the addition of new protocols and organization into collections of mutually supportive protocol suites.

Like modules, they simplify modeling and abstraction, and as such may often be largely "in the eye of the beholder."

A good protocol is one that supplies both robustness and evolvability.

Eight principle tools to employ when designing or analyzing a system for agility



← This Presentation Focus

It is suggested that new initiates begin at 12 o'clock and move clockwise

Agility - Fundamentally

The Ability to Thrive in a Continuously Changing, Unpredictable Environment.

**Agility is *effective response* to opportunity and problem,
within mission ... always … no matter what.**

An *effective response* is one that is:	**Metric**
■ **timely (fast enough to deliver value),**	**time**
■ **affordable (at a cost that leaves room for an ROI),**	**cost**
■ **predictable (can be counted on to meet expectations),**	**predictability**
■ **comprehensive (anything/everything within mission boundary).**	**scope**

You can think of Agility as Requisite Variety.

You can think of Agility as proactive Risk Management.

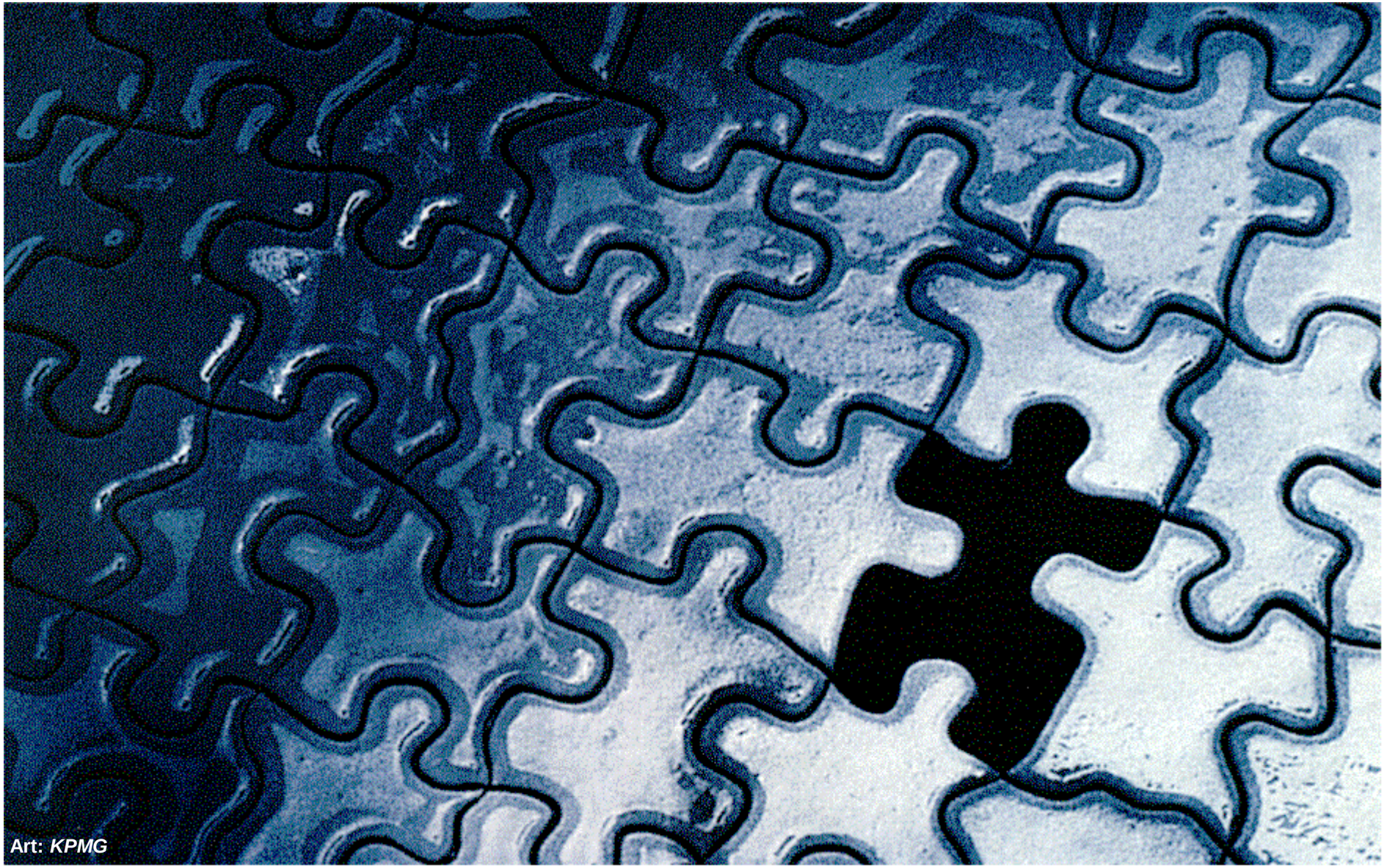
You can think of Agility as Innovative Response in unpredictable situations.

You can think of Agility as Life Cycle Extension.

**The trick is understanding the nature of agile-enabling fundamentals,
and how they can be applied to any type of system/process.**

Domain Independent

Modular – But Not Agile



Agile System and Project Management by Design

Risk and Uncertainty Management Through:

- **Creation of drag-and-drop response options**
- **Enabling effective plug-and-play use of options**
- **Agility management through active & passive infrastructure responsibility that evolves the system constantly**

System X-Ray Vision

The bone structure is depicted in the Agile Architecture Pattern.
All truly agile systems have the same basic structure and strategy.
Knowing this will change the way you "see" and evaluate a system.

http://awespendo.us/animemangacomics/kermit-at-the-doctor/

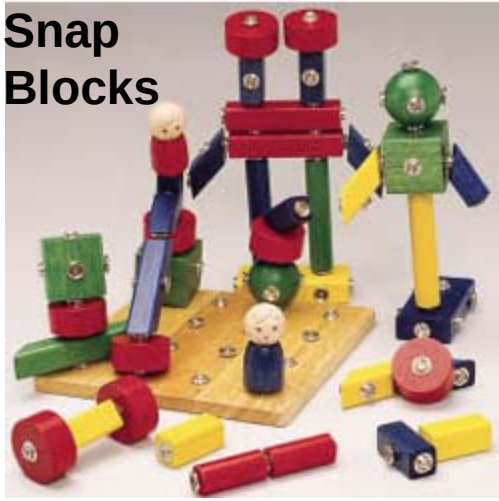
Design the Architecture of Your Construction Set

Bendables

Straws and Connectors

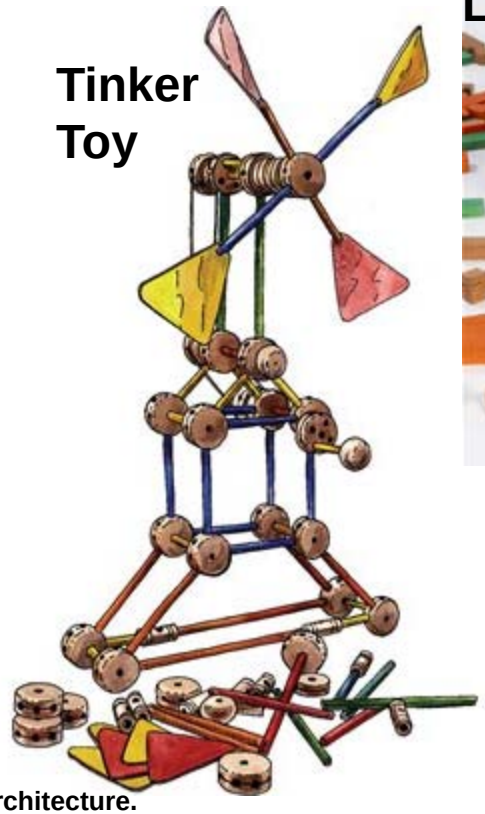
Marble Run

Snap Blocks



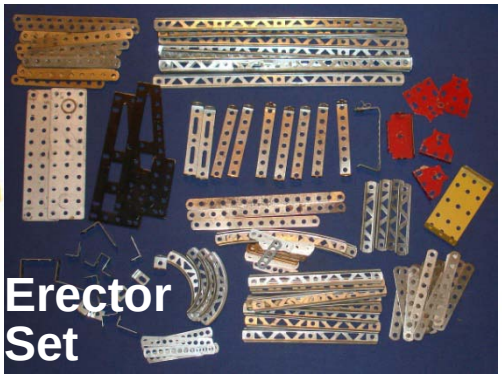
Woodbuilders

Tinker Toy



Log Builder

Lego



Bristle Blocks

Construction (response) architecture different from system functional architecture.
Response architecture is a domain-focused engineering architecture

Agile Systems and Systems Engineering (AS&SE) Working Group

An INCOSE Working Group

(International Council on Systems Engineering)

On Request to rick.dove@parshift.com:

1. **Get on mail list for general announcements.**
2. **Participate in WG remote-collaboration projects.**
3. **Get working group charter.**

Chair: Rick Dove
Co-Chair: Ron Lyells, Honeywell
Co-Chair: Mike Coughenour, Lockheed Martin

References and Supportive Readings

(Alberts 1996) David S. Alberts. Revised 2002. *Information Age Transformation – Getting to a 21st Century Military*. DoD Command and Control Research Program (CCRP). www.dodccrp.org/html4/books_downloads.html.

(Alberts 2011) David S. Alberts. *The Agility Advantage: A Survival Guide for Complex Enterprises and Endeavors*. DoD Command and Control Research Program (CCRP). www.dodccrp.org/html4/books_downloads.html.

(Bohem 2004) B. Boehm and R. Turner, R., *Balancing Agility and Discipline – A Guide for the Perplexed, Addison*-Wesley, 2004.

(Boehm 2014) Barry Boehm, 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.

(Boss 2010) Jason Boss and Rick Dove. Agile Aircraft Installation Architecture In a Quick Reaction Capability Environment. INCOSE International Symposium 14Jul2010, Chicago. www.parshift.com/Files/PsiDocs/Pap100712IS10-AgileAircraftInstallationArchitecture.pdf

(Dove 1996) Rick Dove, Sue Hartman and Steve Benson. An Agile Enterprise Reference Model – with a case study of Remmele Engineering. Agility Forum, Report AR96-04. www.parshift.com/Files/PsiDocs/AerModAll.pdf

(Dove 2001a) Rick Dove. Response Ability – The Language, Structure and Culture of the Agile Enterprise. Wiley.

(Dove 2001b) Rick Dove. Design Principles for Highly Adaptable Business Systems, With Tangible Manufacturing Examples. Book chapter in Maynard's Industrial Handbook, McGraw Hill. http://www.parshift.com/Files/PsiDocs/Rkd8Art3.pdf

(Dove 2005) Rick Dove. Fundamental Principles for Agile Systems Engineering. Conference on Systems Engineering Research (CSER), Stevens Institute of Technology, Hoboken, NJ, March. www.parshift.com/Files/PsiDocs/Rkd05032.pdf

(Dove 2006) Rick Dove. Engineering Agile Systems: Creative-Guidance Frameworks for Requirements and Design. 4th Annual Conference on Systems Engineering Research (CSER), Los Angeles, CA, Apr 7-8. www.parshift.com/Files/PsiDocs/Rkd060407CserEngineeringAgileSystems.pdf

(Dove 2008a) Rick Dove and Garry Turkington. Relating Agile Development to Agile Operations. Conference on Systems Engineering Research (CSER), Redondo Beach, CA, April. www.parshift.com/Files/PsiDocs/Pap080404Cser2008DevOpsMigration.pdf

(Dove 2008b). Rick Dove. Embedding Agile Security in Systems Architecture. INSIGHT 12(2):14-17, INCOSE. www.parshift.com/Files/PsiDocs/Pap090701Incose-EmbeddingAgileSecurityInSystemArchitecture.pdf

(Dove 2009) Rick Dove and Garry Turkington. On How Agile Systems Gracefully Migrate Across Next-Generation Life Cycle Boundaries. Global Journal of Flexible Systems Management, Vol 10, No 1, pp 17-26, 2009. www.parshift.com/Files/PsiDocs/Pap080614GloGift08-LifeCycleMigration.pdf

(Dove 2010) Rick Dove. Pattern Qualifications and Examples of Next-Generation Agile System-Security Strategies. IEEE International Carnahan Conference on Security Technology (ICCST), San Jose, CA, 5-8 Oct. www.parshift.com/Files/PsiDocs/PatternQualificationsForAgileSecurity.pdf

(Dove 2011a) Rick Dove. Patterns of Self-Organizing Agile Security for Resilient Network Situational Awareness and Sensemaking. 2011 Eighth International Conference on Information Technology: New Generations. www.parshift.com/s/110411PatternsForSORNS.pdf

(Dove 2011b) Rick Dove. Self-Organizing Resilient Network Sensing (SornS) with Very Large Scale Anomaly Detection. IEEE International Conference on Technologies for Homeland Security, Waltham, MA, 15-17Nov. www.parshift.com/s/111115VeryLargeScaleAnomalyDetection.pdf

(Dove 2014), Dove, Rick and Ralph LaBarge. Agile Systems Engineering – Part 1&2. International Council on Systems Engineering IS14 Conference, Los Angeles, CA, 30-Jun-03Jul. www.parshift.com/s/140630IS14-AgileSystemsEngineering-Part1&2.pdf

(Papke 2013) Barry Papke and Rick Dove. Combating Uncertainty in the Workflow of Systems Engineering Projects. INCOSE IS13. www.parshift.com/s/130624Last Planner.pdf

(Sillitto 2013) Hillary G. Sillitto. Composable Capability – Principles, Strategies and Methods for Capability Systems Engineering. INCOSE International Symposium, Philadelphia, PA 24-27 June.

(Turner 2007) Richard Turner. Toward Agile Systems Engineering Processes. CrossTalk, April, pp 11-15.

Download 103 webinar slides: www.parshift.com/s/AgileSystems-103.pdf

Download 102 webinar slides: www.parshift.com/s/AgileSystems-102.pdf

Download 101 webinar slides: www.parshift.com/s/AgileSystems-101.pdf

(updated asynchronously from time-to-time)

Agile Modeling and Modeling Agile Systems

3 Part MBSE Break Out Session
IW15 – Saturday 24 Jan 2015

Part 3 –
Domain Independent
Agile Systems Engineering Life Cycle Model Fundamentals
Discovery Project

Rick Dove

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.**

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)

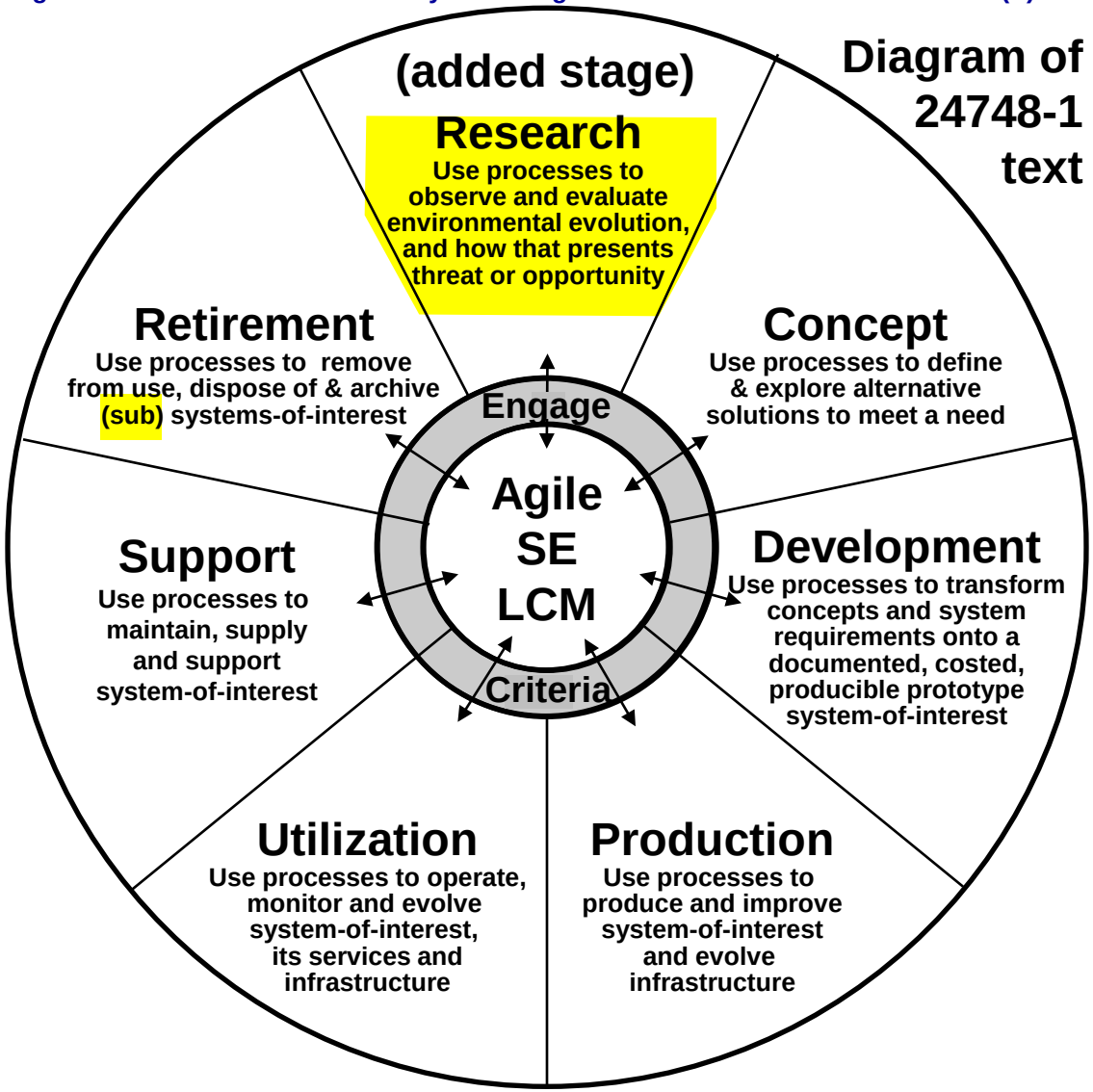
Section 5.5.5 (p. 32):

"… 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:

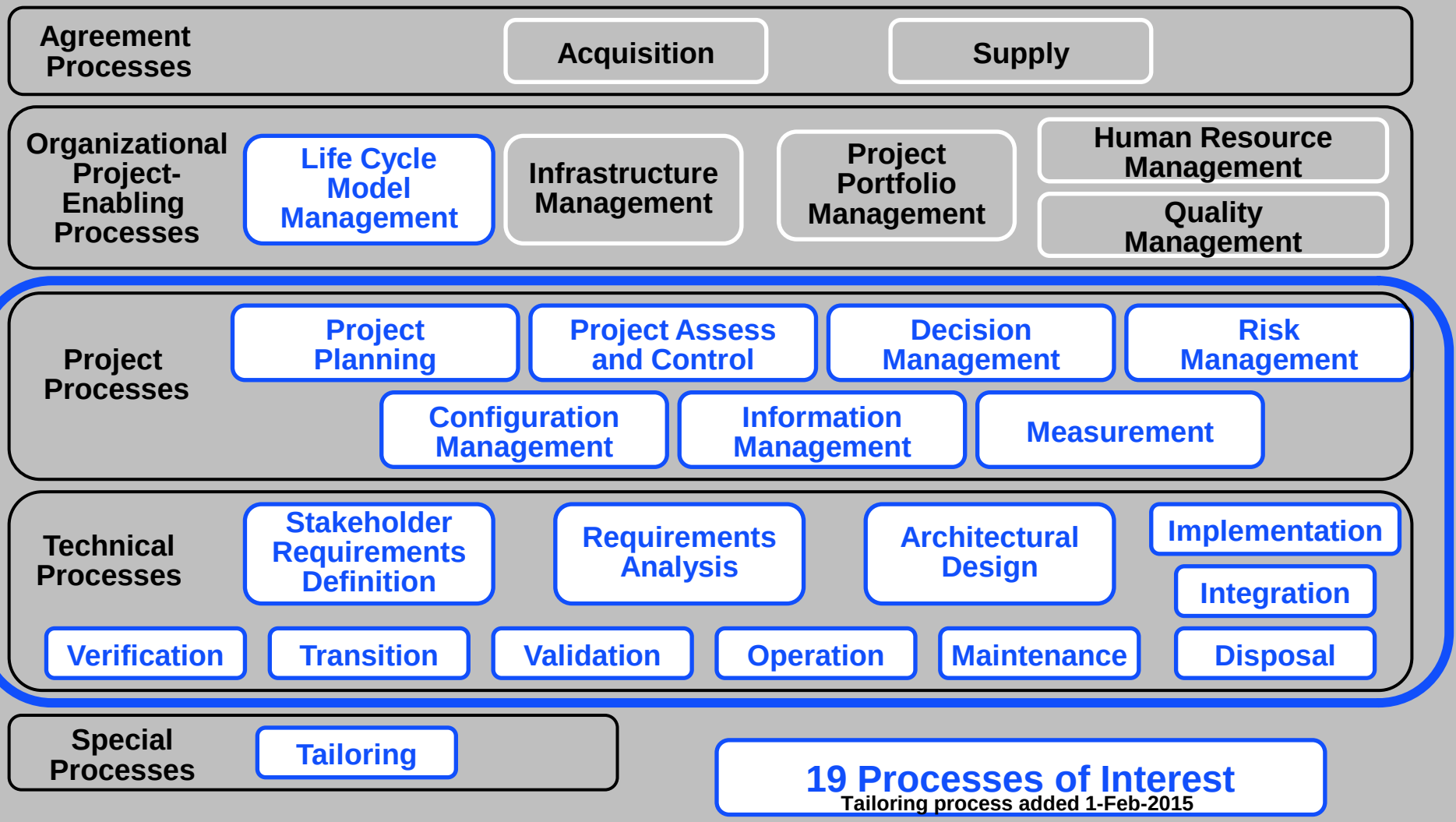
A. **Discover generic principle-based life-cycle stages/processes/activities that can be intuitively embraced and applied, rather than compromised by situational reality factors, for dealing with uncertain, unpredictable, evolving SE environments.**

B. **Cover several varieties of agile SE projects, e.g.:**

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.**

ISO/IEC/IEEE 15288–2008 Processes



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. **Pattern Based SE Modeling (PBSE) will illustrate configurations aligned to the case studies (next slide).**

3. **Supplemental guidance for application and/or tailoring of SE processes contained in ISO/IEC/IEEE 15288 (potential future Annex or part of guides) and INCOSE SE Handbook.**

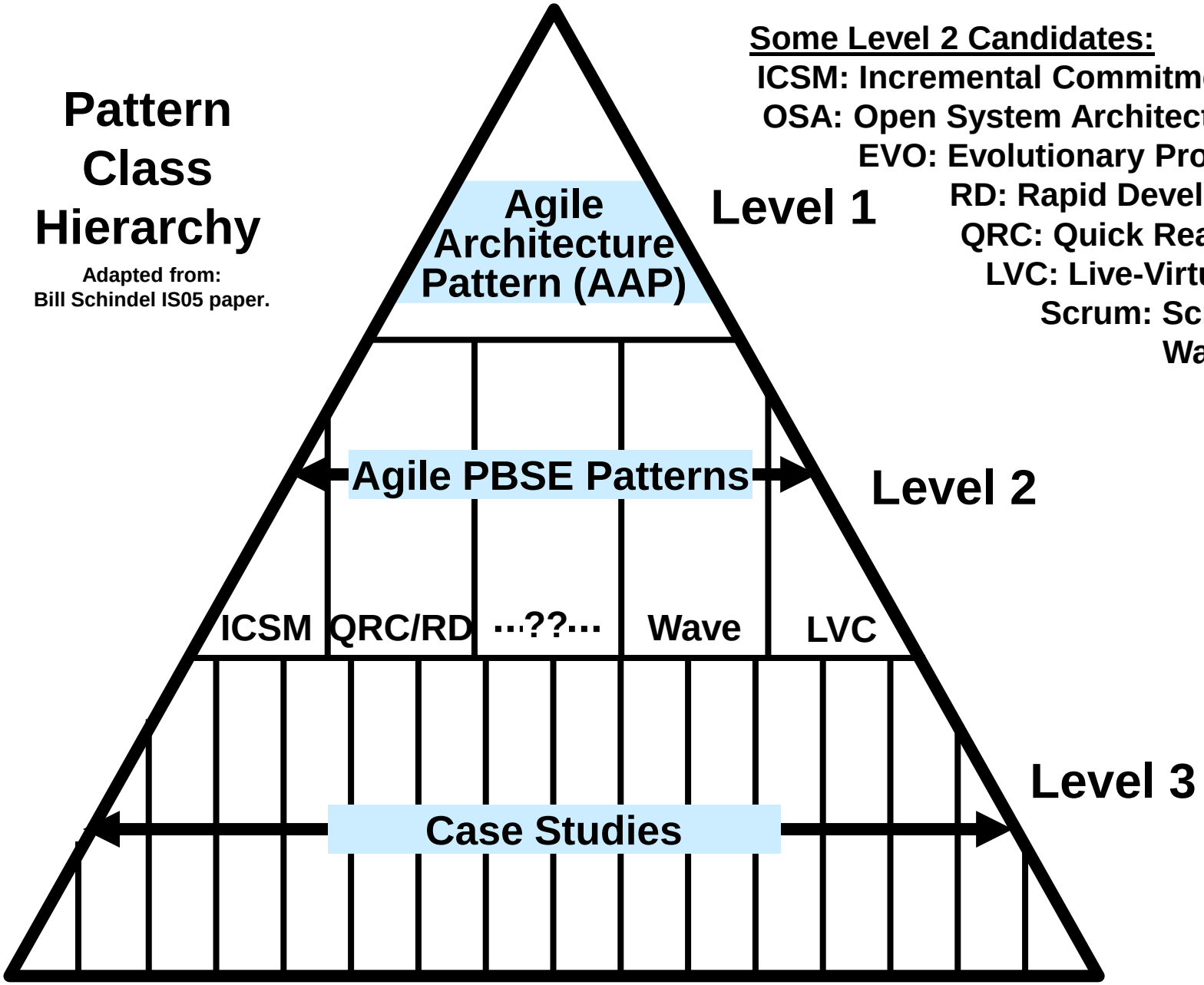
4. **Collateral technical information in briefer form and focus is anticipated as papers targeted for relevant SE journals and conferences.**

Estimated project report completion is later half of 2016

Pattern-Based System Engineering (PBSE)

Pattern Class Hierarchy

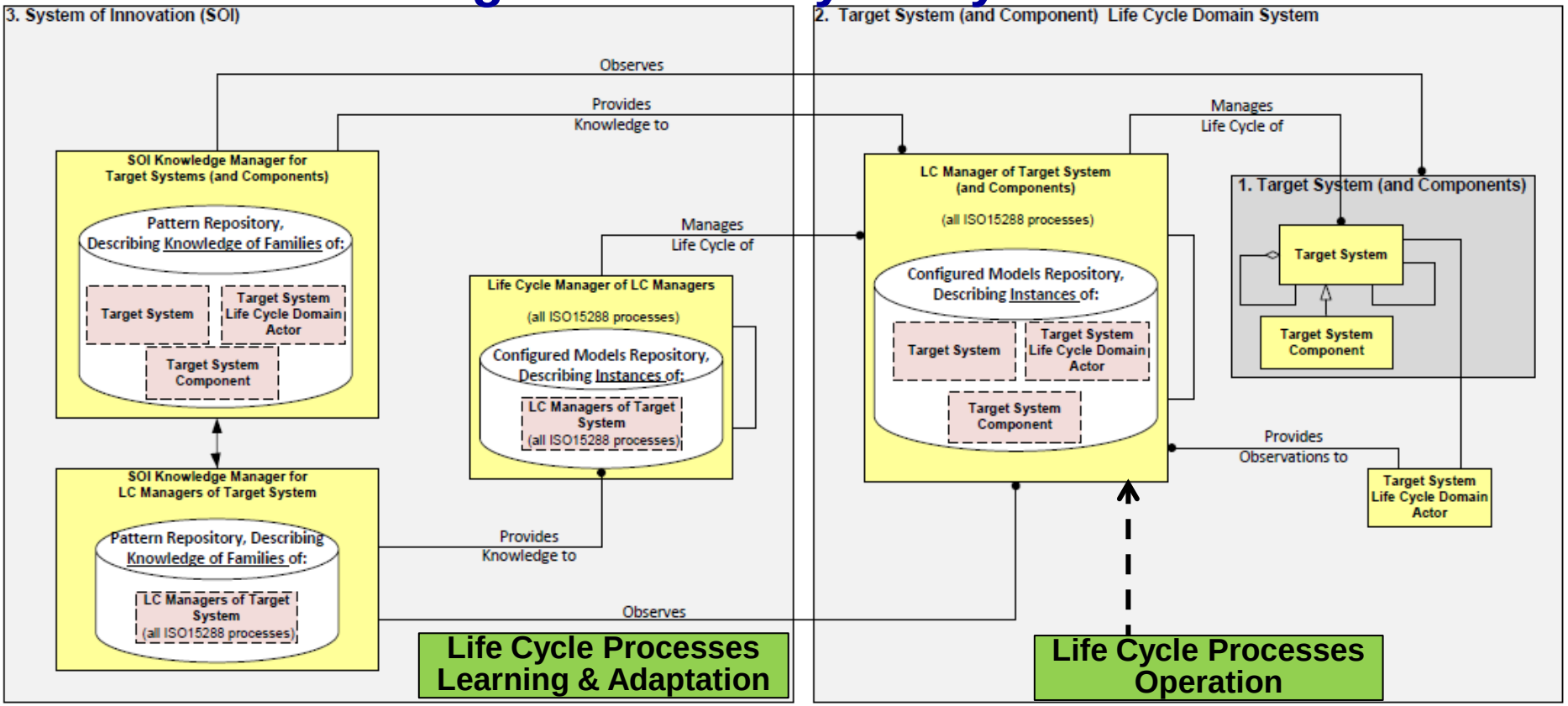
Adapted from:
Bill Schindel IS05 paper.



Some Level 2 Candidates:

ICSM: Incremental Commitment Spiral Model
OSA: Open System Architecture PM concept
EVO: Evolutionary Project Managment
RD: Rapid Development/Fielding
QRC: Quick Reaction Capability
LVC: Live-Virtual-Constructive
Scrum: Scrum PM concept
Wave: Wave model

Pattern Framework for the Three High-Level Agile SE Life Cycle Systems



- **System 1 Features: Stakeholder capabilities of the Target System—the system we ultimately want to respond (with help from Systems 2 and 3) in agile fashion.**
- **System 2 Features: Stakeholder capabilities of the Target System Life Cycle Management System. This includes all aspects of its LC, a subset of which are relevant to the Agile Systems LC Pattern.**
- **System 3 Features: Stakeholder capabilities of the three subsystems of System 3—concerned with observing and learning about the Target System and its Environment, and about the Target System LC Manager; also responsible for managing the LC of the Target System LC Manager.**

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 15288-2008 to specifically analyze 19 selected Processes.**

2. **Workshop Hosts will provide discussion and presentation of one completed agile-SE experience 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. **Non-Host Traveling Participants may fill out workshops to max 20 total participants, with each participant, Host and non-Host, 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.

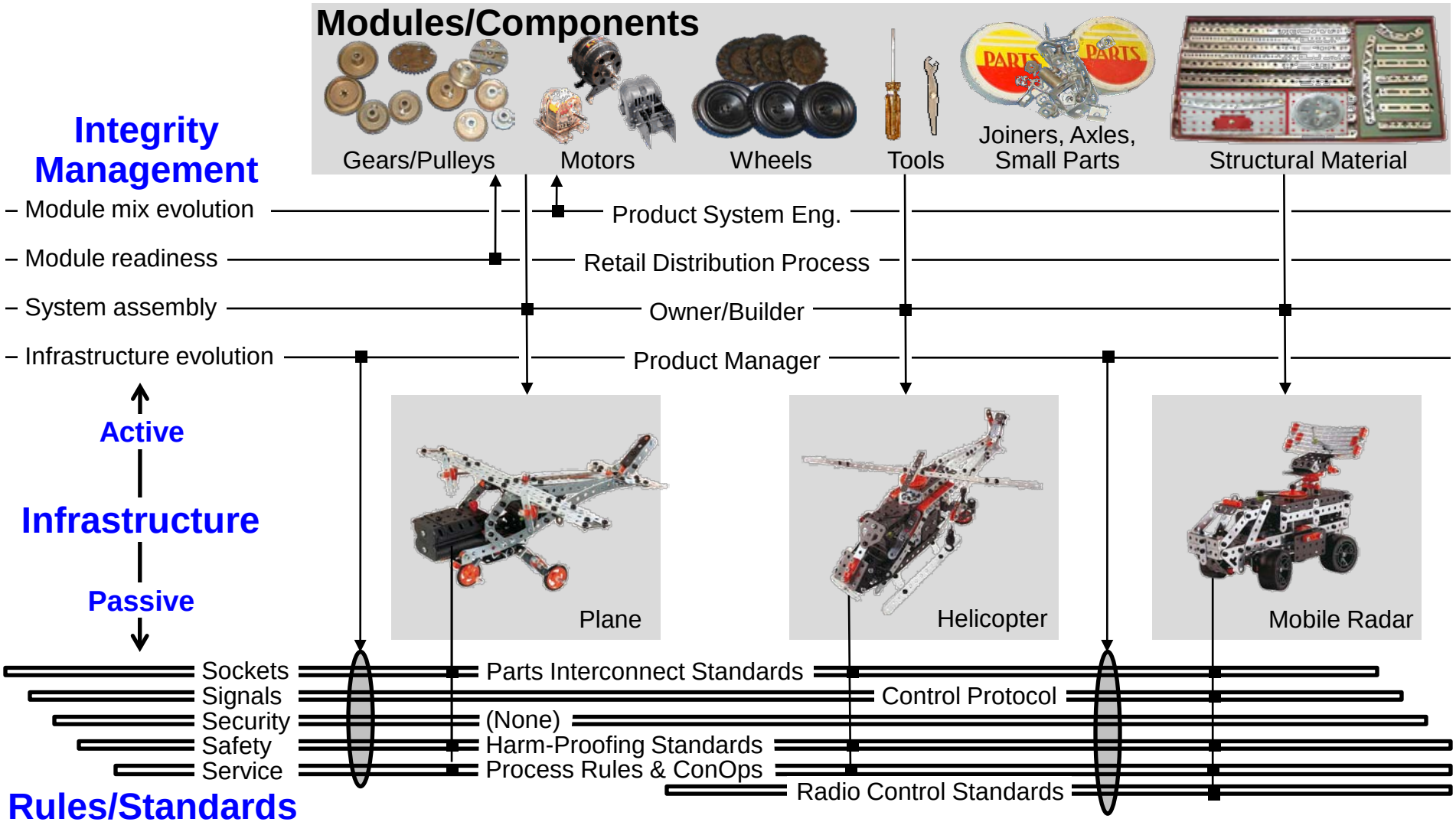
5. With a structured synthesis approach, apply discovery and provide benefit to workshop hosts and participants with an application of accumulated learning to a relevant host opportunity or problem.

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.

Notional Concept: Agile Architecture Pattern (AAP)

System Response-Construction Kit

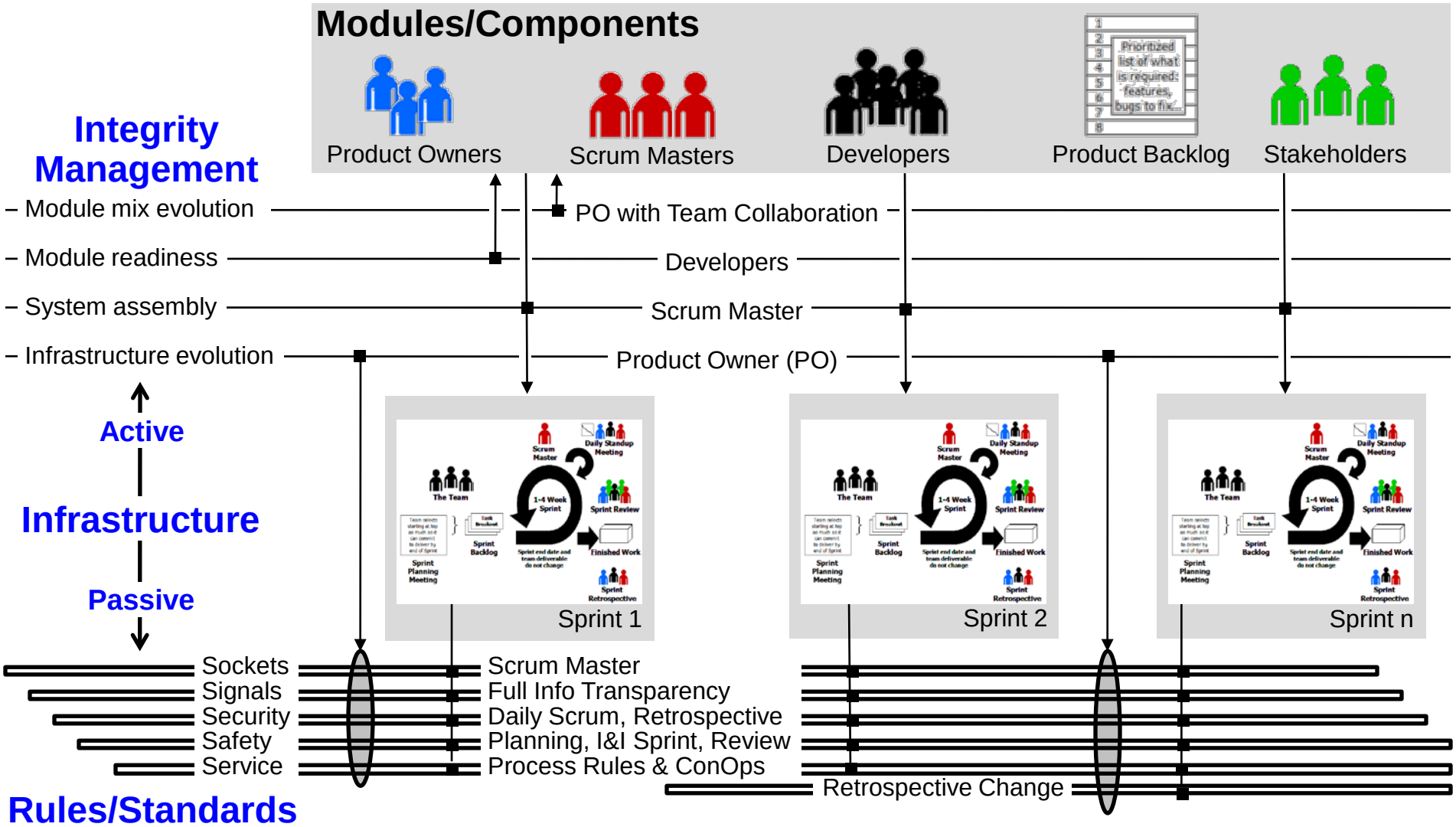
Details in www.parshift.com/s/140630IS14-AgileSystemsEngineering-Part1&2.pdf



Participants will construct AAP from Host discussion

Example: Scrum Agile Architecture Pattern (AAP)

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



Pro forma only – not expected to survive the project analysis work

Participants will construct RSA from Host discussion

Example: Scrum Response Situation Analysis (RSA)

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

Change Domain		Pro forma only – not expected to survive the project analysis work	
Proactive	Creation (and Elimination)	• requirements • experiments • next sprint activity	• shared team knowledge • customer satisfaction
	Improvement	• process effectiveness • risk/uncertainty reduction	• effort estimating • completion to schedule
	Migration	• new technology/tools that will impact infrastructure • lean SE process principles	
	Modification (of Capability)	• new team member unfamiliar/uncomfortable with agile SE • new environmental situation	
Reactive	Correction	• wrong requirement • wrong design • inadequate implementation	• non-compliant supplier • inadequate developer
	Variation	• expertise and skill levels among team members • allowable deliverable performance range • customer availability, interaction, 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 • system/sub-system design change	

Participants will construct Reality Factors from Host discussion

Example: Scrum 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	Pro forma only – not expected to survive the project analysis work
Human Behavior: Non-team behavior, error, expediency, uncommitted customer rep, …	
Organizational Behavior: Change in stakeholders, organizational priorities, resource access, ...	
Technology Pace: Evolving technology, testing trade-offs, ...	
Complexity: Large project with many involved simultaneously, emergent interaction affects, ...	
Globalization: Partners/teams with different ethics, cultures, infrastructures, …	
Partially-Agile Enterprise Concepts: Outsourcing, COTS affects, COTS supply/supplier affects, agile software practice-thinking dominance on HW/SW project...	
Agile Customers/Competitors/Adversaries: Continuous external-knowledge evolution, continuous external innovation, …	

Planned (Roughly) Workshop Agenda

---------- Day 1 – 8 hours of structured work starting at 8:00am, room open at 7:30.

2.00 – Introductions, objectives, workshop agenda structure, tools and processes, accumulated learning review.
2.00 – Host process presentation/discussion of SE UURVE situation and SE process (guide provided to host, analysis forms provided to participants).
Lunch(one hour lunch allows informal conversation)
2.00 – Break-out analysis of RSA/RF/AAP (two separate teams doing identical analysis on total SE process overview).
2.00 – Brief-out: Analysis results, discussion, and refinement.
Dinner (host-funded for all participants) at time TBD.

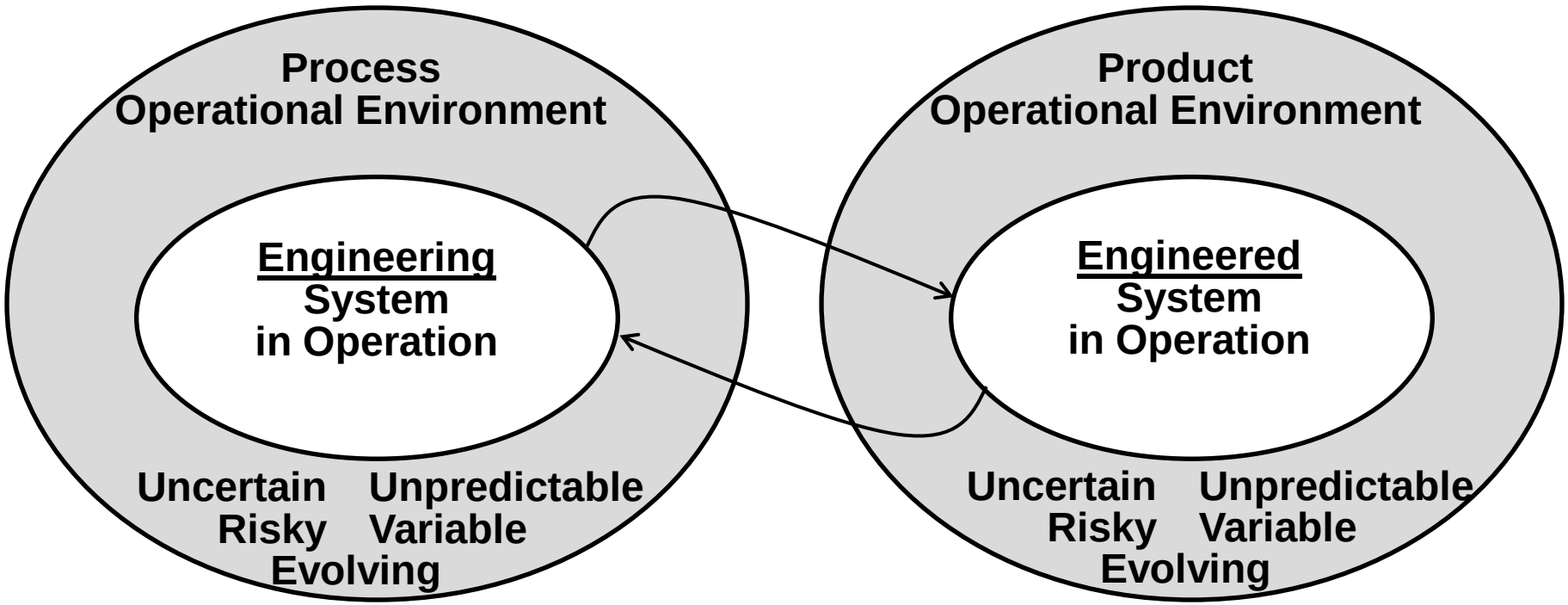
---------- Day 2 – 8 hours of structured work starting at 8:00am, room open at 7:30.

1.00 – Review of yesterdays salient learning.
3.00 – Host presentation and Q&A of 19 processes (guide and discussion templates provided to host outlining the points we need to hear and discuss).
Lunch(one hour lunch allows informal conversation).
2.00 – Break out ties 19 processes to RSA/RF with issue closure, and refines AAP of SE process overall.
2.00 – Brief-out: Analysis results and discussion.

---------- Day 3 – 8 hours of structured work starting at 8:00am, room open at 7:30.

1.00 – Review/discussion of yesterday's salient learning (with process/issue closure relations).
2.00 – Host presentation/discussion and Q&A of process challenge (in any form wished).
1.00 – Break out synthesis exercise – Synthesis exercise at overall process level – converge on key RSA issues with suggested process activity closure relations and general AAP elements.
Lunch(one hour lunch allows informal conversation).
2.00 – Break out cont. – Synthesis exercise at overall process level – converge on key RSA issues with suggested process activity closure relations and general AAP 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 have an agile product architecture

Action Plan

- ~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 March of 2015, approximately one/month.
- Traveling participants must participate in at least 3 workshops. Host sites must provide at least two 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 20 participants plus briefers. Participants are favored to be mostly from various Hosts.
- 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

Workshop Hosts:

- **Diagnostic analysis of an agile SE process experience for fundamentals that enable effective response in uncertain, unpredictable, evolving SE environments.**
- **Action-learning synthesis applied to a host situation in need of more agile capability.**
- **Understanding of necessary and sufficient enabling principles for any type of agile SE process on any type of project.**
- **Insightful competency developed among at least a few host participants for knowledgeable internal leadership.**
- **Influence where things are going, compatible with your environment.**

Traveling Participants:

- **Insightful competency for transformational leadership.**
- **Bench-mark exposure to HW/SW/WW agile SE processes.**

Systems Engineering Community:

- **Generic principle-based framework for knowledgably evaluating, choosing, tailoring, integrating, and evolving agile SE.**
- **Means to address SE dynamics with resilient & composable processes.**
- **Clarified agile-SE compatibility with 15288 and INCOSE Handbook.**

Status

INCOSE-PROJ-2014-01 Technical Project Plan approved 13-Oct-2014.

Project website with additional info at www.parshift.com/ASELCM/Home.html

Next

Host identification and scheduling.

Workshops will occur approximately one per month.

Identify and secure relevant host sites (yours?).

Workshops anticipated to begin in March of 2015.

Project Leadership:

- **Rick Dove, prior agile-fundamentals workshop series involvement**
- **Kevin Forsberg, V diagram and INCOSE Handbook involvement**
- **Bud Lawson, systems engineering text-book involvement**
- **Jack Ring, prior agile-fundamentals workshop involvement**
- **Garry Roedler, 15288 involvement**
- **Bill Schindel, PBSE concept involvement**

Ask us to schedule a Webinar to help your organization get involved

Breaking News

Local Motors Looks To Disrupt the Auto Industry With 3D-Printed Car Bodies

Local Motors solicits design ideas through crowdsourcing, allows anyone to use open source software to contribute ideas, and then 3D prints car bodies according to the chosen specs in a matter of days.

To prove they mean business, Local Motors 3D-printed a car on the floor of the Detroit Auto Show last week. "It took 44 hours to print the Strati's 212 layers. Once 3D printing is complete, the Strati moves to a Thermwood CNC router—a computer-controlled cutting machine that mills the finer details—before undergoing the final assembly process, which adds the drivetrain, electrical components, wiring, tires, gauges, and a showroom-ready paint job."

Here's another big difference from the current auto industry: "Customers can also bring their vehicles in at any time for hardware and software upgrades, or they can choose to melt their vehicle down and, for instance, add a seat.

Because Local Motors uses a distributed manufacturing system to make only what is purchased, it doesn't stock inventory. Anyone can come into a Local Motors microfactory, use its design lab, and work on a vehicle project free of charge."

INCOSE Project: Agile SE Life Cycle Model Fundamentals

Addressing SE UURVE: Uncertainty, Unpredictability, Risk, Variation, Evolution.

Objectives – Identified/justified necessary/sufficient fundamentals:

- **That can be intuitively embraced and applied.**
- **Compatible with 15288, any agile SE process, existing organizational cultures.**

Fifteen 3-day "discovery" workshops in US and Europe 2015/2016.

- **Workshop Hosts in defense and commercial sectors.**
- **Analyze SE processes dealing with UURVE in mixed HW/SW/WW projects.**
- **Immediately apply action-learning to an SE process in need of (more) agility.**
- **Workshop Hosts must send 2 participants to 2 other-Host workshops.**
- **Host cost ~$20k USD, which covers facilitation, synopsis reports, materials, estimated participant travel costs, workshop lunches, and one dinner.**

"Tell me and I forget. Teach me and I remember. Involve me and I learn."
Benjamin Franklyn

Town Hall 8:00-9:00 Tuesday provides details with Q&A.

Active In-Process Workshop Sites: Honeywell, General Dynamics, Lockheed, Northrop Grumman, Rockwell Collins, SPAWAR/MITRE, … You?

References

Bezdek, William J., Joel Maleport, Robert Z Olshan. 2008. Live, Virtual & Constructive Simulation for Real Time Rapid Prototyping, Experimentation and Testing using Network Centric Operations. American Institute of Aeronautics and Astronautics.

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&2.pdf

Gilb, Tom. 2005. (Re: Evolutionary Management) Competitive Engineering - A Handbook for Systems Engineering, Requirements Engineering, and Software Engineering Using Planguage. Elsevier.

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.