Agile Systems,
Agile Systems Engineering,
and
Self-Organizing Agile Systems Security

ILTAM / INCOSE IL, Tel Aviv
8-Jan-2014

Rick Dove
Paradigm Shift International
Stevens Institute of Technology

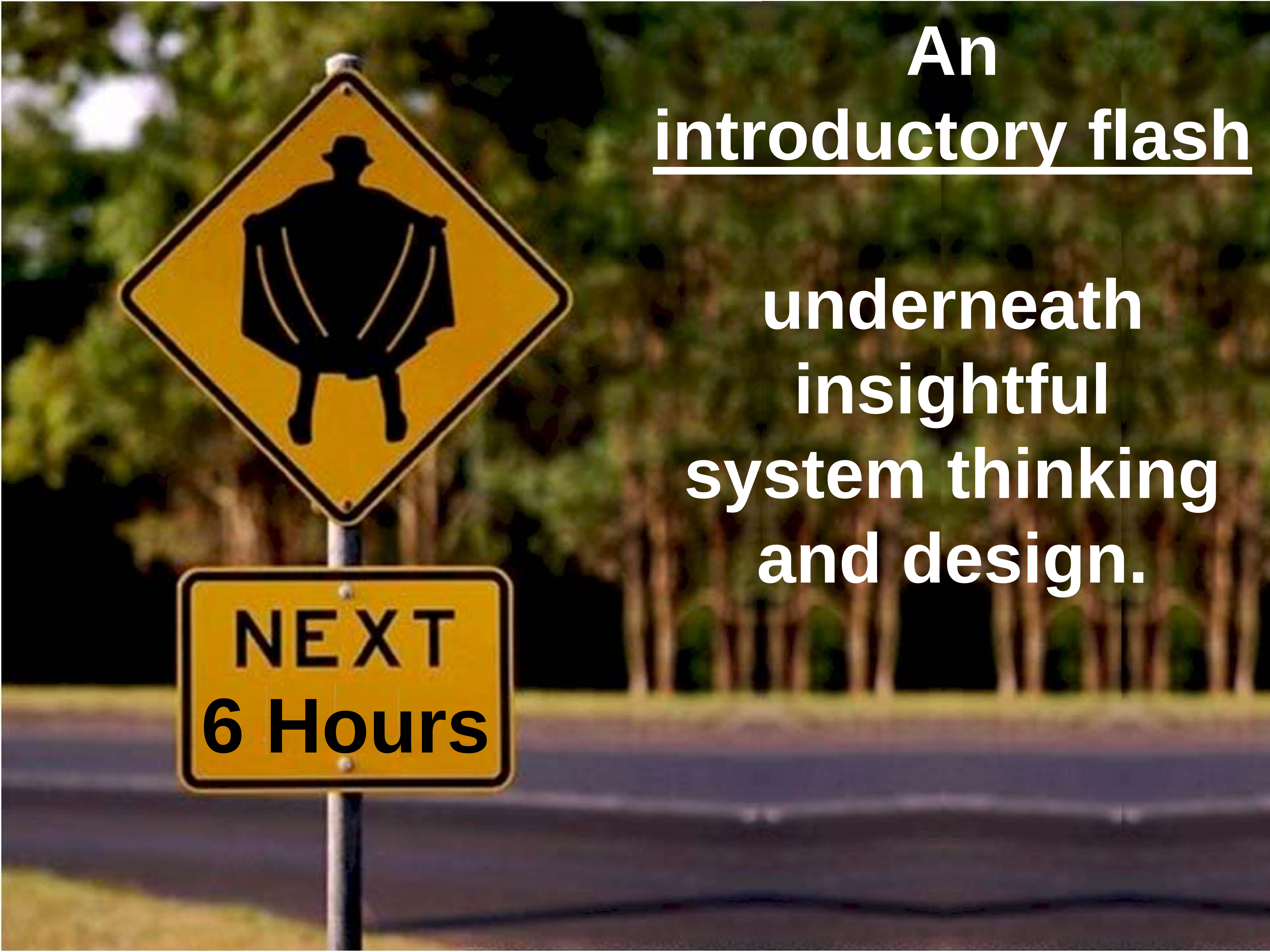
Objective and Agenda

The objective of this one day seminar is to review the basic engineering concepts that enable agility in systems of all kinds, and to introduce advanced concepts of self-organizing agility in the systems-security domain.

09:00-11:00	Agile 101 – Architecture Fundamentals
11:00-11:15	Break
11:15-12:45	Agile 102 – Requirements Fundamentals
12:45-13:45	Lunch
13:45-15:15	Agile 103 – Design Fundamentals
15:15-15:30	Break
15:30-17:00	Agile Security: Self Organizing Patterns and New Enabling Technology
17:00-17:15	Q&A

Part 1
Agile 101 – Architecture Fundamentals

Agile Systems and Processes:
Necessary and Sufficient Fundamental Architecture


An
introductory flash
underneath
insightful
system thinking
and design.
NEXT
6 Hours

Objective: System X-Ray Vision (seeing the bones)

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

Content

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

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

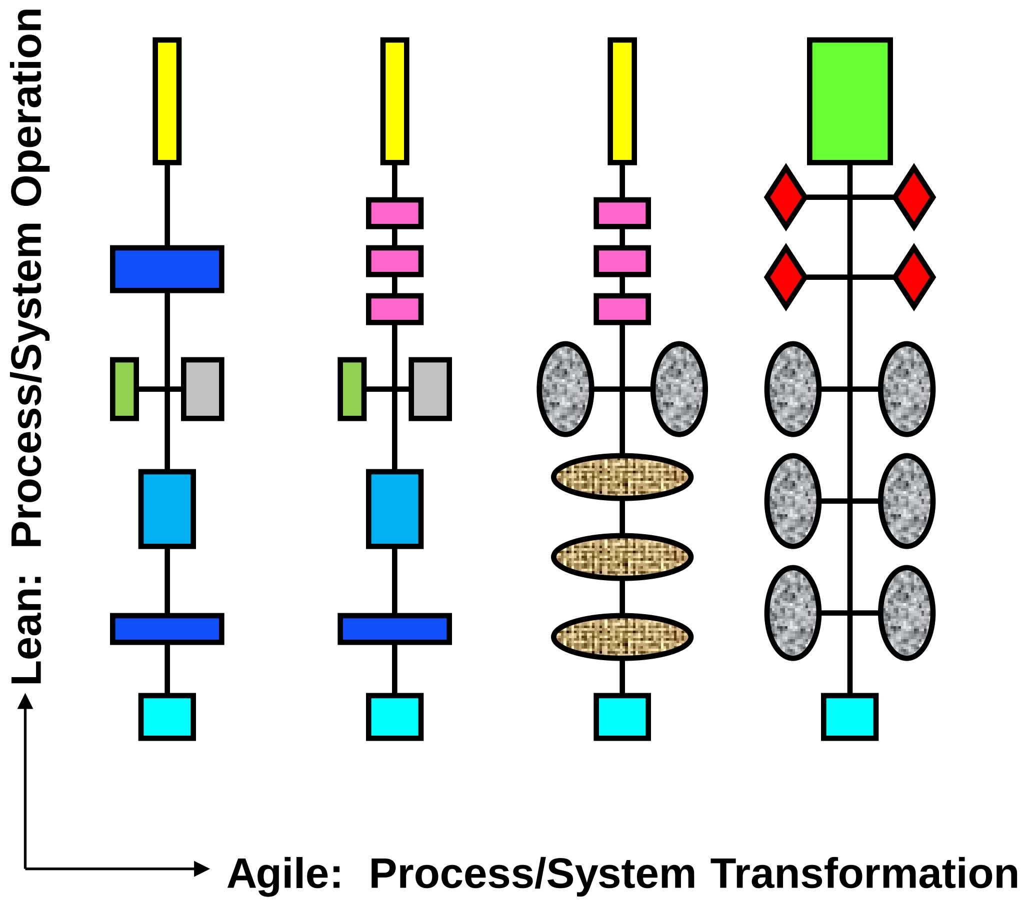
Lean & Agile: Orthogonal Focus

Agility deals with "design-for-transformation".

In a very general interpretation,

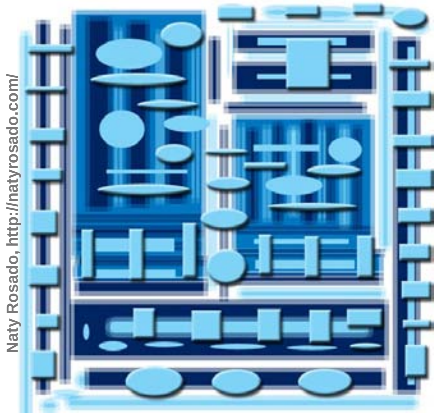
Lean values efficiency of operation and achieves this mainly through operational principles;

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.

Class 1 Agile Systems are <u>Reconfigurable</u>

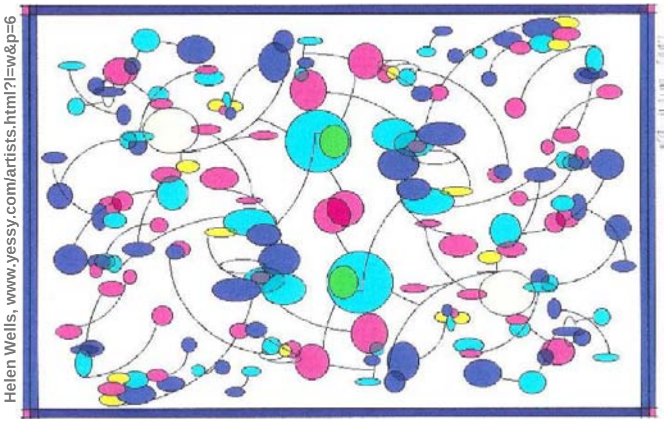


Useful Metaphors:

Plug-and-Play – Drag-and-Drop

(focus of this presentation)

Class 2 Agile Systems are <u>Reconfiguring</u>



Useful Metaphors:

Ecologies and Evolution

(not in this presentation)

Both employ the same fundamental agile architecture pattern

RAP* – 7 Thought-Guiding Frameworks

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)

Sustainable agility responsibilities (4 elements):
Module Readiness, System Assembly & Re-configuration
Module Evolution, Infrastructure Evolution

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

*RAP: Response Ability Principles

RAP – 7 Thought-Guiding Frameworks

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

fundamentals focus

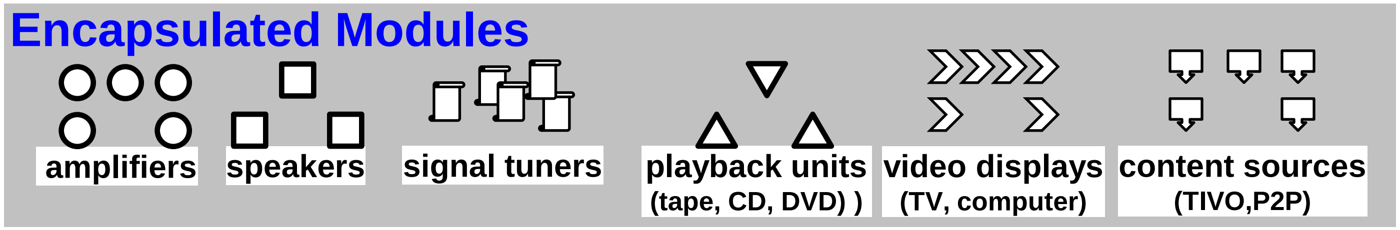
An overarching architectural philosophy (3 elements):
Reusable modules Reconfigurable in a Scalable architecture (RRS)

Agility-sustaining responsibilities (4 elements):
Module Readiness, System Re-configuration
Module Mix Evolution, Infrastructure Evolution

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

Case: Home Entertainment Technology Migration

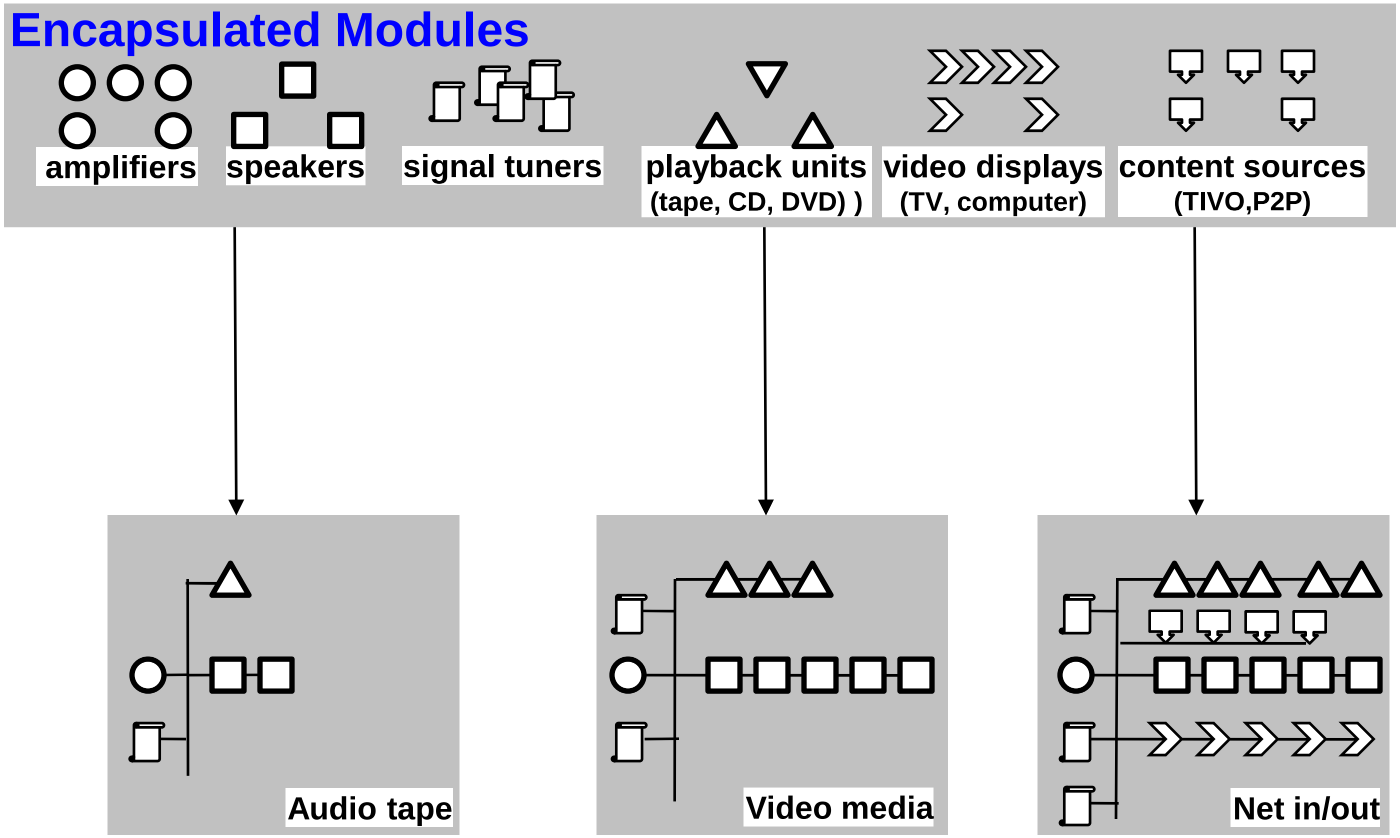
agile architecture pattern: drag-and-drop, plug-and-play



Drag-and-Drop
Reusable
Components

Case: Home Entertainment Technology Migration

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

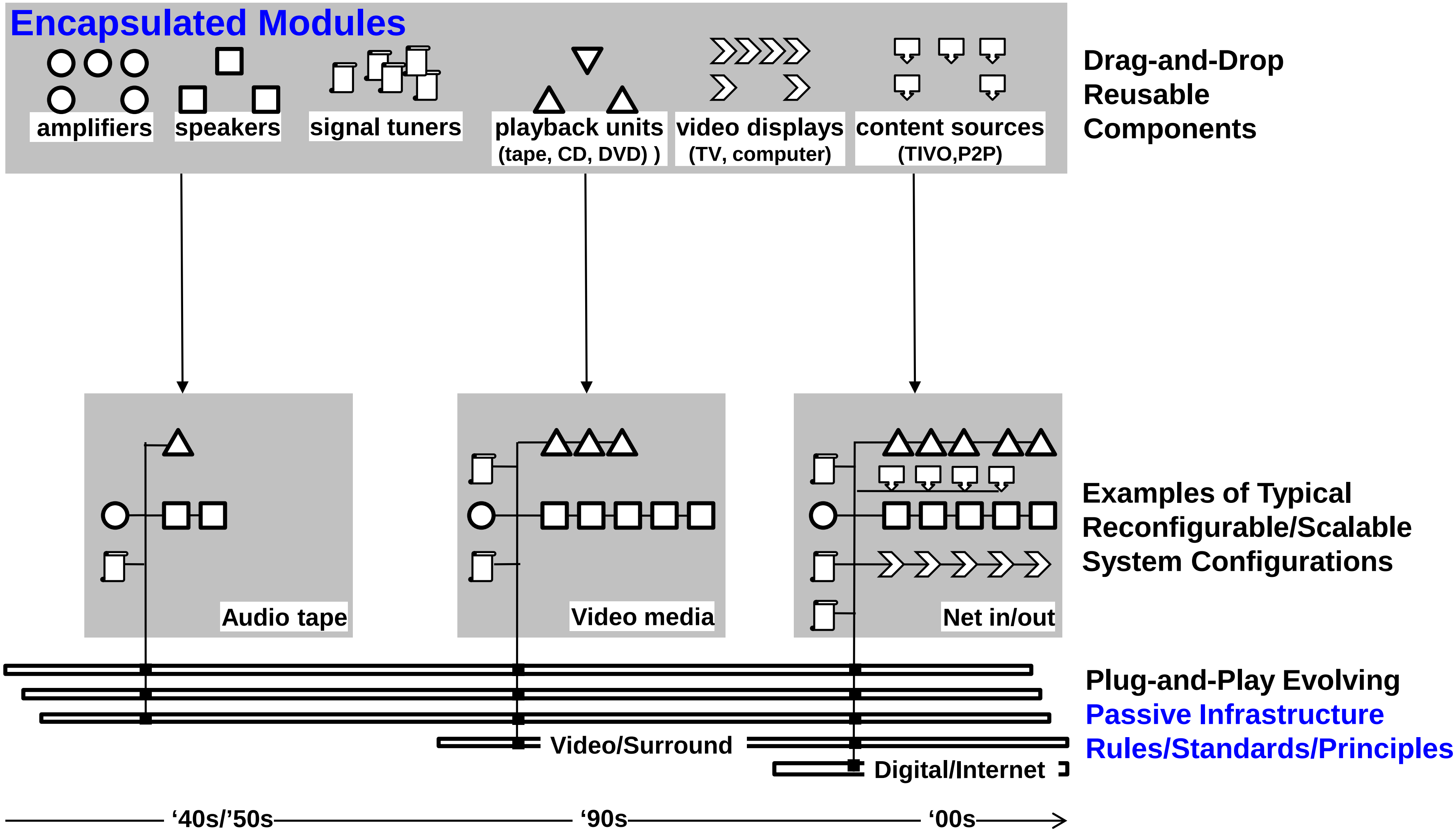


Drag-and-Drop
Reusable
Components

Examples of Typical
Reconfigurable/Scalable
System Configurations

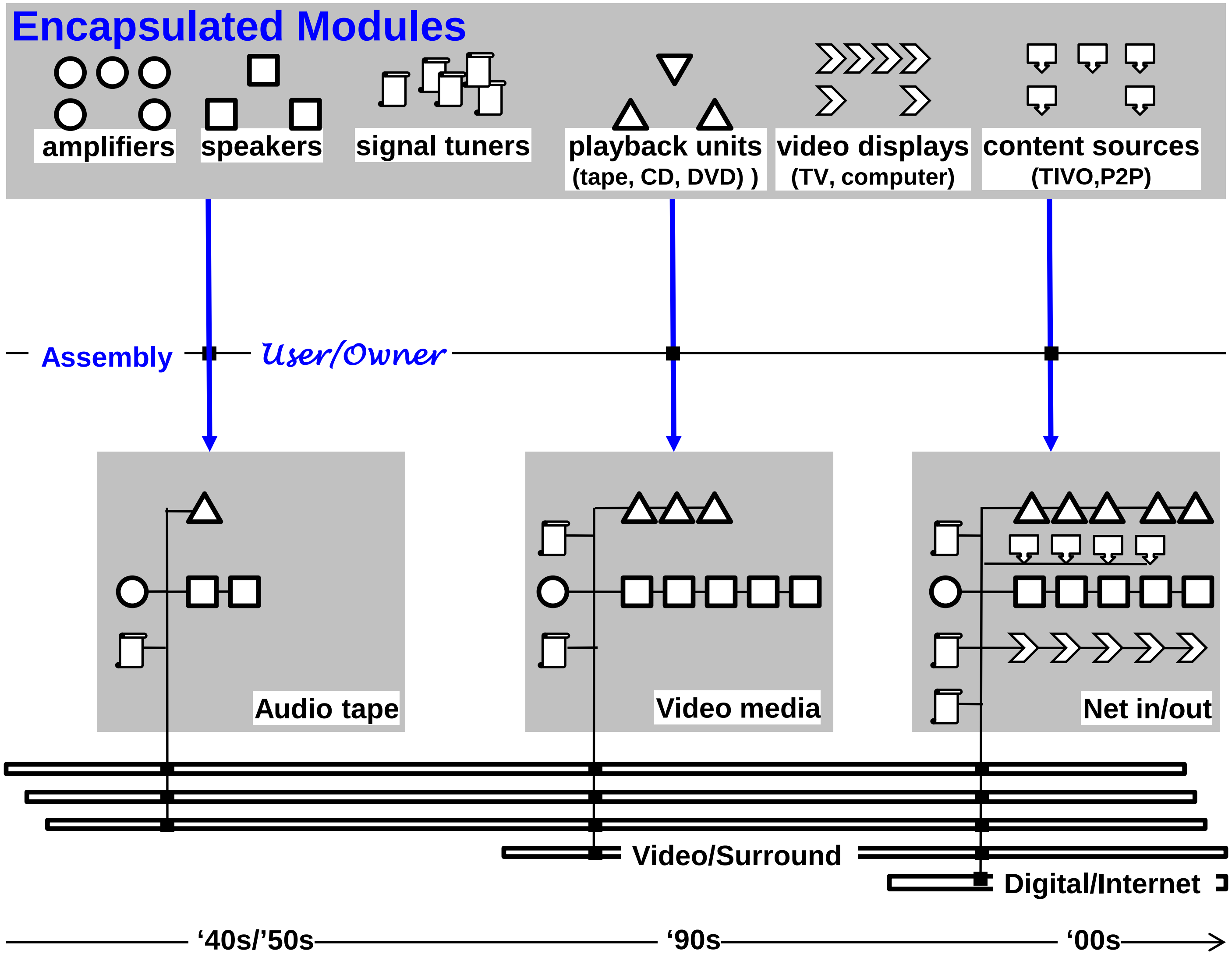
Case: Home Entertainment Technology Migration

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



Case: Home Entertainment Technology Migration

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



Drag-and-Drop
Reusable
Components

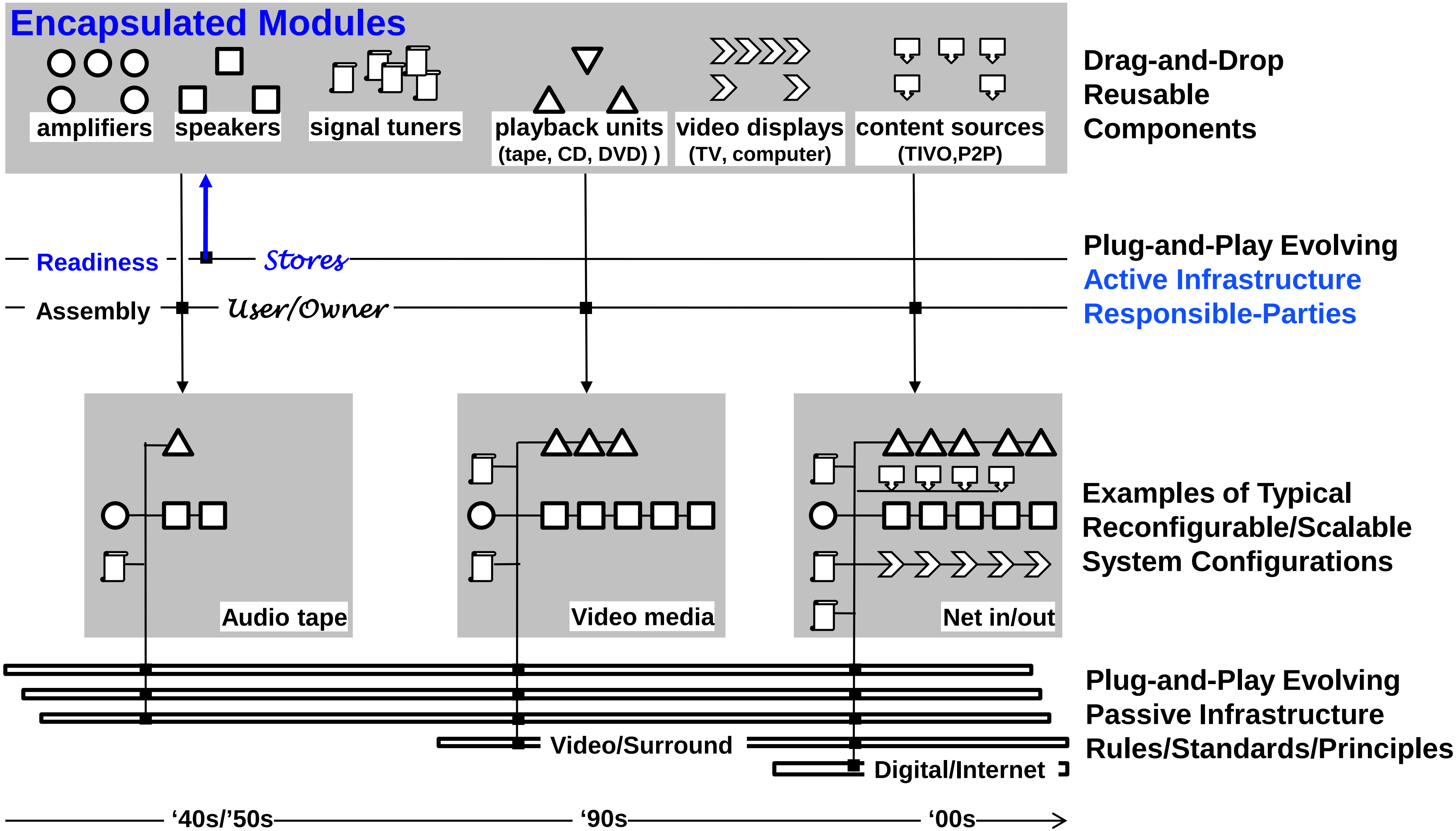
Plug-and-Play Evolving
Active Infrastructure
Responsible-Parties

Examples of Typical
Reconfigurable/Scalable
System Configurations

Plug-and-Play Evolving
Passive Infrastructure
Rules/Standards/Principles

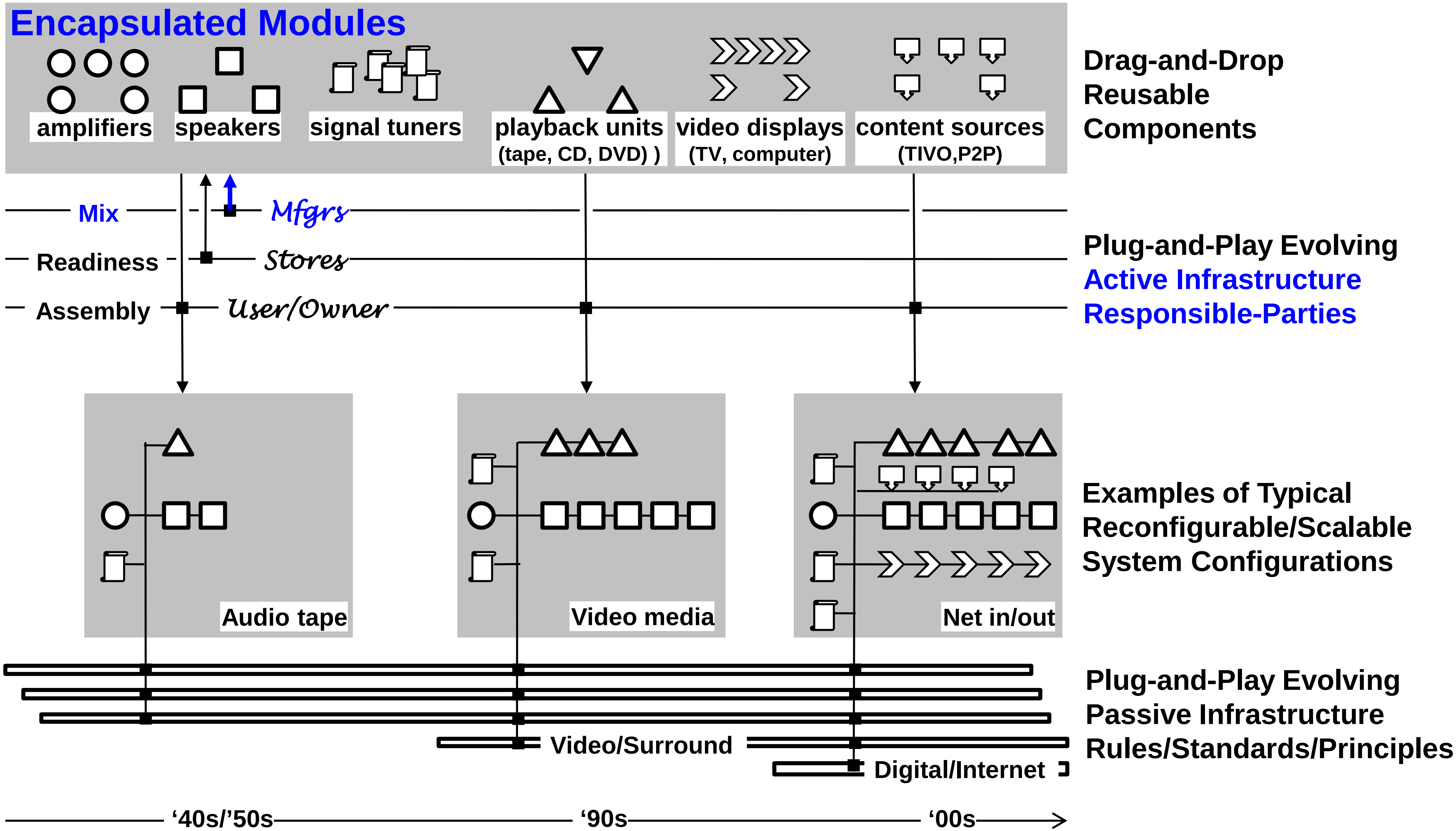
Case: Home Entertainment Technology Migration

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



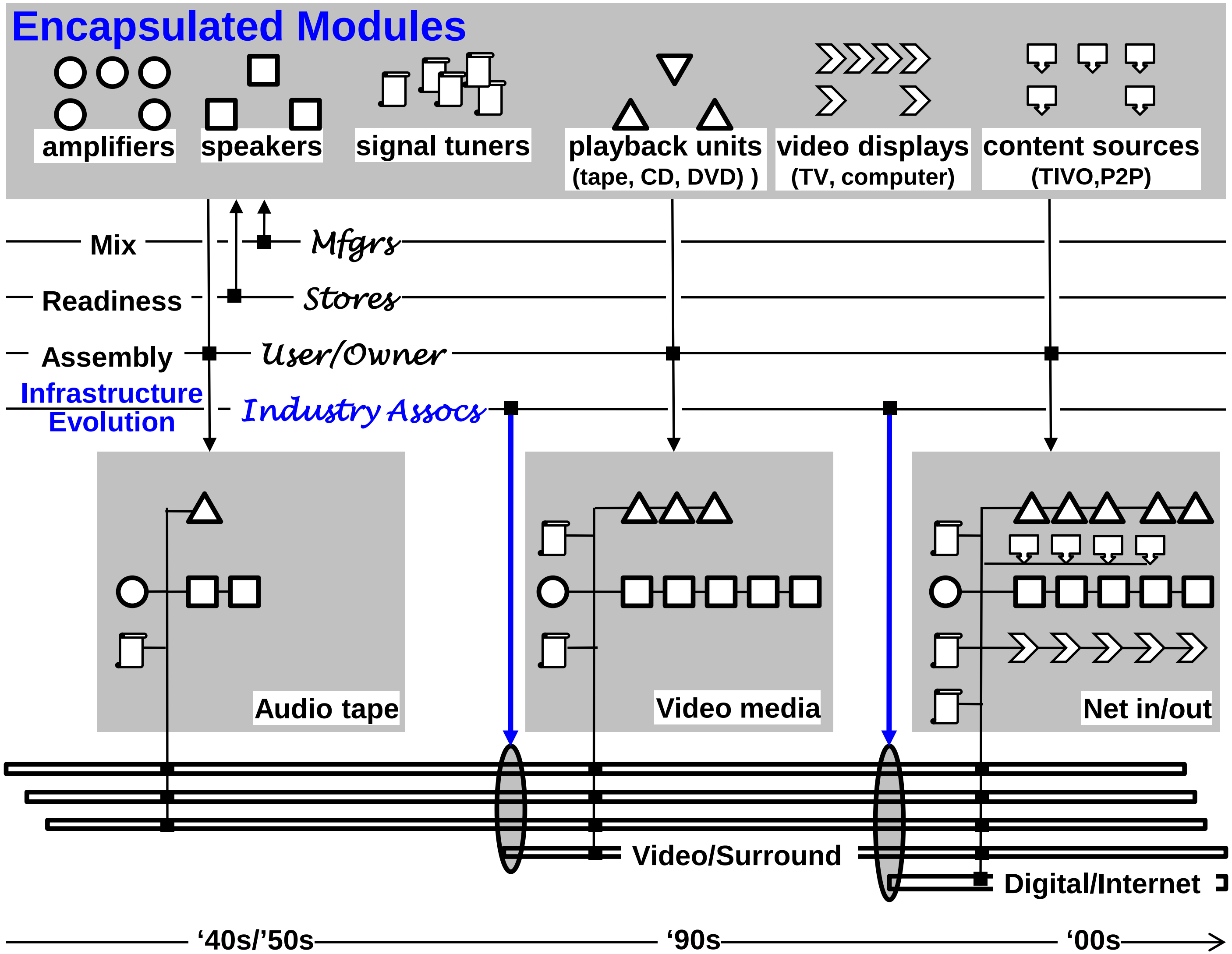
Case: Home Entertainment Technology Migration

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



Case: Home Entertainment Technology Migration

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



Drag-and-Drop
Reusable
Components

Plug-and-Play Evolving
Active Infrastructure
Responsible-Parties

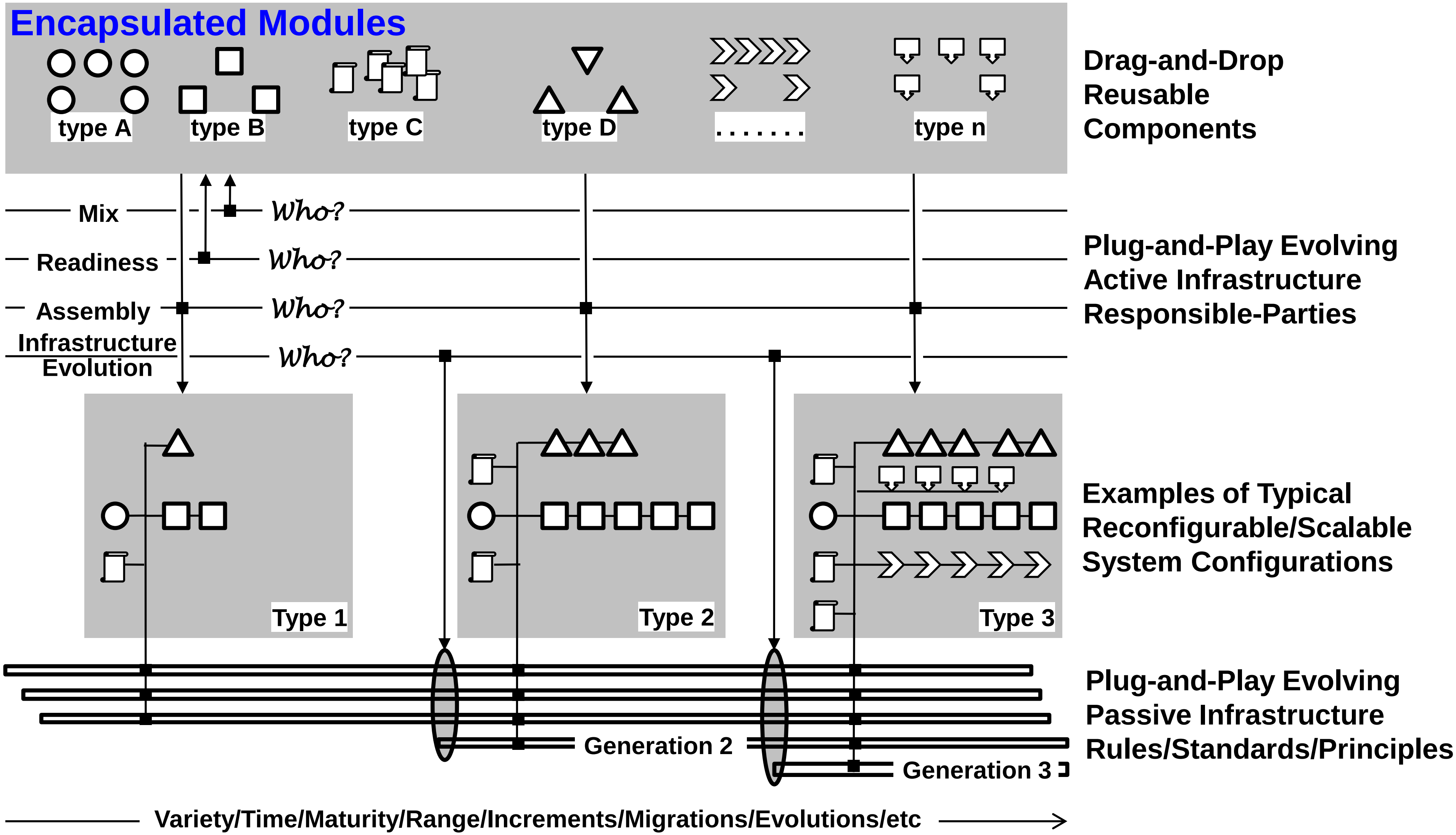
Examples of Typical
Reconfigurable/Scalable
System Configurations

Plug-and-Play Evolving
Passive Infrastructure
Rules/Standards/Principles

Fundamental Concept

Reusable modules Reconfigurable in a Scalable architecture (RRS)

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



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.**

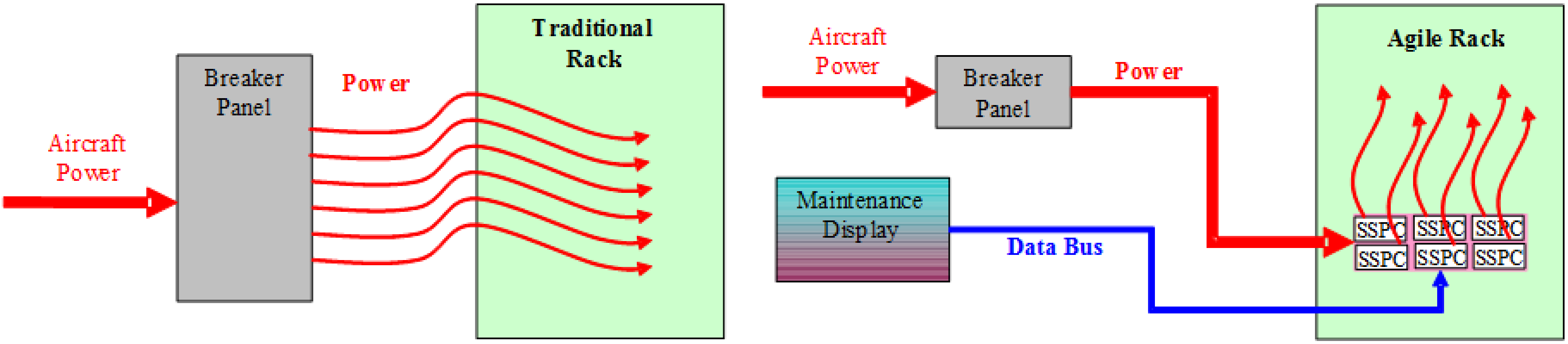
***QRC: Quick Reaction Capability**

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.

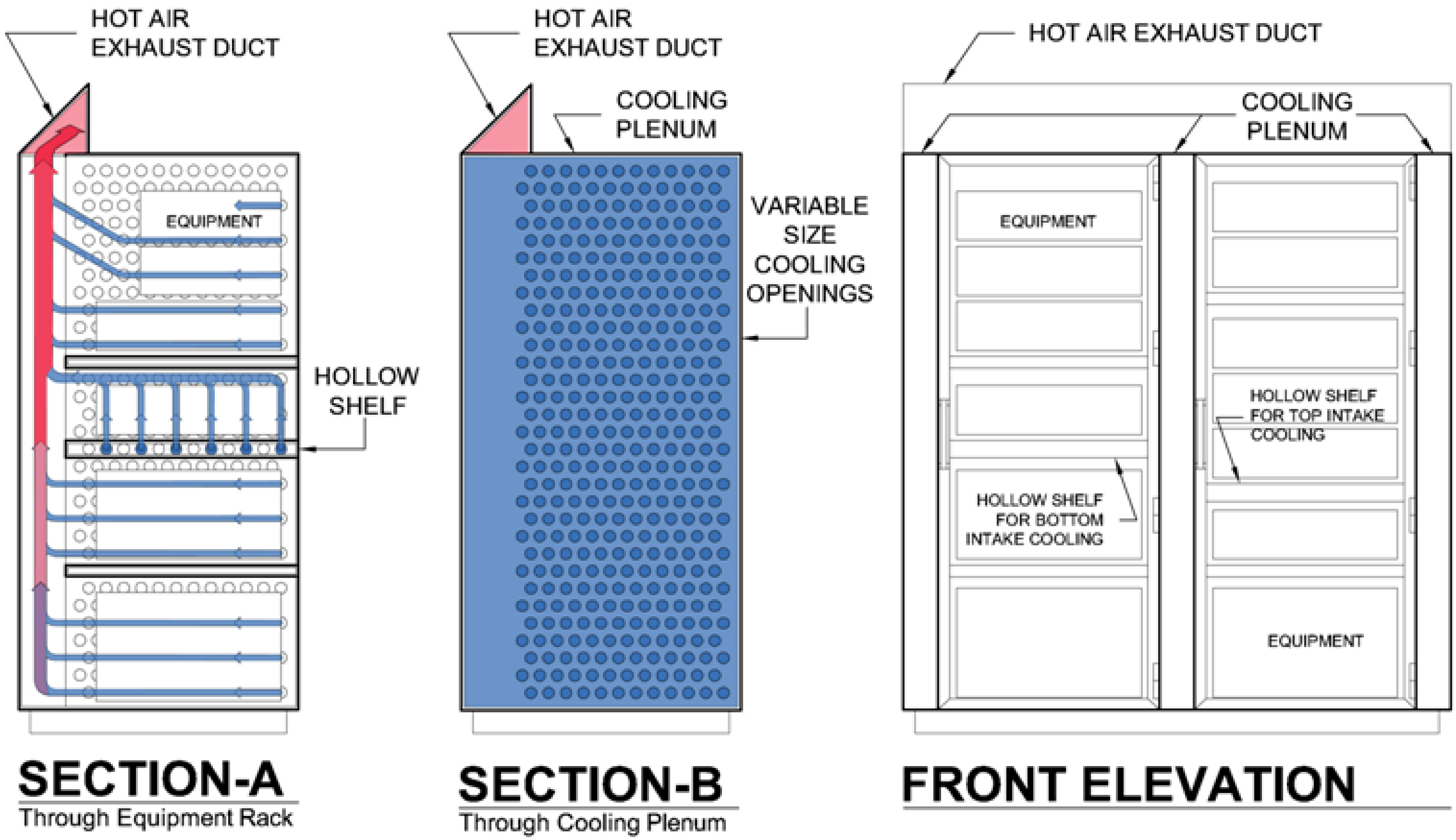
To reduce the number of interfaces, decrease wire routing effort, and allow rack modularity, the power distribution can be moved from the aircraft to within the rack itself.

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.

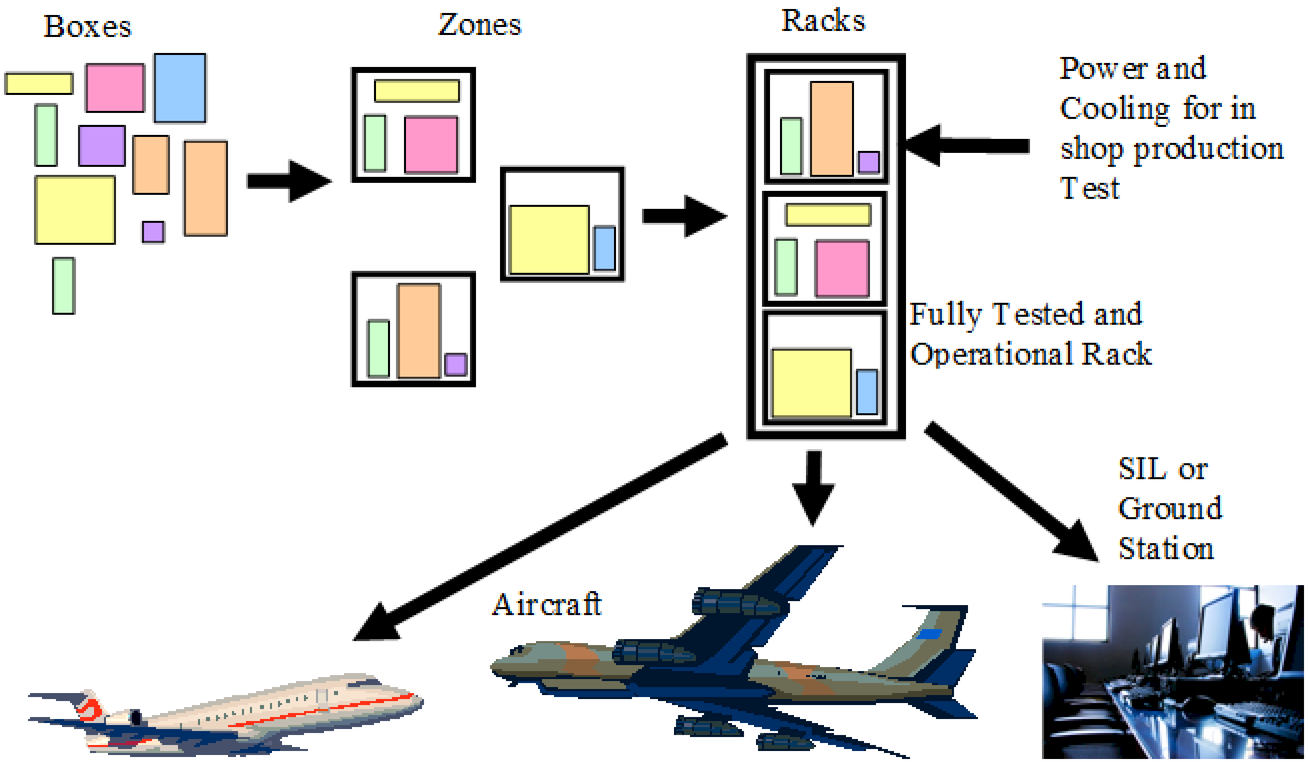
Example: Modular Rack Cooling



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.

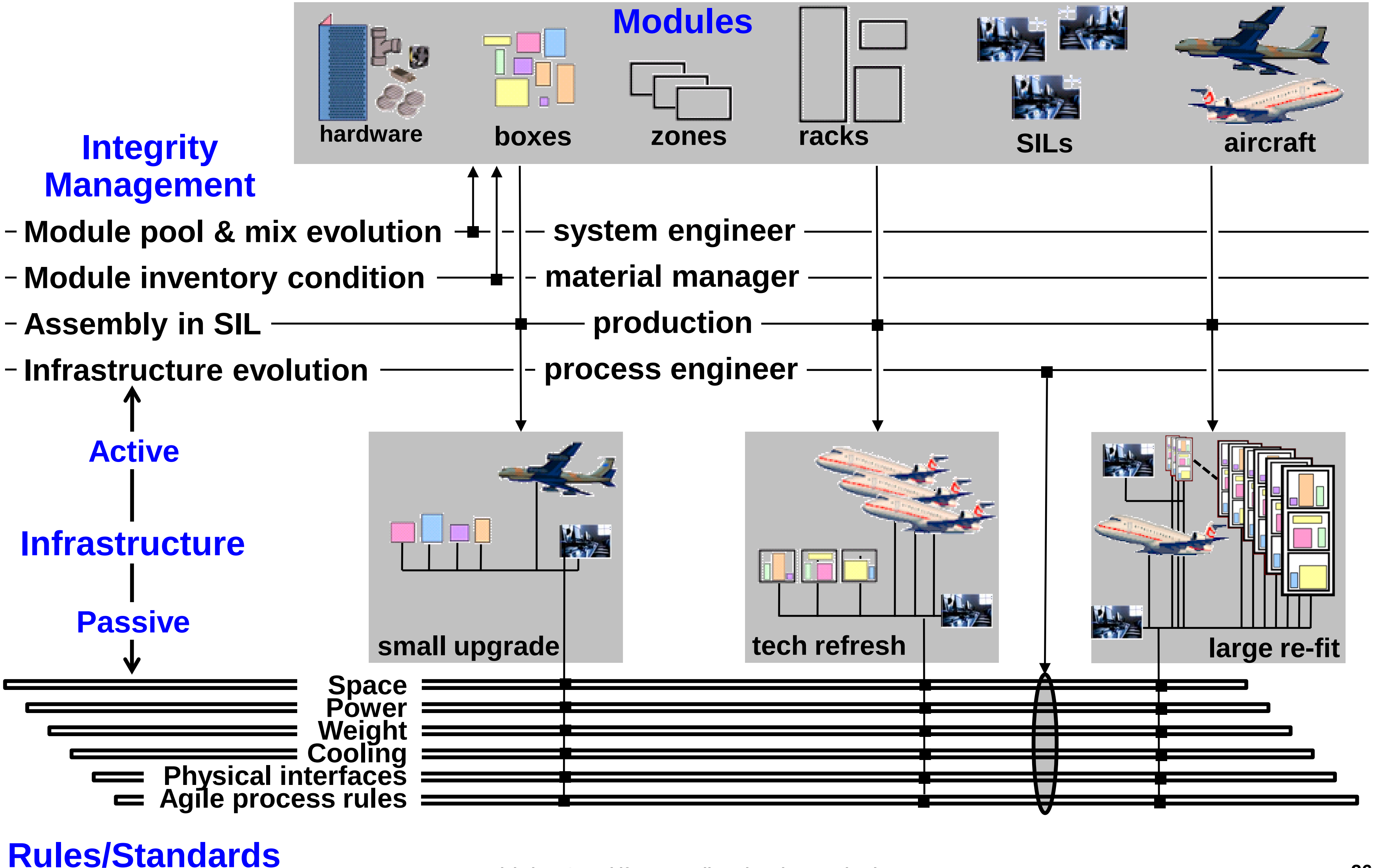
Aircraft installation is a simple relocation of pluggable modules.

Minimizes aircraft downtime and eliminates custom installation work.

Parameter	Nature of Standard
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.

***SIL: System Integration Lab**

QRC Aircraft Installation – Agile Architecture



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

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?

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.

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?

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?

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 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:

Rule 1: Allow scheduled activities to remain in the master schedule unless positive knowledge exists that the activity should not or cannot be executed when scheduled.

Rule 2: 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.

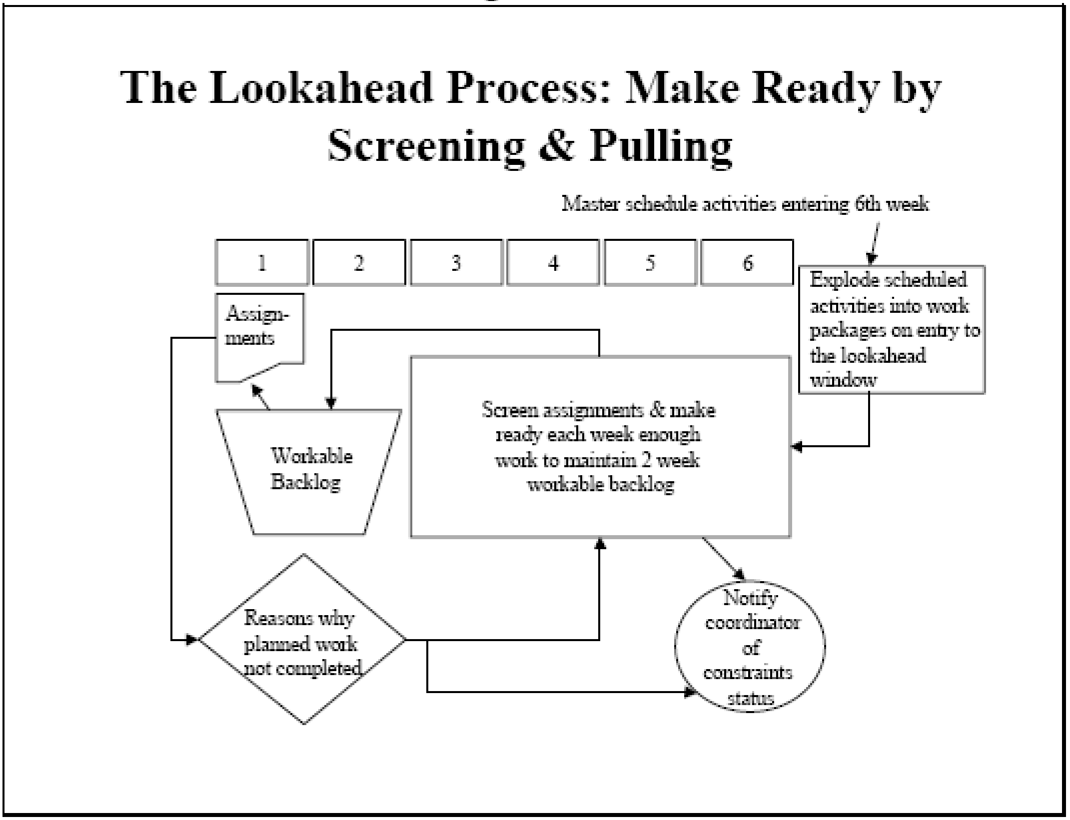
Rule 3: Allow scheduled activities to be released for selection into weekly work plans only if all constraints have been removed; i.e., only if the activity has in fact been made ready.

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.

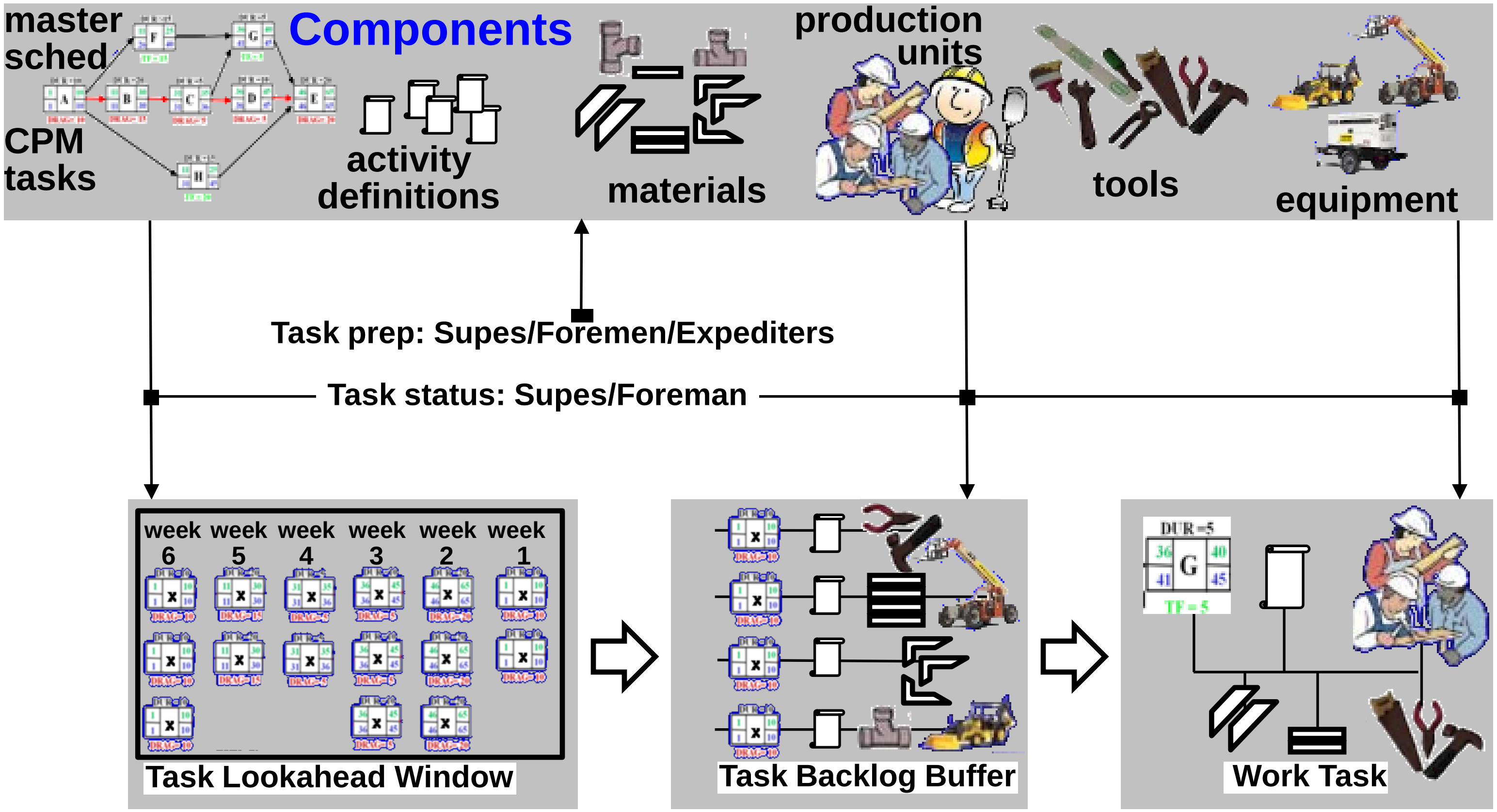
[13] Deliberately building inventories, inventories of ready work in this case, may seem contradictory to the goals of just-in-time. To clarify, inventories of all sort are to be minimized, but as long as there is variability in the flow of materials and information, buffers will be needed to absorb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

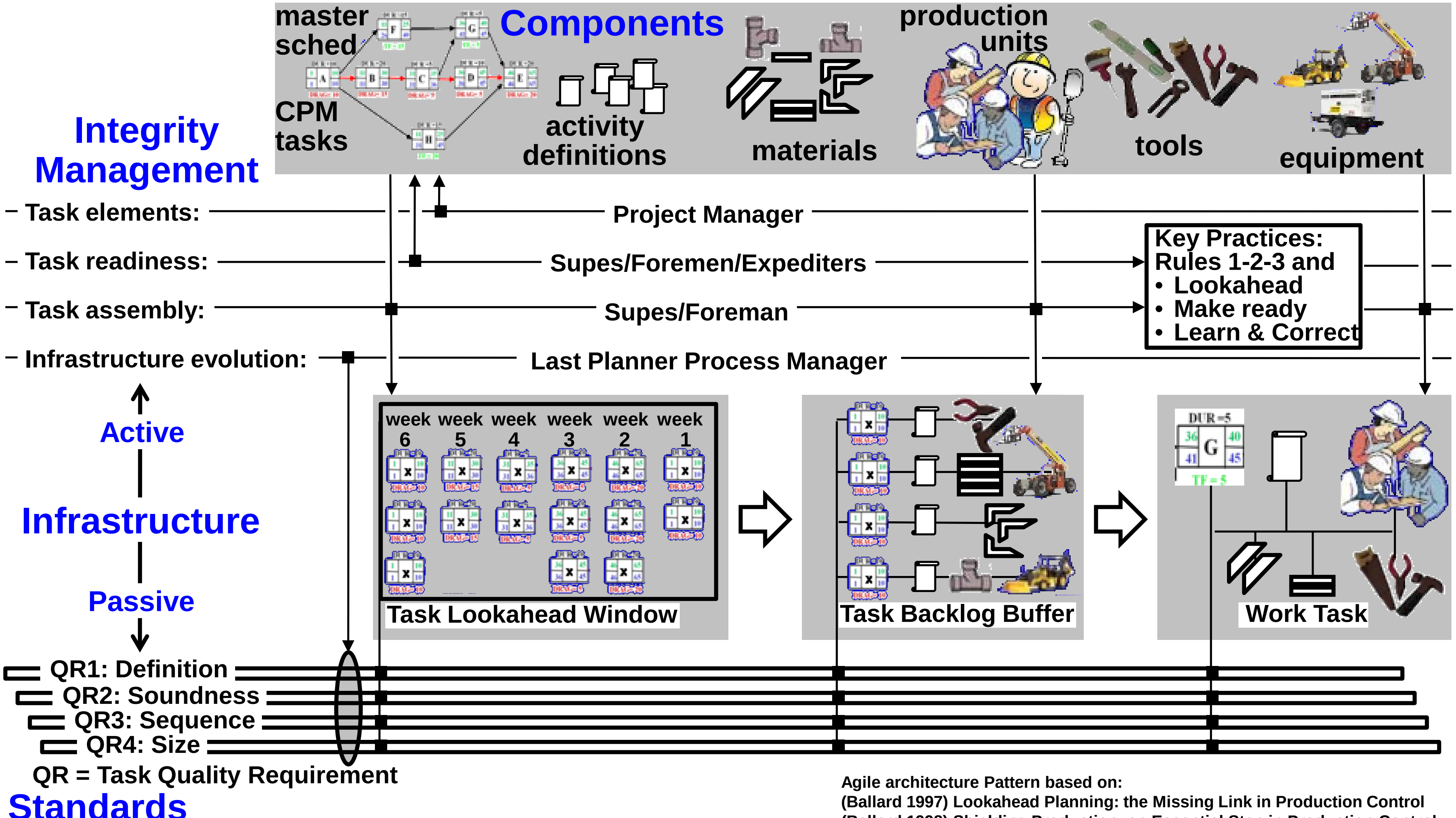
Weekly work tasks are drawn from readiness backlog, keeping crews fully employed.

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.



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

Reusable, Reconfigurable, Scalable (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 Minimal Standards (Infrastructure)

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

Encapsulated Modules	Reusable	Scalable	Evolving Minimal Standards
Plug Compatibility (Facilitated Interfacing			Redundancy and Diversity
Facilitated Reuse			Elastic Capacity
Reconfigurable			
Flat 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 <u>custom</u> electronic "partner" relationships
is enabled rather than inhibited.

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

<u>All</u> 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:

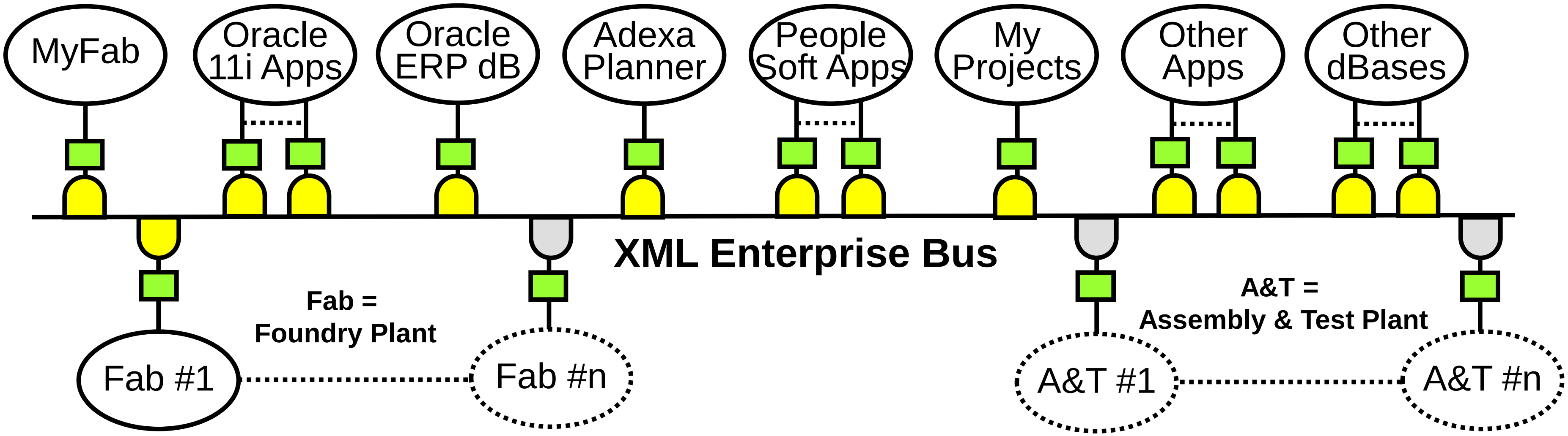
- **Mandatory Response Situation Analysis (agility-tool)**

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 Design



- = Bus 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 Process – 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

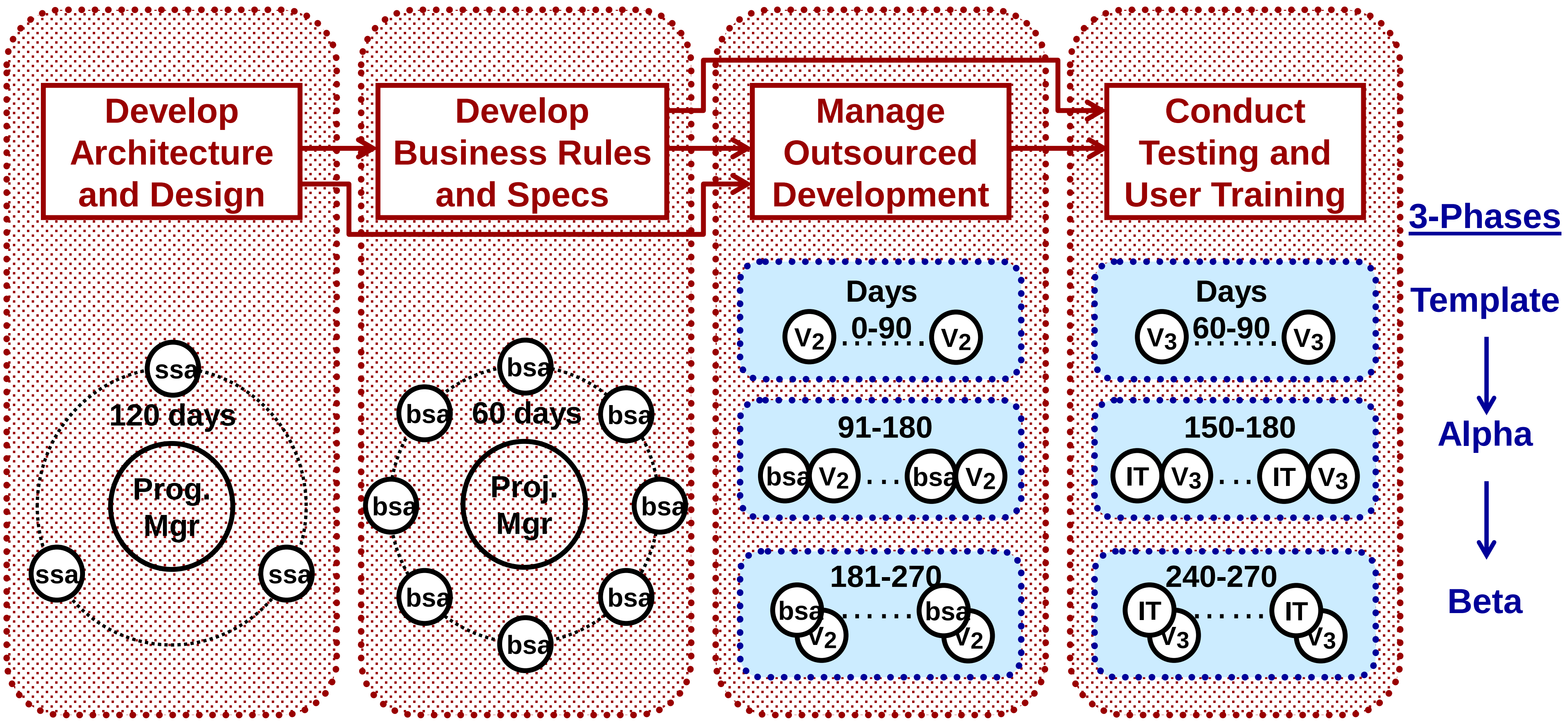
P2: BSA-developed business process rules
Vendors + BSAs configure the applications

P3: Refined 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)

Encapsulated Development Process



- Designed to Accommodate Requirements Evolution -

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

Effective Predictability

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

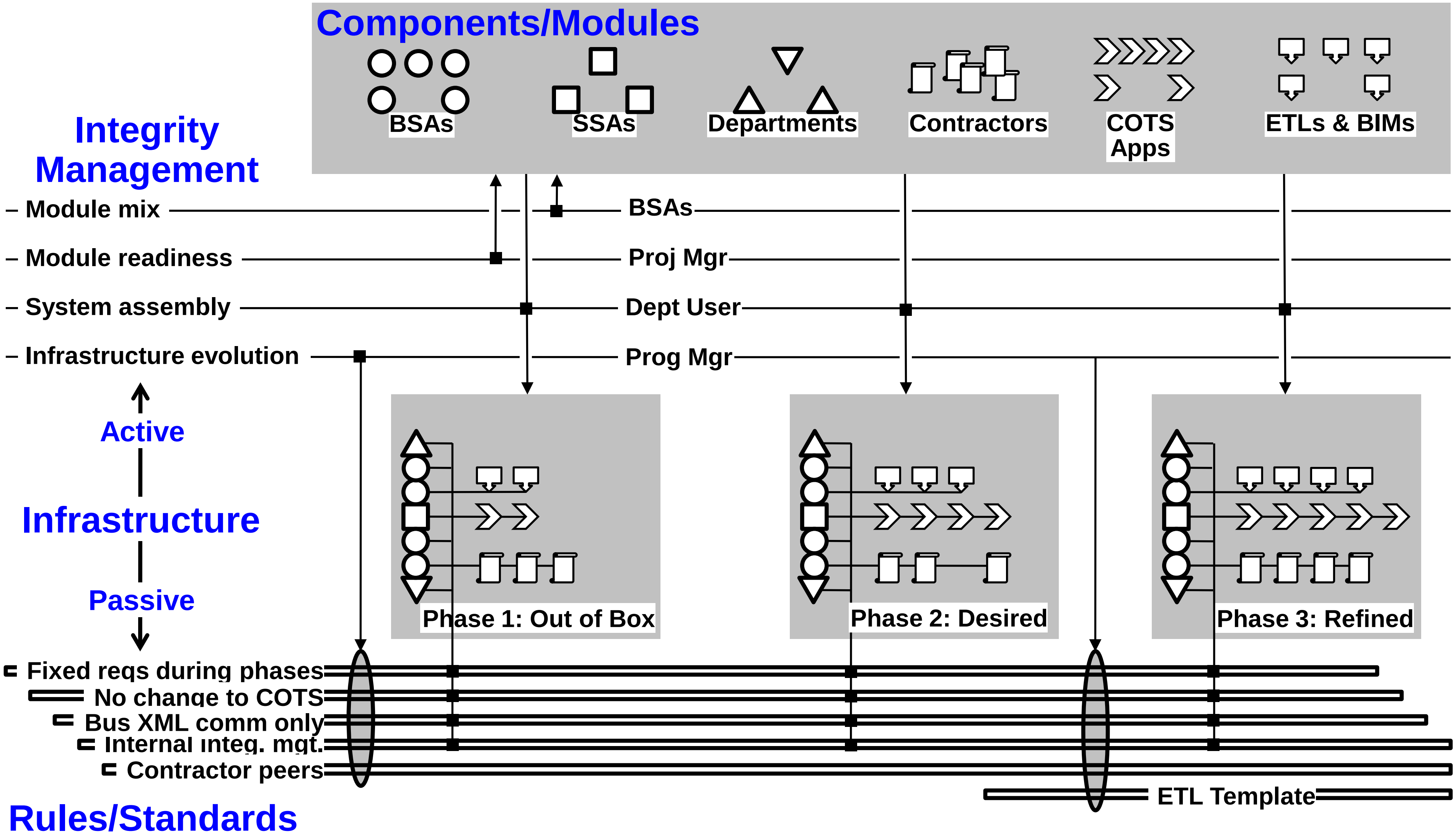
Wish	Typical Imp	Actual Imp
ERP in 12 mos total	24-36 mos	12[1,2]
75% of license budget	200-300%	75%
$10 Million (5 + 5)	$15-25 Million	$9 Million
HRM in 6 mos	12-18 mos	5 mos

HOW??

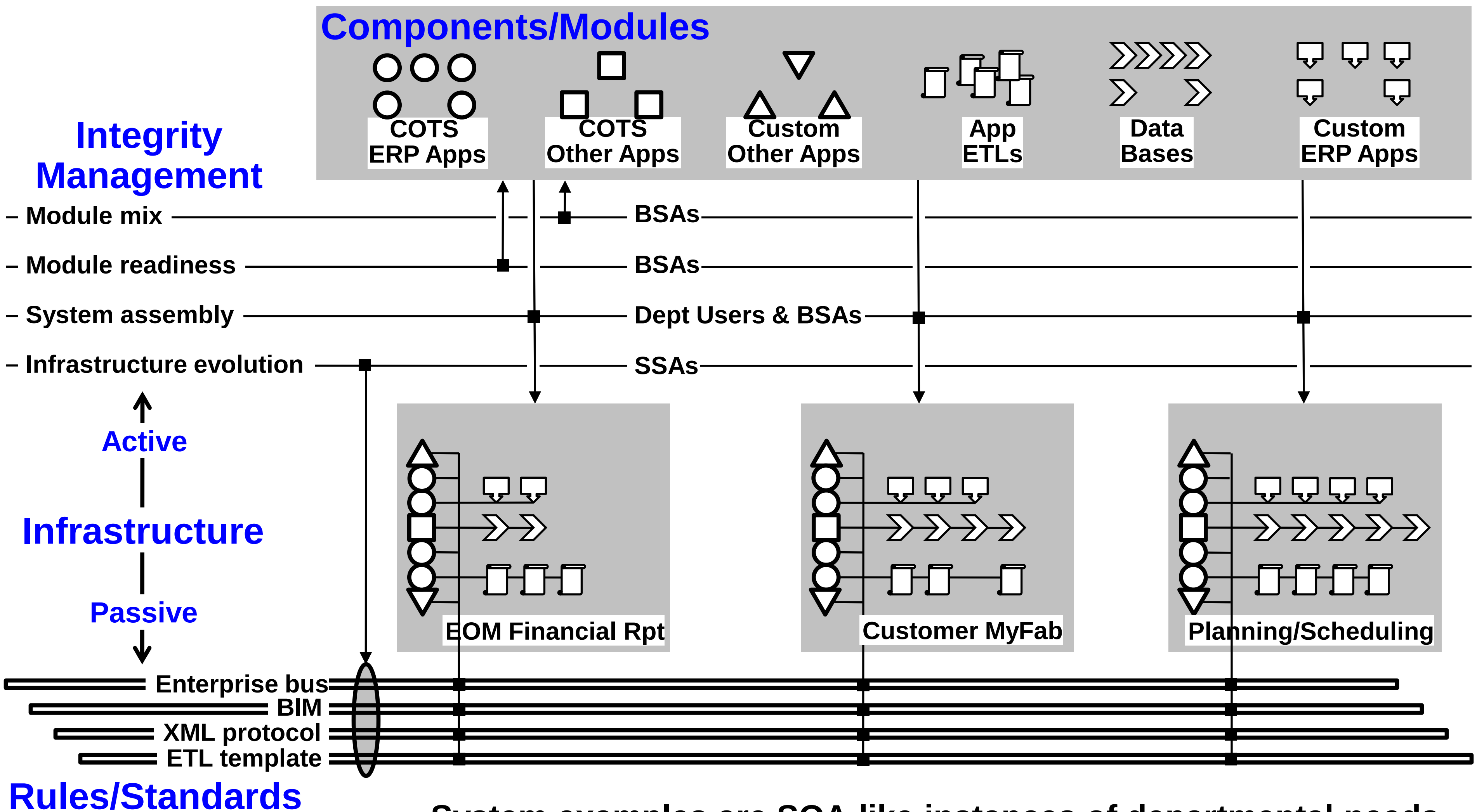
- Principle-based installation/integration methodology and management
- Adherence to methodology (ie, effective management)
- BSAs utilizing MBW tool to develop and capture business processes
- BSAs taking responsibility for integrating ERP with users
- Bus architecture connecting ERP with HRM
- Experienced 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

"Classic" Scrum

Ken Schwaber, Jeff Sutherland. 2011. 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

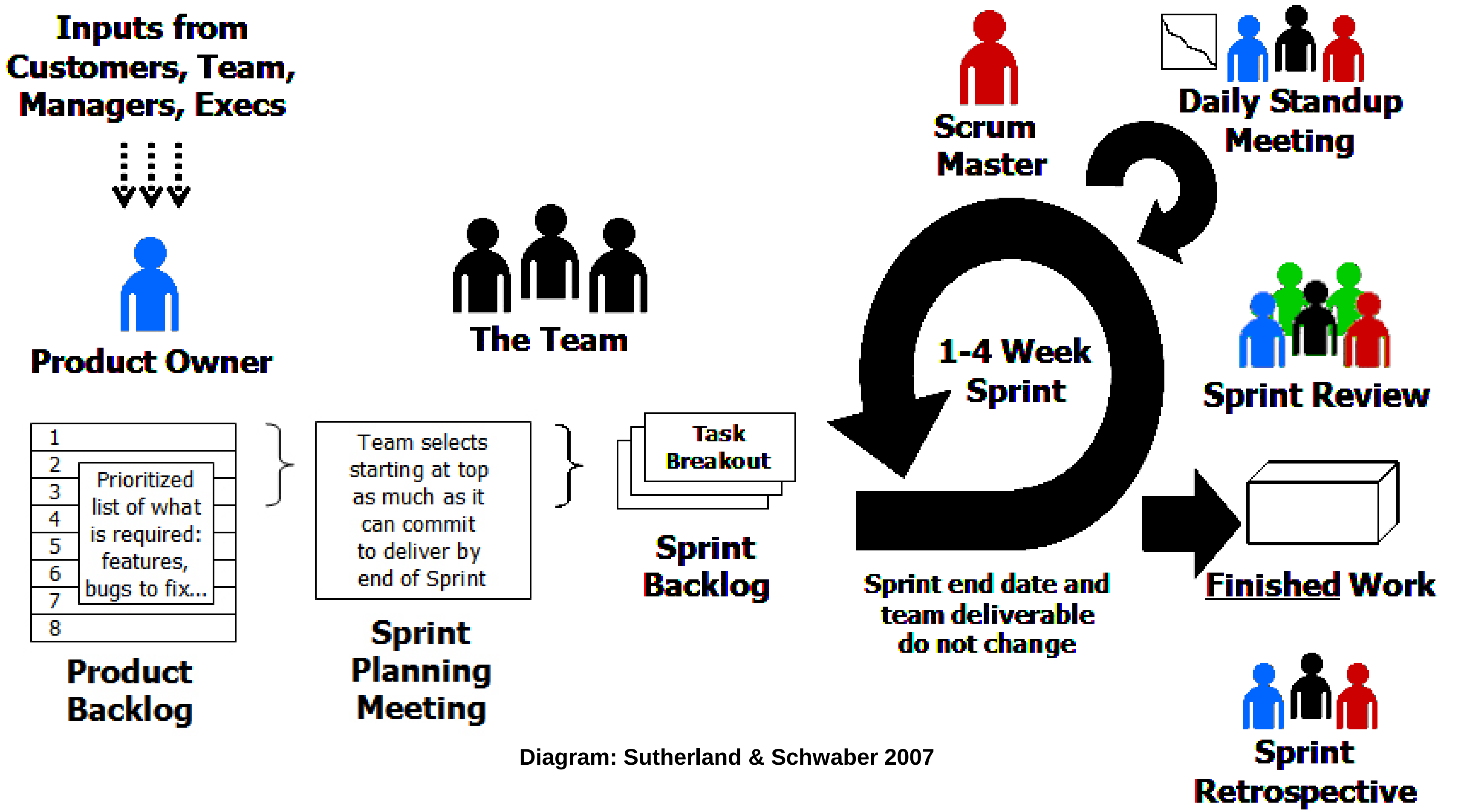


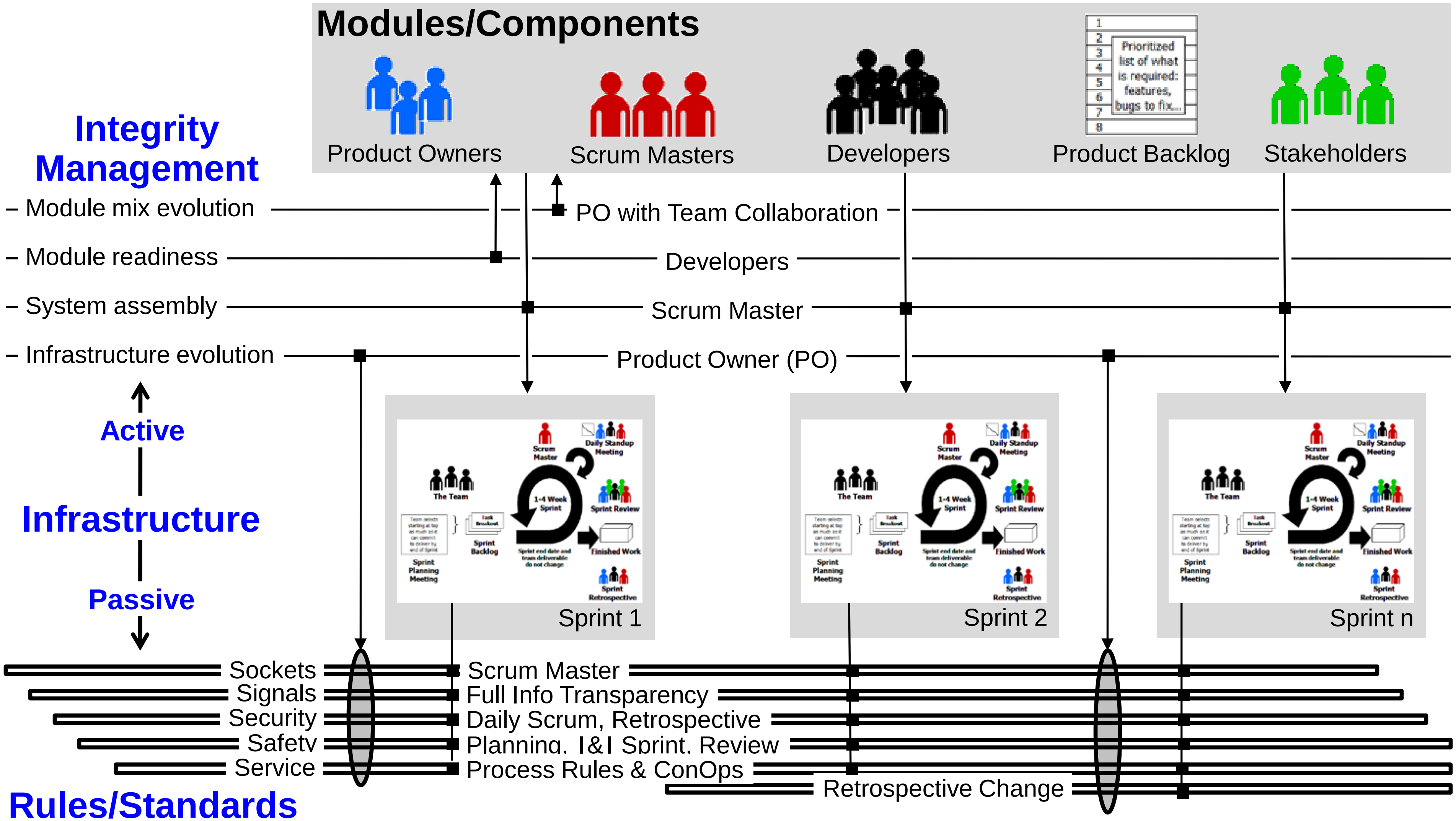
Diagram: Sutherland & Schwaber 2007

"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)

Scrum has an Agile Architecture Pattern (AAP) *Structure*

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



… because the RSA is different for an agile systems-engineering process, and the Scrum AAP *strategy* is inadequate for systems engineering

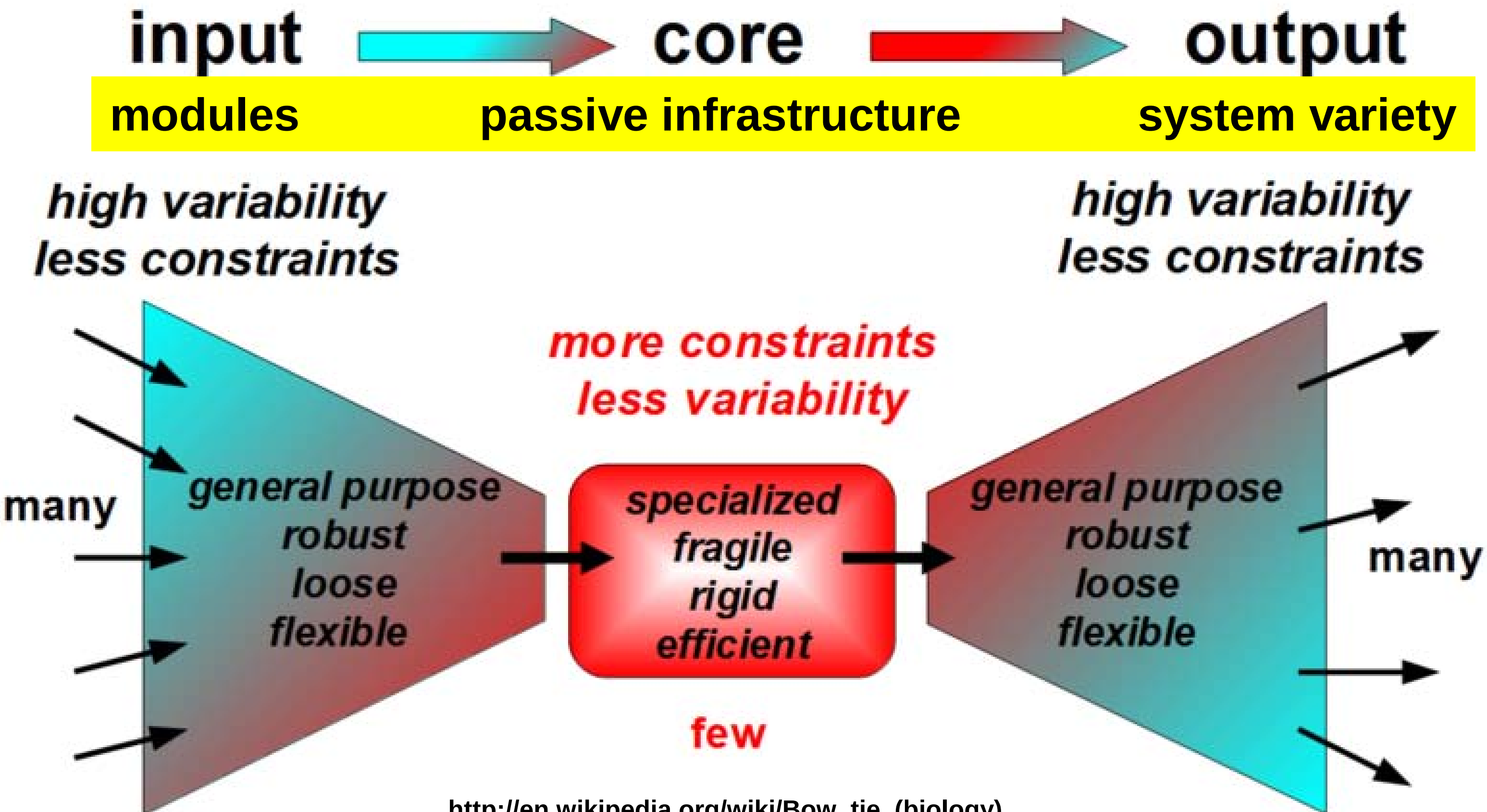
Echoes from
Biology and Complex Systems

Bow Tie Process Echo from Science

Our work was based on observation of many real systems that exhibited agile characteristics in a large variety of enterprise domains.

Since then, we have discovered a body of science behind the architecture, with work carried out by collaborators:

- **John Doyle, John G Braun Professor of Control & Dynamical Systems, Electrical Engineering, and BioEngineering at Caltech**
- **Jean Carlson, Department of Physics, UC Santa Barbara**
- **Marie Csete, (now) Staff Physician at UC San Diego Anesthesiology**

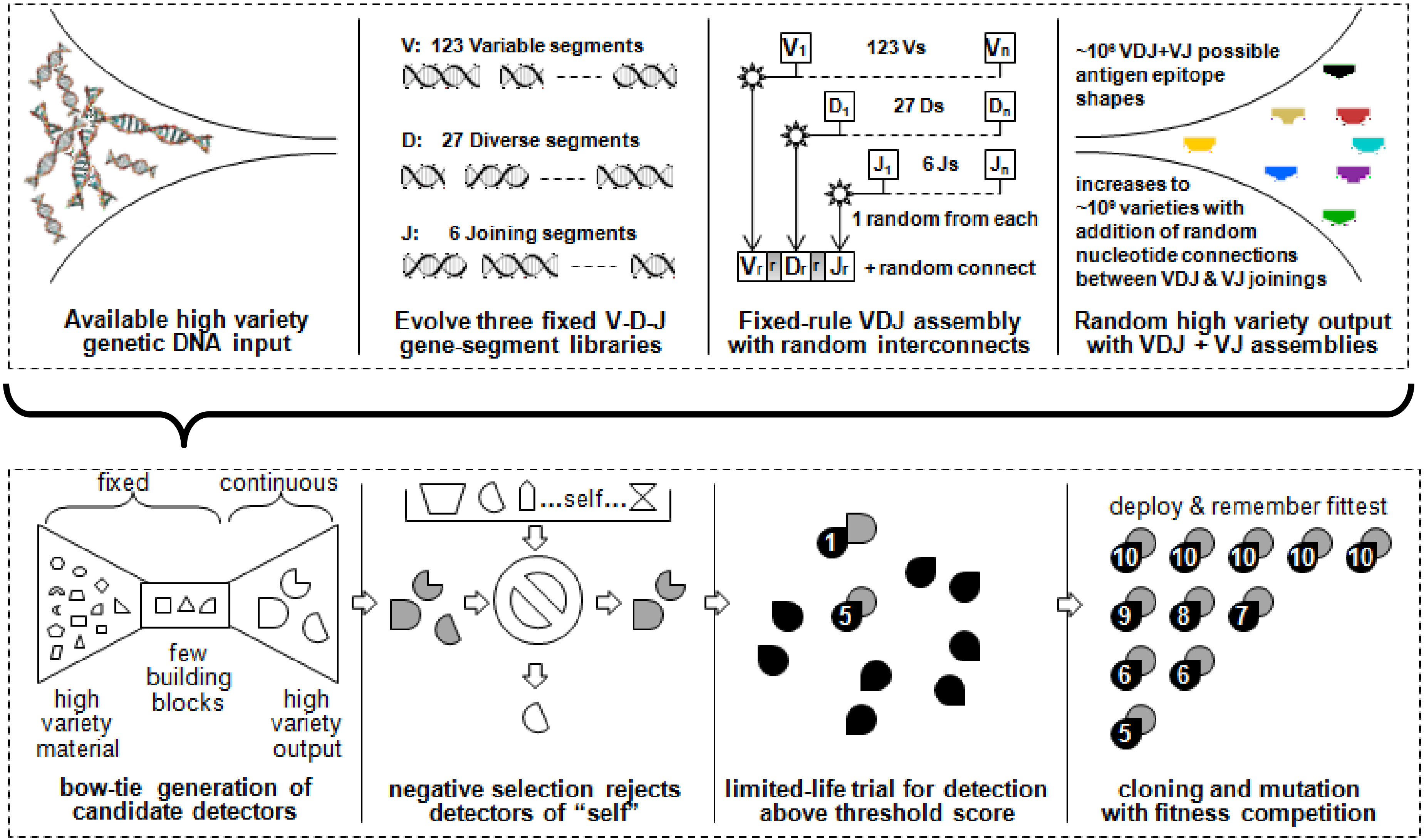


http://en.wikipedia.org/wiki/Bow_tie_(biology)

Bow Tie Pattern in the Immune System

Millions of random infection detectors generated continuously by fixed rules and modules in the "knot"

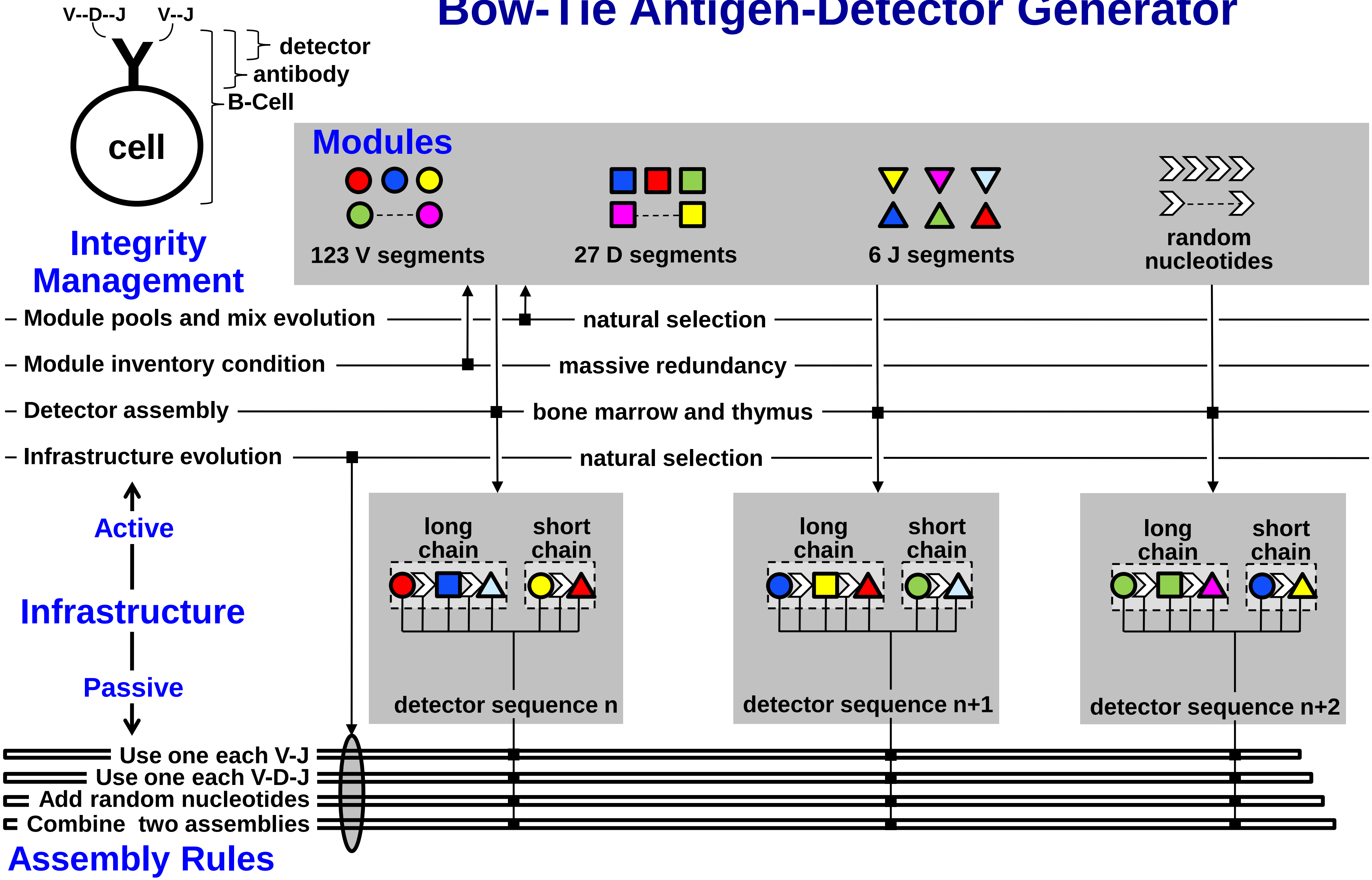
(Dove 2010). Pattern Qualifications and Examples of Next-Generation Agile System-Security Strategies.



Speculative generation and mutation of detectors recognizes new attacks like a biological immune system

(Dove 2011a) Patterns of Self-Organizing Agile Security for Resilient Network Situational Awareness and Sensemaking

Adaptable Immune System
Bow-Tie Antigen-Detector Generator



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.

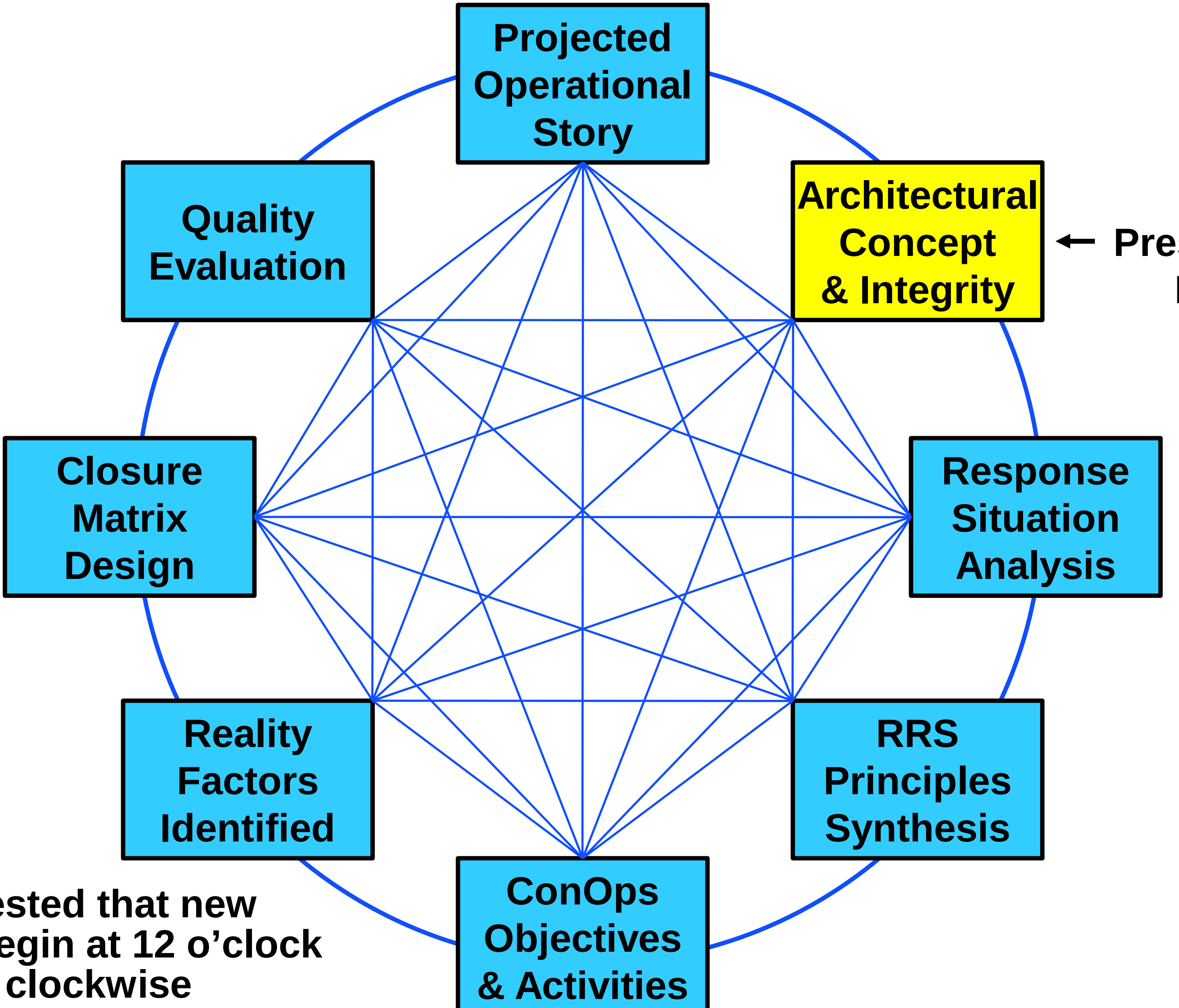
Protocols also facilitate the addition of new protocols and organization into collections of mutually supportive protocol suites.

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.

Wrapping it Up

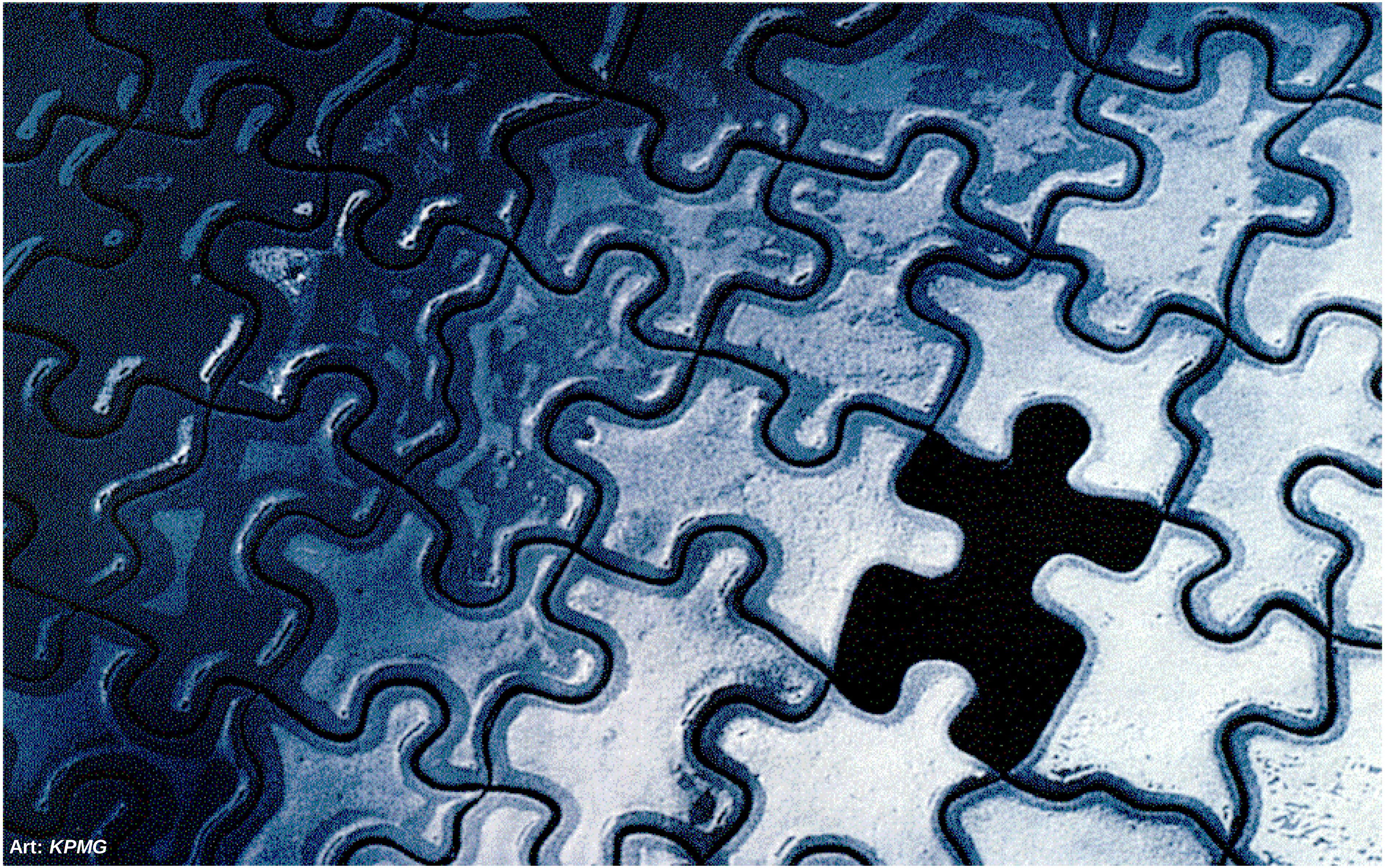
Eight principle tools to employ when designing or analyzing a system for agility



This Presentation Focus

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

Modular – But Not Agile



Agile System and Project Management by Design

<u>Risk and Uncertainty Management Through:</u>

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

References and Supportive Readings

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

(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

(Ballard 2000) Herman Ballard. The Last Planner System of Production Control. PhD Thesis at Birmingham University. www.leanconstruction.org/pdf/ballard2000-dissertation.pdf

(Csete 2002) Marie E. Csete and John C. Doyle. Reverse Engineering of Biological Complexity. Vol 295 SCIENCE, 1 March. www.cds.caltech.edu/~doyle2/wiki/images/7/7a/Science1664-2002.pdf

(Csete 2004) Marie Csete and John Doyle. Bow Ties, Metabolism and Disease. TRENDS in Biotechnology 22(9), September. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.173.3019&rep=rep1&type=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. http://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. http://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. http://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

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

(Schumacher 2011) Col. Ludwig J. Schumacher. Dual Status Command for No Notice Events Integrating Military Response to Domestic Disasters. Homeland Security Affairs, Vol 7, Feb. www.hsaj.org/?download&mode=dl&h&w&drm=resources/volume7/issue1/pdfs/&f=7.1.4.pdf

Part 2
Agile 102 – Requirements Fundamentals

Agile Systems and Processes –
Driving Architecture with
ConOps & Response Situation Analysis

Recapping Agile 101…

**A system's "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.**

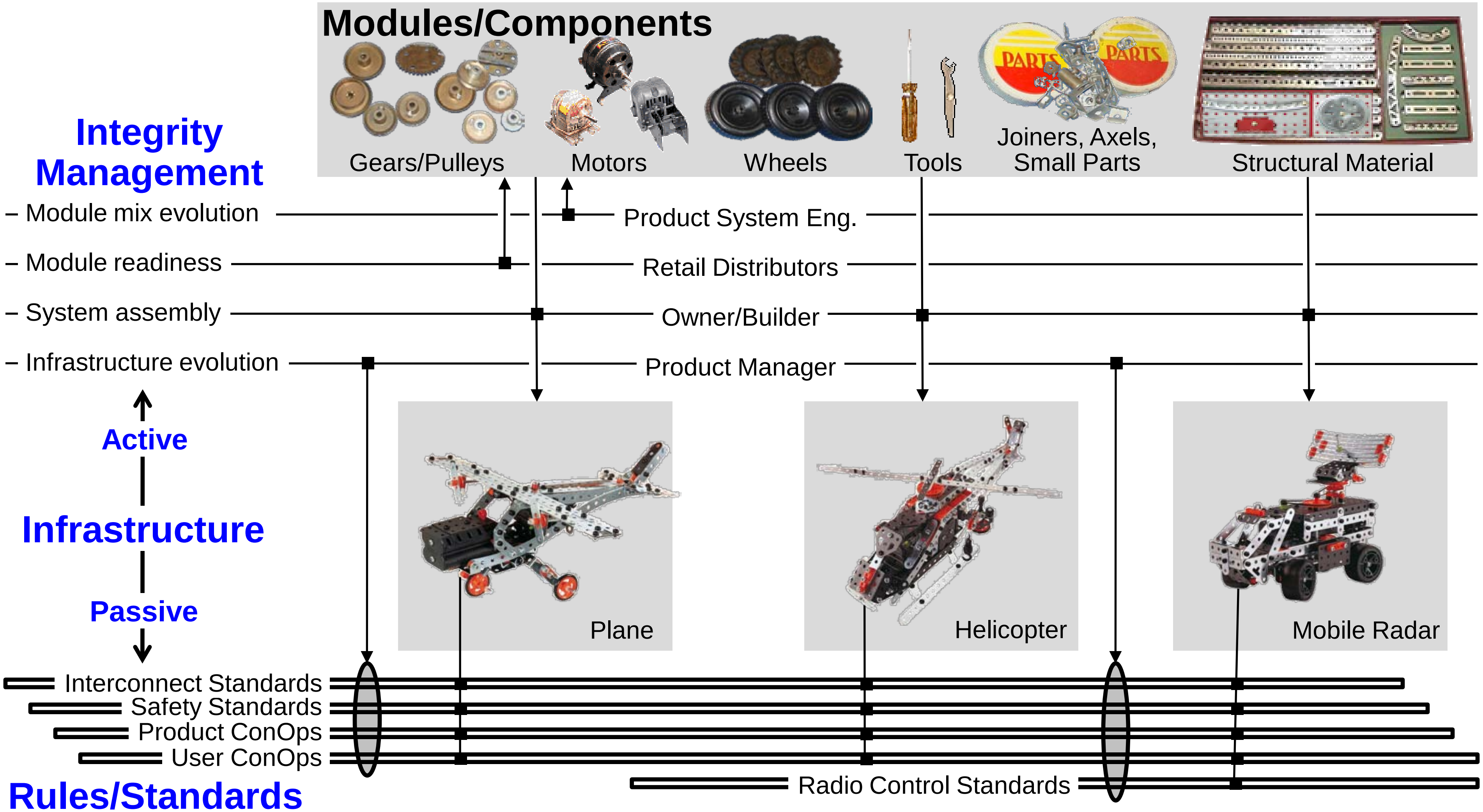
http://awespendo.us/animemangacomics/kermit-at-the-doctor/

Here’s a Box of Bones

ERECTOR=MECCANO

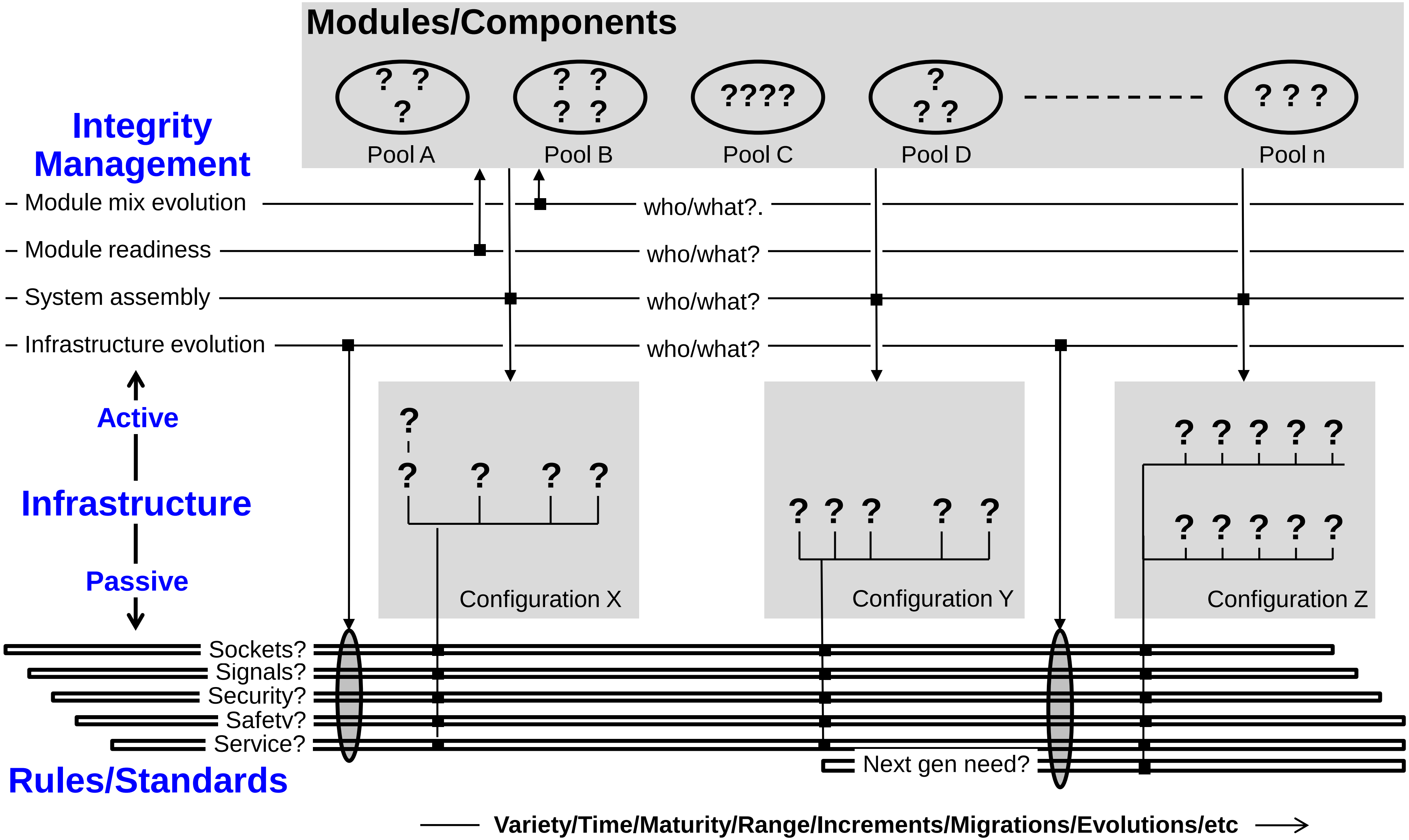
Here is a System-Construction-Kit System

this *agile architecture pattern* provides adaptable structure (Agile 101)



Developing the System-Construction-Kit System Architecture

…how do we answer the questions? (Agile 102)



Sorting Out the Architectures

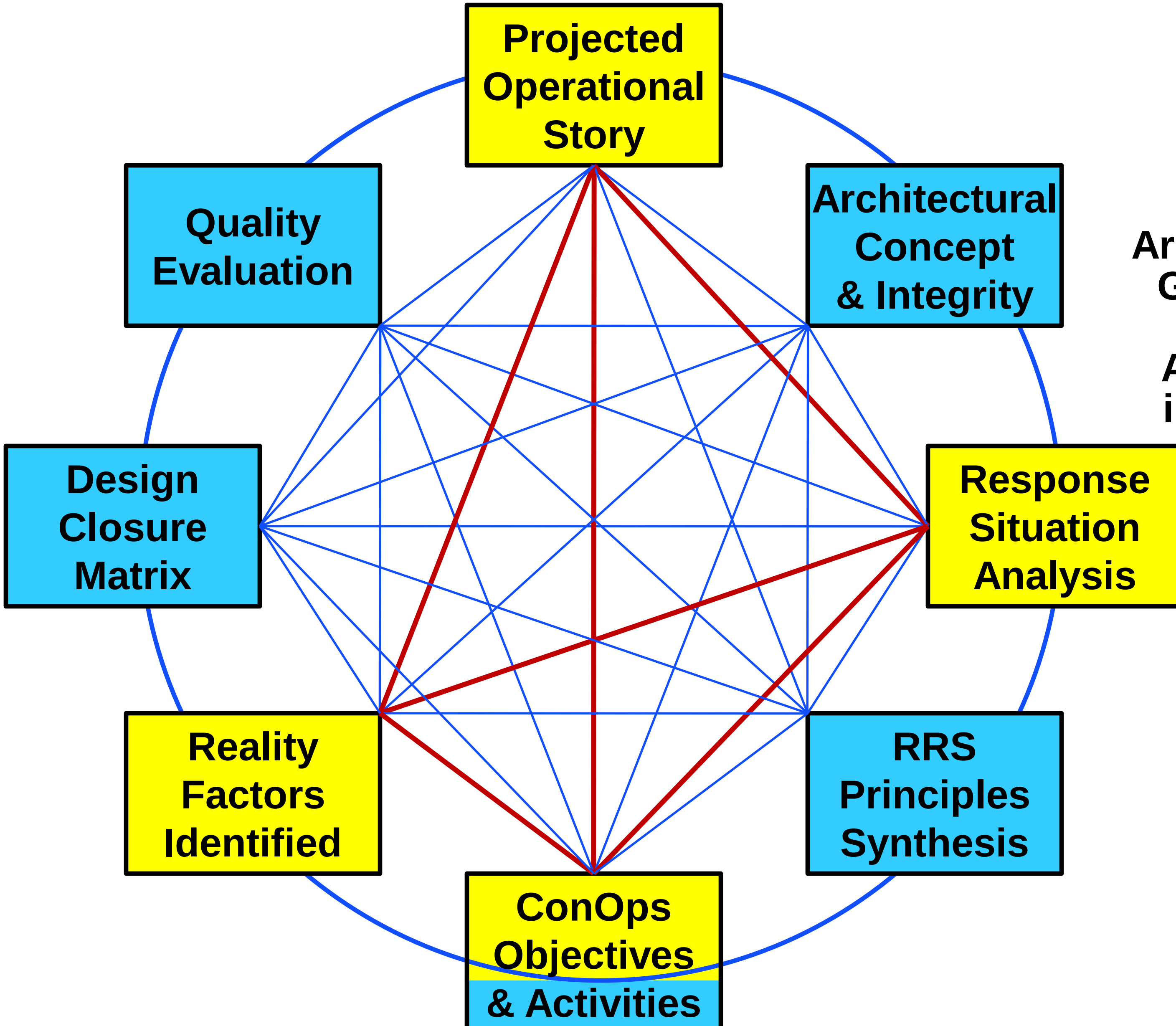
Ferris wheel has a functional architecture.

Erector set has an adaptable architecture.

The adaptable architecture enables the building and changing of the functional architecture.

One could argue that the adaptable architecture is also a functional architecture.
(but why bother?)

Four tools for guiding *agile architecture pattern* development



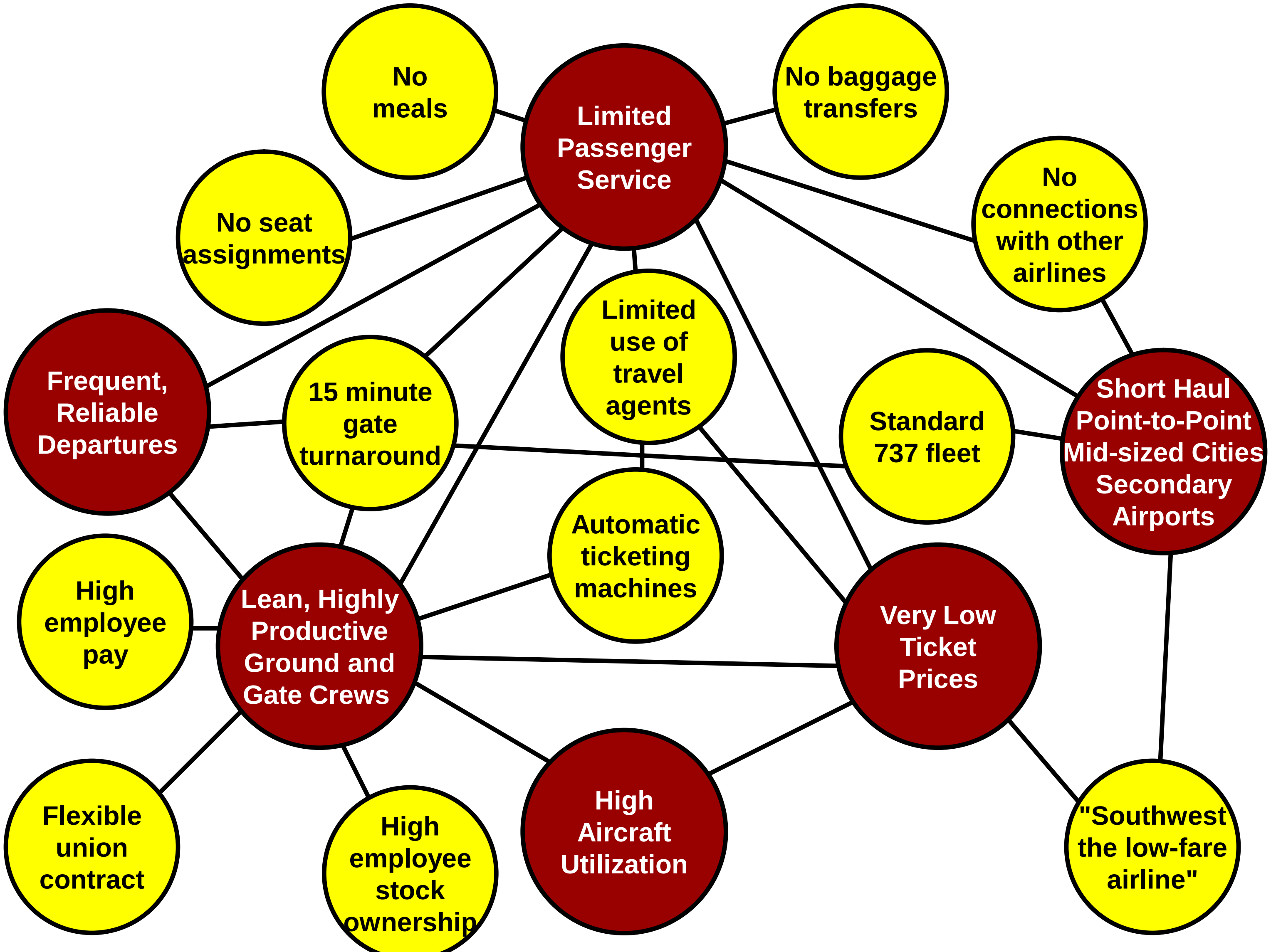
Architecture Guidance
…
Agile 102 in Yellow

Developing a
Concept of Agile Operations

Strategy Web
Operational Story

Southwest Airlines
(Concept of Operations)

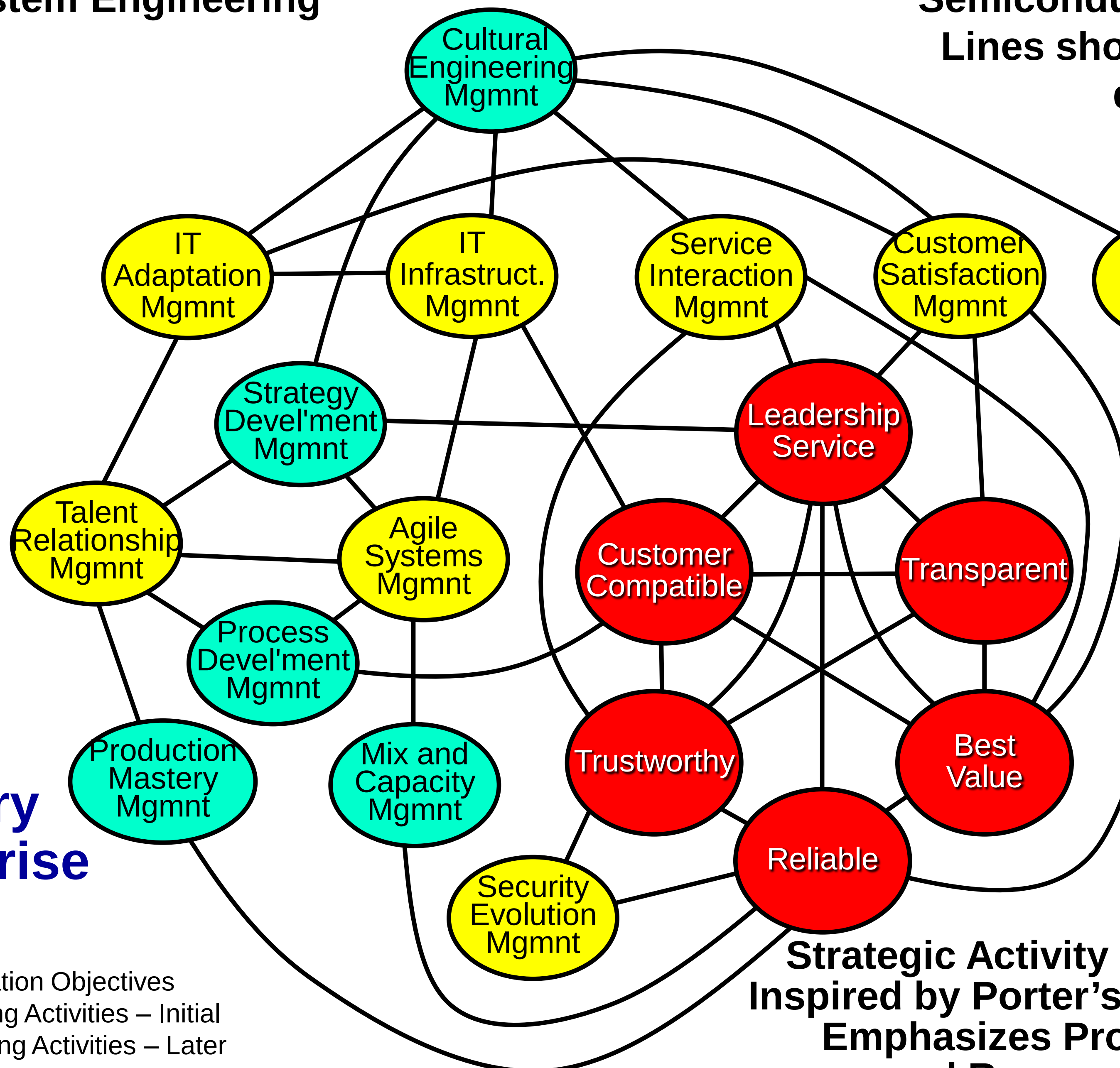
■ Reputation Perceptions (Objectives)
□ Causal Enablers (Activities)



"What is Strategy?", Michael Porter, Harvard Business Review, Nov-Dec '96

Agile-System Engineering

Semiconductor Foundry
Lines show synergistic dependencies

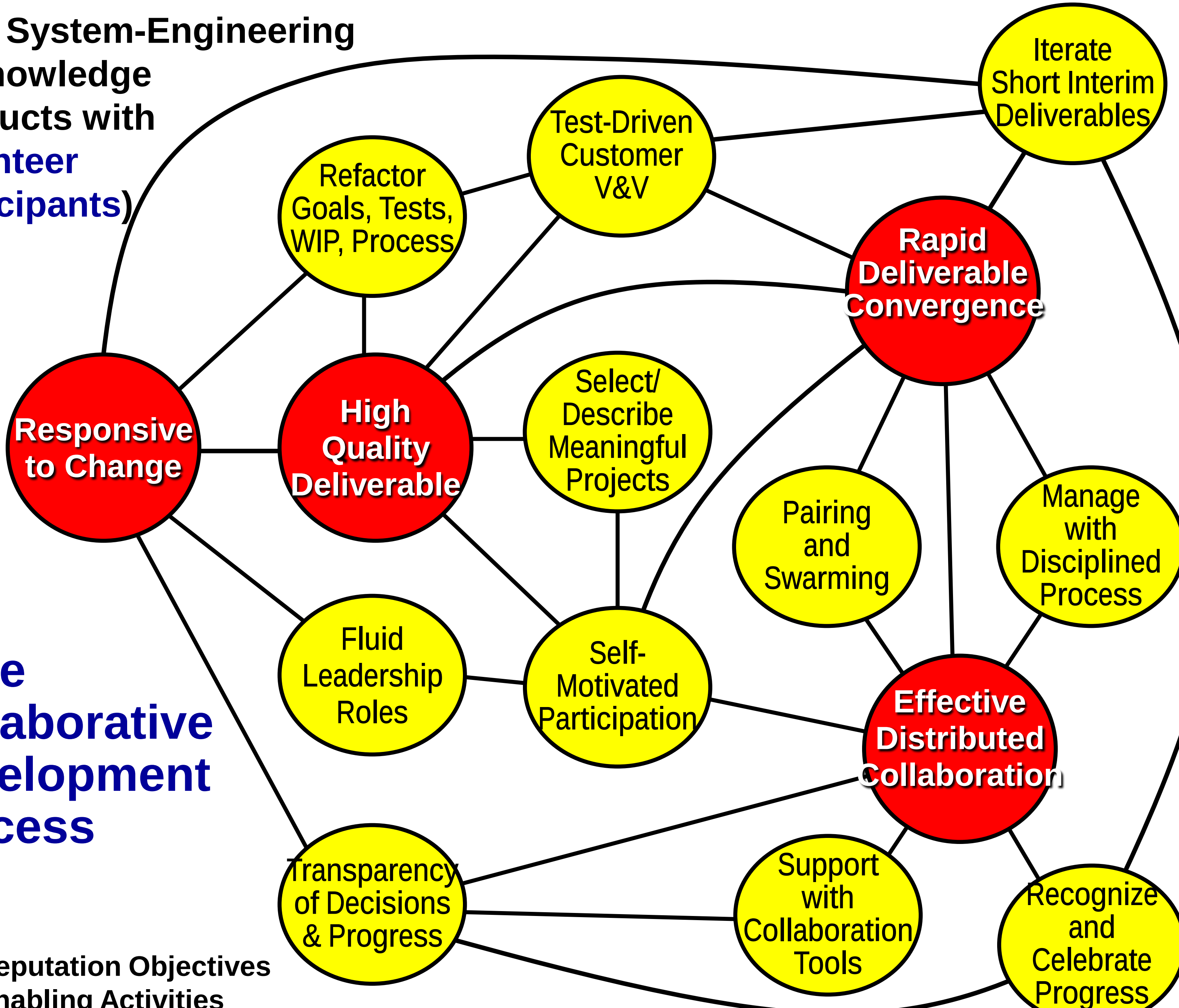


Agile Foundry Enterprise

- Reputation Objectives
- Enabling Activities – Initial
- Enabling Activities – Later

Strategic Activity ConOps Web
Inspired by Porter's Activity Web
Emphasizes Process Activity and Response Objectives

Agile System-Engineering (of knowledge products with volunteer participants)



Agile Collaborative Development Process

Reputation Objectives
Enabling Activities

On the Strategic Activity ConOps Web

This web of synergistic activities should depict the desired <u>value-based public reputation</u>, and the activities that cause that reputation.

It is the architecture of a ConOps.

Reputation objectives (red): a few, 3-7, or focus is lost.

Activities (yellow): these are continuous day-in-and-day-out processes that ensure the objectives are realized. They are not things or concepts. Again, keep the number smallish or the critical activities get lost in the noise.

The few words used to label a red or yellow bubble are critical – they must capture and focus the essence of intent succinctly.

Synergistic Dependencies: more is (often) better - multiple lines attached to every bubble – this provides robustness. And, according to Porter, makes it a lot harder for any competitor to duplicate.

You Are There – Inside The System, Watching it Work

al-Qaeda: An Agile Terrorist Enterprise

By Nicole Long and Vicente Tur-Rojas, Analytic Services Inc., Arlington, Virginia

An al-Qaeda Agent Handler receives instructions to carry out a suicide bombing at a local market at a specified time. The future martyr is a member of the Agent Handler's Operational Cell "A". The Agent Handler selects and activates an Improvised Explosive Device (IED) maker operational cell "B" and a surveillance team from operational cell "C".

During the cell's surveillance of ... local citizens recognize unusua... enforcement intercepts and deta... team. The Agent Handler pulls ... team from Sleeper Cell "A" and c... surveillance tactics to better avoid... before the attack, military operatio... maker; the Agent Handler select... from Operational Cell "B".

Two days later, the attack is carried out successfully, killing four people and injuring dozens; several nearby businesses sustain damage. The market is deserted for days and local businesses do not recover financially for 2 months following the attack.

al-Qaeda is a focused, effective, and agile enterprise that can not be defeated using conventional means

More than six years post-September 11, images of the attacks still remain vivid for most Americans: the shocking footage of planes destroying our national landmarks and killing thousands of innocent people in a systematic process. The heroic response from our first response community. And then the recognition that this was the work of a terrorist group that was not a popularized or well-known threat. Up until then, many thought al-Qaeda to be a stereotypical group of religious zealots living primitively and occasionally causing minor problems far from our shores. But what we have slowly come to realize is that al-Qaeda is a focused, innovative, effective, and agile organization that can not be quickly or easily defeated using our convention approaches.

The cost of the September 11 attacks to America? Nearly 3000 people killed instantaneously, several billions of dollars in infrastructure damage, and hundreds of billions of dollars of collateral effects to an already waning financial market. The cost to al-Qaeda? Nineteen minimally-equipped and moderately-trained terrorists, for a cost of less than $500,000. The planning for the operation was conducted covertly in only two years.

Al Qaeda's two primary founding fathers, Abdullah ...

... important starting point for the discovery of what makes al-Qaeda so agile. Operatives are assigned to cells of varying sizes depending on their experience, know-how, and availability. They typically are given little information and are directed to live normal lives; they only sporadically perform operational tasks. A cell will be facilitated by an Agent Handler—a "commando" of considerable experience, training, and trust—who receives basic instructions from the al-Qaeda central command, evaluates intelligence from his theater of operations, conducts planning, obtains resources, and then carries out operation.

The commandos are spread out in various theaters of operations. Cells are known to be established in North America, Latin America, Europe, the Middle East, Africa, Asia, and Australia. Some cells fall under the al-Qaeda Central Command while some are only loosely affiliated groups aligned under other terrorist organizations. However, al-Qaeda can fund these "sister" organizations to conduct operations in the event their own operatives are unable to carry them out.

Another hallmark of the al-Qaeda enterprise is the training provided to their operatives. Many of them receive rigorous training, to include espionage, concealment, communication, counterfeiting, transportation, and weapons training. This training is offered to prospective operatives for al-Qaeda and its sister organizations alike. Thus, all potential operatives receive the same basic education, and this gives al-Qaeda the opportunity to reinforce its ideology, which helps maintain retention, sustains morale, and reinforces

Another hallmark of the al-Qaeda enterprise is the training provided to their operatives. Many of them receive rigorous training, to include espionage, concealment, communication, counterfeiting, transportation, and weapons training. This training is offered to prospective operatives for al-Qaeda and its sister organizations alike. Thus, all potential operatives receive the same basic education, and this gives al-Qaeda the opportunity to reinforce its ideology, which helps maintain retention, sustains morale, and reinforces individual focus to complete difficult and potentially life-ending tasks.

... the best of his/her abilities. In the event the local al-Qaeda labor pool is lacking in a specialized expertise necessary to conduct a particular operation, the agent handlers can leverage external cells, sleeper cells, sister organizations or request that the command structure quickly provide the proper personnel.

None of the al-Qaeda operation requires a large bureaucracy; instead operational units exist more as a decentralized network. Their ideology reinforces a Spartan lifestyle for their operatives. Thus, commandos are not hindered by the requirement to sustain large resource requirements, and innovative means of operations are considered to obtain spectacular results from limited means. For example, it is estimated that the full al-Qaeda costs to plan and execute the 9-11 attacks cost them anywhere from $250,000 to $500,000. In the U.S. government, that amount might buy the services of two contractors for one year. This high level of innovative use of available resources allows requisite variety to carry out operations while still allowing parsimony within the organization, meaning that their resources are generally flexible.

This level of ingenuity with resources is also contributes to al-Qaeda's ability as an organization to change tactics and techniques quickly in response to external factors or to take advantage of environmental changes. For example, al-Qaeda operatives who learn of planned attacks through connections with other Islamic terrorist organizations are able to "piggy back" on these organizations. Because they have more flexible resources, al-Qaeda can provide appropriate additional capabilities to other organizations when needed for a planned attack. Likewise, because most of their tactics involve plentiful and inexpensive multipurpose components, if law enforcement or military actions change so that one technique is no longer feasible (e.g. mitigation techniques have been created) or that the likelihood of the attack being detected is increased (e.g. new chemical detectors are utilized), then operatives can quickly and easily adjust their strategy to counter these obstacles.

...should come as no surprise that al-Qaeda's ...der, Osama bin Laden, has an education ... in economics and business administration, ... from a family that owns one of the largest ... companies in the Islamic world. Both of ...grounds suggest an early exposure and ... to agile practices. In addition, studies by ...man show that a significant portion of al-...erational members also have engineering ...s and/or business experience, which helps to engender understanding of agile principles throughout the organization.

In essence, al-Qaeda's agility is the primary contributor to its generally accepted resilience. Without its ability to be agile in the face of threats from the military and law enforcement might of the strongest nations in the world, it would not have been able to plan, coordinate, and execute intricate and effective operations such as the 9-11 attacks, the 2000 USS Cole bombing, or the 1998 U.S. Embassy bombings in Kenya and Tanzania. The world's superpowers continue to utilize traditional military and law enforcement means to counter this threat, but these methods are not designed to counter such newer, more agile threats. It seems that al-Qaeda will, for now, remain a constant menace.

The intent of this analysis is not to admire or laud al-Qaeda, but to characterize it, without bias, as an agile enterprise. By understanding what elements of the al-Qaeda organization create agility, it may be possible to destabilize the organization and as al-Qaeda's agility is weakened, its resilience may also diminish. It is the authors' hope that by contributing to the discussion and debate, we can assist in al-Qaeda's eradication.

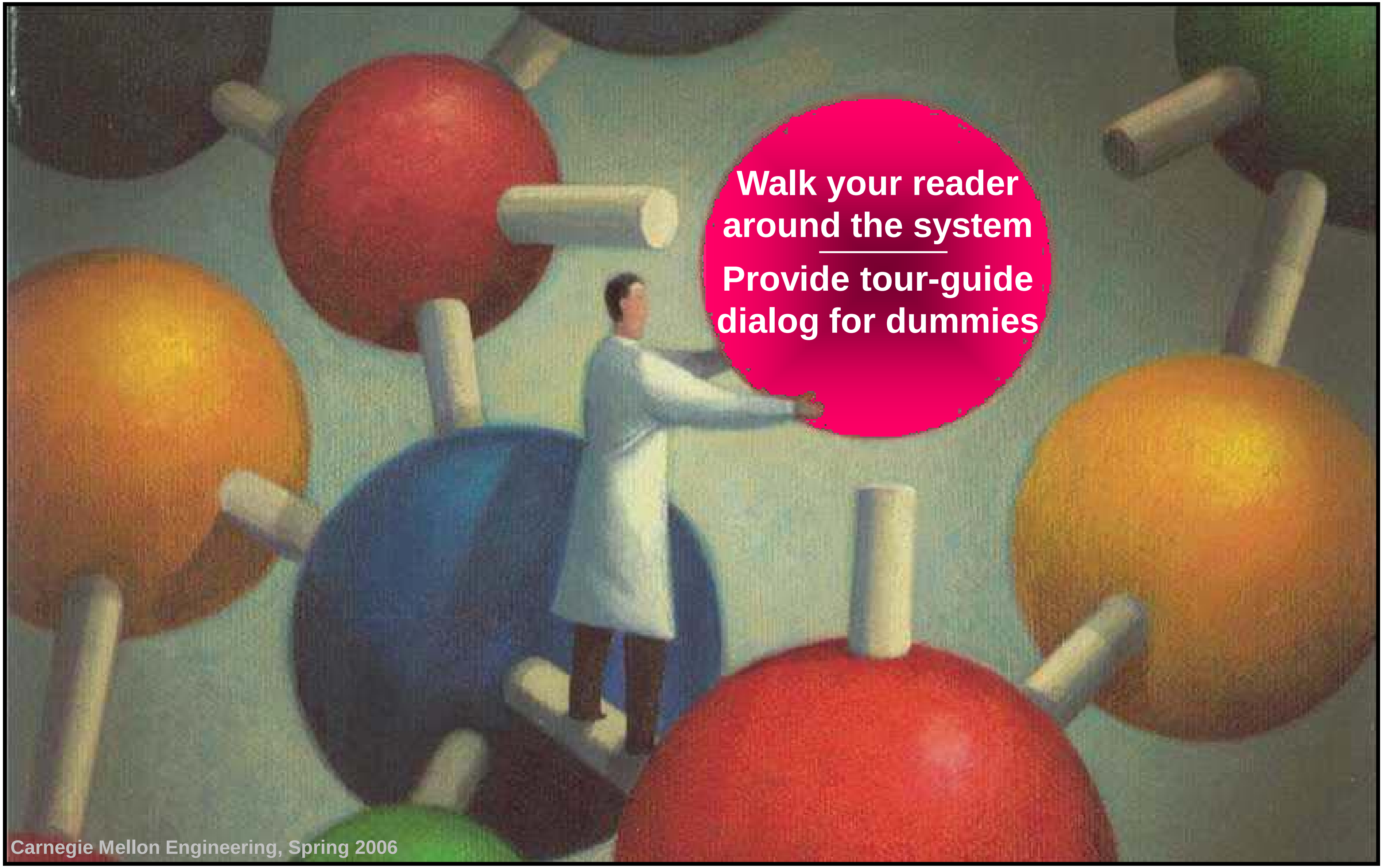
The Operational Story: Imagine yourself as the person who IS ASSEMBLING the drag-and-drop, plug-and-play responses to a variety of "situations" in real time … and tell us what you are doing.

Analytical Services, Oct. 2007

Nicole Long

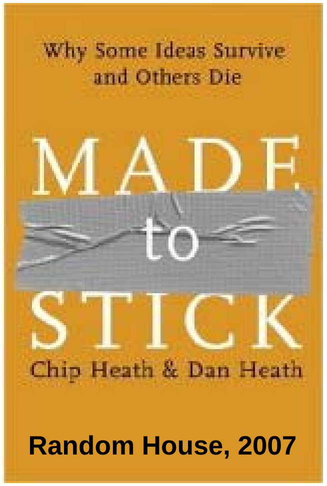
Vince Tur Rojos

Reusable Modules, Reconfigurable, In A Scalable Framework



Your Operational Story Should be *Sticky*
- bring it to life -

http://heathbrothers.com/books/made-to-stick/



- **Simplicity**: the idea must be stripped to its core, and the most important concepts should jump out.
- **Unexpectedness**: the idea must destroy preconceived notions about something. This forces people to stop, think, and remember.
- **Concreteness**: avoid statistics, use real-world analogies to help people understand complex ideas.
- **Credibility**: if people don't trust you, they'll ignore you. In some cases, they will be openly hostile, which means they'll actively try to dispute your message!
- **Emotional**: information makes people think, but emotion makes them act. Appeal to emotional needs, sometimes even way up on Maslow's hierarchy.
- **Stories**: telling a story [gets] people into paying closer attention, and feeling more connected. Remember the Jared Subway commercials?

One or two pages of sticky U-R-There,
and your "proposed" future-vision will be understood (and funded)

Response Situation Analysis (RSA)

UURV
Metrics
8 domains

The UURV Environment Drives the 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:

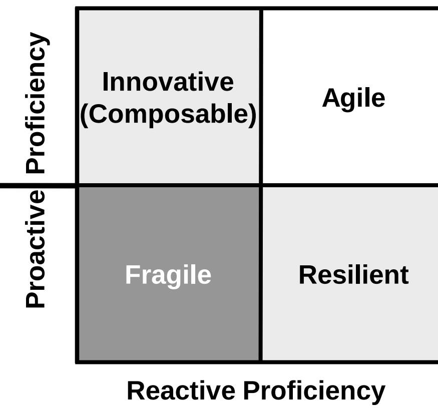
- **Unpredictability: randomness among unknowable possibilities.**
- **Uncertainty: randomness among known possibilities with unknowable probabilities.**
- **Risk: randomness among known possibilities with knowable probabilities.**
- **Variation: randomness among knowable variables and knowable variance ranges.**

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).

Knight, Frank H. 1921. *Risk, Uncertainty and Profit.* Hart, Schaffner & Marx.
Full text available at: www.econlib.org/library/Knight/knRUPCover.html

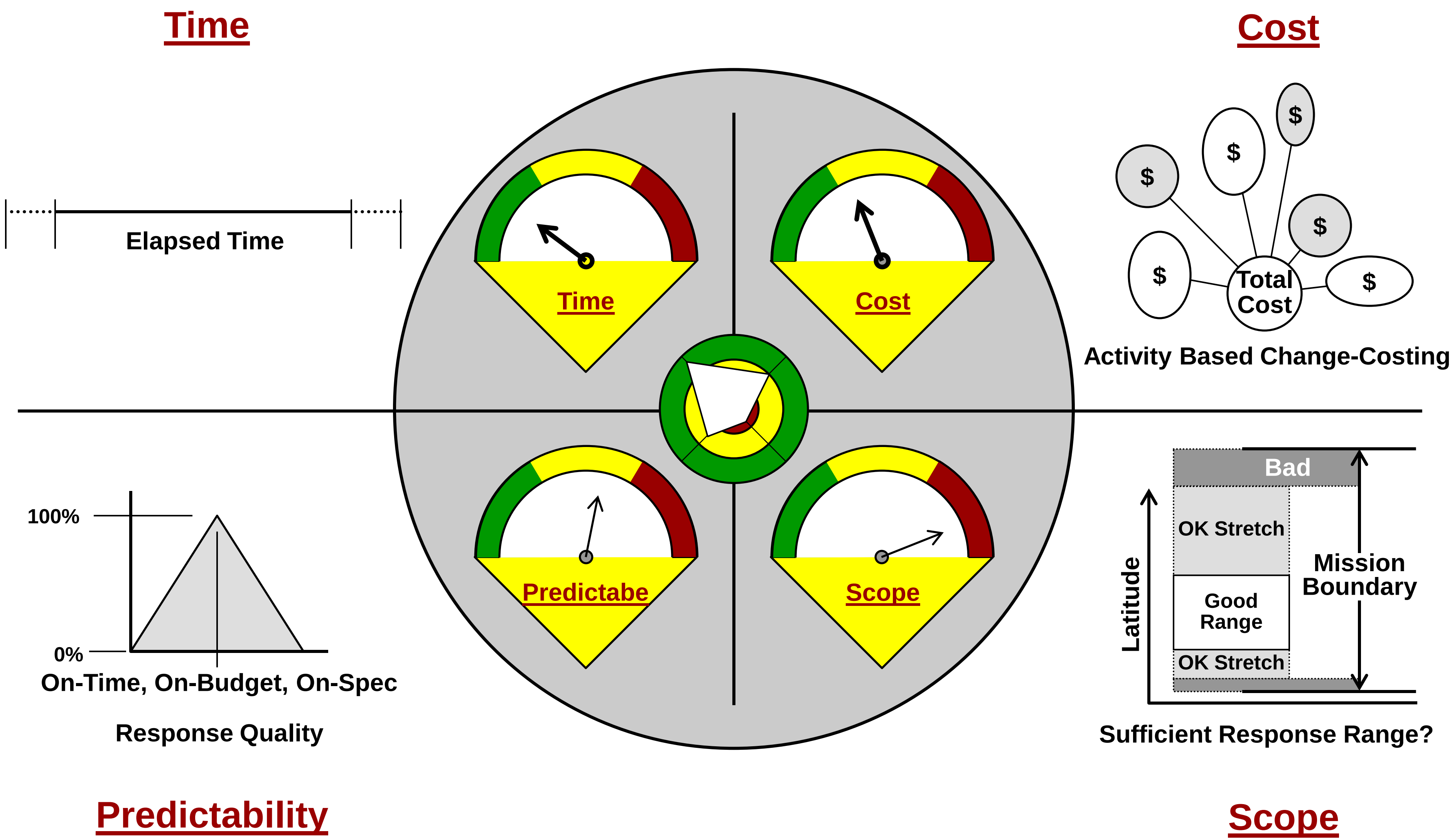
Change/Response Domains

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



Change/Response Metrics

Identify initial metric priorities (t-c-p-s) for each response situation



Change/Response Domains

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

Creation/Elimination

What range of opportunistic situations will need modules assembled into responsive system configurations; what elements must the system create during operation that can be facilitated by modules and module pools; what situational evolution will cause obsolesce of modules which should be removed?

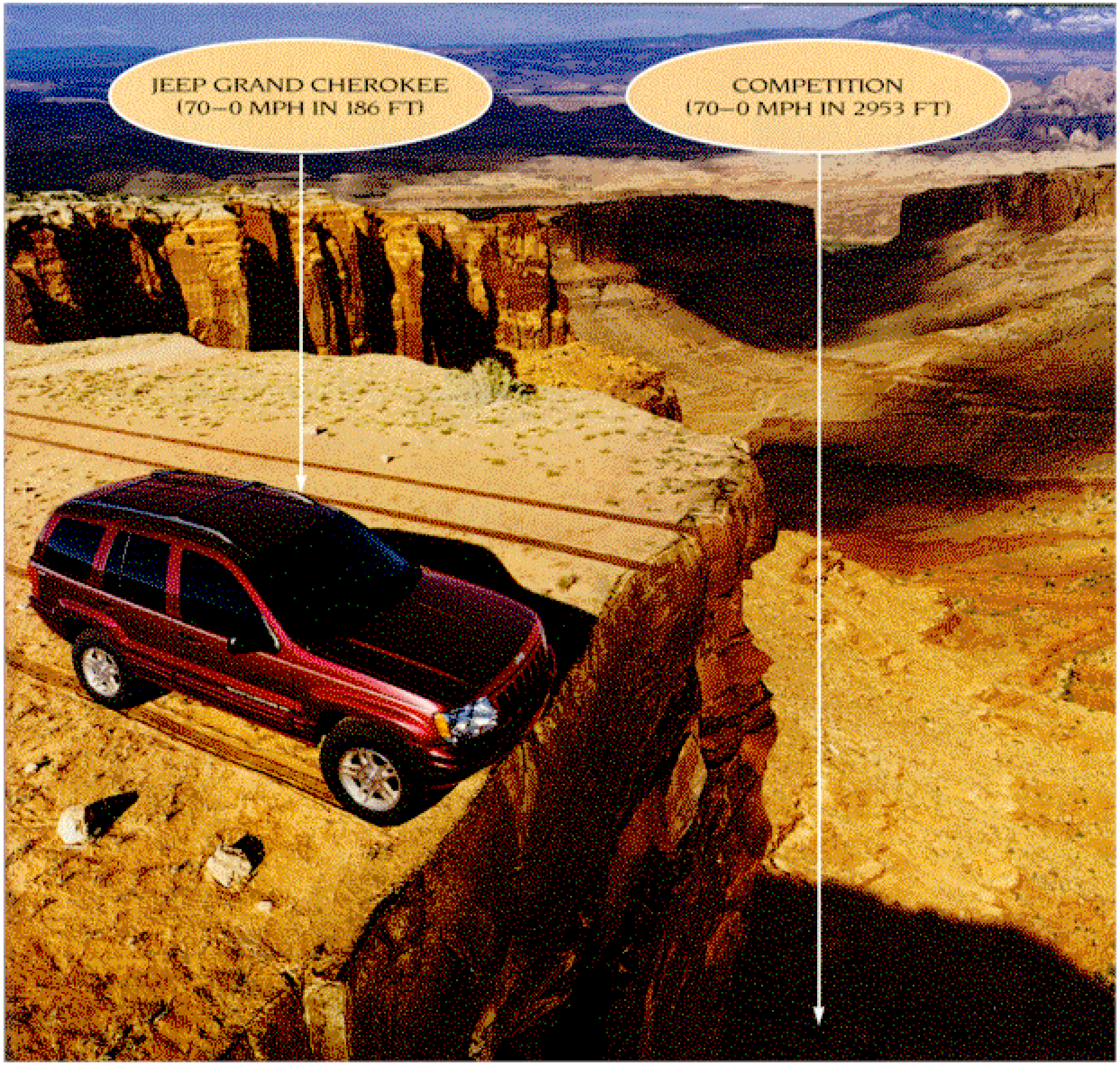
The distinguishing feature is the creation of something new or reincarnated that is not currently present. To note, this is not about the situation that calls for the original creation of an agile system, but rather about the evolution of the agile system during its operational period.

Situations to identify are those that require system configuration assemblies during operation, and those that require new modules for employment in those assemblies.

Agile Systems-Engineering (Project Mgmnt)

- project management strategy (t);
- project team (t, c); • system requirements (t, p);
- system architecture (t, s);
- system design (t, c, p);
- development activity plans (t);
- V&V/test plans (t);
- team collective understanding and learning (t, p);
- product development [software code, hardware build documentation] (t, c, p).

Inertia – The Bane of Agility



Ceasing prior activity

quickly and cleanly

is just as important as

starting new activity.

Bane: a cause of death, destruction, ruin (Webster)

Improvement

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

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

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

Agile Systems-Engineering (Project Management)

- activity effort estimating (p);
- activity completion to plan (t, c, p);
- reducing uncertainty and risk (t, p, s).

Migration

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

The distinguishing feature is a need to change the nature of the plug-and-play infrastructure, not the addition of new modules.

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

Agile Systems-Engineering (Project Management)

- compelling new technology availability (t, c, s);
- project scope change (s);
- lean process principles (p).

Modification (of capability)

What evolving technologies and opportunities might require modification of the available modules and roster of module pools?

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

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

Agile Systems-Engineering (Project Management)

- new added team member unfamiliar/uncomfortable with management strategy (t);
- new environmental dynamics (t, c, p, s).

Change/Response Domains

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

Correction

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

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

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

Agile Systems-Engineering (Project Management)

- wrong requirement (t);
- inadequate developer (t);
- failed V&V/test (t, c);
- non-compliant supplier (t, c).

Variation

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

The distinguishing feature is predictable but uncertain variance.

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

Agile Systems-Engineering (Project Management)

- expertise and skill levels among team members (p);
- grace period on schedule (t, c);
- deliverable performance range (p);
- availability, interaction, and expertise of customer involvement (s).

Expansion/Contraction

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

The distinguishing feature is capacity scalability.

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

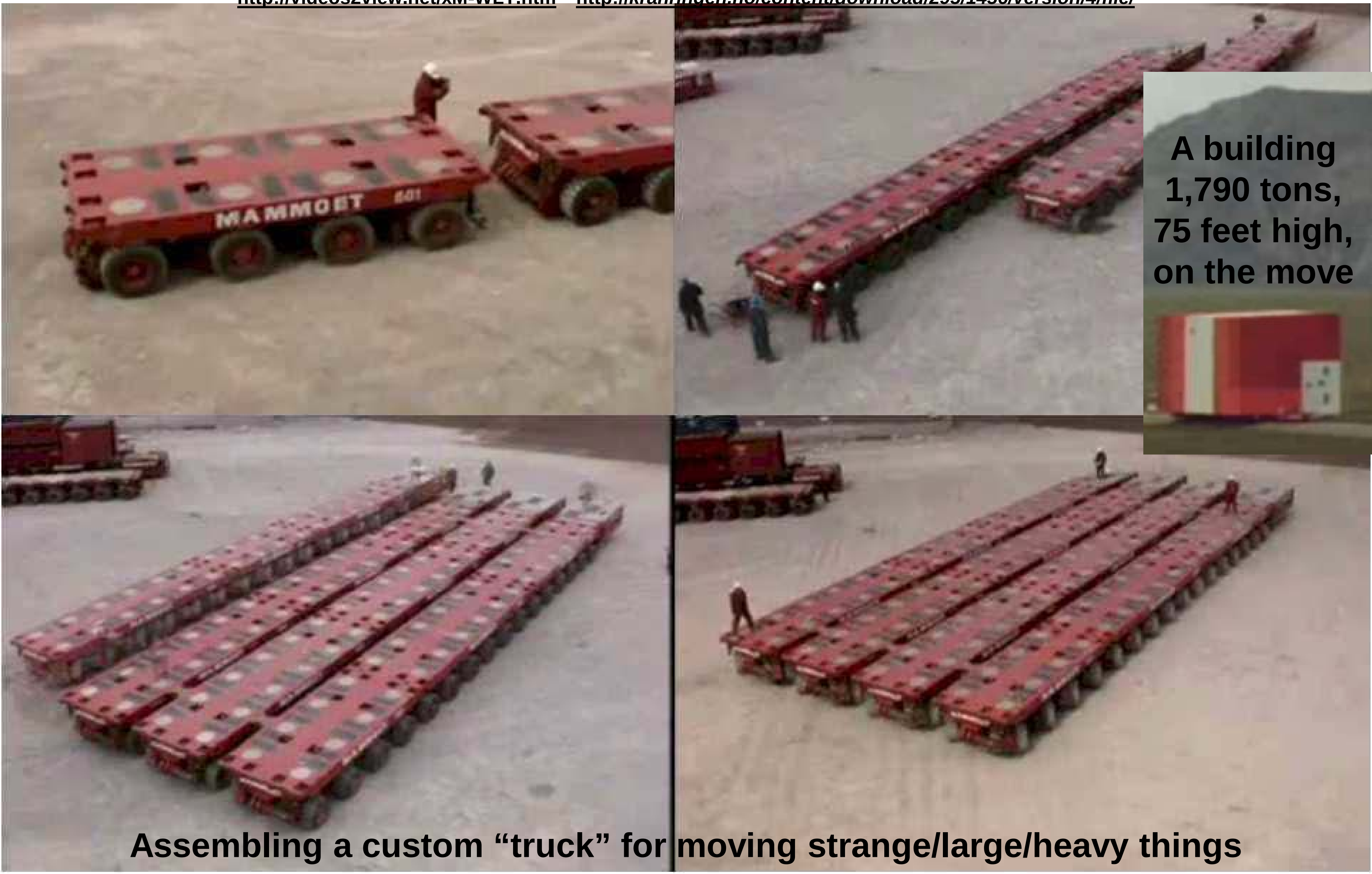
Agile Systems-Engineering (Project Management)

- 2x project scope change (t, c, p, s);
- team-size changes of x-y engineers distributed across n-m locations (t, s).

Reusable, Reconfigurable, Scalable Expansion/Contraction: Unbounded Capacity

File5.5
File8.25

http://videos2view.net/xM-WLT.htm   http://kranringen.no/content/download/295/1456/version/4/file/



A building 1,790 tons, 75 feet high, on the move

Assembling a custom “truck” for moving strange/large/heavy things

Reconfiguration

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

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

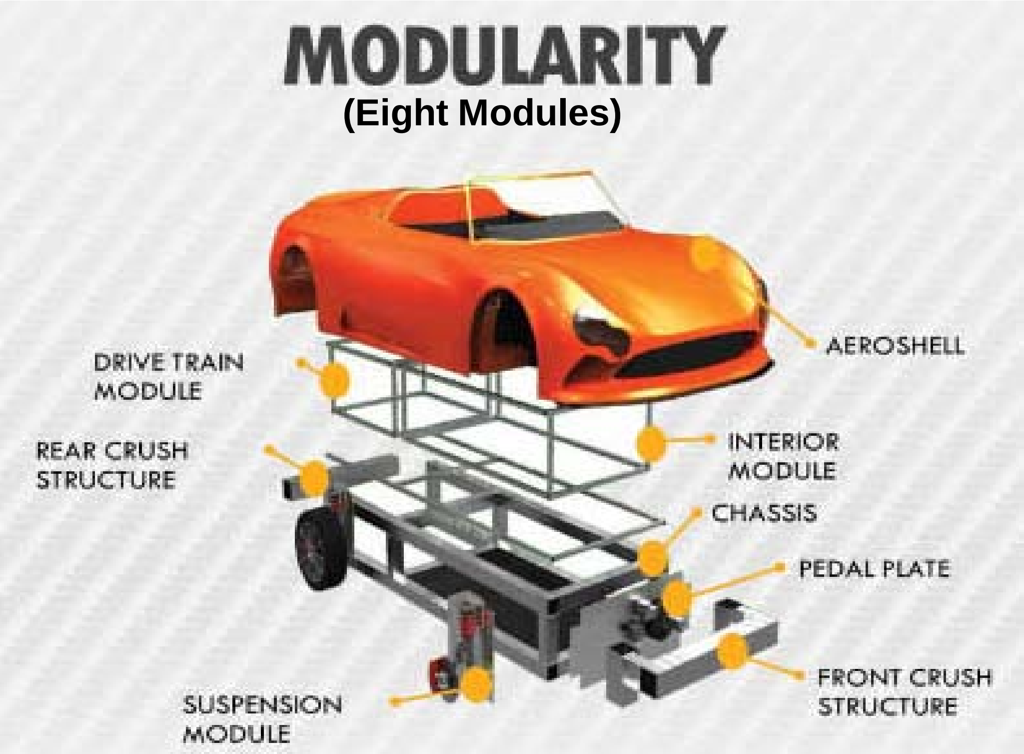
Situations to identify are those that are within the system mission boundaries, and that may require a reconfiguration of an existing system assembly, perhaps augment with removal of modules or addition of available modules.

Agile Systems-Engineering (Project Mgmnt)

- unanticipated expertise requirement (t);
- development activity-sequence priority change (t).

Wikispeed's Modular Cars

Detroit Auto Show, 11Jan2011



Modular design – Development is rapid because the design of the car is modular.

The engine is able to be switched from a gasoline to an electric engine in about the time it takes to change a tire. The car body can be switched to a pickup truck.

Modular design enables Wikispeed to make changes and develop quickly.

Simplicity and modularity reduce costs in making changes, in tooling, in machinery and in complexity.

Accelerating the response to problems – Wikispeed has steadily increased its velocity in resolving issues. For instance, on one occasion, within hours of getting a video back from a side impact test, the team realized that there was four inches of penetration into the cabin. It was still survivable, and still road-legal, but it wasn't the five star crash rating that the team wanted. So within hours, they had a volunteer team update the side impact crash structure and bolt it onto the car. The first time Wikispeed did a safety iteration like this, it took many weeks. Now they are able to accomplish it within a seven day sprint cycle.

Video and text at: www.youtube.com/watch?v=tTDCQMjbc40

Pro Forma RSA for Agile SE Project Management

with [t,c,p,s] metric-priorities for each issue, t = time of change, c = cost of change, p = predictability of change, s = scope of change

Change Domain		
Proactive	Creation (and Elimination)	• project management strategy (t); project team (t, c); system requirements (t, p); • system architecture (t, s); system design (t, c, p); development activity plans (t); • V&V/test plans (t); team collective understanding and learning (t, p); • product development [software code, hardware build documentation] (t, c, p).
	Improvement	• activity effort estimating (p); • activity completion to plan (t, c, p); • reducing uncertainty and risk (t, p, s).
	Migration	• compelling new technology availability (t, c, s); • project scope change (s); • lean process principles (p)
	Modification (of Capability)	• new added team member unfamiliar/uncomfortable with management strategy (t); • new environmental dynamics (t, c, p, s).
Reactive	Correction	• wrong requirement (t); • inadequate developer (t); • failed V&V/test (t, c); • non-compliant supplier (t, c).
	Variation	• expertise and skill levels among team members (p); • grace period on schedule (t, c); • deliverable performance range (p); • availability, interaction, and expertise of customer involvement (s).
	Expansion (of Capacity)	• 2x project scope change (t, c, p, s); • team-size changes of x-y engineers distributed across n-m locations (t, s).
	Reconfiguration	• unanticipated expertise requirement (t); • development activity-sequence priority change (t).

Getting it Right

Requirements *shall statements* define

exactly what must be accomplished.

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

For Response Situation Analysis…

you do not need to develop a *comprehensive* list of shall statements, but

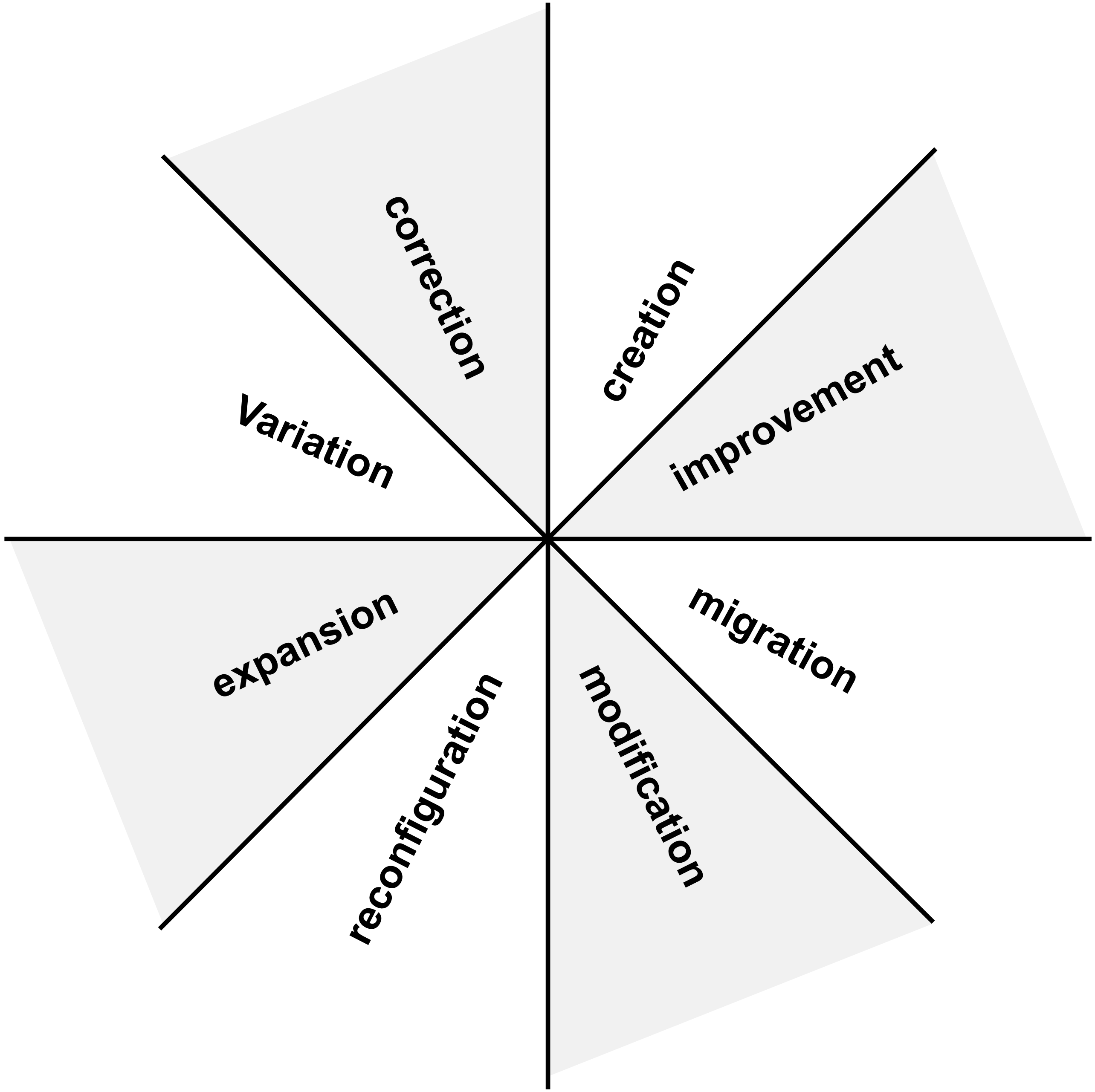
rather a *sufficient* list of *response capabilities* –

which if accomplished,

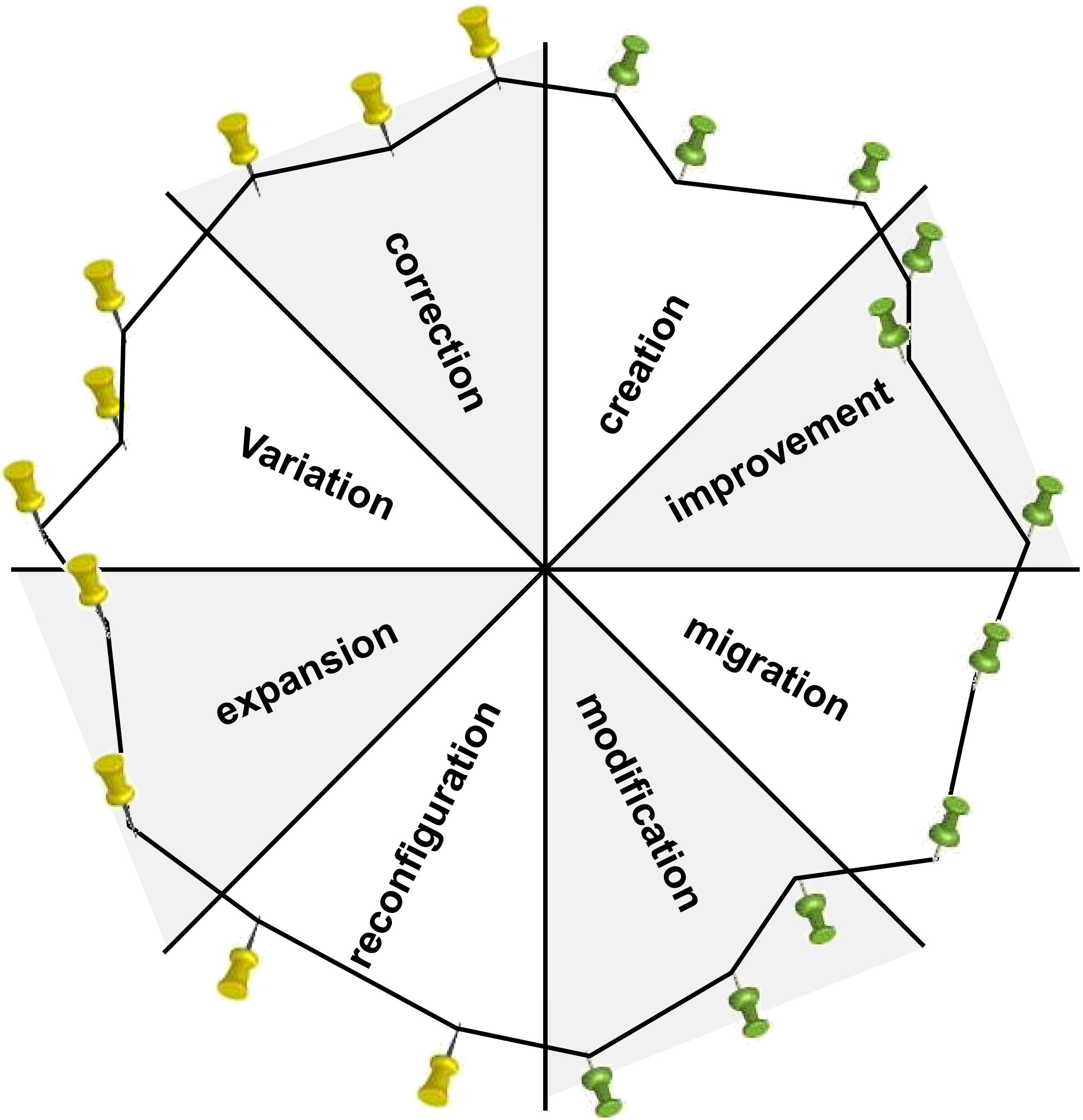
will stretch the envelope of agile response capability

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

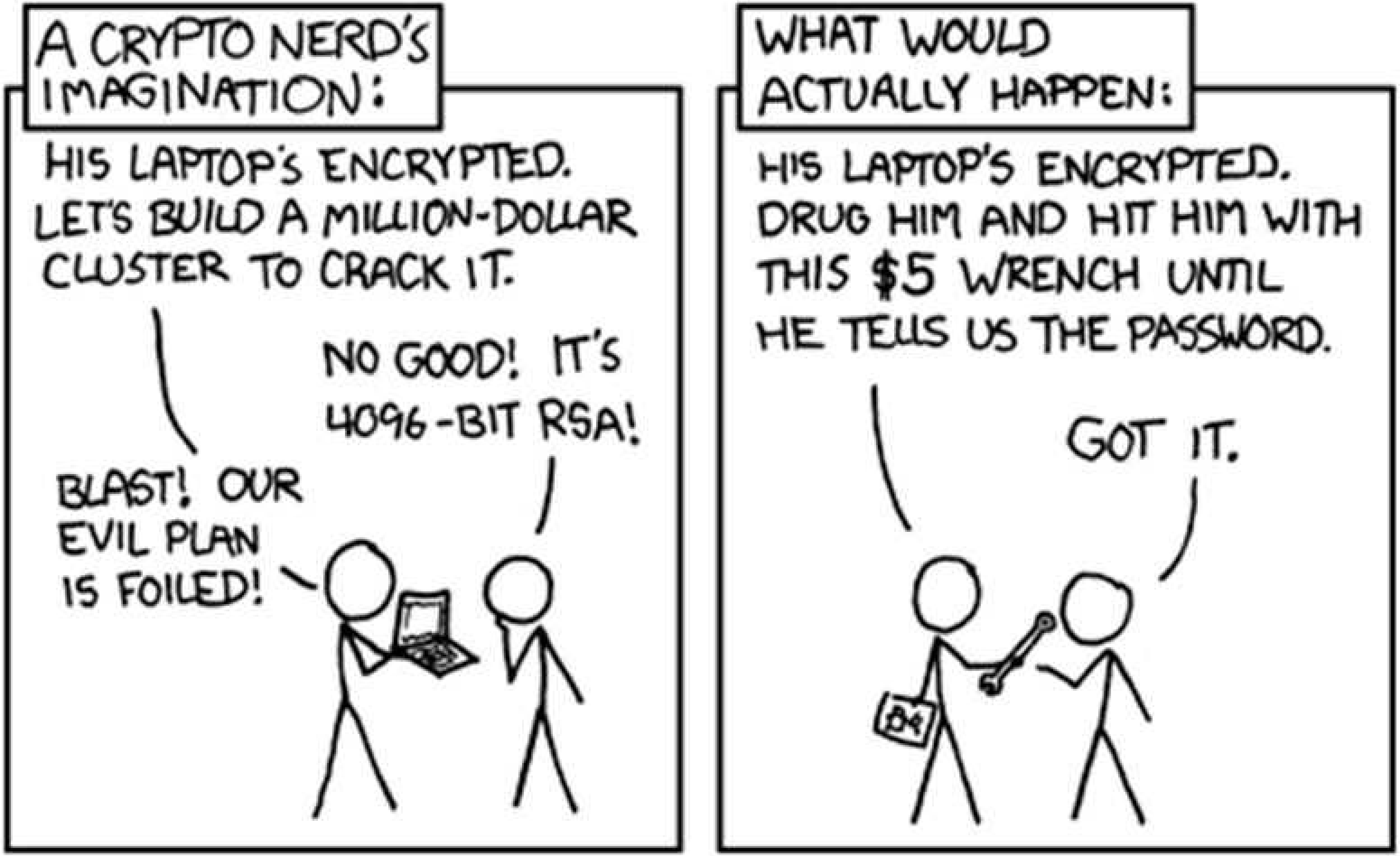
even if they were not on the list.



correction
creation
Variation
improvement
expansion
migration
reconfiguration
modification


correction
creation
Variation
improvement
expansion
migration
reconfiguration
modification

Reality Factors



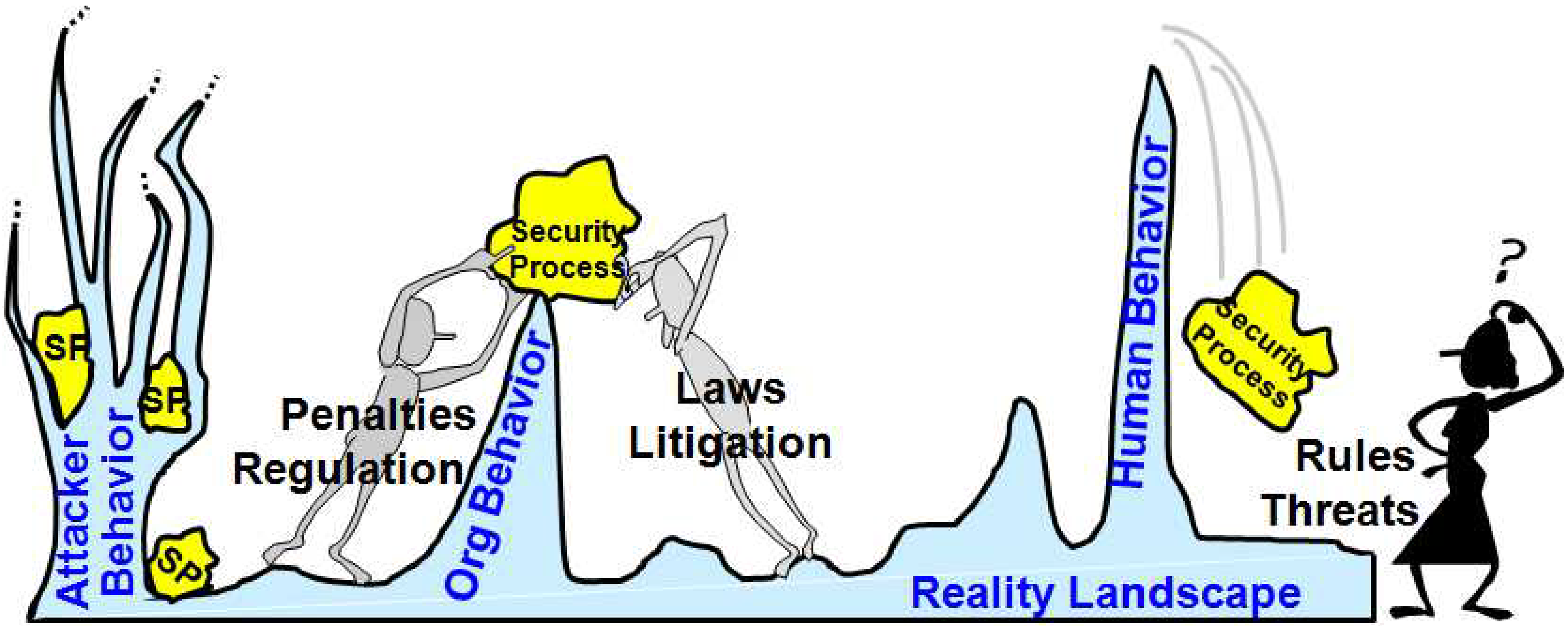
A padlock on a straw hut

http://xkcd.com/538/

Agility is All About Dealing With Reality

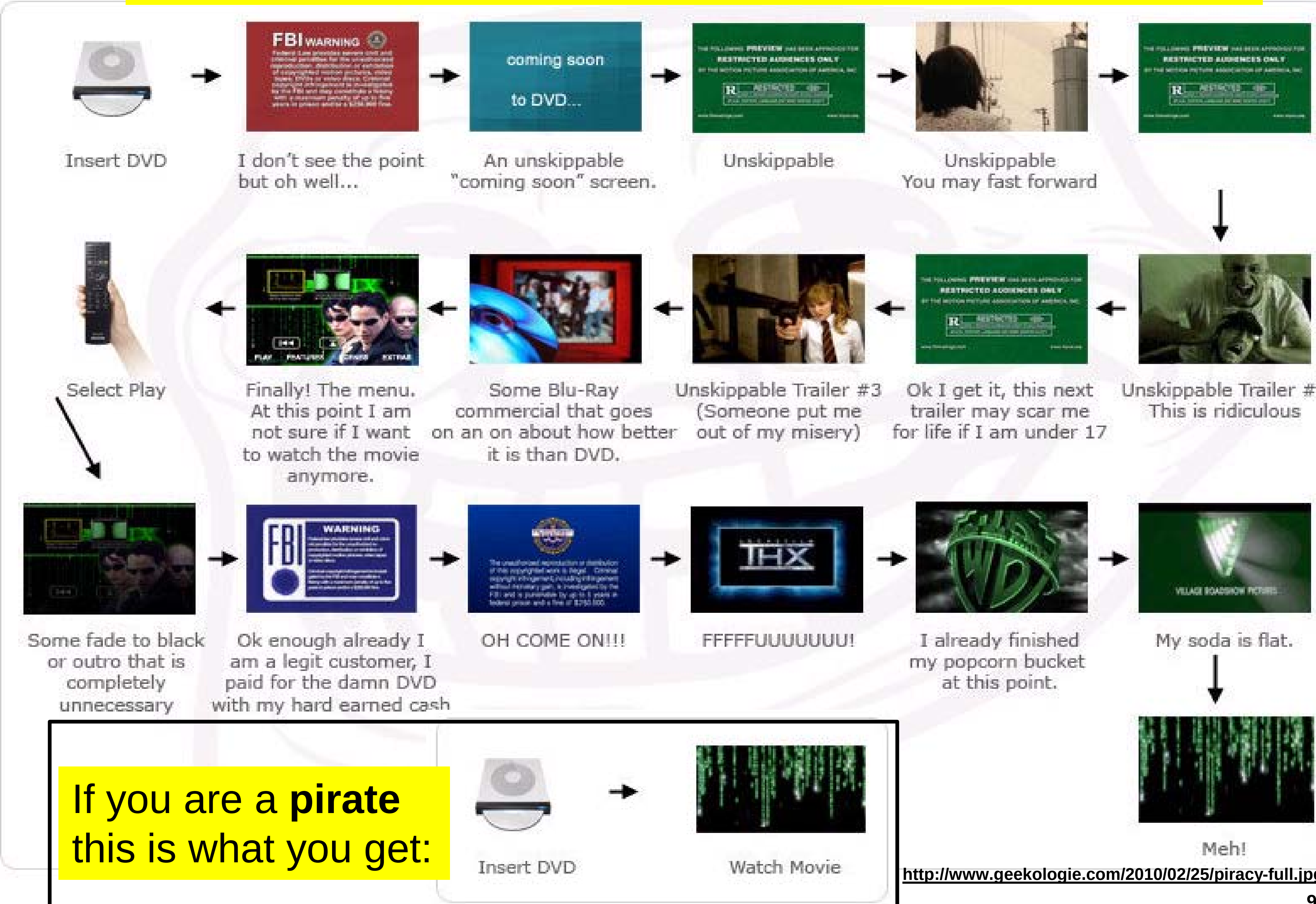
Security strategy often ignores reality:

eg., maintaining systems in unstable states takes constant energy input



Expecting or enforcing ideal and repetitive behavior ignores reality... not a substitute for effective strategy

If you are a **paying customer**, this is what you get:



http://www.geekologie.com/2010/02/25/piracy-full.jpg

Reality Factors – Framework

Think like a red team.
Identify uncooperative and malicious environmental factors.
(fold the results back into the RSA)

Human Behavior – Human error, whimsy, expediency, arrogance…
Organizational Behavior – Survival rules rule, nobody's in control…
Technology Pace – Sparse testing, faster bug/vulnerability-introductions…
System Complexity – Incomprehensible, unintended consequences…
Globalization – Partners with different ethics, values, infrastructures…
Faddish Ill-Thought Practices – Outsourcing, web services, cots, transparency…
Agile Adversaries/Competitors/Customers – Distributed, collaborative, self organizing, proactive, innovative…

Example: Agile SE For Aircraft Refurb Company

Reality Factors
Human Behavior – Human error, whimsy, expediency, arrogance… • Lack of attention to detail by production and engineers. • Fatigue due to long hours with short breaks. • Arrogant less experienced mechanics not seeking the advice of more experience mechanics.
Organizational Behavior – Survival rules rule, nobody's in control… • The “get it done, I don’t care how” attitude that is sometimes present. • The idea to keep work load high but manpower minimal. • Counterproductive monetary incentives that put schedule over quality.
Technology Pace – Accelerating vulnerability-introductions, sparse testing… • Electronic drilling machine. • Work instruction-delivery changing with technology pace.
System Complexity – Incomprehensible, highly networked, unintended consequences, emergence… • High knowledge/experience required. • Level of complexity leads to unintended consequences such as additional damage by the mechanics.
Globalization – Partners with different ethics, values, infrastructures… • Diverse ethnicities and cultural beliefs and values among customers. • Priority lists are different due to diversity.
Faddish Ill-Thought Practices – Outsourcing, web services, transparency, COTS, SOA… • Outsourcing manufacturing. • Using Components off the shelf (COTS).
Agile Adversaries/Competitors/Customers – Distributed, self organizing, impatient, innovative… • Competitors may provide a more agile operation. • Customer desires maximum agility.

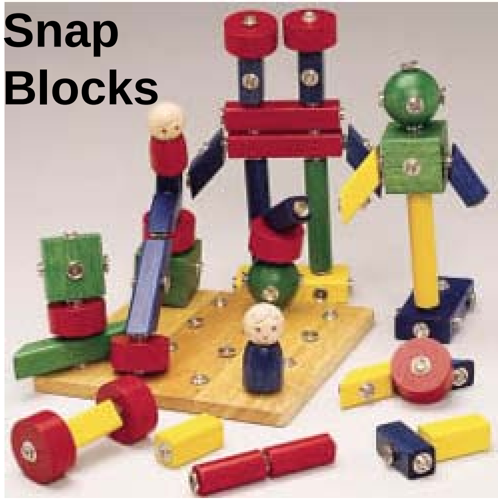
Wrapping it Up

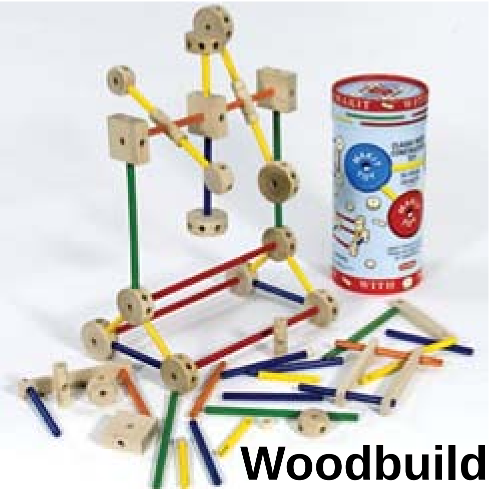
Bendables

Straws and Connectors

Marble Run

Snap Blocks

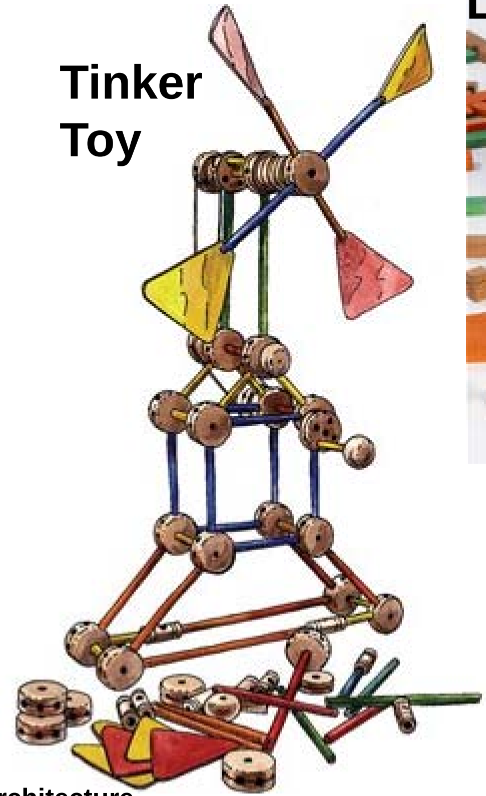




Woodbuilders

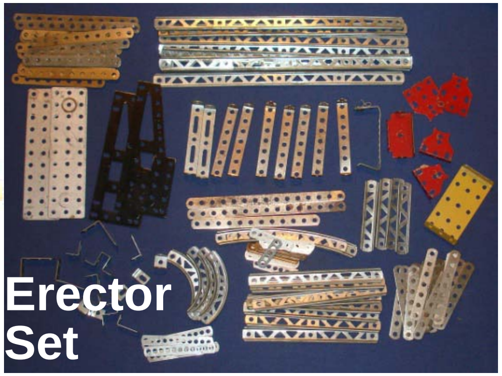
Design the Architecture of Your Construction Set

Tinker Toy



Log Builder

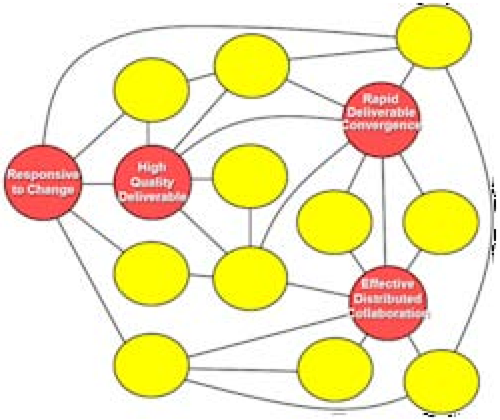
Lego



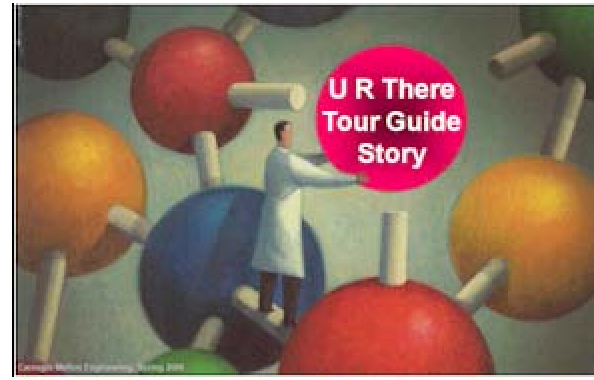
Erector Set

Bristle Blocks

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

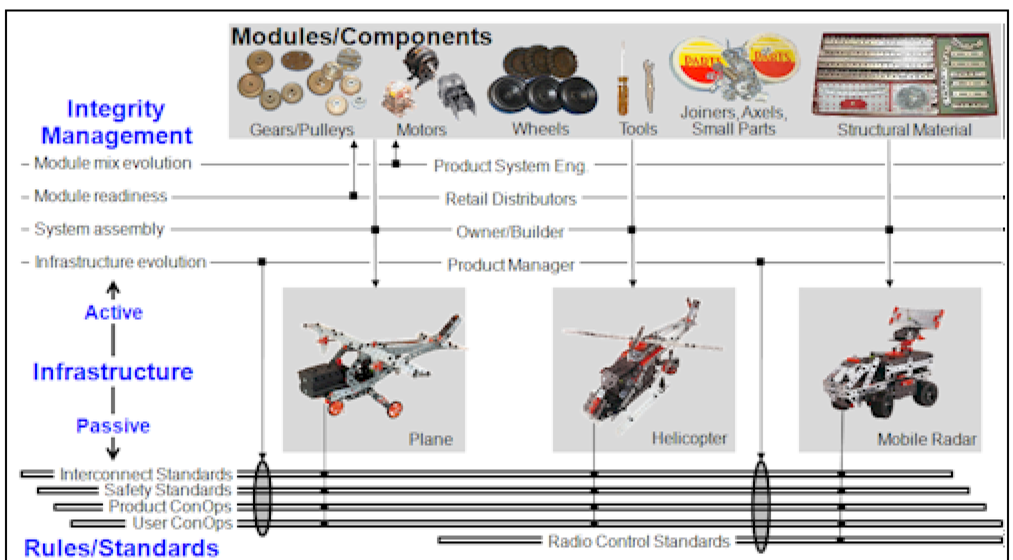


Agility ConOps

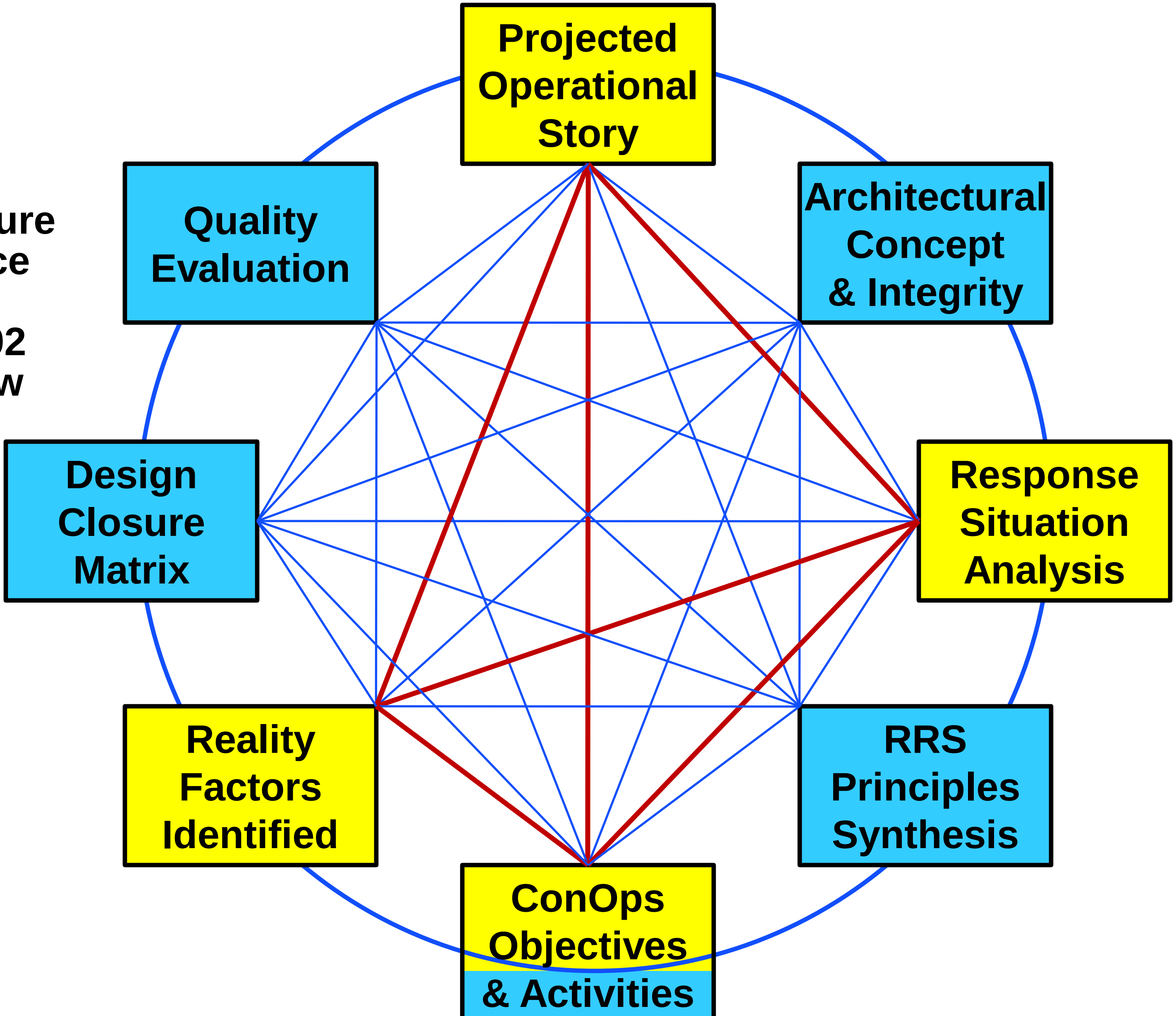


Response Requirements

4 Thinking Tools for Guiding Architecture Pattern Development



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



Architecture Guidance
...
Agile 102 in Yellow

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

A Working Group of INCOSE

(International Council on Systems Engineering)

On request to rick.dove@parshift.com:

1. **Get on WG mail list for general announcements.**
2. **Get access to WG shared documents.**
3. **Participate in WG remote-collaboration projects.**

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.

(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

(Csete 2002) Marie E. Csete and John C. Doyle. Reverse Engineering of Biological Complexity. Vol 295 SCIENCE, 1 March. www.cds.caltech.edu/~doyle2/wiki/images/7/7a/Science1664-2002.pdf

(Csete 2004) Marie Csete and John Doyle. Bow Ties, Metabolism and Disease. TRENDS in Biotechnology 22(9), September. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.173.3019&rep=rep1&type=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

(Papke 2013) Barry Papke and Rick Dove. Combating Uncertainty in the Workflow of Systems Engineering Projects. Paper submitted for INCOSE IS13 review. 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 102 webinar slides: www.parshift.com/s/AgileSystems-102.pdf

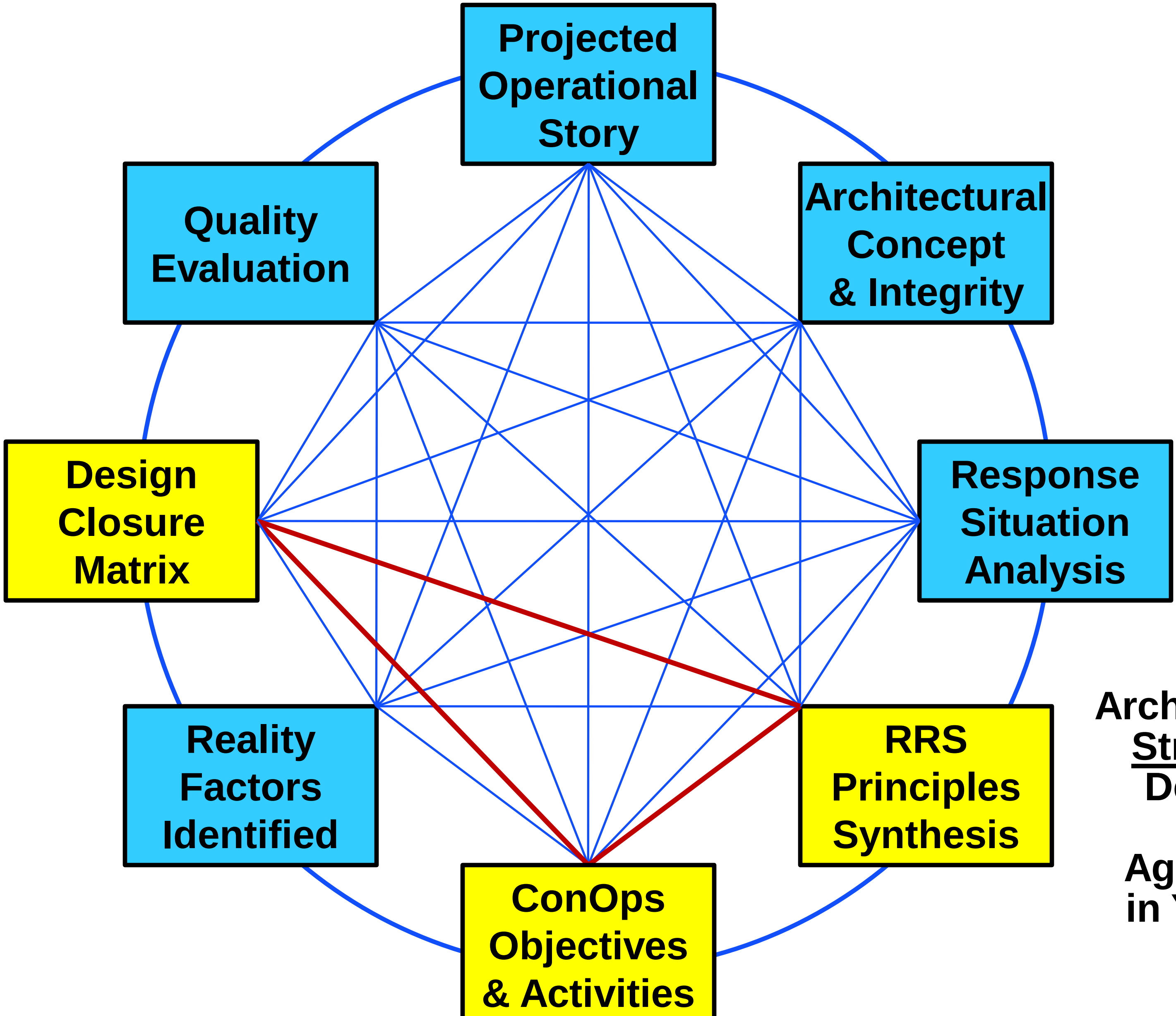
Download 101 webinar slides: www.parshift.com/s/AgileSystems-101.pdf

(updated asynchronously from time-to-time)

Part 3
Agile 103 – Design Fundamentals

Agile Systems and Processes –
Fleshing out the Architecture with
RRS Design Principles, Activities, and Closure Matrix

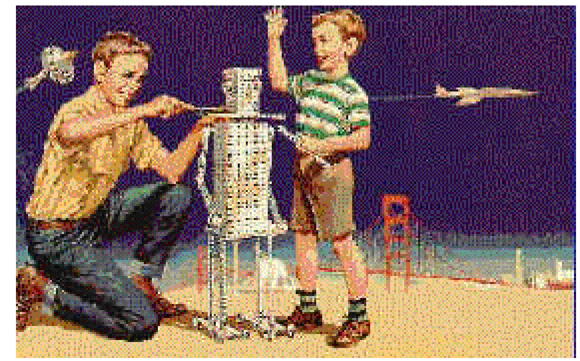
Eight principle tools are brought to bear when designing or analyzing a system for agility

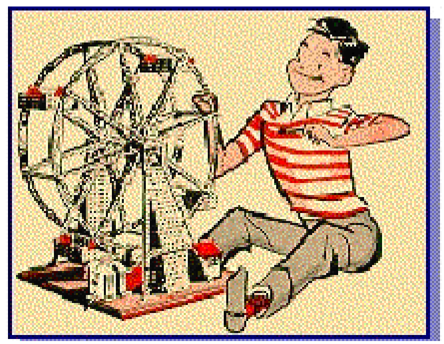


Architecture
<u>Strategy</u>
Design
...
Agile 103
in Yellow

ERECTOR/MECCANO
A Modular Construction System

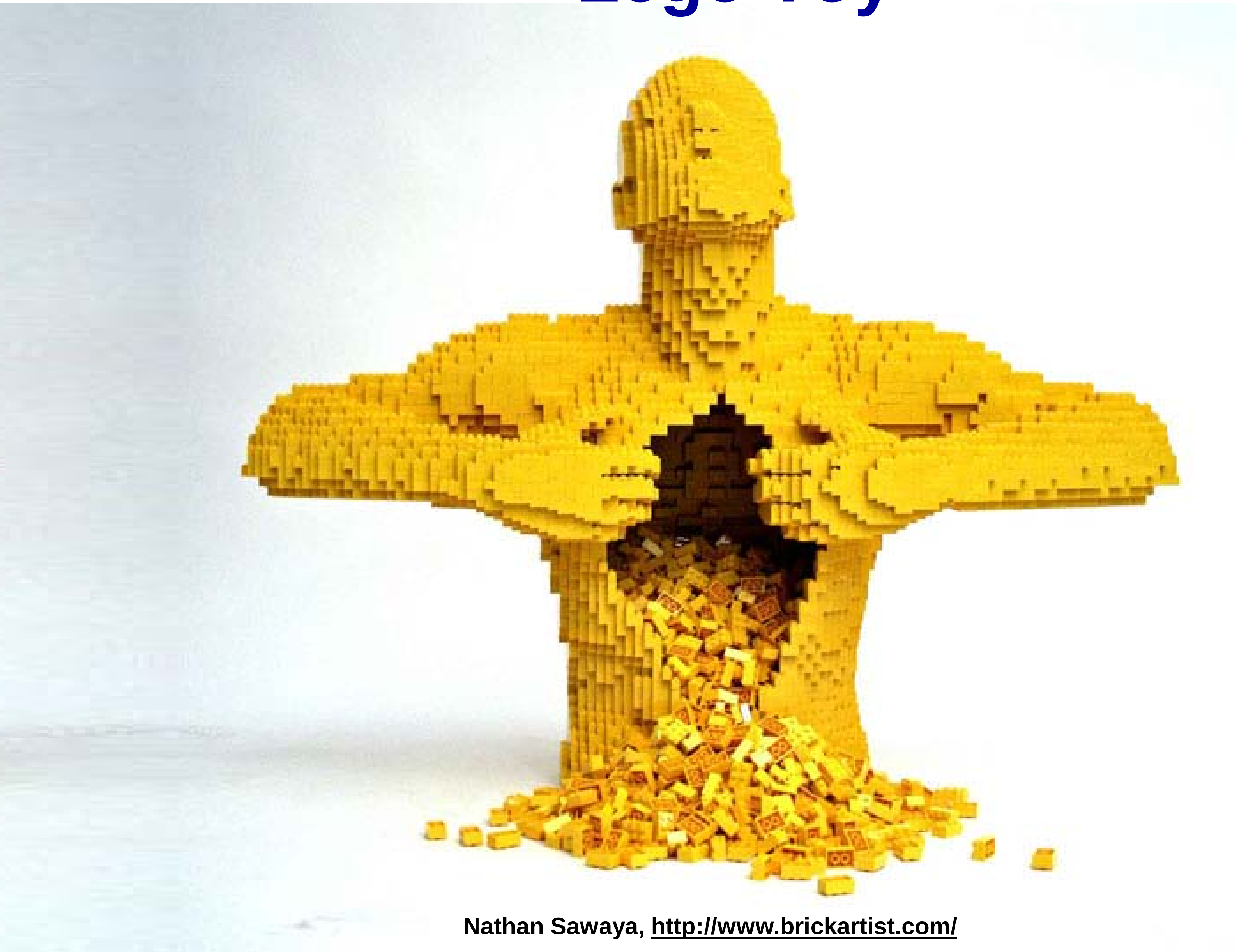
Restored 10-1/2 Amusement Set.
You wanted this one.





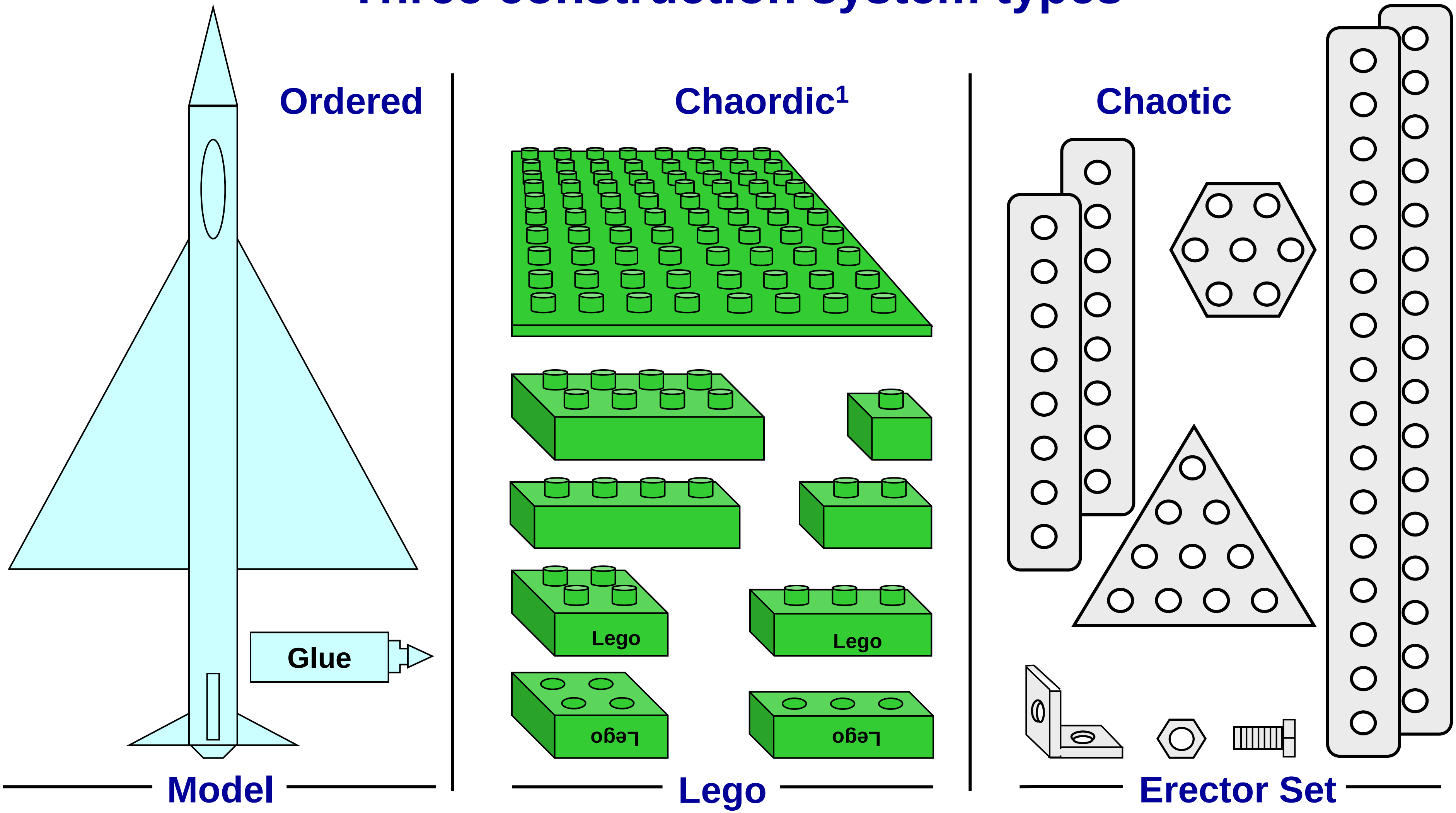
Marion Designs
8-1/2 Restoration

Lego Toy



Nathan Sawaya, http://www.brickartist.com/

Frameworks and Modules
Three construction system types



[1] Dee Hock (Visa Corp) coined the word *chaord* for organisms, organizations, and systems which harmoniously exhibit characteristics of both order and chaos.

Framework & Modules

Erector Set

Features	Effects (plus/minus values)
• open system • tools needed • plan needed • more degrees of freedom • safety is a concern • more difficult to take apart • some training required • robust construction	• ?

Lego

Features	Effects (plus/minus values)
• closed system • no tools needed • plan emerges • safe • colorful • no brain required • multi-level interest/functionality	• constrained to use intended modules • ?

Response Able System Principles – RRS

Reusable	
Self-Contained Units (Modules) Components are distinct, separable, loosely-coupled, self-sufficient units cooperating toward a shared common purpose.	**Evolving Standards (Framework)** - Frameworks standardize inter-component communication and interaction; define component compatibility; and are monitored/updated to accommodate old, current, and new components.
Plug Compatibility (Facilitated Interfacing) Components share defined interaction and interface standards; and are easily inserted or removed.	**Redundancy and Diversity** Duplicate components provide capacity right-sizing options and fail-soft tolerance; diversity among similar components employing different methods is exploited.
Facilitated Reuse Components are reusable and replicable; and responsibilities are specifically designated for inventory management, module maintenance, and upgrade of module inventory.	**Elastic Capacity** Component populations in *response able* systems may be increased and decreased widely within the existing framework.

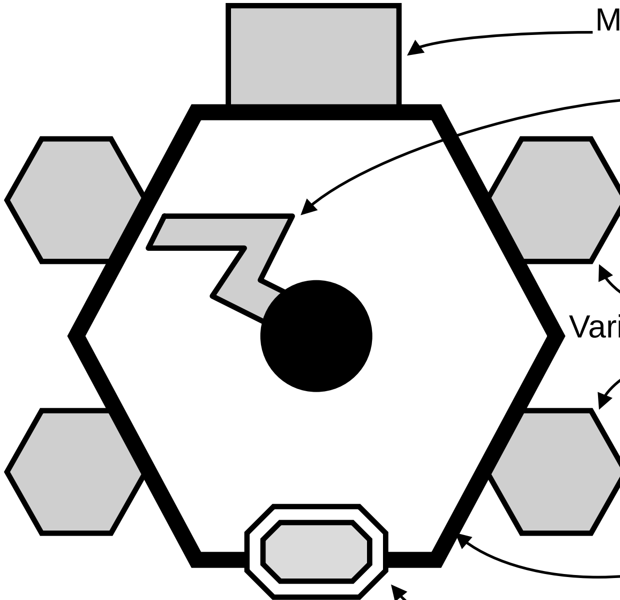
Scalable

Reconfigurable

Peer-Peer Interaction Components communicate directly on a peer-to-peer relationship; and parallel rather than sequential relationships are favored.	**Distributed Control and Information** Components are directed by objective rather than method; decisions are made at point of maximum knowledge; information is associated locally, accessible globally, and freely disseminated.
Deferred Commitment Component relationships are transient when possible; decisions and fixed bindings are postponed until immediately necessary; and relationships are scheduled and bound in real-time.	**Self-Organization** Component relationships are self-determined; and component interaction is self-adjusting or negotiated.

Cluster Machine

Depiction of *Precision 5000* Family from Applied Materials Inc.



Reusable

- ❑ Material interfaces, transfer robots, process modules, utility bases, docking modules, and user controls are independent units.
- ❑ Common human, mechanical, electrical, gas, and hydraulic framework.
- ❑ A growing variety of processing modules may be mixed or matched within a cluster.

Reconfigurable

- ❑ Wafer path determined in real-time by availability of appropriate process modules.
- ❑ New process modules may be added when new capability is required, and not before.
- ❑ Clusters may begin as 4 sequential processes and evolve to a single 4-unit process as product demand grows.
- ❑ Process-specific control is contained within the process module, traveling with it when redeployed.
- ❑ User control modules are custom configurable for proprietary processing.

Scalable

- ❑ Within a cluster 1 to 4 process modules may be installed.
- ❑ Clusters may be interconnected into larger super-clusters using docking modules in place of process modules.
- ❑ Clusters and super-clusters can be interconnected without limit.

Response Ability

- ❑ Test & Introduce new process modules incrementally.
- ❑ Custom process individual wafers and prototype runs.
- ❑ Repair/replace faulty module while cluster operates.
- ❑ Add modules and machine clusters as/when needed.
- ❑ Reconfigure clusters and redeploy process modules as product-line demand cycle changes.
- ❑ Create super-clusters as contaminant sensitivity requires.

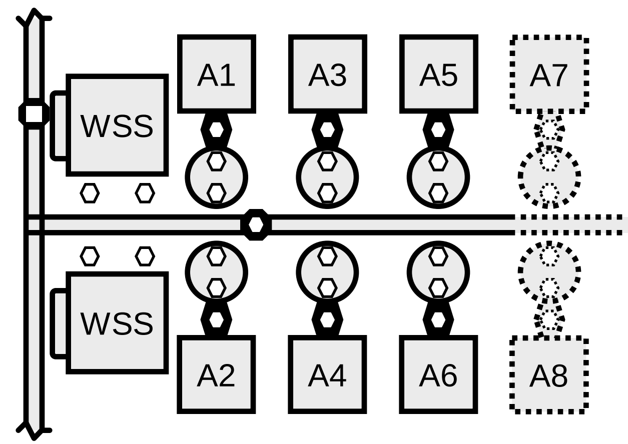
Cluster Machine

(*Responsibility – The Language, Structure and culture of Agile Enterprise*, Wiley 2001 – chapters 2 and 6)

<table>
<tr><td>Self-Contained Units (Modules)
Material interfaces, transfer robots, process modules, utility bases, docking modules, and user controls are independent units.</td><td rowspan="3">Reusable</td><td rowspan="3">Scalable</td><td>Evolving Standards (Framework)
Standardization focused on individual module interconnect only: mechanical coupling, communication protocols, and utility connections.</td></tr>
<tr><td>Plug Compatibility (Facilitated Interfacing)
Common human, mechanical, electrical, gas, vacuum, hydraulic, and control system interfaces.</td><td>Redundancy and Diversity
Machine utility bases are all identical, duplicate processing chambers can be mounted on same base or different bases.</td></tr>
<tr><td>Facilitated Reuse - Processing modules may be mixed or matched within a cluster. Machine manufacturer extends/replicates process module family. Customer manages reuse of all modules.</td><td>Elastic Capacity - 1-4 process modules per cluster. Docking modules can interconnect clusters into super-clusters. Transport bay can interconnect clusters and super-clusters without limit.</td></tr>
<tr><td colspan="4">Reconfigurable</td></tr>
<tr><td colspan="2">Flat Interaction
Scheduler in one base unit may access process history data for a process module on another base - perhaps to correct for a wafer’s prior process steps.</td><td colspan="2">Distributed Control and Information
Process history and tight-loop control located in process module, traveling with it when redeployed. Cluster controller manages macro-process and material transfer.</td></tr>
<tr><td colspan="2">Deferred Commitment - Process modules custom configured when installed. New process modules added when new capability required. User control modules are custom configurable for proprietary processing.</td><td colspan="2">Self-Organization
Wafer path within cluster determined in real-time according to the availability of appropriate process modules.</td></tr>
</table>

Production Cell

(*Responsibility – The Language, Structure and culture of Agile Enterprise*, Wiley 2001 – chapters 2 and 6)



Concept Based on LeBlond Makino A55 Cells at Kelsey-Hayes

Reusable

- Machines, work setting stations, pallet changers, fixtures are all standard, independent units.
- Common human, mechanical, electrical, and coolant framework.
- Machines do not require excavated pits or special foundations, and are relatively light and easy to move from one cell to another.

Reconfigurable

- Cell control dynamically changes work routing as machines are removed or added, on the fly.
- Autonomous part machining, non-sequential.
- Machines and material scheduled by cell control software in real time per current cell status.
- Part programs downloaded when needed.
- Machine's history stays with its controller.
- Machines ask for appropriate work when ready.

Scalable

- Cell may have any number of machines and up to four work setting stations.
- Cells may have multiple unit instances in operation.
- Machines capable of duplicate work functionality.
- Utility services and vehicle tracks can be extended without restrictions imposed by the cell or its units.

Response Ability

- Install and set up a new cell in 4-8 weeks.
- Reconfigure a cell for entirely new part in 1-4 weeks.
- Duplicate cell functionality in another cell in 1-2 days.
- Add/calibrate machine in 1-2 days while cell operates.
- Remove or service machine without cell disruption.
- JIT part program download.
- Insert prototypes seamlessly.

Production Cell

Reusable	
Self-Contained Units (Modules) Flexible machines, guided vehicles, rail sections, work-setting stations, loader/unloaders, pallet changers	**Evolving Standards (Framework)** – General manager responsible for component commonality, and interconnect standards for mechanical coupling, communication protocols, and utility connections.
Plug Compatibility (Facilitated Interfacing) Common human, mechanical, electrical, and coolant system interfaces. Common inter-module mechanical interfaces.	**Unit Redundancy and Diversity** Cells have multiples of each component, all cells made from same types of components, machines have full work functionality.
Facilitated Reuse - Machines do not require pits or special foundations, and are easy to move. Account mgrs with P&L responsibility add/subtract resources as needed. Ops manager maintains resource pool.	**Elastic Capacity -** Cell can accommodate any number of machines limited only by physical space for rail extension. A part can be made in multiple cells. One cell can make multiple parts.

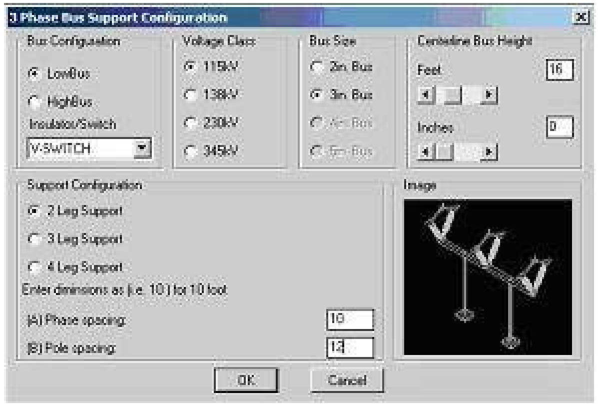
Scalable

Reconfigurable

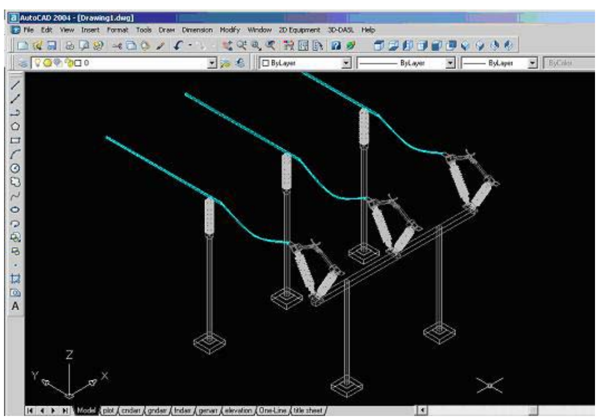
Flat Interaction Complete autonomous part machining, direct machine-repository program download negotiation.	**Distributed Control and Information** Part programs downloaded to machines, machine history kept in machine controller and accompanies machine as it changes location, machines ask for work when ready.
Deferred Commitment Machines and material scheduled in real-time, downloaded part programs serve individual work requirements.	**Self-Organization** Cell control software dynamically changes work routing for status changes and for new, removed, or down machines on the fly.

Substation Designs in 6 Hours
(normally 6 months)

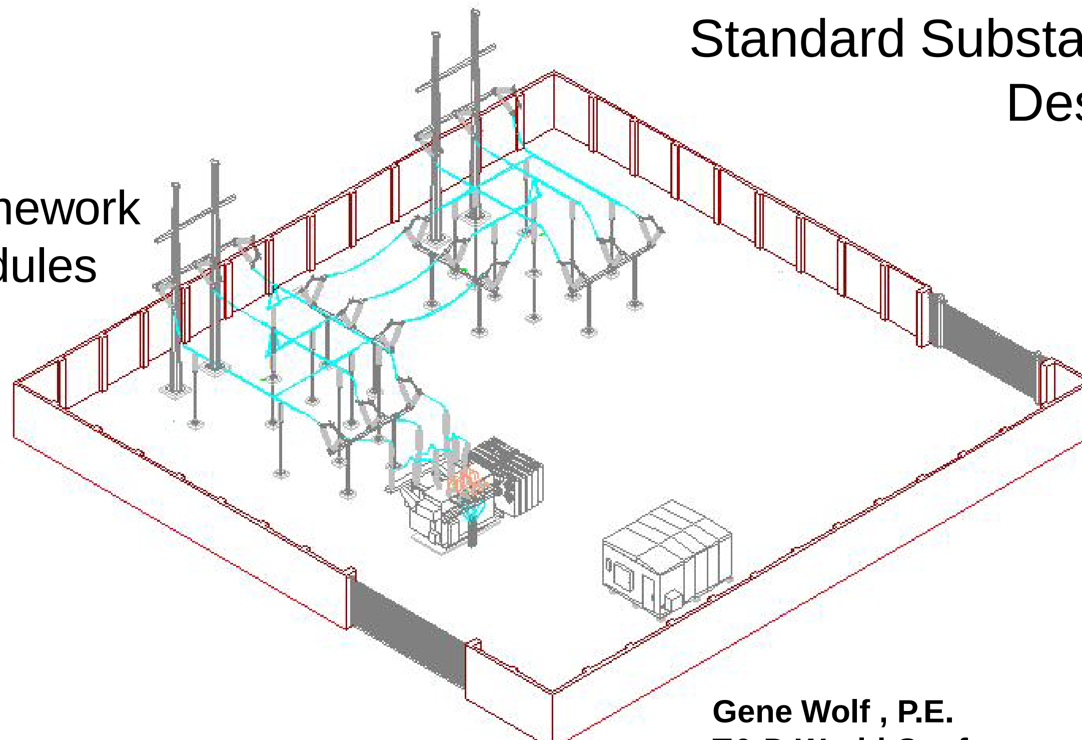
File



DASL provides common framework and common equipment modules



PNM's Second Standard Substation Design



Gene Wolf , P.E.
T& D World Conference, 2004

Details: **www.tdworld.com/mag/power_pointandclick_substation_matures/index.html**

1- Proposed Site

58 Days
from Signing of Contract to Energization of El Cerro Substation

Usually 12-18 months

Gene Wolf , P.E., PNM, T& D World Conference, 2004

2- Superimposed Computer Graphic

3- Completed Project

Self Contained Units (Encapsulated Modularity)

(PNM Substation - www.parshift.com/Files/Essays/Essay069.pdf)

Encapsulated modularity shares most-important-factor status with frameworks. These two principles alone provide basic agility. Without both, *effective* agility is doubtful.

- **PNM's prime module types include engineers, transformers, switchgear, transmission termination structures, low-voltage feeder circuits, and station steel. In each module type there are generally a few varieties, allowing configurations customized to a particular substation need.**

- **Transformer specification is what determines substation delivery capability. PNM found three varieties to be sufficient: 16, 22, and 33MVA. Limiting transformer types to a minimal three reduces spares inventory requirements while increasing the likelihood of a necessary spare on-hand.**

- **The *encapsulated* requirement for modules requires that they be functionally self-sufficient to meet their objective, and that the methods employed for meeting objectives are of no concern to the greater system. In the case of transformers, should technology evolve, a superior performing version may be substituted without unintended consequences from integration.**

Evolving Standards (Frameworks)

(PNM Substation - www.parshift.com/Files/Essays/Essay069.pdf)

- **PNM standardized a sub-station architecture that accommodates almost all needs. This provides the framework for reconfiguration, and includes an embedded infrastructure of conduits, standard conduit physical interfaces, specified space limits for equipment, and standardized concrete pads that can accommodate all transformer and switchgear options.**
- **Important for any agility framework are two deeper principles, in purposeful tension: *requisite variety* insists that a framework have standards for everything *necessary*, and *parsimony* insists that a framework not have any *unnecessary* standards. One too many will decrease agility. One too few pushes toward chaos.**
- **The nature of the framework both enables and limits agility. Maintaining and improving agility relies on managing framework evolution ... prudently. PNM's substation framework evolved through T, H and fly-through variations. Prudence in this evolution maintained conduit interface standards, important for continued *module reuse*; but added new module options for transmission input configurations and feeder output configurations. The third "fly-through" version changed the perimeter configuration to fit within a transmission line right-of-way; reducing difficulties with acquiring land and permits. Prudent evolution did not impact the *plug-compatibility* of existing equipment modules.**

Facilitated Plug Compatibility

(PNM Substation - www.parshift.com/Files/Essays/Essay069.pdf)

Plug compatibility simply means that modules can be plugged into the framework infrastructure—with no modification to anything: a standardized plug/socket wiring interface specification, and a standardized pad installation mechanical interface regardless of transformer size.

Facilitated **is the operable word, and means the utilization of plug compatibility is** ***natural*** **and** ***readily/easily/simply accomplished*****, and that responsibility for conformance to and evolution of the infrastructure standards is designated.**

- **PNM has provided an invariant standard interface spec to the transformer manufacture, and the manufacture delivers a plug compatible unit.**
- **Regardless of power ratings, hook-up interfaces are all identically located and identically specified, ready to mate with the concrete-pad infrastructure and compatible with standardized equipment space allowance.**
- **No deviation from or changes to standards are permitted w/o the express authorization of the chief engineer.**

Facilitated Reuse

(PNM Substation - www.parshift.com/Files/Essays/Essay069.pdf)

Reusability of modules is a paramount advantage of agile systems – but *facilitated* is the operable word.

***Basic* reuse-facilitation comes from *plug compatibility* and *encapsulated modularity*. Beyond that is the need to facilitate acquisition, configuration and assembly by ensuring that modules are both *naturally and readily reusable* and *ready for reuse*.**

Note that design has become a configuration and assembly activity, rather than a custom and expert design-from-scratch activity with attendant human-error risk.

- **PNM developed a custom AutoCAD-extension solution (3D-DASL) as their substation design tool—facilitating ready reuse with added built in menus for quick drag-and-drop placement of stored pre-drawn modules, pre-drawn standard layouts as frameworks, and built-in configuration restrictions that ensure the chosen modules are compatible with the power requirements.**
- **3D-DASL is structured to enforce framework and module standards; reducing the design time from six months to six hours—while reducing risk by eliminating vulnerabilities.**
- **Ensuring that modules are ready for reuse is important in construction and operational activities after design is done. This is accomplished with processes and responsibilities that enable timely acquisition of modules, and ensures module inventory is sufficient and maintained in a state of readiness.**

Redundancy and Diversity

(PNM Substation - www.parshift.com/Files/Essays/Essay069.pdf)

Module redundancy means identical proven units are available for reuse—with no surprises or unintended consequences.

Module diversity means there are variations within a given module type—offering configuration options for custom needs.

- **Rather than increasing capacity with a custom designed higher-power transformer, two standard modules can increase power delivery capacity without the risks of new design and first-time equipment. The three-variety transformer diversity also provides the ability to mix any variety for efficiently achieving the capacity needed.**
- **The greater substation process includes people as working modules, particularly in design engineering. Here we see the natural diversity among engineers being leveraged—less experience and less training is required, making a broader pool of capable engineers available when peak needs or retirements require new or additional resources.**
- **Redundancy also plays a key role in minimizing inventory costs, while maximizing inventory effectiveness and reducing the risk of prolonged power outage.**

Elastic Capacity

(PNM Substation - www.parshift.com/Files/Essays/Essay069.pdf)

Effective capacity-demand response is often a prime driver for agile process development, and rears its ugly head when demand falls outside planned expectations. Fixed costs and capital investments often make downsizing uneconomical, while on the flip side, added capability can't be built fast enough.

- **PNM has effective options to accommodate unexpected capacity demand. If demand does not materialize as expected, they can easily replace a larger transformer with a smaller one, and redeploy the larger one where it is more economic.**

- **For increased demand they can upgrade the transformer, add an additional transformer, or even add a duplicate substation relatively quickly.**

- **On the peopled-side of the equation, peak design demands can employ additional engineers easily. And since the design engineering time has been reduced so dramatically, existing engineers already spend the bulk of their time in other engineering activities—a reduced substation design-load is barely noticeable.**

Flat (Non-Hierarchical) Interaction

(PNM Substation - www.parshift.com/Files/Essays/Essay069.pdf)

Seeking approvals and sign-offs, and filtering communications through hierarchical silo managers, is both time consuming and knowledge reducing.

- **The alliance with PNM's transformer manufacturer encourages direct engineer-to-engineer collaboration, eliminating the prior purchasing dept knowledge-filtering communication channel.**
- **Standardized ordering and standardized design eliminates both internal and external time-consuming approval cycles and review sign-offs.**
- **Risks of miscommunication, inadequate communication, altered communication, and protracted approval cycles are eliminated.**

Facilitated Deferred Commitment

(PNM Substation - www.parshift.com/Files/Essays/Essay069.pdf)

In order to avoid rework and waste when a situation changes mid-course, this principle insists on just-in-time decision making, and "system designed" facilitation of both decision deferment and decision-implementation time reduction.

- **PNM's reduction of design time from six months to six hours considerably reduces implementation time and postpones the need for procurement and construction commitments, reducing economic risk in the process.**
- **Module standardization permits construction to proceed with spares inventory before replacement modules are received.**
- **PNM negotiated a collaborative alliance with a single transformer and switchgear manufacturer, which facilitated a shortened procurement cycle by eliminating bid procedures, and facilitated a shortened manufacturing cycle by ordering units identical to previous ones. Orders for new transformers do not have to be placed a long time in advance of projected needs that may not materialize.**

Distributed Control and Information

(PNM Substation - www.parshift.com/Files/Essays/Essay069.pdf)

One of the three cornerstones of agility is knowledge management, another is decision-making support. These rely on information and decision control being in the right place at the right time.

Effective decisions are made at the point of most knowledge. The most knowledge is available at the point of knowledge application and feedback learning.

- **PNM's transformer and switchgear manufacturer has the most knowledge about unit cost and performance options, and is expected and empowered by PNM to employ what they know to provide the best components to achieve objectives.**

Self-Organization

(PNM Substation - www.parshift.com/Files/Essays/Essay069.pdf)

Self organization is an advanced principle employing modules that can make decisions and change the nature of their relationships with other modules by themselves. Two cases at PNM:

- **Active trust development -- Trust is a self-organizing driver in relationships. Trust develops or deteriorates as parties interact and as the parties in a relationship change. A permit agency scrutinizes plans with a healthy degree of skepticism, with people who are spread thin with other priorities. As trust grows, agency relationships evolve and self organize to accelerate successive permitting activity. Facilitated by: Standard plans that have been approved in the past, delivering finished construction consistent with approved plans, reinforcing trust development with post-construction meetings that show plans and promises that match finished results.**

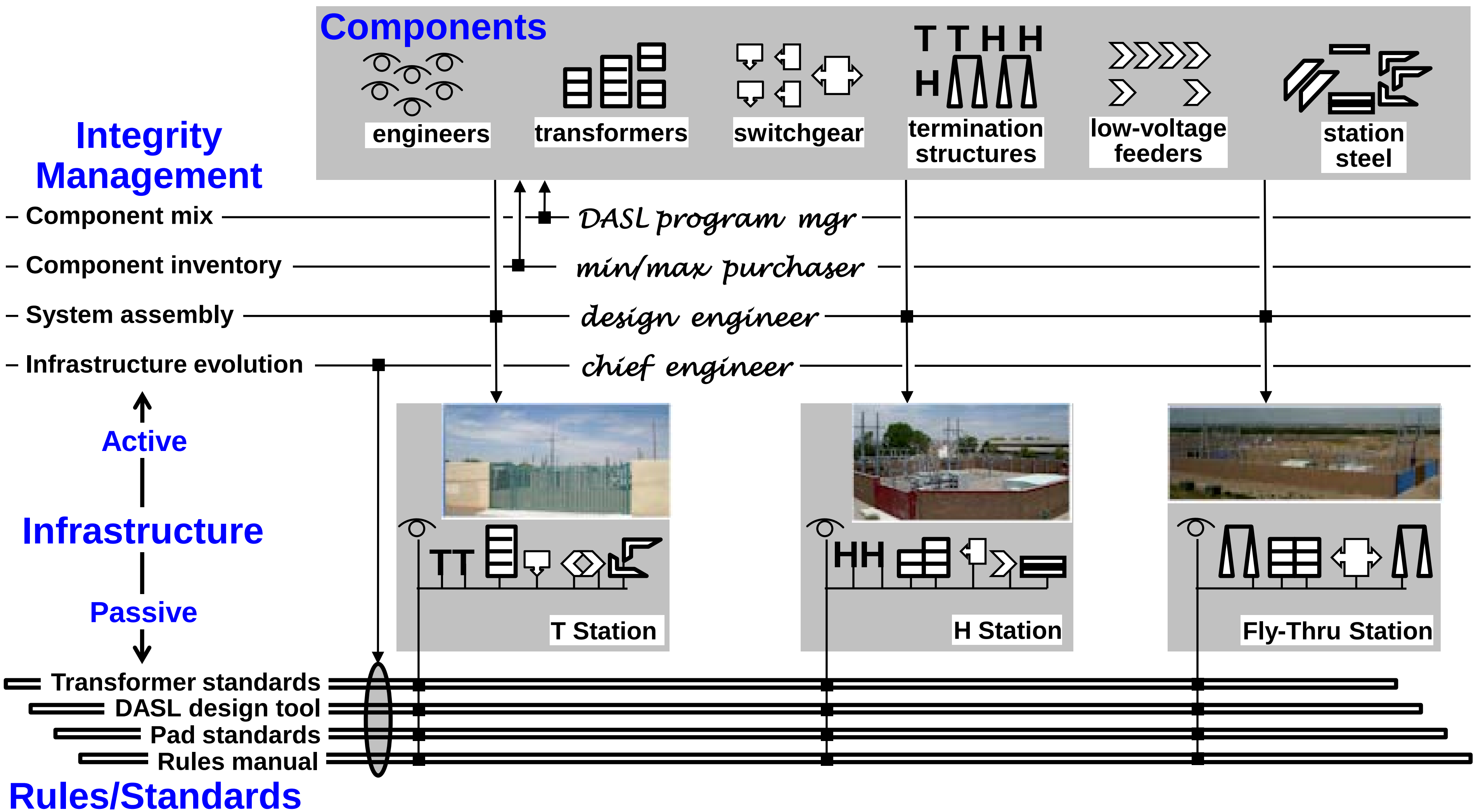
- **Collaborative improvement -- PNM's process is being tested at Long Island Power Authority and at Kansas City Power and Light, (December 2004). PNM's purpose for broadened usage is to develop a community of users, with new and diverse needs, that will collaborate in a self-organizing fashion toward improved functionality.**

Note: This set of RRS slides mixes elements from three systems: design, construction, and operation. Not generally a good practice. Done here for instructive RRS exposure.

PNM Agile Substation System

www.parshift.com/Files/PsiDocs/Pap080404Cser2008DevOpsMigration.pdf

Architectural Concept Diagram



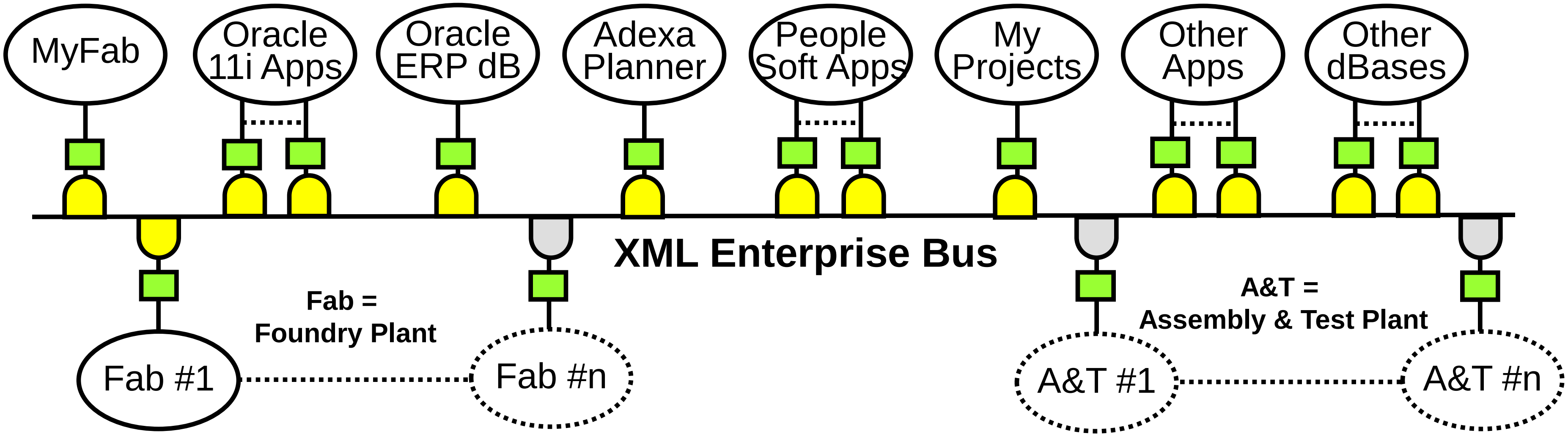
For PNM – Agility Costs Less

The PNM case study demonstrates that agility can reduce bottom-line costs while reducing response-sufficiency risk and response-predictability vulnerability.

Reengineering existing processes and systems for agility does incur some costs, but a far greater cost is incurred with an inefficient and poorly-responsive status quo.

When migration toward more agile processes is done incrementally and knowledgeably, extreme ROI can be realized, with short-term bottom-line effect.

Silterra IT Infrastructure Design



- = Bus Interface Module (BIM)
- = ETL Interface Modules
- MyProjects = Web-accessible strategic-project portfolio manager
- MyFab = Web-accessible operations transparency

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

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**

Response Requirements – IT Infrastructure

Response Metrics: c=cost, t=time, q=quality, s=scope

Proactive Dynamics

- Creating new customer/supplier/partner business net-link [t,q,s]
- Creating acquisition business net-link [t,q,s]
- Creating interface to a new application [t,c,s]
- Improvement of interface performance [t,s]
- Migration to NT and COM/DCOM [c,q]
- Addition of new foundry facility [q,s]
- Addition of new customer/supplier/partner data interface [t,s]
- Addition of new industry data-standards [t,s]
- Replacing the bus vendor [c,t,s]

Reactive Dynamics

- Correcting an interface bug that surfaces later in time (original engineer gone) [t,q]
- Variation in quality of data from production MES system [t]
- Variation in competency/availability of infrastructure operating personnel [t,s]
- Variation in real-time on-line availability of applications [t,s].
- Expand the number of interfaced applications and business net-links [s]
- Reconfiguration of an interface for an application upgrade/change [t,c,q,s]

RRS Principles Applied for Silterra Enterprise IT

Evolving Standards (Framework) - SSA group, XML protocol, message data definitions, ETL-interface specs, ETL template spec, BMI spec.

Encapsulated Modules - Applications, data bases, ETL table-driven templates, bus-interface modules (BIMs), BSAs, SSAs.

Facilitated Plug Compatibility - XML, message-data definitions, BIM spec, ETL-interface spec, rule on COTS.

Facilitated Module Reuse - BSA group, business process maps, ETL templates, mandatory rule on COTS.

Redundancy and Diversity - Multiple app versions, multiple bus paths, replicated apps at each physical locations, ERP multiple-vendor apps, rule on mandatory user collaboration, cross-trained BSA departmental responsibilities.

Elastic Capacity - Virtually unlimited bus extension and capacity with compartmented parallelism.

Distributed Control and Information - Separate apps and data bases at each physical location, BSA independence and team collaboration, SSA/BSA separation, rule on mandatory user collaboration.

Facilitated Deferred Commitment - Publish subscribe asynchronicity, ETL created after app is stable, rule that response-requirements be developed before solutions considered.

Flat Interaction - Direct app-to-app dialog, BSA group user/management access and team collaboration.

Self-Organization - BSA autonomy, BSA teaming, SSA autonomous control, publish-subscribe options to pull information as needed.

ETL=extract/transform/load, BSA=business systems analyst, SSA=strategic systems analyst, BIM=bus interface module, COTS=common off the shelf.

Key Points

- ETL homegrown as reusable framework template to reduce the level of expertise and time required for new-application ETL development
- BIM homegrown in order to isolate the bus as an encapsulated module that could be replaced if necessary in the future
- Oracle Apps – Initially implemented with Oracle's direct inter-application communications as API documentation not available – transition to encapsulated apps with API/ETL/BIM interface later
- PeopleSoft Apps – encapsulated right off

Implementation Process – 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

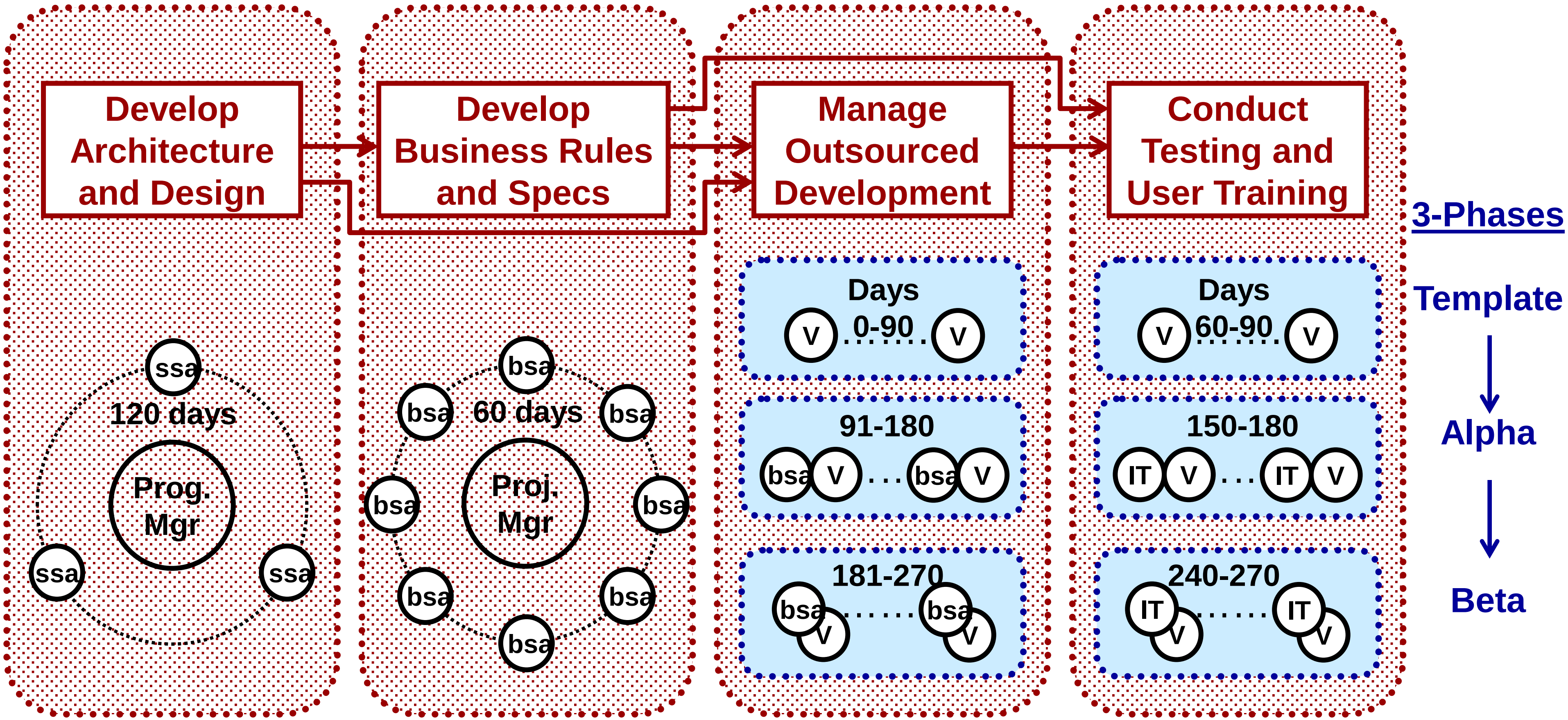
  P2: BSA-developed business process rules
  Vendors + BSAs configure the applications

  P3: Refined 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)

Encapsulated Implementation Process Modules
(text book chapter 8 for details)



- Designed to Accommodate Requirements Evolution -

Also see paper at www.parshift.com/Files/PsiDocs/Rkd050324CserPaper.pdf

RRS Principles Applied for Implementation Process

Evolving Standards (Framework) – 3-phase implementation (out-of-box, desired, refined), 90-day phases max, no spec/requirement changes once phase begins, internal total infrastructure design responsibility, vendor total application responsibility (HW/SW).

Encapsulated Modules – Bus vendor (BEA), ERP app vendors (Oracle, PeopleSoft, Adexa), database vendor (Oracle), app requirements developers (BSAs), infrastructure requirements developers (SSAs), infrastructure implementers (IT).

Facilitated Plug Compatibility – vendor rules clear, agreed in advance, and managed.

Facilitated Module Reuse - BSA group, business process development system.

Redundancy and Diversity - Cross-trained BSA dept responsibilities, mixed outsource/insource resources and expertise.

Elastic Capacity – Outsource implementers managed by small internal group.

Distributed Control and Information - BSA business rule development autonomy, SSA infrastructure rules/design autonomy, vendor implementation autonomy.

Facilitated Deferred Commitment – Implementation doesn't begin until requirements are firm.

Flat Interaction – All vendors are peers, BSAs have direct access to everyone.

Self-Organization - BSA team relationships and assignments.

ERP=enterprise resource planning, BSA=business systems analyst, SSA=strategic systems analyst, HW/SW=hardware/software

Effective Response

- Bus vendor team (Australian to USA switch)
- ERP vendor team (USA to Malaysian switch)
- Planner Choice (Oracle to Adexa)
- Added Planner system
- Added Time and Accounting system
- Added HRM system
- ETL design evolution
- CIM integration (major data integrity problems)
- MyFab (operational transparency) integration
- Unstable company ($1.5 Billion massive start-up scramble)
- Unstable ERP (new, buggy, undocumented)
- Undefinable business processes (inexperienced company staff/mgmnt)
- Under experienced IT staff (Malaysian resource inadequacy)

Employment of Principles...

Forces consideration of each principle: better design-for-agility

Values: increases scope of response options,
reduces future cost and time

Defines clear framework: integration rules don't change

Values: increases predictability of project,
reduces current cost and time

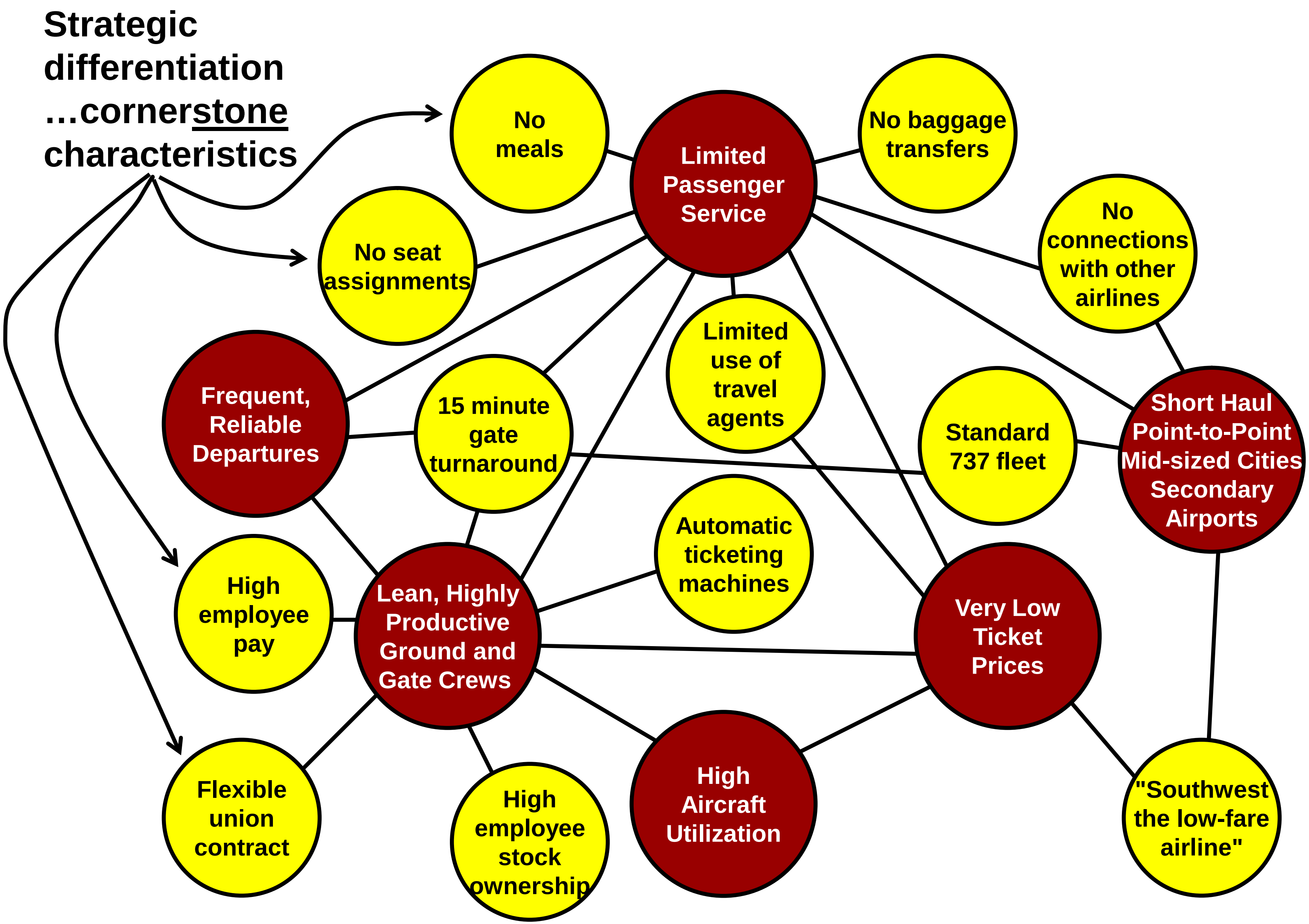
Defines encapsulated modules: requirements don't change

Values: increased predictability of project,
increased options for alternatives,
reduces current cost and time

www.datacenterknowledge.com/inside-the-box-container-video-tours/
www.datacenterknowledge.com/archives/2010/08/11/the-blackbox-lives-or-at-least-is-not-dead/
www.zdnet.com/blog/datacenter/suns-datacenter-container-forgotten-but-not-gone/398

file

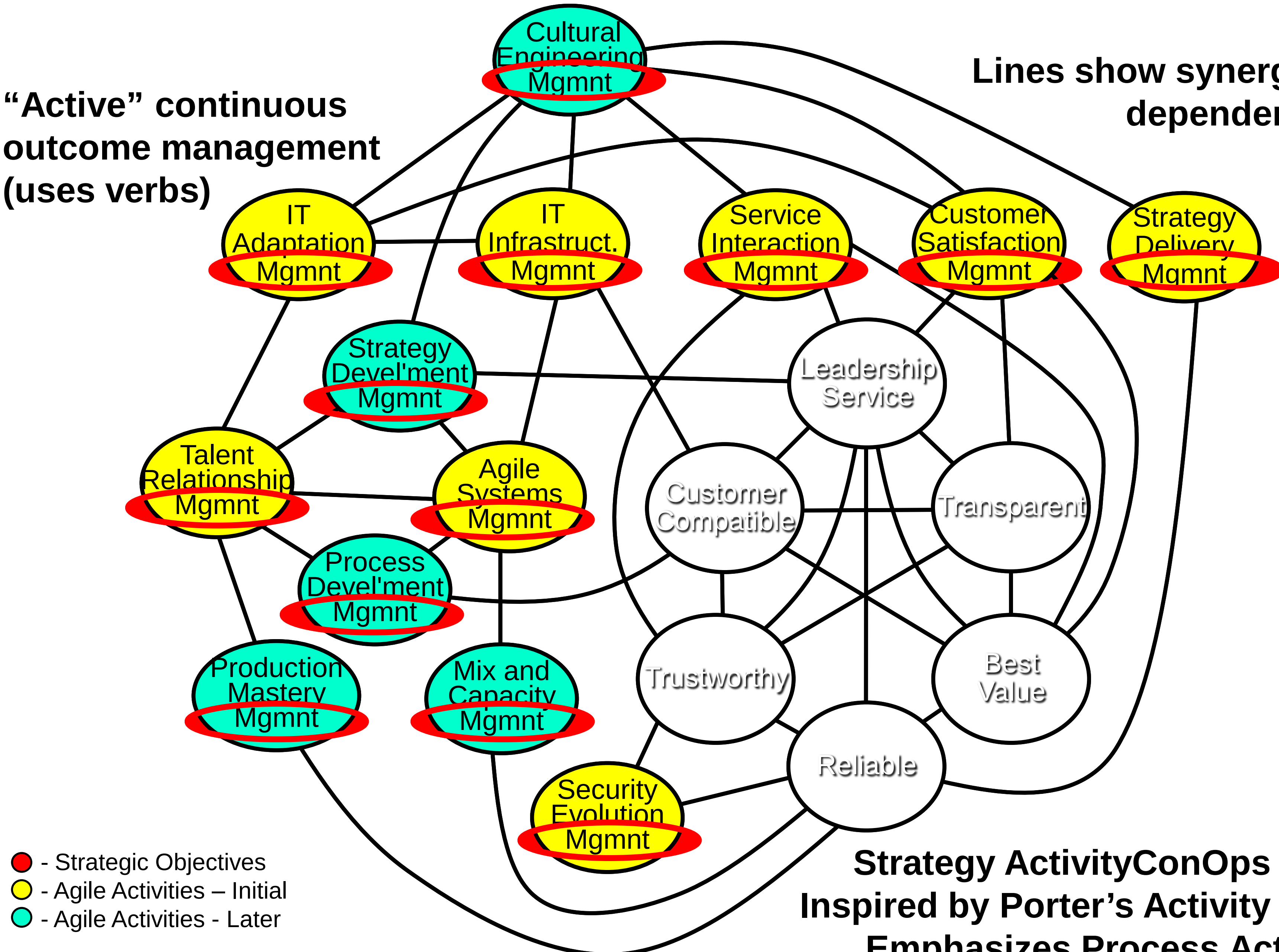
Porter on Strategy



Semiconductor Foundry

Lines show synergistic dependencies

"Active" continuous outcome management (uses verbs)



Strategy ActivityConOps Web
Inspired by Porter's Activity Web
Emphasizes Process Activity

On the Strategic Activity ConOps Web

This web of synergistic activities should depict the desired <u>value-based public reputation</u>, and the activities that cause that reputation.

It is the architecture of the agility ConOps.

Reputation objectives (red): a few, 3-7, or focus is lost.

Activities (yellow): these are continuous day-in-and-day-out processes that ensure the objectives are realized. They are not things or concepts. Again, keep the number smallish or the critical activities get lost in the noise.

The few words used to label a red or yellow bubble are critical – they must capture and focus the essence of intent succinctly.

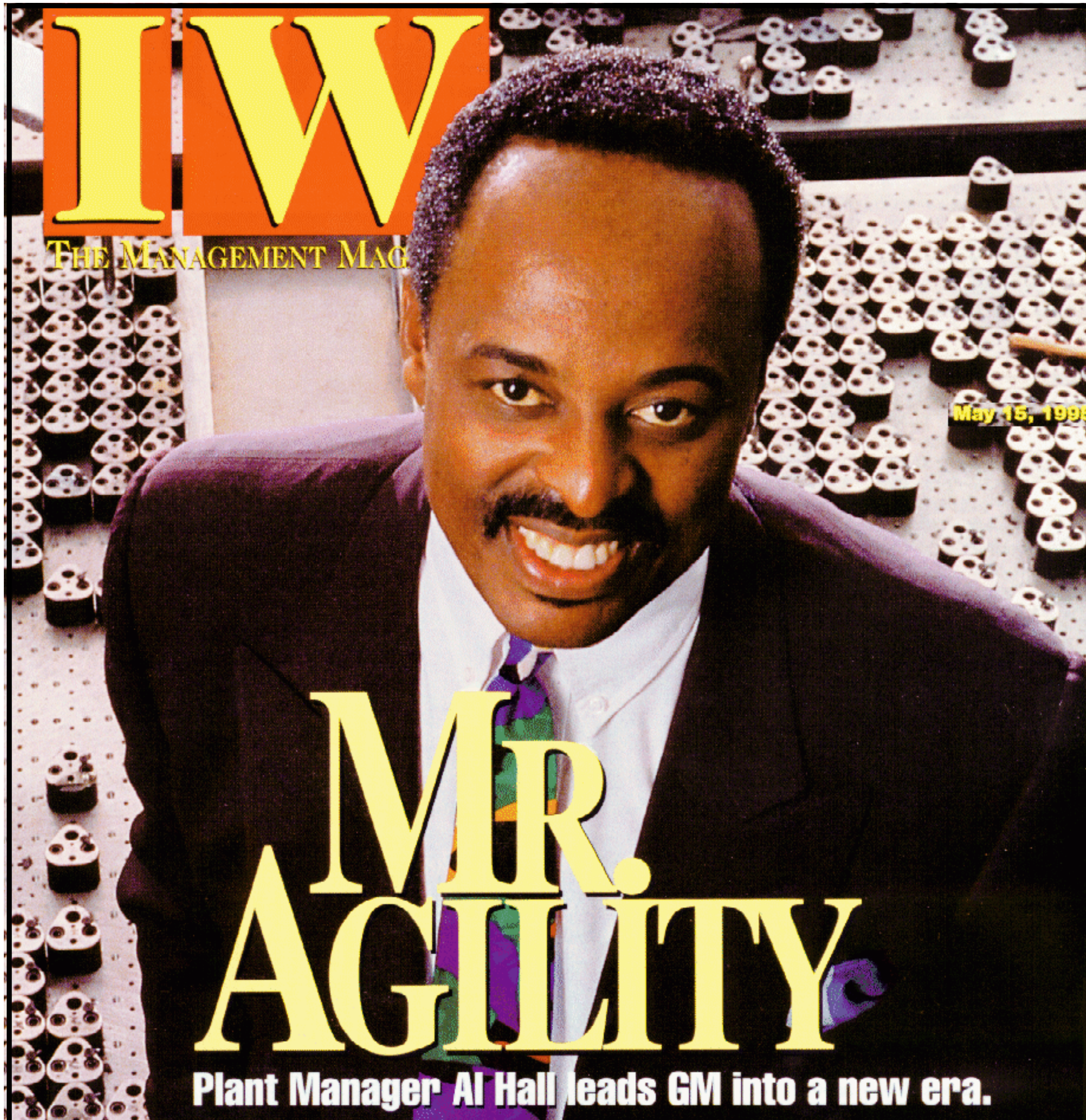
Synergistic Dependencies: more is (often) better - multiple lines attached to every bubble – this provides robustness. And, according to Porter, makes it a lot harder for any competitor to duplicate.

Note that this is not an agile architecture if Porter's advice is taken.

Porter encourages dependencies and tight coupling as ways to make competitor duplication difficult – providing a meaningful *strategy*.

Not a good idea if the ConOps values (environment) evolve faster than the ConOps activities (system) can.

So … carefully choose timeless values (102); and design activity-relationships that are synergistic but not tightly dependent (103).



Agility Workshop

General Motors
West Mifflin, PA

Situation:

Many highly-agile unique processes and practices in this low-volume high-variety production environment.

All were the design efforts of a few "naturals", who cannot articulate to others how to continue this necessary practice after they retire.

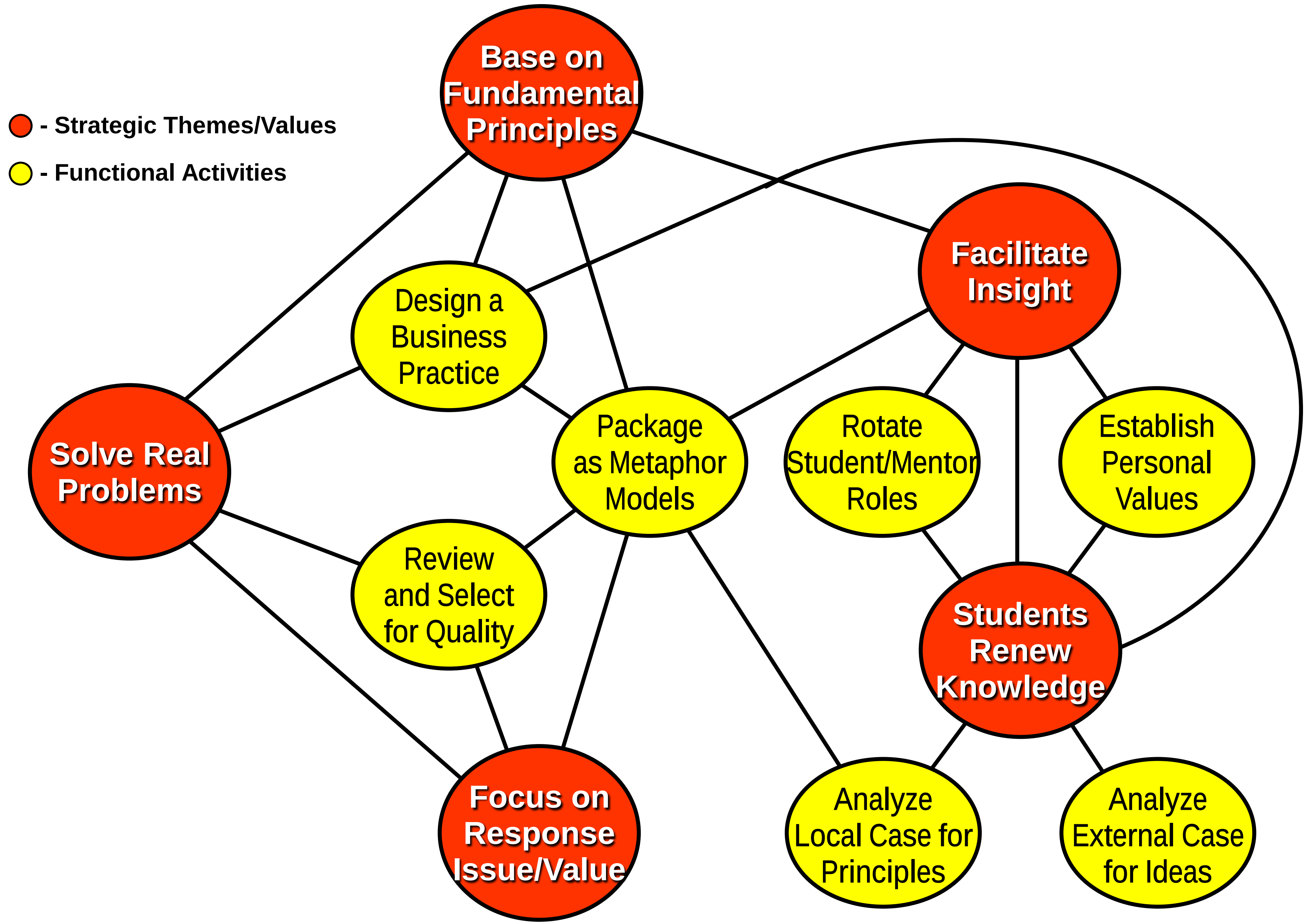
General Motors Workshop

- **Analyzed: The JIT Assembly Line process.**
- **Analyzed: The Pittsburgh Universal Holding Device.**

The workshop group included about 10 management and executive level participants from the plant. First they analyzed the two things above that they were very familiar with and respected highly – looking through the lens of the 10 RRS principles to identify how these were employed to enable agility. Then they were guided through an exercise that applied these principles to the design of a new process.

- **Exercise: Core Competency Insight Diffusion…one of the most important problems facing all companies today: how to make good intuitive knowledge in a few employee-heads in one part of the company explicit, so that it can be taught to new employees and taken to other parts of the company.**

Core Competency Diffusion Strategy



Core Competency Insight Diffusion

Change Proficiency

Key Proactive Issues:

Creation:
- Tacit Knowledge Capture
- Student Interest/Value

Improvement:
- Knowledge Accuracy
- Knowledge Effectiveness

Migration:
- Knowledge Focus

Addition/Subtraction:
- Student Types
- Fresh Outside Knowledge

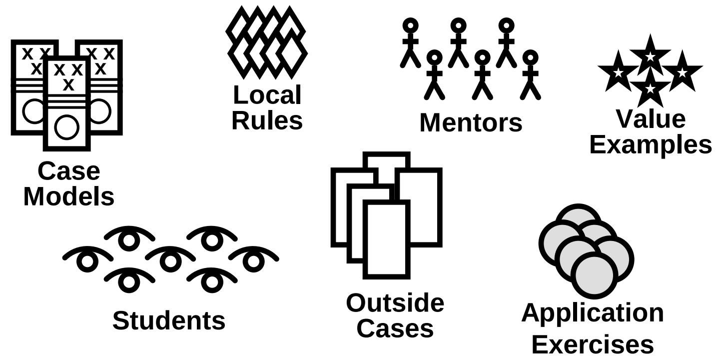
Key Reactive Issues:

Correction:
- Incorrect Knowledge
- Poor Value Knowledge

Variation:
- Flexible Student Schedule

Expansion:
- Any Size Group

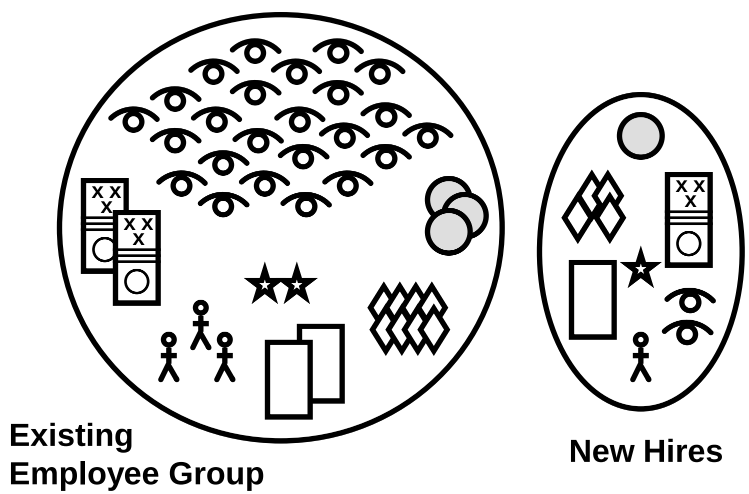
Reconfiguration:
- Rules«–»Applications



Operational/Integrity Management

Knowledge Mgnt Committee: Framework/Process.
HR/OD: Students, Mentors, IDG formation.
QA Committee: Rules/Problems/Models.
Students: Outside Cases, Value Examples.

Sample Insight Development Groups (IDGs)



Drag & Drop Modules:
- Mentors
- Students
- Case Models (Knowledge)
- Local Rules (Knowledge)
- Outside Cases (Knowledge)
- Application Exercises
- Personal Value Examples

Plug & Play Framework:
- Fundamental-Principle Based
- Solve Real Problem
- Students Renew Knowledge
- Change Issue/Value Focus
- Insight Facilitation

Closure Matrix – Where Deep Design Begins

Details: http://www.parshift.com/Essays/essay039.htm

(Case: An Insight Development System)

Activities (Functions)	
Establish personal values	1
Analyze external case for ideas	2
Analyze local case for principles	3
Design a business practice	4
Package as *response ability models*	5
Rotate student / mentor roles	6
Review and select for quality	7

RRS Principles — **Principle-Based Activities, and Issues Served**

	Issues (Requirements)		Self Contained Units	Plug Compatibility	Facilitated Re-Use	Flat Interaction	Deferred Commitment	Distributed Control & Info	Self Organization	Elastic Capacity	Redundancy & Diversity	Evolving Standards
Proactive	Capturing hidden tacit knowledge	3567	35	356	57	3	37	6	3		3	37
Proactive	Creating student interest and value	124	1	1	1	12	124		124	1	1	
Proactive	Improving knowledge accuracy	367		6		3	37	6	3		3	7
Proactive	Improving knowledge effectiveness	1245	45	245	45	1			12	5	2	
Proactive	Migrating the knowledge focus	247	27	4	2		4	7	247		4	47
Proactive	Accommodating different student types	(all)	25	6			347	2	12345	1	17	2
Proactive	Injecting fresh outside knowledge	26	26	26		2		6	2			
Reactive	Finding and fixing incorrect knowledge	367	7		7	3	3	6	3		3	7
Reactive	Excising poor value knowledge	2357	7		7	3	3	2	23		35	257
Reactive	Allowing flexible student schedules	34	34			34			34			
Reactive	Accommodating any size group	2345	2345	234				2	25	34	234	
Reactive	Reinterpret rules for new applications	23457	27		5		2	357				23457

Creating Conceptual Design Closure

The closure tool is where design thought gets deep. Here the preliminary issues, principles, and activities are sifted for relevance and related for synergy.

The tool is first used to specify which activities will address which issues, and why; and to verify (in the mind of the designer) that the set of issues and the set of activities are necessary and sufficient. It is a time to step back from the preliminary, somewhat brainstormed, formulation of the problem and the solution-architecture, and do a sanity check before specifying design-principle employment. Not explored further here, Chapter 7 of the text book can assist.

The real work with the closure tool is generally on the employment and purpose of principles - the ones that would compromise potential if they are not employed as design elements. This we will explore further here.

Issue-Focused, Principle-Based Design - The General Process

1) Pick an activity, and describe its general process sequence.

2) Focusing on one issue: sequentially think if/how each of the ten principles might be employed by the activity to address the issue meaningfully. Then write a paragraph that describes the key principles and what they achieve.

3) Loop through all issues for item 2.

4) Loop through all activities for item 1.

The Activity

Analyze Local Case for Principles

(strong similarities to the GM workshop process that designed this activity)

This is the primary mechanism for capturing core-competency knowledge, and uses the students to analyze and describe the features and underlying principles of an existing highly adaptable system.

Typically the original designers of these existing systems employ techniques that they are unable to articulate to others sufficient for duplicating the expertise.

The purpose of this analysis is twofold: first, it turns tacit knowledge into explicit knowledge, and second, it is a warm-up exercise for the group which subsequently employs what they have learned to solve the workshop application problem.

Students choose the subject for analysis from candidates suggested by mentors. Mentors provide process guidance, aiming the group toward the eventual descriptive requirements for consistent knowledge representation.

1a) Describe the activity and …

Activity Process Sequence Described

(Analyze Local Case For Principles)

1. **Explain in presentation/tour the case under analysis.**
2. **Full group Q&A and discussion.**
3. **Breakout sub-groups *identify issues and values*.**
4. **Full group discussion on sub-group results.**
5. **Breakouts *build activity diagram* and *identify framework, modules, and system responsibilities*.**
6. **Full group discussion on sub-group results.**
7. **Breakouts build *closure matrix* with *RRS examples*.**
8. **Full group discussion on sub-group results.**
9. **Mentors lead consensus-making among sub-group differences where possible – as a transition into the next activity: *Metaphor Model Packaging*.**

1b) … its process sequence

Issue: Capture Hidden Tacit Knowledge

Employing the *flat interaction* principle we encourage the sub-groups to independently question and probe the people involved in designing or operating the system under analysis without restricting this to a full group discussion and Q&A activity.

Importantly, *deferred commitment* is at work by first examining issues and activities before identifying the underlying principles that are important - which tends to broaden the perspective while focusing it on priorities at the same time.

***Unit redundancy* is employed by purposely having multiple sub-groups go after the same analysis independently so that if one gets in a hole another will surly succeed.**

By the same token, we let these sub-groups exercise a high degree of *self-organization* as to how they will schedule their analysis activity, how they will interpret the principles, what libraried cases they will study for guidance, and how they will arrive at a *self-contained unit* conclusion - requiring no dependence on other sub-groups.

Of course their conclusion is going to be *plug compatible* with the full group because the analysis structure is a given: the metaphor model is the template.

This independent work by multiple groups will develop a broader and deeper set of alternative views, guard against single-view dogma, and generally make progress even if some of the people in the group are confused and lost.

Finally, *evolving standards* will modify our understandings of the principles and their usage, and the change issue/value focus to keep up with new learnings and perspectives.

2) Pick an issue, write a paragraph showing RRS-principle usage

Issue: Improving Knowledge Accuracy

Redundant **sub-groups and even duplicate analyses by whole groups refines the knowledge.**

Self organization **of the sub-groups and allowing direct** ***flat interaction*** **between teams and sources increases the likelihood that some teams will uncover knowledge overlooked by others who approach the process differently.**

As before, ***deferring*** **the close look at principles focuses the priorities; and allowing direct team/source interaction broadens the total perspective.**

Issue: Improving Knowledge Effectiveness

Chartering each sub-group as a ***self-contained unit*** **means that they must build a complete stand-alone analysis, and not split up the effort with another – meaning they will learn a full system with all its checks and balances and not simply a few odds and ends about something that appears to work.**

2-loop) Pick another issue, write a paragraph showing RRS-principle usage

Issue: Different Student types

The issue of *different student types* is accommodated by *deferring* the selection of the local case until the participant profile is known - and at the same time letting the group *self-decide* what the case shall be from among their own candidates as well as those offered by mentors.

Issue: Finding and Fixing Incorrect Knowledge
Issue: Excising Poor Value Knowledge

Though they are two distinct issues, *finding and fixing incorrect knowledge* and *excising poor value knowledge* are both achieved identically in our case here - and in a similar manner to *improving knowledge accuracy*.

***Redundant* sub-groups and even duplicate analyses by whole groups is bound to produce differing points of view and even expose a sacred cow now and then.**

***Self organization* of the sub-groups and *flat interaction* increases the likelihood that some teams will look at things differently than others.**

Finally, *deferring* the close look at principles until a sound set of issues and values is developed is likely to ferret out bad assumptions

2-loop) Pick another issue, write a paragraph showing RRS-principle usage

Issue: Flexible Student Schedules

The issue of *flexible student schedules* is enabled by *self-organizing* sub-groups that stand-alone as *self-contained* teams and are able to interact *peer-to-peer* (flat interaction) in their analysis work. Though there are some times when an entire workshop group must meet together, the bulk of the time consuming work is spread over weeks and can occur asynchronously

Issue: Accommodating Any Size Analysis Group

The issue of *accommodating any size analysis group*, from a few new hires to a large retraining class, relies on the *flexible capacity* afforded by splitting a total group into any number of sub-group teams, chartering these teams as independent *self-contained units* that work to a common *plug-compatible* process structure, and having them all work *redundantly* on the same objectives.

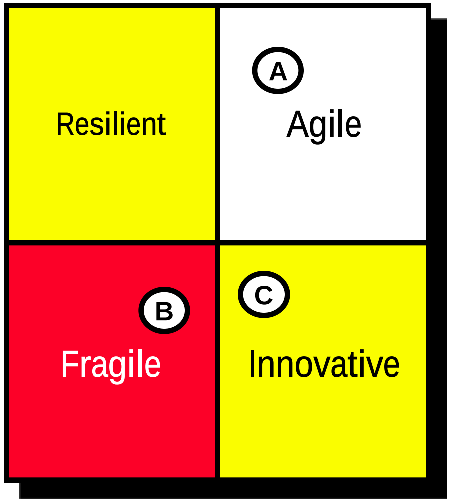
Issue: Reinterpret Rules for New Applications

Technology, applications, and corporate strategy change with time. By *distributing control* of this total process to the points of maximum knowledge we vest *evolving standards* responsibility in the hands of the *knowledge management committee*, for they have the current strategies and future goals of the organization in sight.

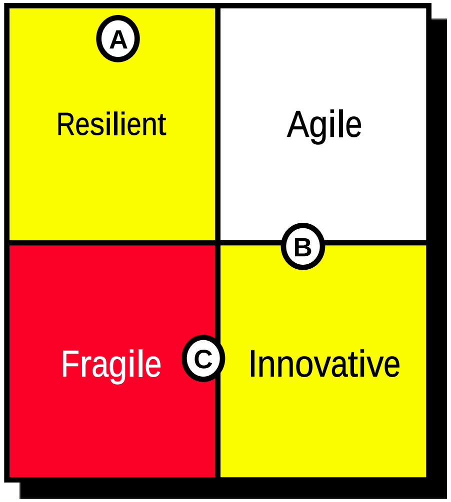
2-loop) Pick another issue, write a paragraph showing RRS-principle usage

Maturity Modeling

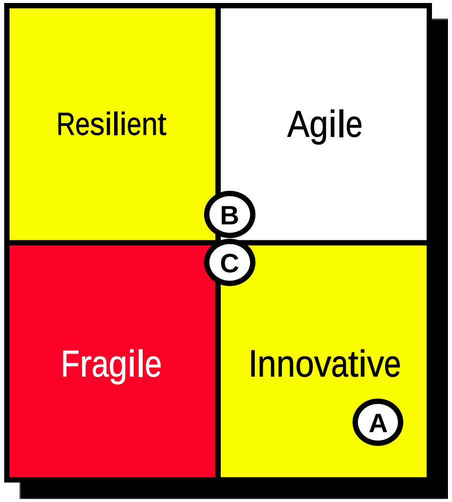
Outage Management



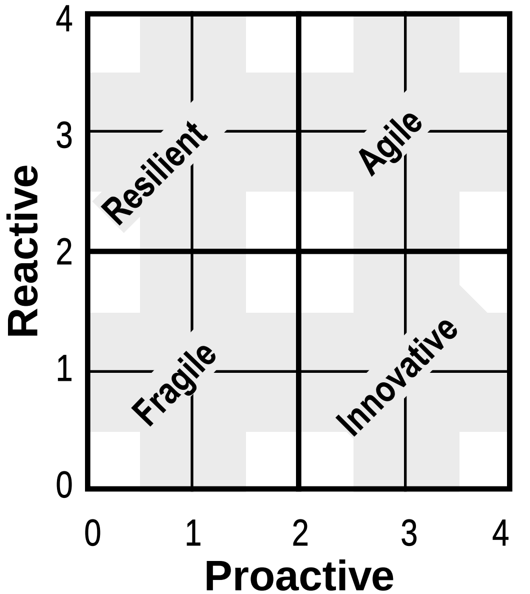
Customer Responsiveness



Competency Development



Assessment and Competitive Evaluation



Comparing Companies A, B, C

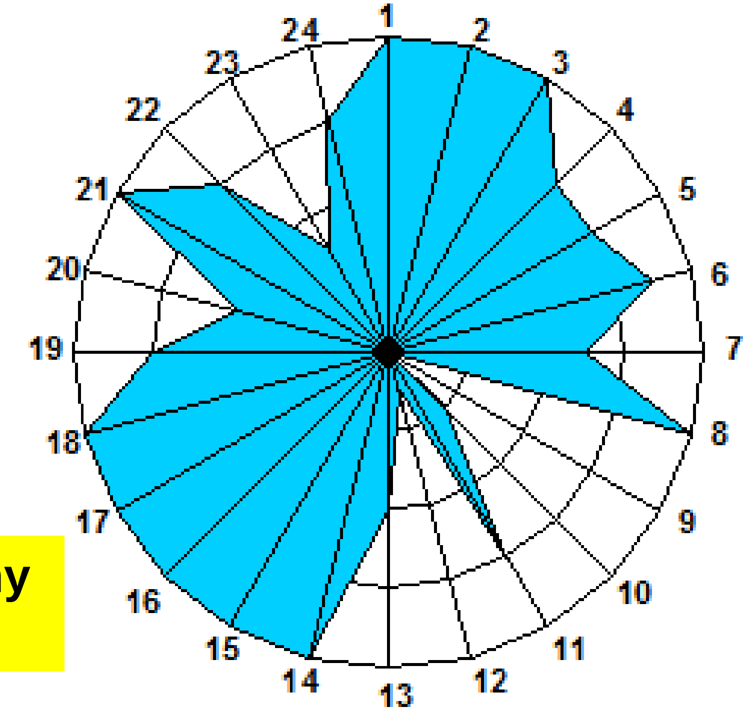
Response Proficiency Maturity Model

Stages	Metric Focus	Working Knowledge	Competitive Development Proactive	Competitive Development Reactive
0 Accidental	Pass/Fail	Examples	Lucky	None
1 Repeatable	Time	Concepts	Creation	Correction
2 Defined	Cost	Metrics	Improvement	Variation
3 Managed	Quality	Rules	Migration	Expansion
4 Mastered	Scope	Principles	Modification	Reconfig'tion

www.parshift.com/docs/MaturityModel00.htm

Enterprise Change Proficiency Profile

Analytical Guide
for Establishing
Competitive Strategy
and
Improvement Targets



Could show where your company and/or its processes are agile!

Maturity Stage	Working Knowledge	Metric Focus	Change Competencies Proactive	Change Competencies Reactive
0 Accidental	Examples	Pass/Fail	None	None
1 Repeatable	Concepts	Time	Creation	Correction
2 Defined	Metrics	Cost	Improvement	Variation
3 Managed	Responsibilities	Quality	Migration	Expansion
4 Mastered	Principles	Scope	Modification	Reconfiguration

www.parshift.com/docs/MaturityModel00.htm

Change Proficiency Maturity Stages

Stage 0 - Accidental - Characterized by:
- The lack of any change-process recognition, yet change manages to occur.
- The process is ad hoc: exhibiting false starts and retries, unpredictable completion dates and costs, surprising results and side effects, and undesirable reactions from, and effects on, the personnel involved.
- Examples: Downsizing, management fad-of-the-day, grueling overtime, fire-fighting, multiple reengineering attempts, expediting.

Stage 1 - Repeatable - Characterized by:
- Anecdotal "lessons learned" from past change activities.
- The time it takes to make a change is under control.
- Specific individuals and teams are recognized for repeatable success with effective change projects.

Stage 2 - Defined - Characterized by:
- The emergence of formal change processes with documented procedures.
- The base of practitioners is broadened as process rather than intuitive talent becomes appreciated.
- Metrics for the change process are identified, cost of change is under control, and predictability becomes a known but elusive desire.
- Typically procedures at this stage are rigid and based on studied experience and analysis.

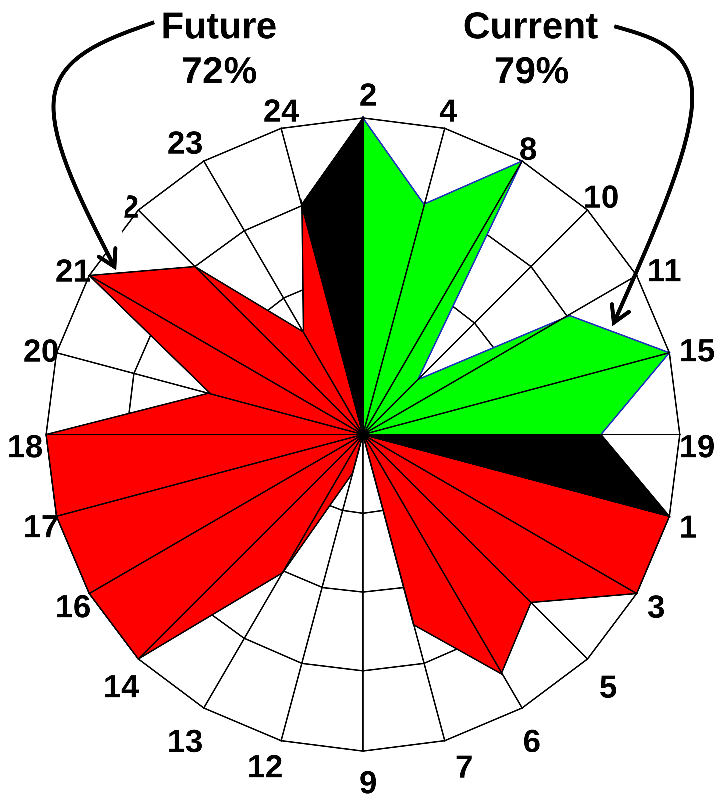
Stage 3 - Managed - Characterized by:
- The appointment of change managers (business engineers) with established responsibilities, though they may neither be called such nor recognized as such.
- An evolving knowledge base of change process fundamentals and rules begins to emerge.
- Rigid procedures are loosened, and predictable change processes are the norm.
- Appreciation for and participation in the corporate change process is widespread.

Stage 4 - Mastered - Characterized by:
- A principle-based, deep appreciation of adaptability.
- An understanding that process alone is not sufficient.
- A conscious engineering and manipulation of the business practice structures and organizational infrastructures.
- Like a flock of birds swooping and turning as a unit, corporate change loses its event status and takes on a constant fluid motion.

www.parshift.com/docs/MaturityModel00.htm

Maturity Fit to the Times/Needs

Grouped by Industry Priority



Remmele

www.parshift.com/docs/aermodA0.htm

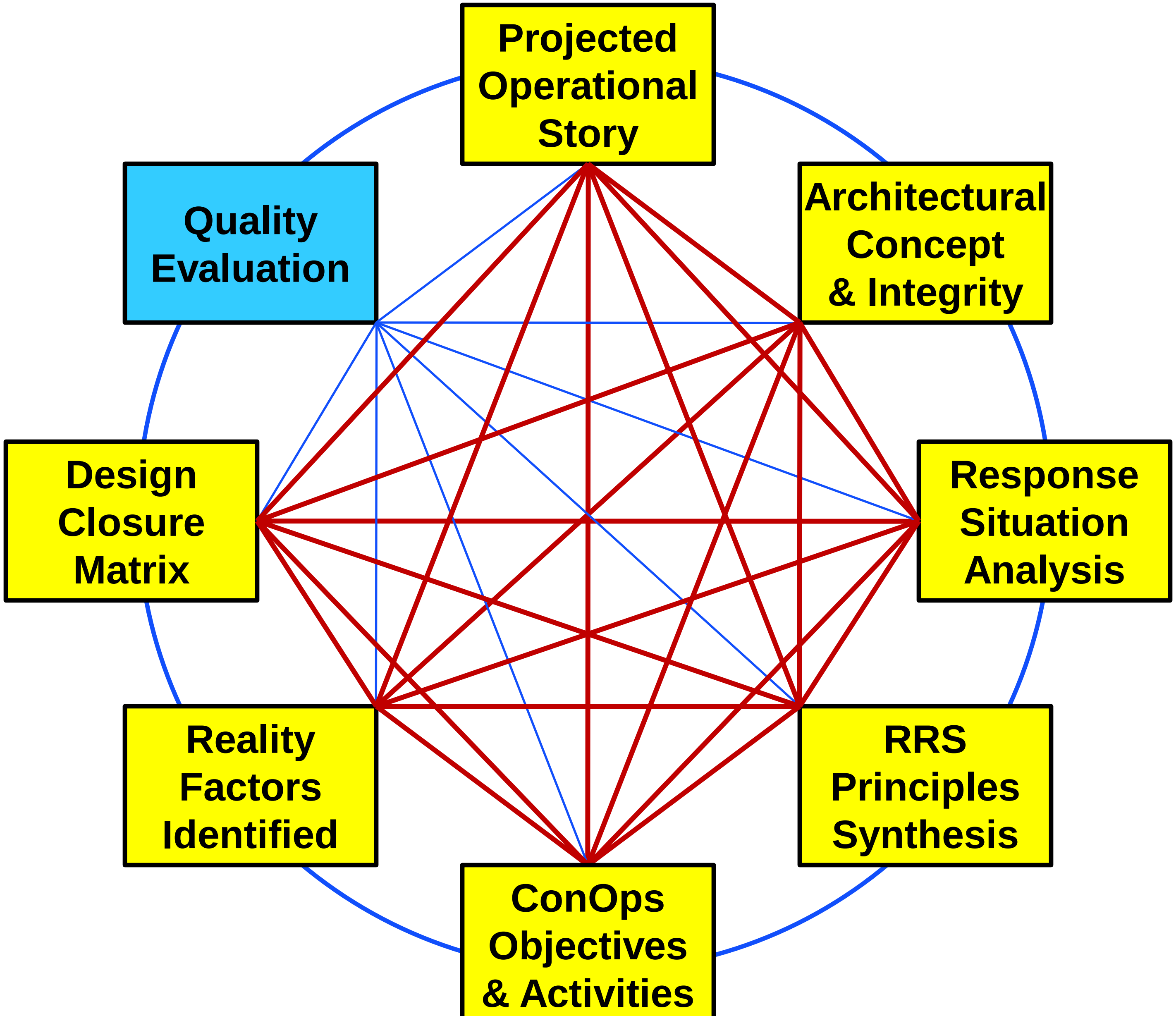
Industry Priority

Future	Current	Maturity		Critical Business Practice
4.00		4.0	1	Strategic Plan Vision
	4.00	4.0	2	Strategic Plan Dissemination
4.00		4.0	3	Strategic Plan Buy-In
	3.00	3.0	4	Capital Investment Justification
3.00		3.0	5	Infrastructure Investment Just.
3.50		3.5	6	Business Eng. Investment Just.
2.50		2.5	7	Business Unit Relationships
	4.00	4.0	8	Employee Relationships
0.00		0.0	9	Partner Relationships
	1.00	1.0	10	Supplier Relationships
	3.00	3.0	11	Customer Relationships
0.50		0.5	12	Information Sys. Unit Relationships
2.00		2.0	13	Production Unit Relationships
4.00		4.0	14	Product Innovation Management
	4.00	4.0	15	Process Innovation Management
4.00		4.0	16	Procedure Innovation Mgmnt.
4.00		4.0	17	Strategy Innovation Management.
4.00		4.0	18	Knowledge-Portfolio Strategy
	3.00	3.0	19	Knowledge Generation
2.00		2.0	20	Knowledge Capture
4.00		4.0	21	Knowledge Mobilization
3.00		3.0	22	Leading Indicator Metrics
1.50		1.5	23	Operating Metrics
3.00		3.0	24	Valuation Metrics

www.parshift.com/docs/MaturityModel00.htm

Wrapping it Up

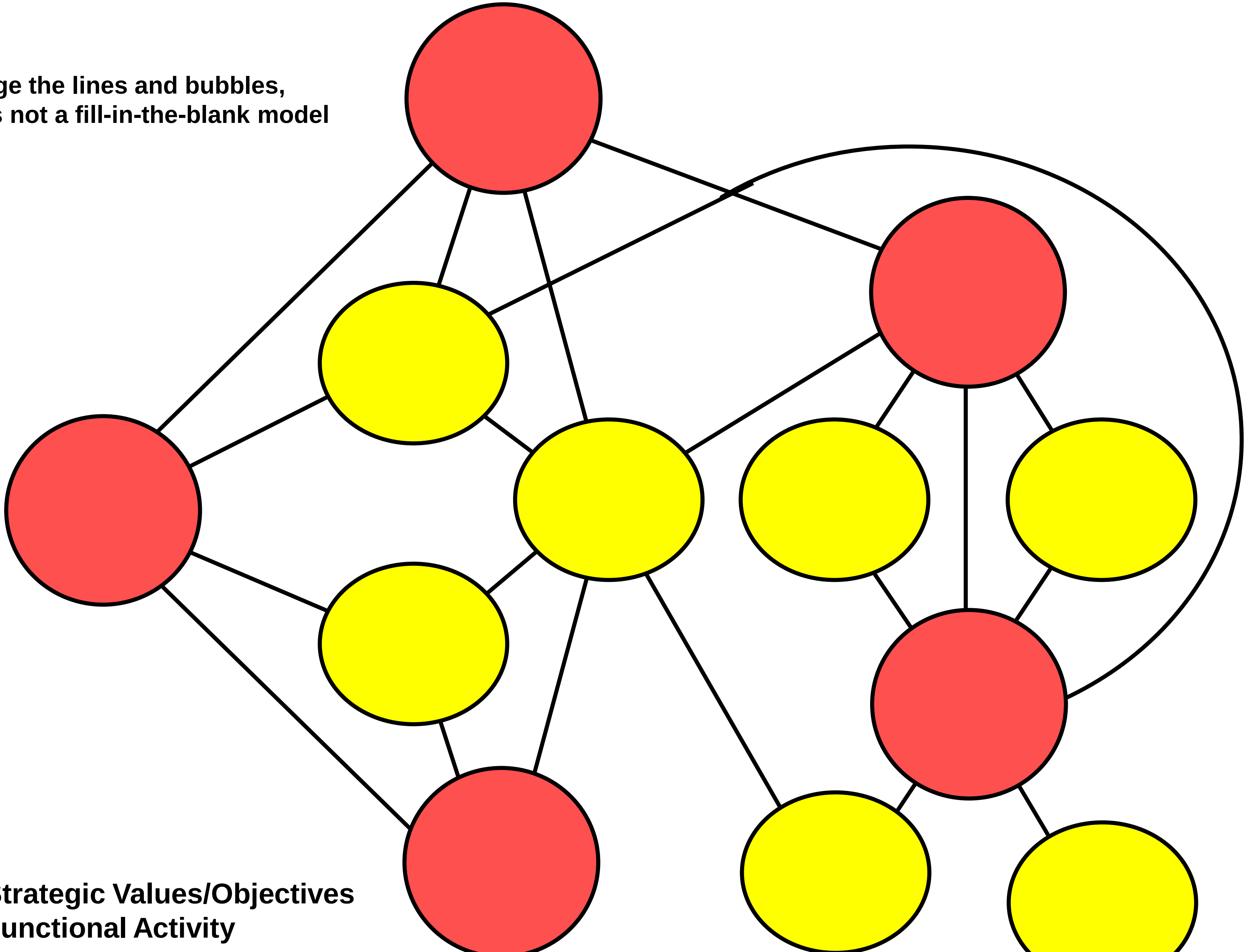
Eight principle tools are brought to bear when designing or analyzing a system for agility



form tools

System: ______________________________

Change the lines and bubbles,
this is not a fill-in-the-blank model



Strategic Values/Objectives
Functional Activity

Strategic Activity ConOps Web

RS Analysis for System ______________________

with [t,c,q,s] metric-priorities for each issue, t = time of change, c = cost of change, q = quality of change, s = scope of change

Change Domain		Response Issues
Proactive	Creation (and Elimination)	What must the system be creating or eliminating in the course of its operational activity? • ? • ? **Use as many bullet points as appropriate** • ?
	Improvement	What performance characteristics will the system be expected to improve during operational life cycle? • ? • ? • ?
	Migration	What major events coming down the road will require a change in the system infrastructure? • ? • ? • ?
	Modification (Add/Sub Capability)	What modifications in resources-employed might need made as the system is used? • ? • ? • ?
Reactive	Correction	What can go wrong that will need an automatic systemic detection and response? • ? • ? • ?
	Variation	What process variables will range across what values and need accommodation? • ? • ? • ?
	Expansion (and Contraction of Capacity)	What are “quantity-based” elastic-capacity needs on resources/output/activity/other? • ? • ? • ?
	Reconfigu-ration	What types of resource relationship configurations will need changed during operation? • ? • ? • ?

System: ______________________________

Reality Factors

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

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

Technology Pace – Accelerating vulnerability-introductions, sparse testing...
- ?

System Complexity – Incomprehensible, highly networked, unintended consequences, emergence...
- ?

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

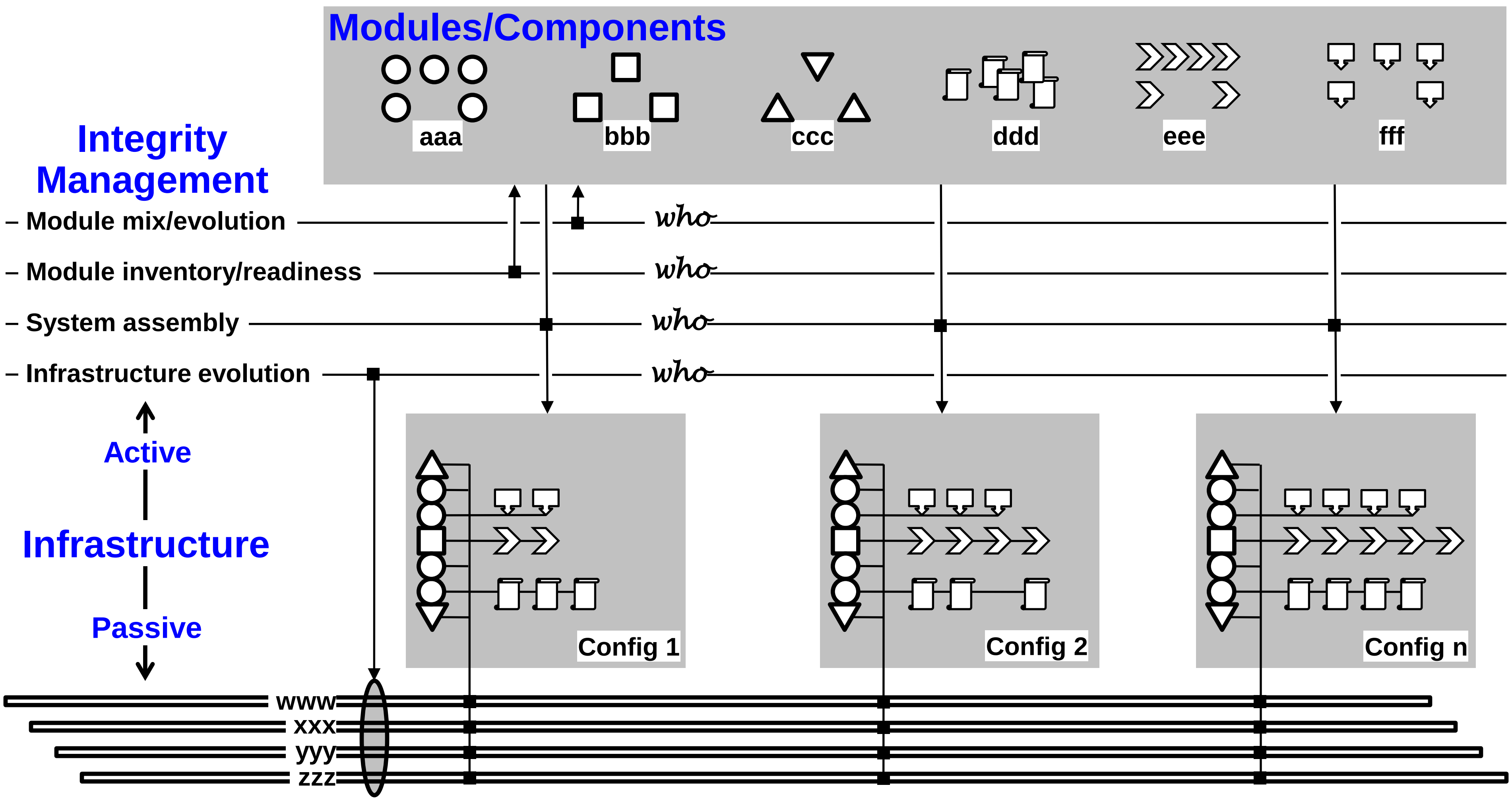
Faddish Ill-Thought Practices – Outsourcing, web services, transparency, COTS, SOA...
- x

Agile Adversaries/Competitors/Customers – Distributed, collaborative, self organizing, proactive, impatient, innovative...
- ?

Other?
- ?

Your System

Sample Graphics for your modification into your system needs



RRS Principles for System: ______________________________

(Think: Plug-and-Play, Drag-and-drop)

Reusable	
Self-Contained Units (Modules) Modules are encapsulated independent units loosely coupled through the passive infrastructure. ?	
Plug Compatibility (Facilitated Interfacing) Modules & infrastructure have features facilitating easy module insertion/removal. ?	
Facilitated Reuse Modules are reusable and/or replicable; with supporting facilitation for finding and employing appropriate modules. ?	

Scalable	
Evolving Standards (Infrastructure) Module interface and interaction standards and rules that evolve slowly. ?	
Redundancy and Diversity Duplicate modules provide fail-soft & capacity options; diversity provides functional options. ?	
Elastic Capacity Module populations & functional capacity may be increased and decreased widely within the existing infrastructure. ?	

Reconfigurable	
Flat Interaction Modules communicate directly on a peer-to-peer relationship; parallel rather than sequential relationships are favored. ?	**Distributed Control & Information** Decisions made at point of maximum knowledge; information accessible globally but kept locally. ?
Deferred Commitment Module relationships are transient when possible; decisions & fixed bindings are postponed until necessary. ?	**Self-Organization** Module relationships are self-determined; and component interaction is self-adjusting or negotiated. ?

System ______________________________
Who Does What to Keep the Systems Effective
(as part of their job description/responsibility/evaluation)

Module Mix/Evolution **(who decides/specifies/adds which new types of modules, excises old useless modules)**:

- ?
- ?
- ?

Inventory Management **(who maintains servicability/readiness and quantity on hand of which modules)**:

- ?
- ?
- ?

Real-Time Configuration **(who assembles/configures/reconfigures the systems):**

- ?
- ?

Framework Evolution **(who reviews and modifies the various framework elements):**

- ?
- ?
- ?
- ?

<your system>

RRS Principles

Activities

Activities	#
??	1
??	2
??	3
??	4
??	5
??	6
??	**7**

Issues

Principle-Based Activities and Issues Served

	Issues	Self Contained Units	Plug Compatibility	Facilitated Re-Use	Peer-Peer Interaction	Deferred Commitment	Distributed Control & Info	Self Organizing	Flexible Capacity	Unit Redundancy	Evolving Standards
Proactive	??										
	??										
	??										
	??										
	??										
	??										
	??										
Reactive	??										
	??										
	??										
	??										
	??										

Part 4
Self Organizing Agility

Agile Security:
Self Organizing Patterns and New Enabling Technology

All Systems are Prey

It used to be that systems were engineered for security first:

Fortress cities, castles, booby trapped tombs…

For protection from individuals and small bands of raiders.

Then came science and technology, not accessible to the general public.

Systems were not so vulnerable, too much knowledge/resource needed.

Systemic security declined in favor of functionality and variety.

Then came parity…

Car hacking, computer hacking, broad system literacy, low cost tech.

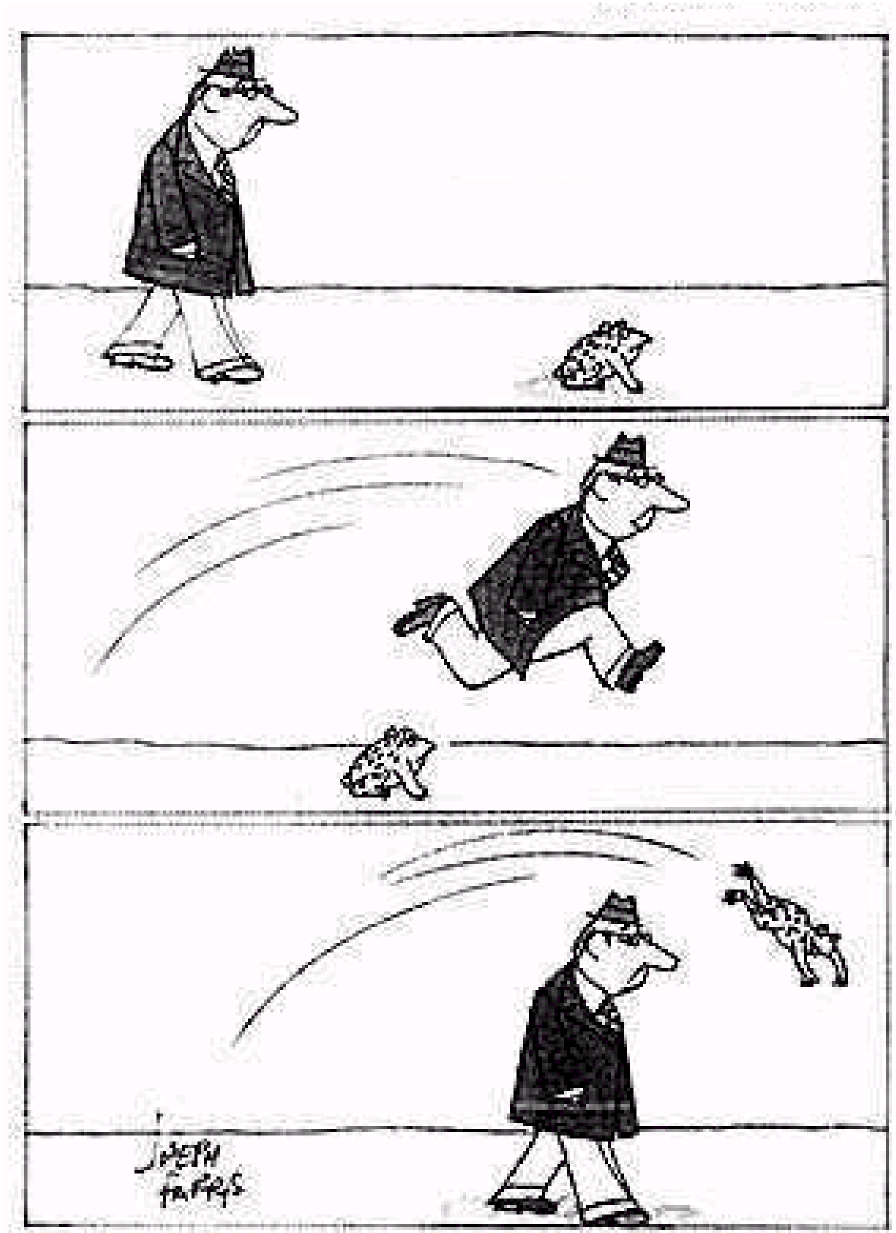
Systems are again vulnerable to the individual and small band.

and of course … the nation state

System engineering again needs to consider security first.

…in a time of increasing uncertainty and unpredictable technology.

Co-Evolution?

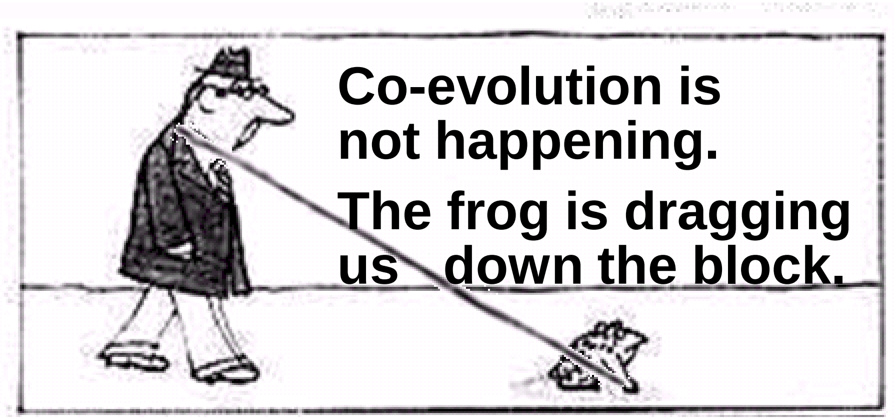


Co-Evolution?

The Adversary:

Architecture:

- Multi-agent
- Loosely coupled
- Self organizing
- Systems-of-systems

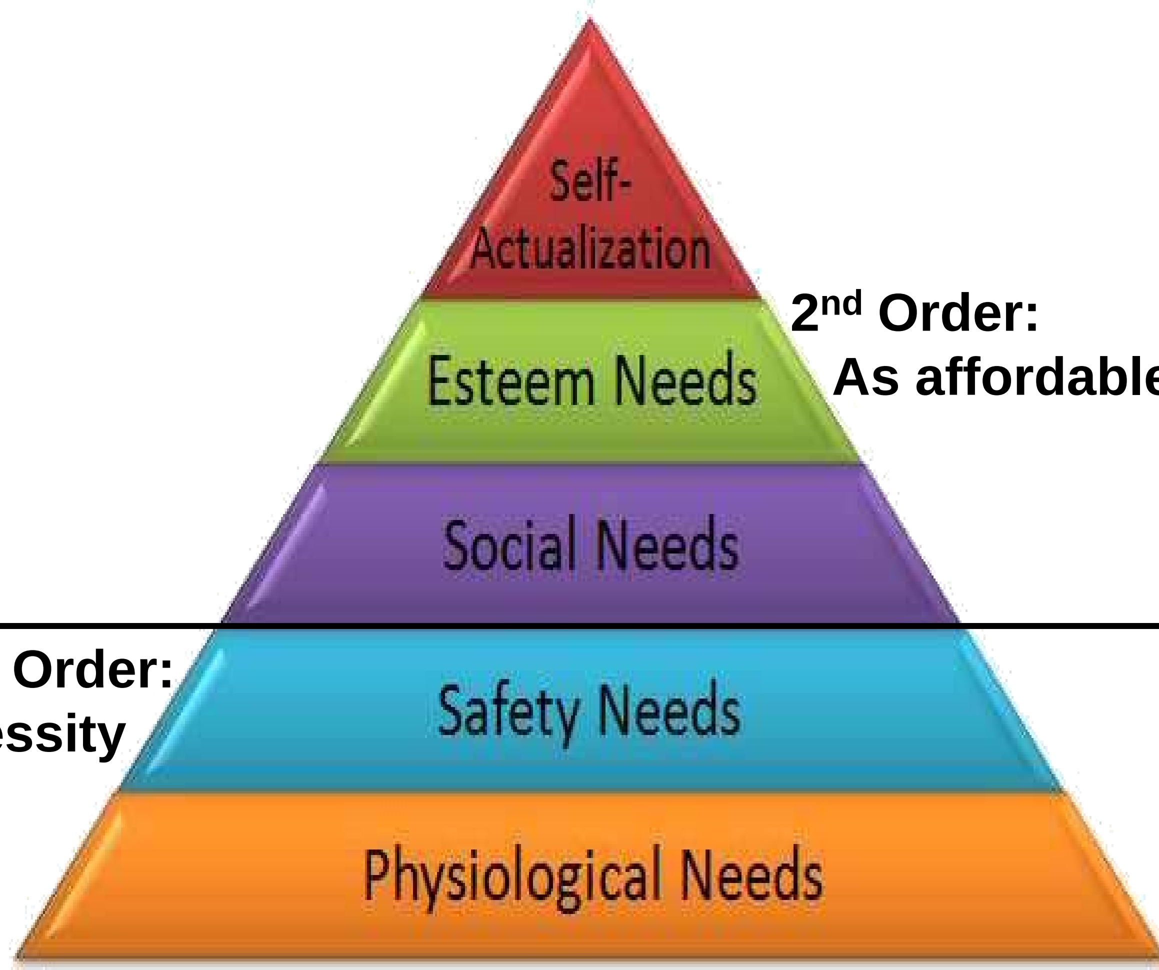


Behavior:

- Swarm intelligence
- Tight learning loops
- Fast evolution
- Dedicated intent
- Broad collaboration

We are not in an arms race
– we haven't engaged.

Maslow’s Hierarchy of Needs
(for systems that would live one more day)



2nd Order: As affordable

1st Order: Necessity

Maslow's Hierarchy of Needs

(for systems that would live one more day)

Its not about Information Security
…all systems are prey

Its about a first priority on
securing and maintaining existence.

- interaction of functional of existence (community impact)
- quality of functional existence
- product of existence
- sustains existence
- enables existence

The system-adversary is extremely competent.

New Reality

Humans are all natural system hackers.

All the world is computer literate.

Big money buys and motivates big talent.

Zero day – attacks never seen before, no detection signatures.

Attacks for sale, sophisticated entrepreneurial markets.

APT – Advanced Persistent Threat.

Nation states with time, resources, and national priority.

Stuxnet – a step function in sophistication.

Study: Advanced Threats a Growing Problem for Security

http://www.cio.com/article/598274/Study_Advanced_Threats_a_Growing_Problem_for_Security?source=CIONLE_nlt_infosec_2010-07-02

Most organizations have been the target of an advanced threat, according to results of a study released this week by Ponemon Institute.

What constitutes an advanced threat still varies within the industry, for purposes of their research, it is as a methodology employed to evade an organizations present technical and process countermeasures which relies on a variety of attack techniques as opposed to one specific type.

"The predominant majority of the threats are represented by unknown, zero-day attacks.

The majority of respondents said advanced exploits and malware have successfully evaded their anti-virus (AV) and intrusion detection system technologies. In addition to the difficulty in preventing advanced threats, 46 percent said it takes one month or longer to detect an advanced threat in their organization.

"Policies and procedures exist but support from personnel and technology seems to be inadequate to address the problem," said Ponemon officials in the summary. Only about one-third report that their security-enabling technologies are adequate and only 26 percent report security personnel are adequate to deal with advanced threats. "

Business Is Booming In the 'Zero-Day' Game

In the NY Times: all over the world, from South Africa to South Korea, business is booming in zero days.

The average attack persists for almost a year before it is detected, according to Symantec, the maker of antivirus software.

The market for information about computer vulnerabilities has turned into a gold rush.

Companies like Vupen charge customers an annual $100,000 subscription fee to shop through its catalog, and then charges per sale to countries who want to use the flaws in pursuit of the kind of success that the United States and Israel achieved three summers ago when they attacked Iran's nuclear enrichment program with a computer worm that became known as 'Stuxnet.'

Israel, Britain, Russia, India and Brazil are some of the biggest spenders but North Korea is also in the market, as are some Middle Eastern intelligence services.

In his own words: Confessions of a cyber warrior

originally published at InfoWorld.com July 2013

“They were offering a lot more money, which surprised me, because I had heard that the guys working on behalf of the feds made a lot less than we did. Not true -- it's certainly not true anymore, if you're any good.

They had thousands of people just like me. They had the best computers. They had multiple supercomputers. They had water-cooled computers. The guys that interviewed me were definitely smarter than I was. I went from always being the smartest guy wherever I worked to being just one of the regular coworkers. It didn't hurt my ego. It excited me.

How many exploits does your unit have access to? Literally tens of thousands -- it's more than that. We have tens of thousands of ready-to-use bugs in single applications, single operating systems.

It's all zero-days. If you can name the software or the controller, we have ways to exploit it. There is no software that isn't easily crackable. In the last few years, every publicly known and patched bug makes almost no impact on us. They aren't scratching the surface.

I work close to DC. There's no lettering or identifiers on the building. There are about 5,000 people on my team. My badge does not say "U.S. government" on it.

Cyber-Physical Systems & Stuxnet

Wake up Call

Landmark case: new era of attack sophistication and target.

Public Domain – But Mainly Studied by the Bad Guys.

Implications: manufacturing systems, energy systems, robots, weapons, medical devices, automotive vehicles, ... the Internet of Things.

Stuxnet Brings New Era of Sophistication

The victim system was Iran's Natanz nuclear-fuel enrichment plant, with thousands of COTS (commercial off-the-shelf) centrifuges and many SCADA (supervisor control and data acquisition) COTS computers that controlled them.

There was an "air gap" between the plant systems and the outside world—with no external network connections.

An infected USB device from the outside is suspected. And/or maybe SCADA computers purchased off the shelf, and placed in a network of devices that likely trusted each other. Even an infection occurring after installation could spread among naïvely trusting networked equipment.

Four so-called zero-day attacks were employed.

Stuxnet came to life once a month for a short period, instructing the centrifuges to spin outside their specifications, slowly damaging them.

Stuxnet spoofed the monitoring system just like the television spy shows, showing normal operating data to the monitors.

This slow, stealthy, periodic attack over many months just looked like accelerated wear, and didn't cause an immediate investigation alert.

The attack method has been reverse engineered, and is now in the public domain—for use by others less resourceful.

Something Major Is Happening

Banks - under major and broad attack

Military - Pentagon Admits Major Data Breach

Infrastructure – back doors hidden in control systems at utilities

Oil - 30,000 computers destroyed (wiped clean) in attack on Saudi Aramco (2013)

Cars – can be told to crash

Medical – Pace Makers can be told to kill

Natanz - Iran admits centrifuges damaged by virus

Weapons - can be told to shoot (or not)

Robots – what would you like me to do?

Manufacturing – how about losing defense-critical semiconductors?

“Dillon Beresford of NSS Labs found that with less than three weeks of work, and spending about $10,000 to replicate his target hardware environment, he was able to successfully exploit a Siemens Simatic S7 PLC. His aim wasn't to create Stuxnet 2.0, [but] to try and show the public that it's really not that complicated,

Systems Engineering

A relatively new "engineering" discipline.

Propelled formally by government acquisition concerns:

- **Big public expenditures.**
- **Complex system.**
- **Systems of systems.**

Needing: accountability, predictability, effectiveness.

Systems Engineering develops the big picture:

- **Requirements – understanding the problem to solve.**
- **Concepts – proposing a path to solution.**
- **Development – managing progress toward solution.**
- **Integration – inter-discipline tradeoffs and interfaces.**
- **Verification and Validation – does it work as planned, as needed.**

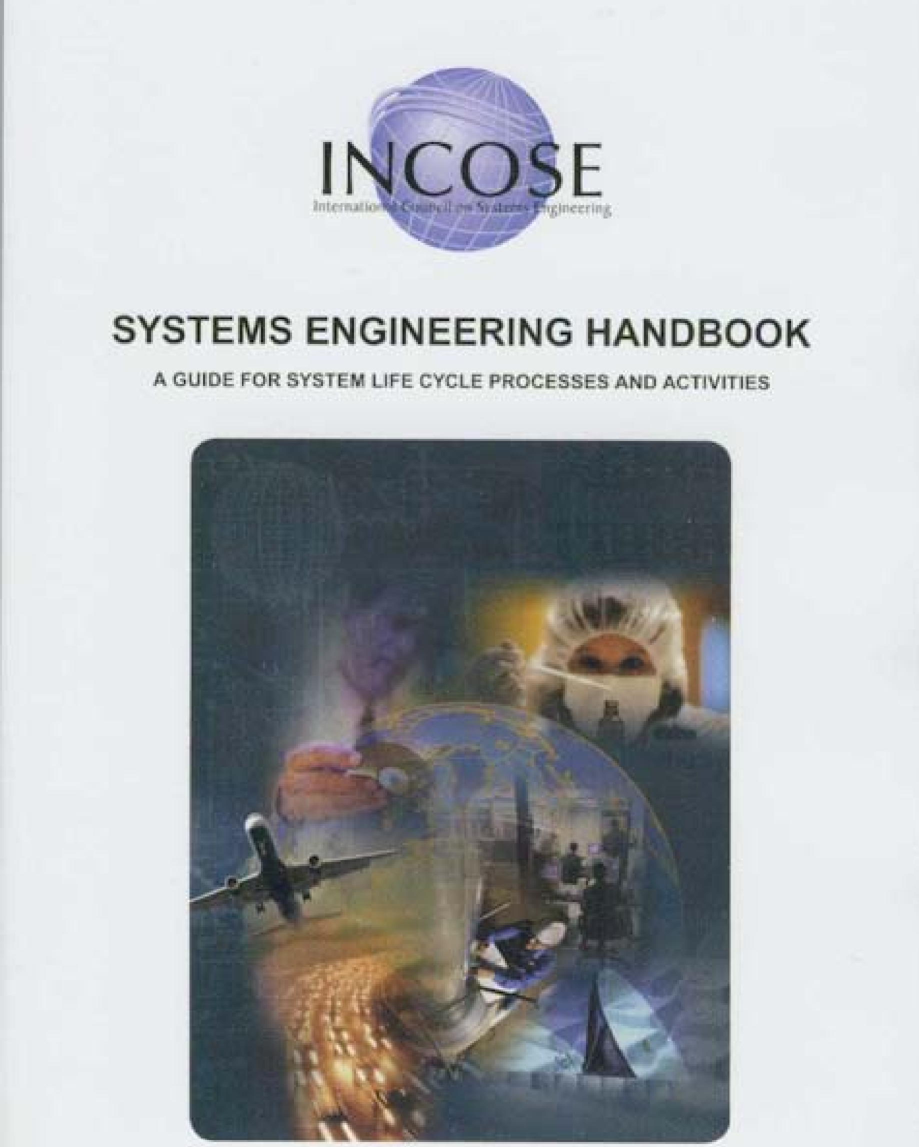
Functional Engineering develops functional performance:

electrical, mechanical, chemical, digital, etc.

Specialty Engineering develops non-functional performance:

producability, usability, safety, security, etc

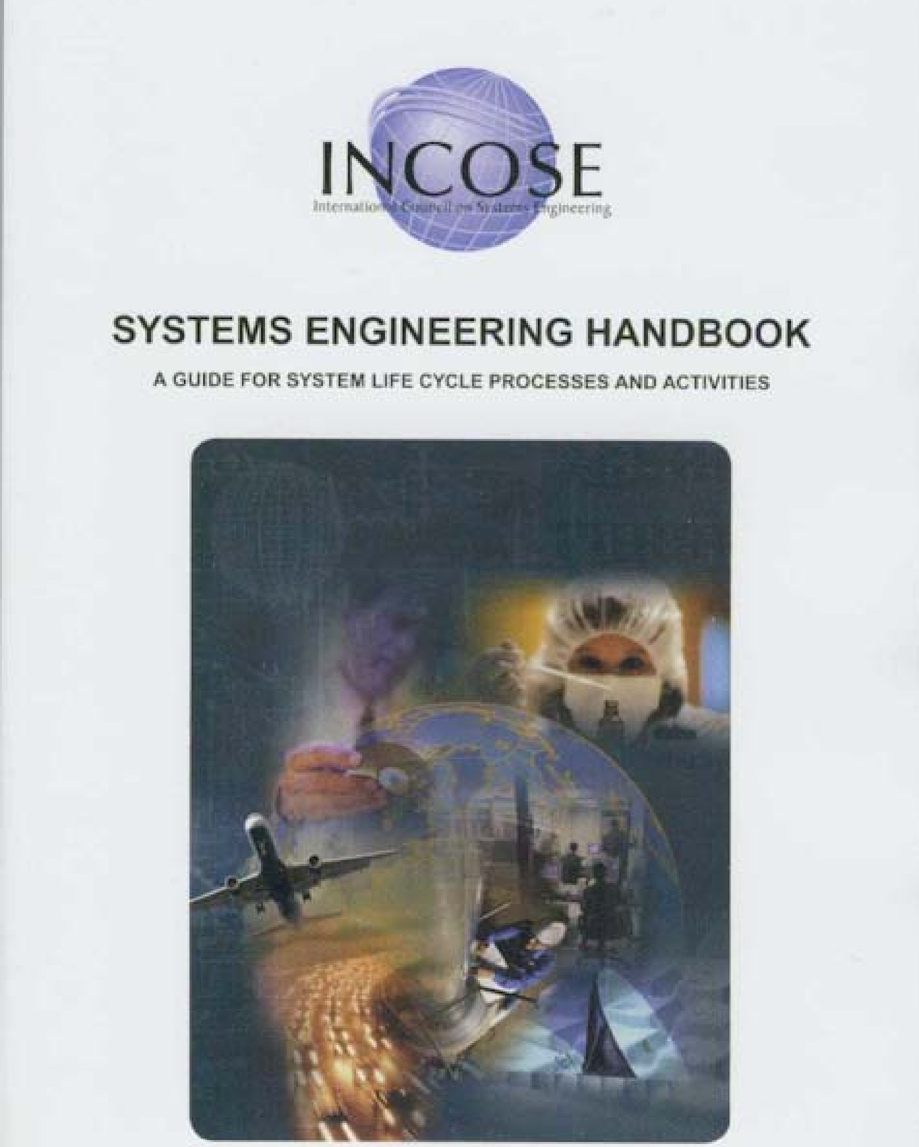
Supports the CSEP exam
(Certified Systems Engineering Professional)



Version 3.2.2 October 2011

Table of Contents

Supports the CSEP exam
(Certified Systems Engineering Professional)



Version 3.2.2 October 2011

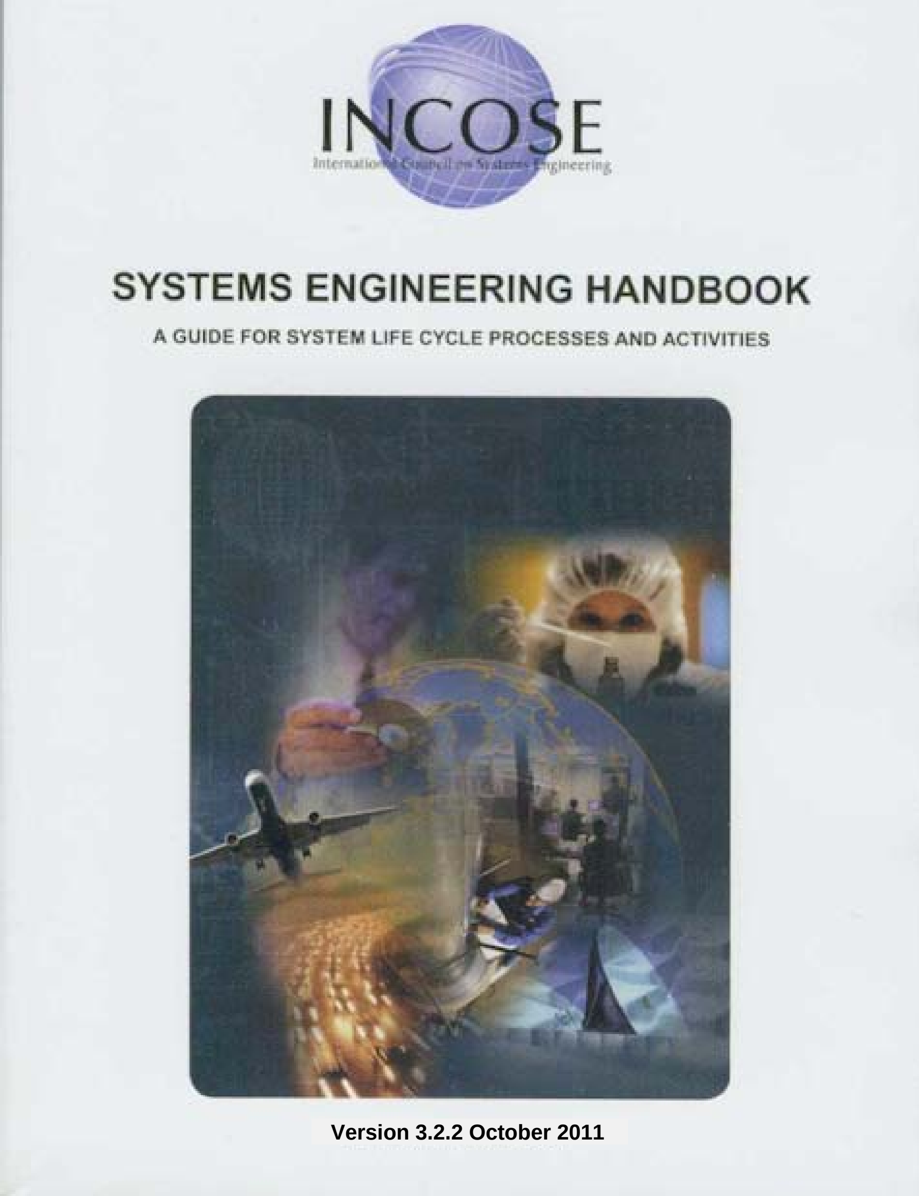
4. Technical Processes
- 4.1 Stakeholder Req. Def. Process
- 4.2 Requirements Analysis Process
- 4.3 Architectural Design Process
- 4.4 Implementation Process
- 4.5 Integration Process
- 4.6 Verification Process
- 4.7 Transition Process
- 4.8 Validation Process
- 4.9 Operation Process
- 4.10 Maintenance Process
- 4.11 Disposal Process
- 4.12 Cross-Cutting Technical Methods

5 Project Processes
- 5.1 Project Planning Process
- 5.2 Project Assessment & Control Process
- 5.3 Decision Management Process
- 5.4 Risk Management Process
- 5.5 Configuration Management Process
- 5.6 Information Management Process
- 5.7 Measurement Process

6 Agreement Process
- 6.1 Acquisition Process
- 6.2 Supply Process

7 Organizational Project-Enabling Processes
- 7.1 Life Cycle Model Management Process
- 7.2 Infrastructure Management Process
- 7.3 Project Portfolio Management Process
- 7.4 Human Resource Management Process
- 7.5 Quality Management Process

Supports the CSEP exam
(Certified Systems Engineering Professional)



Security is missing !

Responsibility/Accountability

Code of Hammurabi (2200 BC)
King of Babylonia Translated by R.F. Harper

If a builder builds a house for a man and do not make its construction firm and the house which he has built collapse and cause the death of the owner of house – **the builder shall be put to death.**

If it cause the death of the son of the owner of the house – **they shall put to death a son of that builder.**

If it destroy property, he shall restore whatever it destroyed, and because he did not make the house which he built firm and it collapsed, he shall rebuild the house which collapsed at his own expense.

Common Law in England (15th Century)

If a carpenter undertakes to build a house and does it ill (not well), an action will lie against him

Napoleonic Code (1804)

If there is a loss in serviceability in a constructed project within 10 years of its completion because of a foundation failure or from poor workmanship, the contractor and architect will be sent to prison

Forensic Engineering, D. Fowler (slide presentation has been removed from the Internet)

Systems Security Engineering Working Group

GOALS

Goal: **Establish the responsibility for security within Systems Engineering, with effective system security accepted and practiced as a fundamental goal of system engineering.**

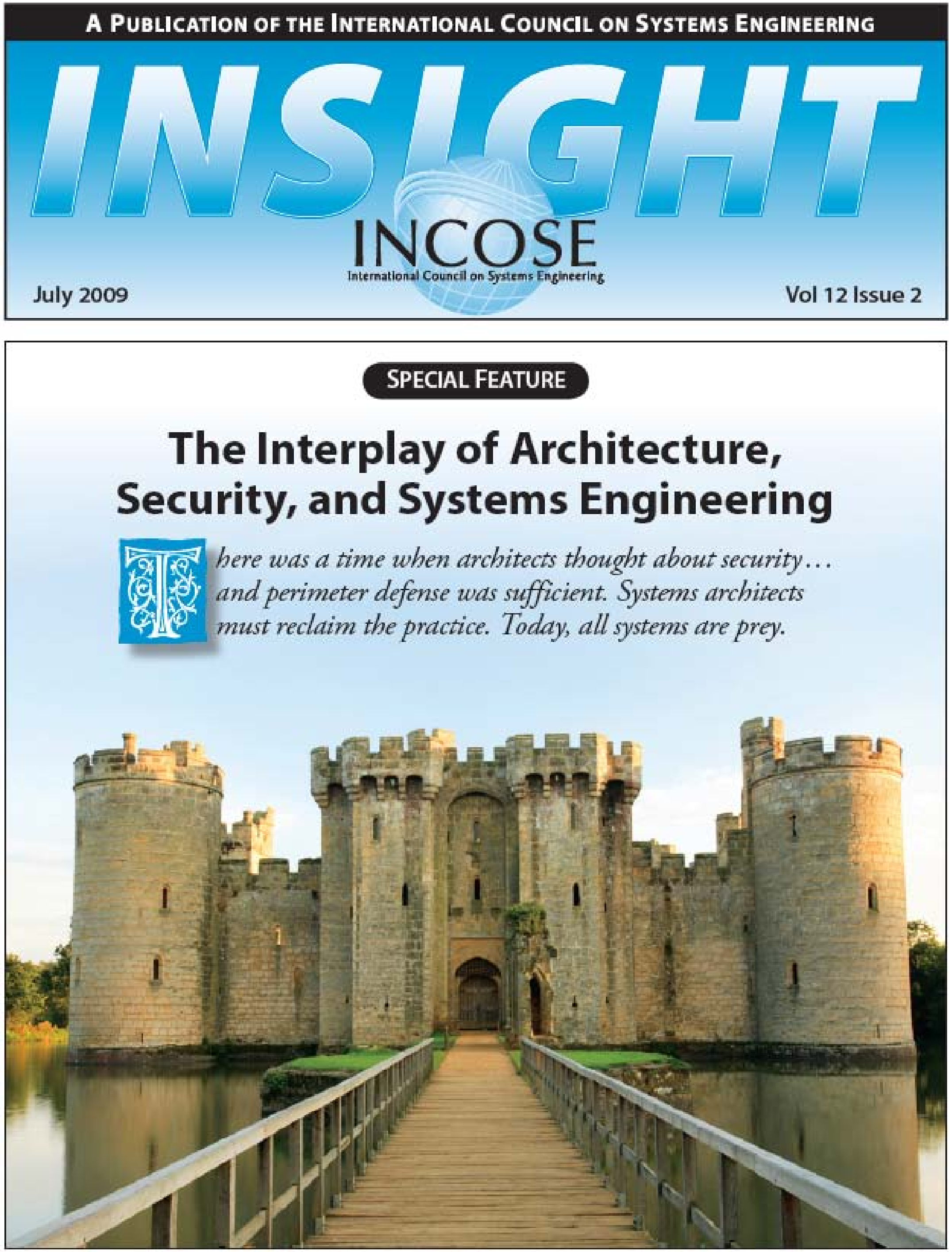
Goal: **Instigate self-sustaining cross-community involvement between systems engineering, security engineering, and system security standards.**

Goal: **Establish exemplar profiles of self organizing system-of-system concepts for next generation security.**

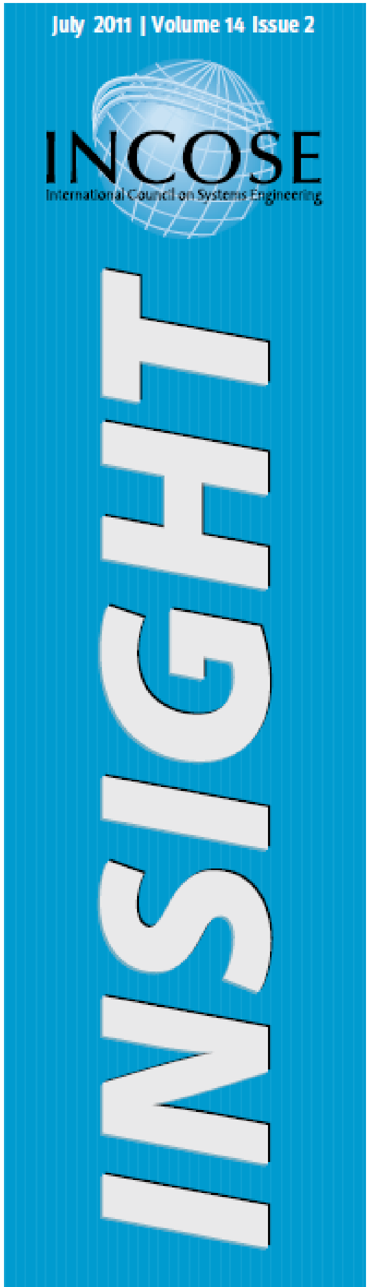
Customer(s)/Stakeholder(s): Systems engineering educators, systems engineering process and standards developers, DoD systems engineering acquisition requirements developers, systems engineering leaders and managers, customers of systems that require effective security.

SCOPE

This WG will address and foster system design concepts, system engineering processes, enabling support (such as standards), and community understanding and acceptance; all relative specifically to next generation system security characterized principally as self organizing, adaptive, resilient, evolutionary, proactive, and harmonious – in at least equal effectiveness as the system-adversarial communities.



- **The Interplay of Architecture, Security, and Systems Engineering** – Rick Dove
- **System Security Engineering: A Critical Discipline of Systems Engineering** – Kristen Baldwin
- **Embedding Agile Security in System Architecture** – Rick Dove
- **Toward a Dynamic System Architecture for Enhanced Security** – Mark De Spain
- **Resilient Control Systems: A Basis for Next-Generation Secure Architectures** – Craig Rieger
- **Secure Architecture and Design of Component-Based Systems** – Karen Mercedes Goertzel
- **Using the U.S. Department of Defense Architecture Framework to Build Security into the Lifecycle** – William P. Mulokey
- **An Architecture of Information Assurance Processes** – Jackson Wynn
- **Standardized Practices for Embedding Security from Concept Through Development** – Susan Albert and Jacqueline Nemeth
- **Balancing Security and Other Concerns within a Systems Architectural Approach** – Mike Wilkinson and Paul King
- **Developing a System Architecture for Managing the Nuclear Weapons Enterprise in the Context of a Comprehensive Policy Portfolio** – Dennis Engi
- **Establishing Security Strategy Using Systems Thinking** – John Wirsbinski and John Boardman



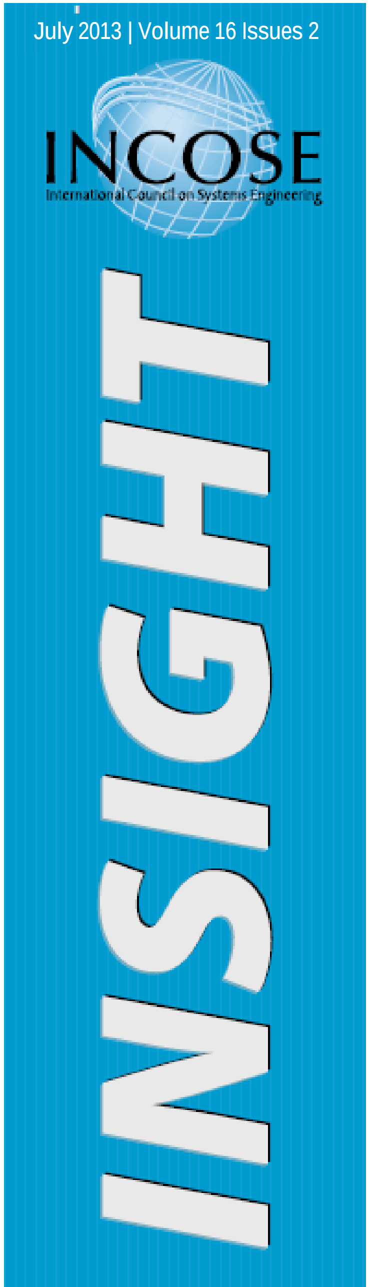
A PUBLICATION OF THE INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING

What's Inside

SPECIAL FEATURE

Systems of Systems and Self Organizing Security

(Photo: Amber Sports Used with permission)



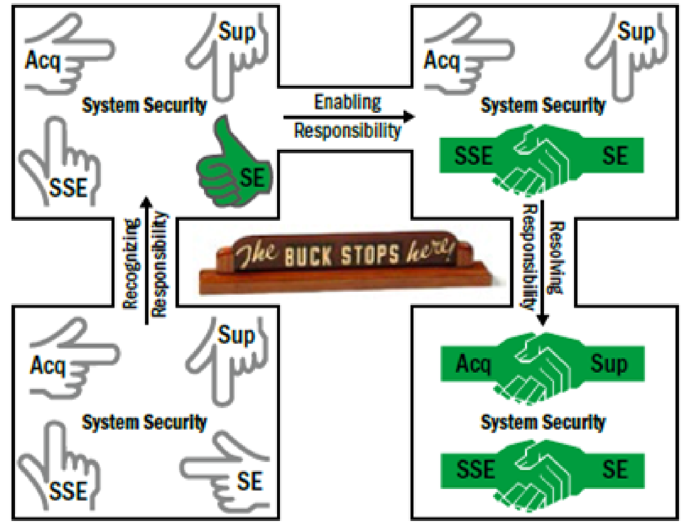
A PUBLICATION OF THE INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING

What's Inside

- **The Buck Stops Here: Systems Engineering Is Responsible for System Security**
  – Rick Dove, Paul Popick, Beth Wilson
- **Management Initiatives to Integrate Systems and Security Engineering**
  – Lori Masso and Beth Wilson
- **Information Security: Shaping or Impeding Systems in the Future?**
  – Ken Kepcahr
- **What Does a Systems-Security Engineer Do and Why Do Systems Engineers Care?**
  – Janet Oren
- **Addressing Attack Vectors Within the Acquisition Supply Chain and the Development Lifecycle**
  – John Miller
- **Requirements Challenges in Addressing Malicious Supply-Chain Threats**
  – Paul Popick and Melinda Reed
- **Uncertainty in Security: Using SE Approaches for Robust System Security Requirements**
  – Ruth Duggan and Mark Snell

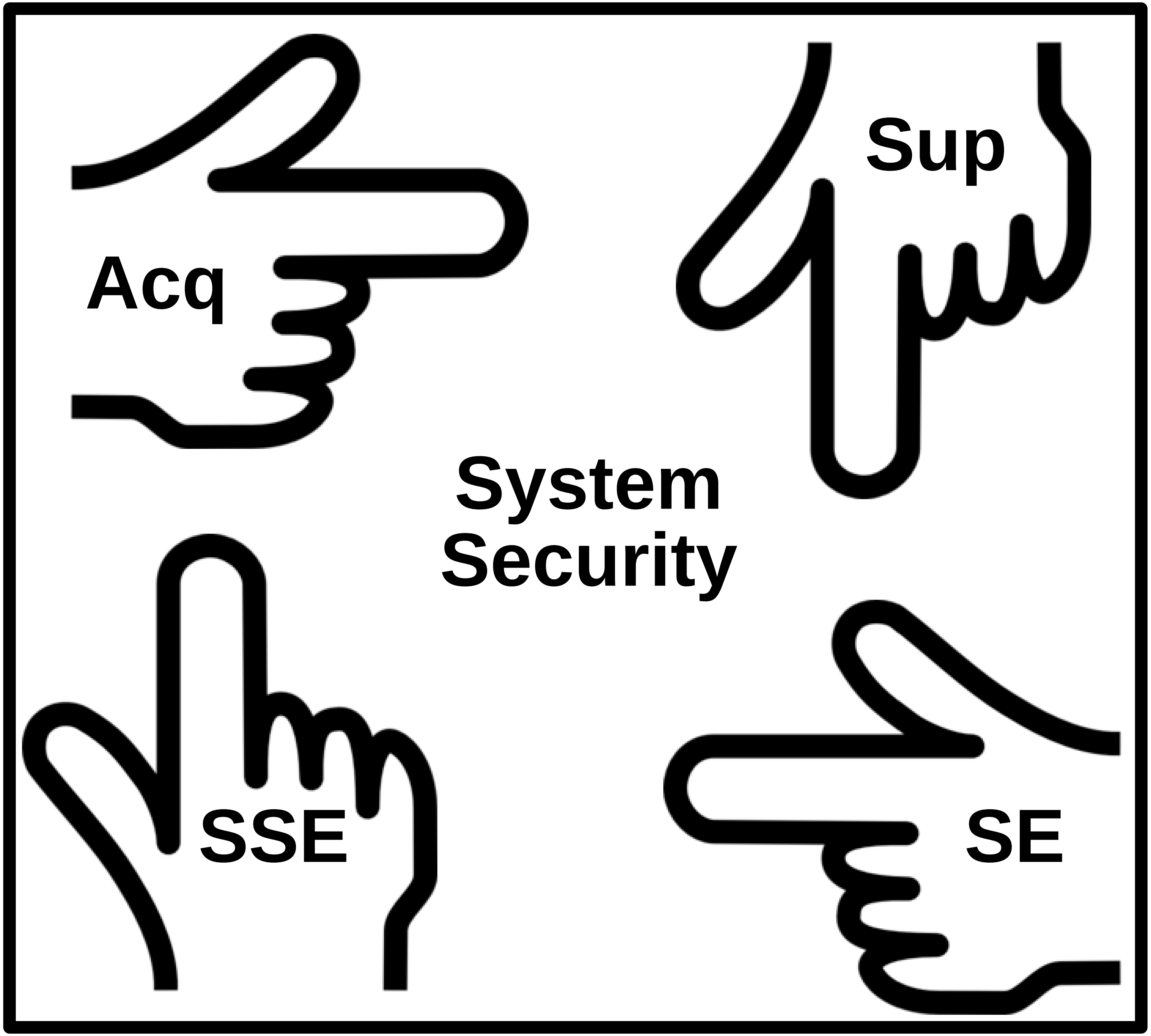
SPECIAL FEATURE

Systems Engineering Is Responsible for System Security

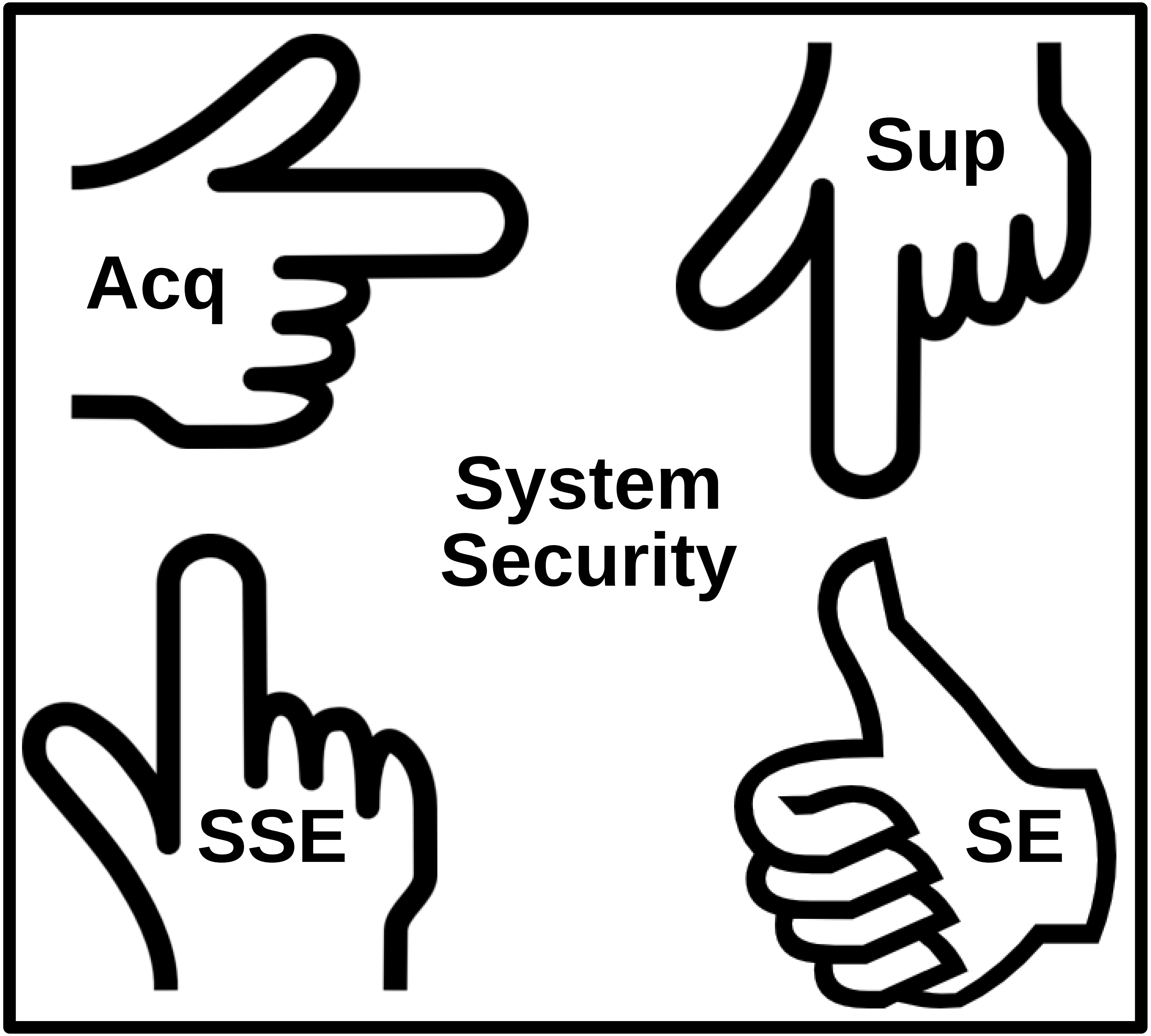


- **Enabling Sustainable Agile Security Through Systems Engineering**
  – Rick Dove
- **Security Engineering Models**
  – Bob Marchant
- **Evaluation of Security Risks using Mission Threads** – Carol Woody
- **System Integration at the Security Interfaces** – Kevin Stoffell
- **Verifying Security Control Requirements and Validating their Effectiveness** – Bruce Hunter
- **An Approach to Integrate Security into a Systems Engineering Curriculum** – Don Gelosh

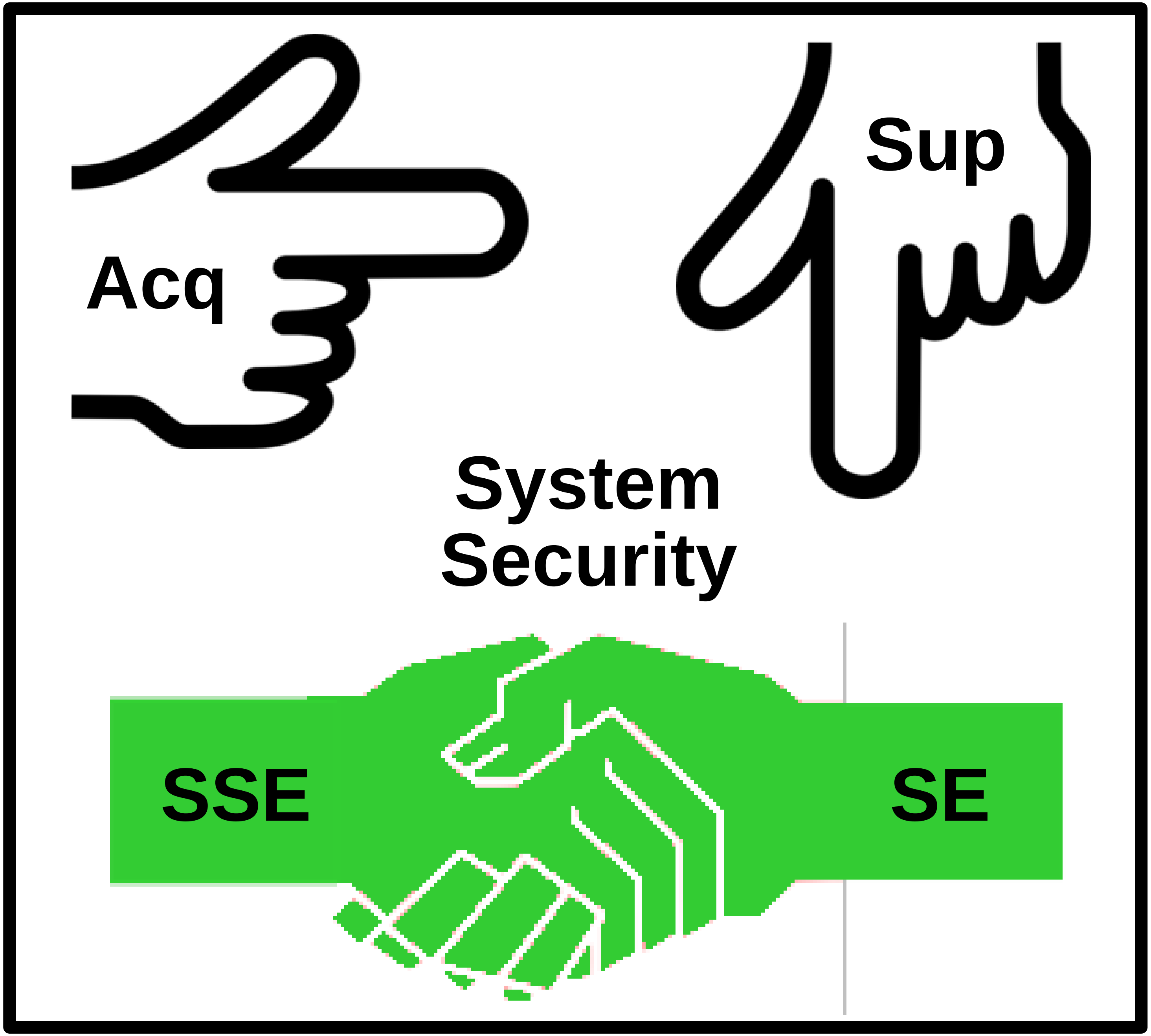
2012



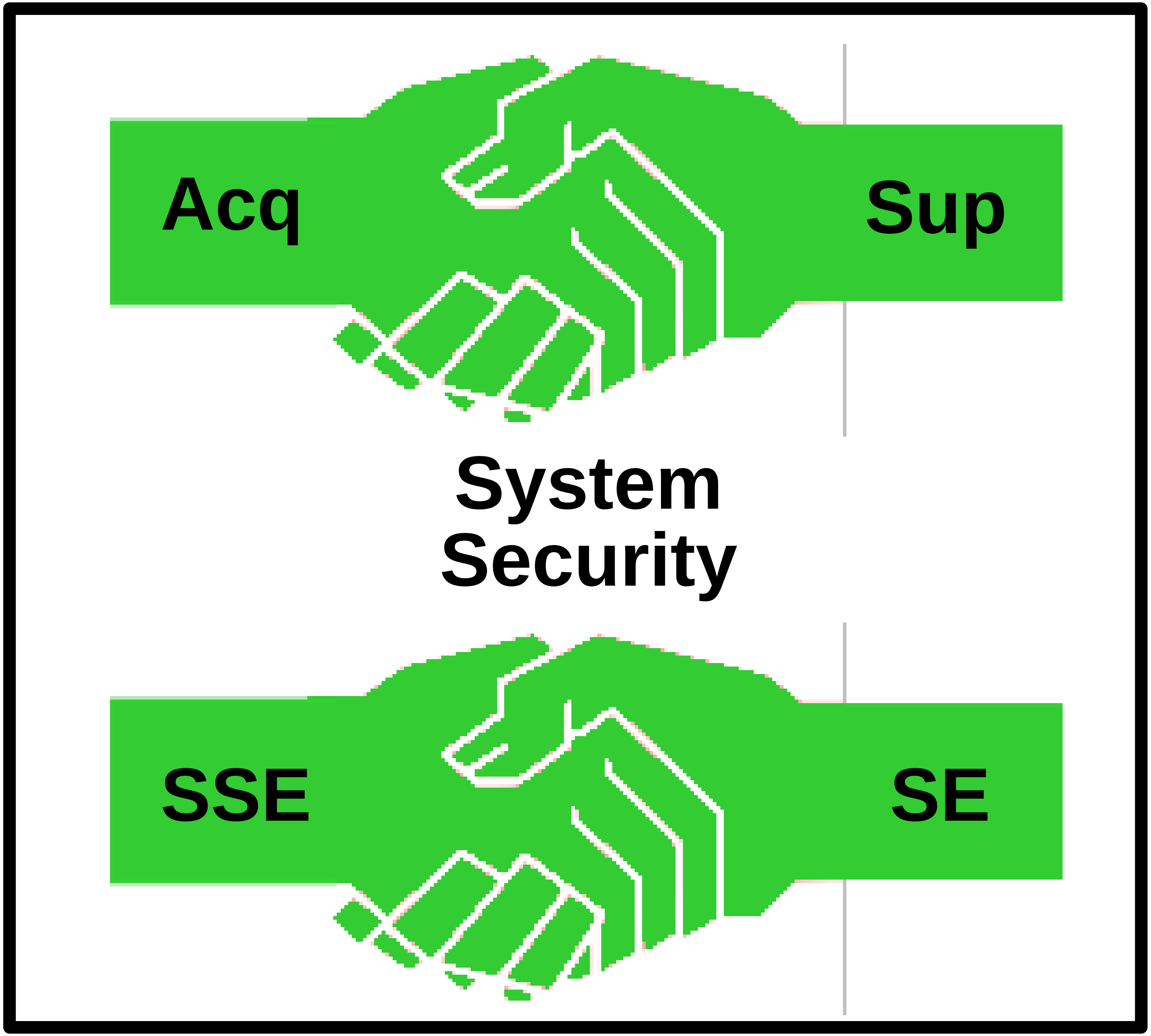
2013



2014



????


Acq
Sup
System
Security
SSE
SE

INCOSE has recognized that system security is the responsibility of Systems Engineering.

The July 2014 version of the Handbook will make this clear.

Two new sections:

3.6.4 Case 4: The Stuxnet Attack on a Cyber-Physical System (in case study section)

10.14 System Security Engineering (in specialty engineering section)

- Roles and Responsibilities
- Stakeholder Requirements
- Architecture Design
- Verification / Validation
- Maintenance and Disposal
- Risk Management Process
- Configuration Management Process
- Acquisition and Supply Processes

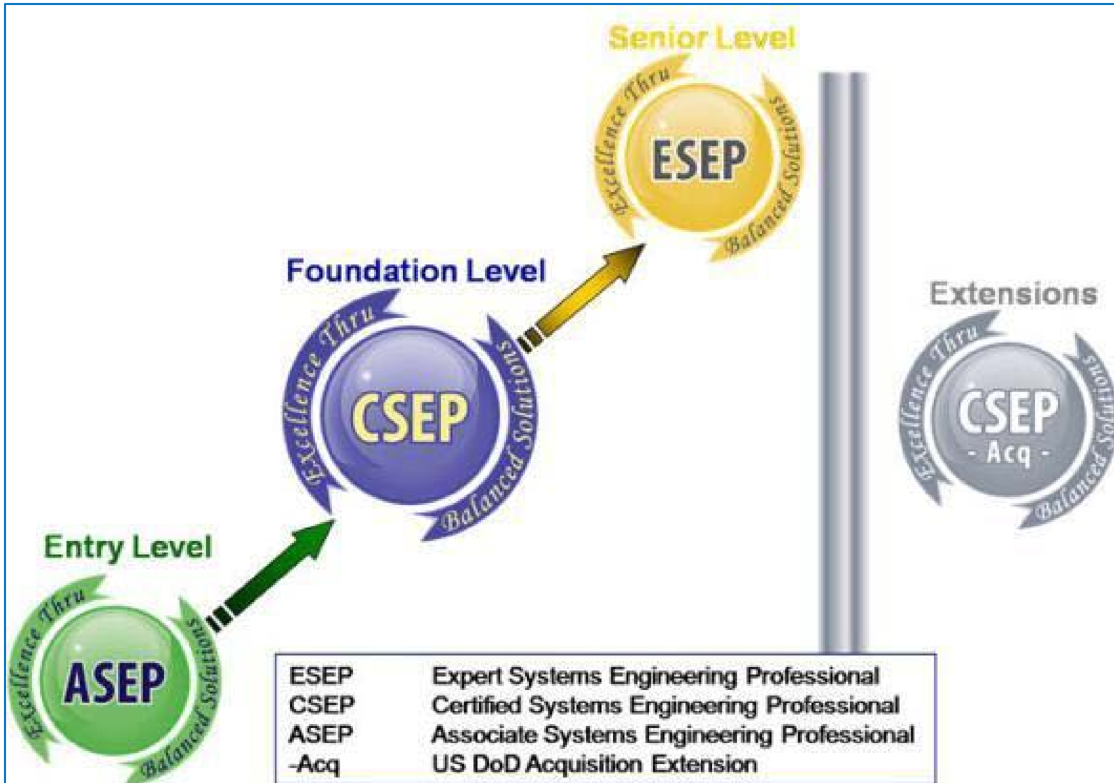
The Challenge

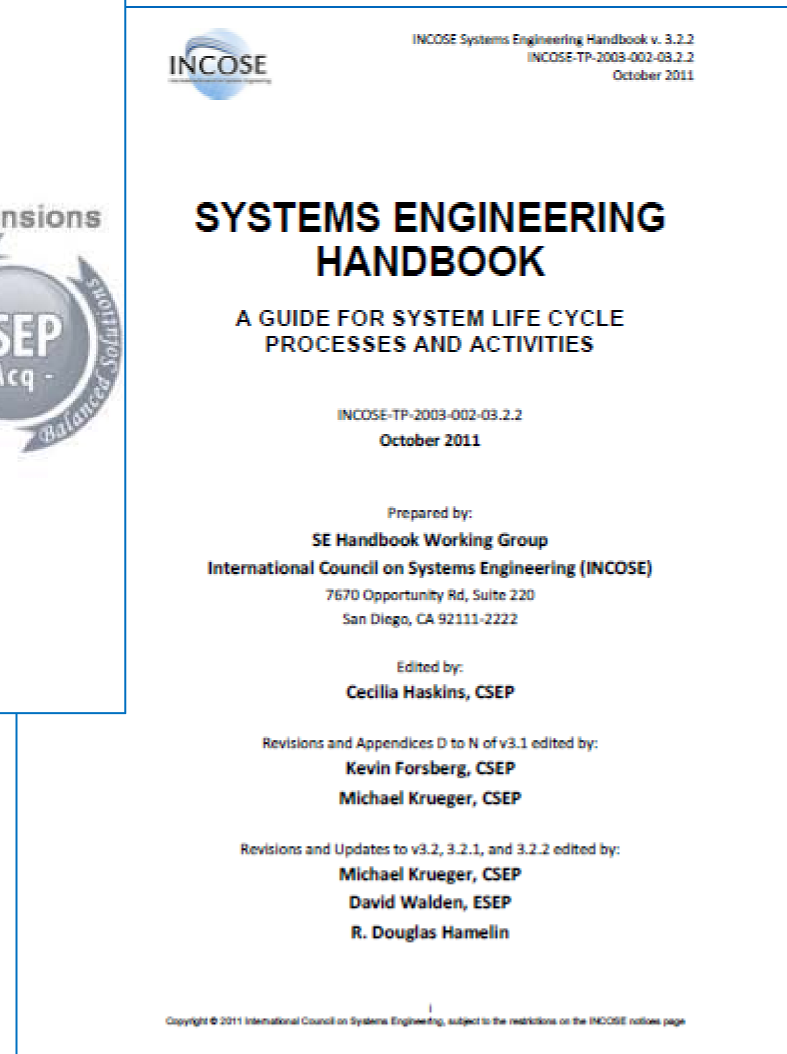
"A wicked problem is a social or cultural problem that is difficult or impossible to solve for as many as four reasons:

- incomplete or contradictory knowledge,
- the number of people and opinions involved,
- the large economic burden,
- the interconnected nature of these problems with *other* problems.

From: *Wicked Problems: Problems Worth Solving*, Jon Kolko, 2012
www.ssireview.org/articles/entry/wicked_problems_problems_worth_solving

INCOSE Systems Engineering Certification Based on SE Handbook





INCOSE

INCOSE Systems Engineering Handbook v. 3.2.2
INCOSE-TP-2003-002-03.2.2
October 2011

SYSTEMS ENGINEERING HANDBOOK

A GUIDE FOR SYSTEM LIFE CYCLE PROCESSES AND ACTIVITIES

INCOSE-TP-2003-002-03.2.2

October 2011

Prepared by:

SE Handbook Working Group

International Council on Systems Engineering (INCOSE)

7670 Opportunity Rd, Suite 220

San Diego, CA 92111-2222

Edited by:

Cecilia Haskins, CSEP

Revisions and Appendices D to N of v3.1 edited by:

Kevin Forsberg, CSEP

Michael Krueger, CSEP

Revisions and Updates to v3.2, 3.2.1, and 3.2.2 edited by:

Michael Krueger, CSEP

David Walden, ESEP

R. Douglas Hamelin

i

Copyright © 2011 International Council on Systems Engineering, subject to the restrictions on the INCOSE notices page

Accepting the challenge starts with security as part of the Certified SE Professional exam

Lockheed Accepts the SE Responsibility

and Leads the Way

New Lockheed Security Engineering Procedure

Applicability and Scope

- **The activities can be tailored to any life cycle methodology, including spiral and agile development methodologies, principles, and practices.**
- **The activities can also be tailored to accommodate varying levels of protection, different trust criteria, and unique customer requirements.**
- **The Capture Manager, Program Manager, and Chief Engineer are responsible for collaborating with the Information Assurance (IA) subject matter expert (SME) for tailoring of security activities, milestones, and work products into engineering deliverables.**

New Lockheed Security Engineering Procedure

Security in the Engineering Life Cycle

IA SMEs have a role in all phases of the life cycle to ensure trade-offs don’t negatively affect the security baseline.

IA SMEs should, as appropriate,

- **Support program planning.**
- **Support development of all engineering deliverables.**
- **Participate in engineering peer reviewsand technical milestone reviews.**
- **Attend status meetings.**
- **Participate in lessons learned.**
- **Support in risk management activities.**
- **Support engineering and program configuration management activities and the configuration control board.**
- **Participate in trade study activities and product assessments.**
- **Share information assurance knowledge with proposal, development, and operations personnel.**

Security Work Products	FP Section	Proposal	ATP	Planning	PR	Requirements	SRR	Design	SDR	Development	TRR	Test	ORR	Deployment	PIR	O&M	RR	Retirement	Responsible Role	Supporting Roles	Engineering Deliverable
Security Engineering Needs Assessment (SENA)	5.1	D	Authorization to Proceed (ATP)	C	Planning Review (PR)	C	System Requirements Review (SRR)	C	System Design Reviews (SDR)	C	Test Readiness Review (TRR)	C	Operational Readiness Review (ORR)	C	Post Implementation Review (PIR)		Retirement Review (RR)		IA	PM, CE, CM	Program Management Proposal & Systems Engineering Management Plan
Security Engineering Cost Estimates	5.2	D		C		C		C		C		C		C					PM	IA, CE	Program Cost Proposal
Proposed Security Technical Solution	5.3	D																	IA	CE, HWE, SE, SWE, SA	Technical Solution Proposal
Security Risk Analysis	5.4	D		C		C		C		C		C		C		C			IA	PM, CE, SA, TE	Risk and Opportunity Management Plan
Security Request for Information (RFI)	5.7	D		C															IA	SE, SA	Request for Information
Security Concept of Operations	5.5			D		C		C		C		C		C		C			IA	HWE, SA	Operational Concept
Security Plan	5.8			D		C		C		C		C		C		C			IA	SE, SA	Systems Engineering Management Plan
Secure Coding Standards	5.9			D		C		C		C		C		C		C			SWE	IA, SA	Operational Concept
Threat Modeling Analysis	5.10			D		C		C		C		C		C		C			IA	CE, SE	Risk and Opportunity Management Plan
Certification and Accreditation (C&A) Implementation Plan	5.11			D		C		C		C		C		C					IA	PM, CE, SA	Systems Engineering Management Plan
Plans of Action and Milestones (POA&M)	5.12			D		C		C		C		C		C		C		CM	IA	PM, CE	Systems Engineering Management Plan
Contingency and Disaster Recovery (DR) Plan	5.13			D		C		C		C		C		C		C			PM	CE, IA, SA	Systems Engineering Management Plan
Security Requirements	5.14					D		C		C		C		C					IA	CE, HWE, SE, SWE, SA	Security Requirements Document
Security Requirements Traceability and Verification Matrix (RTVM)	5.14					D		C		C		C		C					IA	SE, TE	System RTVM
System Security Policy	5.15					D		C		C		C		C		C			IA	SE	System Policy
Security Test Cases	5.20					D		C		C		C		C					IA	TE	System Test Cases
Secure System Design	5.17							D		C		C		C		C			SA	IA, HWE, SWE	System Design
Security Test Plan	5.20									D		C							TE	IA	System Test Plan
Security Risk Assessment Report	5.4											D		C		C			IA	PM, CE,	Risk and Opportunity Management Report
Security Test Results	5.20											D		C					TE	IA	System Test Results
Certification and Accreditation (C&A) Package	5.11											D		C		C		C	IA	PM, CE	N/A
Security Retirement/Transition Plan	5.21																	D	IA	PM, CE	Engineering Retirement/ Transition Plan

Security Work Products	FP Section	Proposal	ATP	Planning	PR	Requirements	SRR	Design	SDR	Development	TRR	Test	ORR	Deployment	PIR	O&M	RR	Retirement	Responsible Role	Supporting Roles	Engineering Deliverable
Security Engineering Needs Assessment (SENA)	5.1	D		C															IA	PM, CE, CM	Program Management Proposal & Systems Engineering Management Plan
Security Engineering Cost Estimates	5.2	D		C															PM	IA, CE	Program Cost Proposal
Proposed Security Technical Solution	5.3	D																	IA	CE, HWE, SE, SWE, SA	Technical Solution Proposal
Security Risk Analysis	5.4	D		C															IA	PM, CE, SA, TE	Risk and Opportunity Management Plan
Security Request for Information (RFI)	5.7	D		C															IA	SE, SA	Request for Information
Security Concept of Operations	5.5			D															IA	HWE, SA	Operational Concept
Security Plan	5.8			D															IA	SE, SA	Systems Engineering Management Plan
Secure Coding Standards	5.9			D															SWE	IA, SA	Operational Concept
Threat Modeling Analysis	5.10		ed (ATP)	D															IA	CE, SE	Risk and Opportunity Management Plan
Certification and Accreditation (C&A) Impleme	5.11			D	PR)														IA	PM, CE, SA	Systems Engineering Management Plan
Plans of Action a (POA&																			IA	PM, CE	Systems Engineering Management Plan
Contingency a Recovery (										C		C		C		C			PM	CE, IA, SA	Systems Engineering Management Plan
Security Req										C	Test Readiness	C	Operational Readir	C	Post Implementa		Retirement		IA	CE, HWE, SE, SWE, SA	Security Requirements Document
Security Req Traceability anc Matrix (R										C		C		C					IA	SE, TE	System RTVM
System Secu										C		C		C		C			IA	SE	System Policy
Security Te										C		C		C					IA	TE	System Test Cases
Secure Syste										C		C		C		C			SA	IA, HWE, SWE	System Design
Security Te										D		C							TE	IA	System Test Plan
Security Risk A Repo												D		C		C			IA	PM, CE,	Risk and Opportunity Management Report
Security Tes												D		C					TE	IA	System Test Results
Certification and (C&A) Pa												D		C		C		C	IA	PM, CE	N/A
Security Retirem Plan																		D	IA	PM, CE	Engineering Retirement/ Transition Plan

Responsible & Supporting Roles

PM - Program Manager
CE - Chief Engineer
CM - Configuration Manager
IA – Information Assurance SME
HWE - Hardware Engineer
SE - Systems Engineer
SWE - Software Engineer
TE - Test Engineer
SA – Systems Architect

D - Security work product is developed
C - Security work product is maintained under version control

Phases & Gates

Proposal
- **ATP (Authorization to Proceed)**

Planning
- **PR (Planning Review)**

Requirements
- **SRR (System Requirements review)**

Design
- **SDR (System Design review)**

Development
- **TRR (Test readiness review)**

Test
- **ORR (Operational readiness review)**

Deployment
- **PIR (Post Implementation review)**

O&M
- **RR (Retirement review)**

Retirement


Static
System
Dynamic
Adversary

A flash mob is a pick-up group of people who assemble suddenly in a designated place to perform some collective activity, generally organized via telecommunications, social media, or viral emails. The first flash mobs were created in Manhattan in 2003 as a social experiment, by Bill Wasik, senior editor of *Harper's Magazine*.

Pillow fight flash mob in Downtown Toronto (2005)

2,000 people converged on Dupont Circle in Washington on 6Feb2010 for a snowball fight of epic proportions -- responding to messages posted on Facebook and Twitter

www.huffingtonpost.com/2010/02/07/dupont-circle-snowball-fi_n_452638.html www.nytimes.com/2010/03/25/us/25mobs.html

24Mar2010: Philadelphia Text-Message Flash Mob

www.nytimes.com/2010/03/25/us/25mobs.html

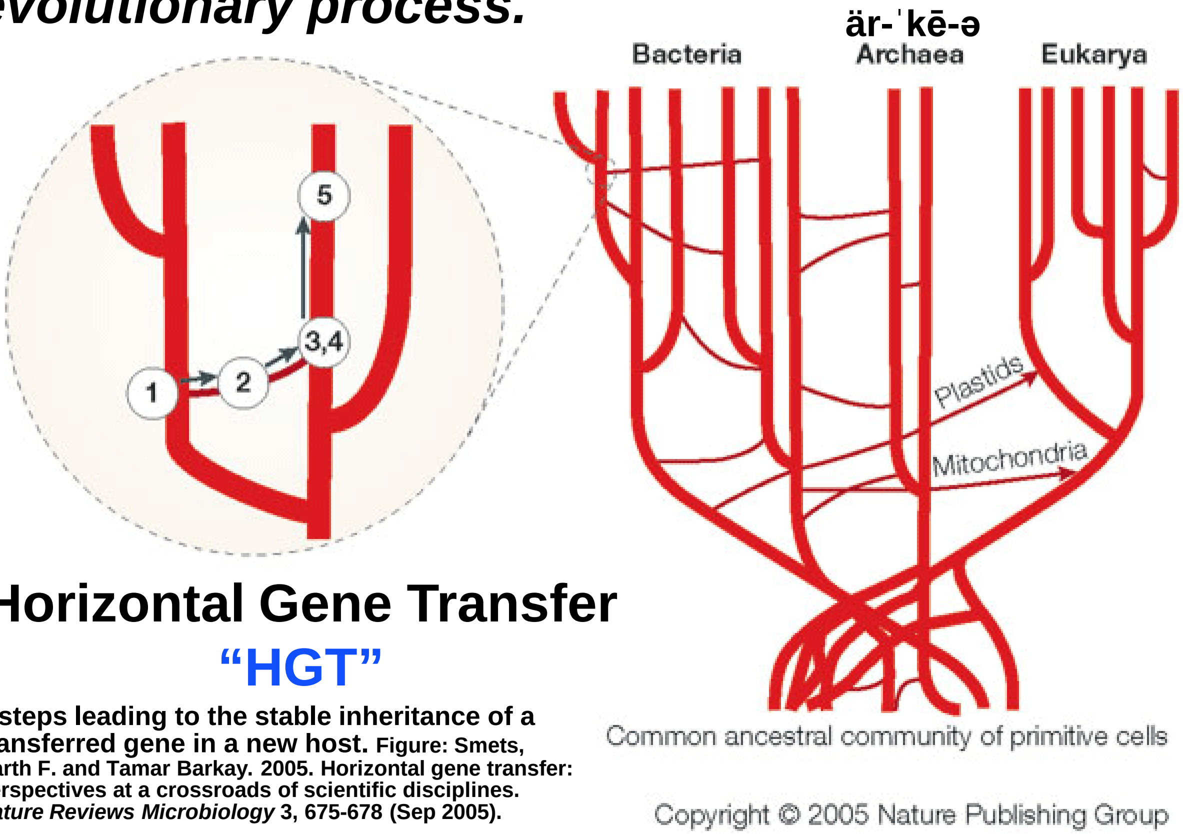
24Mar2010: Philadelphia Text-Message Flash Mob

Horizontal Meme Transfer (HMT)

Evolution and Innovation

Woese, Carl. 2000. Interpreting the universal phylogenetic tree. PNAS. 97(15):8392-6. www.ncbi.nlm.nih.gov/pmc/articles/PMC26958/pdf/pq008392.pdf

Carl Woese: "Vertically generated and horizontally acquired variation could be viewed as the yin and the yang of the evolutionary process.



Horizontal Gene Transfer

"HGT"

5 steps leading to the stable inheritance of a transferred gene in a new host. Figure: Smets, Barth F. and Tamar Barkay. 2005. Horizontal gene transfer: perspectives at a crossroads of scientific disciplines. *Nature Reviews Microbiology* 3, 675-678 (Sep 2005).

"The vast majority, between 88% and 98%, of the expansions of protein families are due to HGT [in eight studied prokaryote clades]"

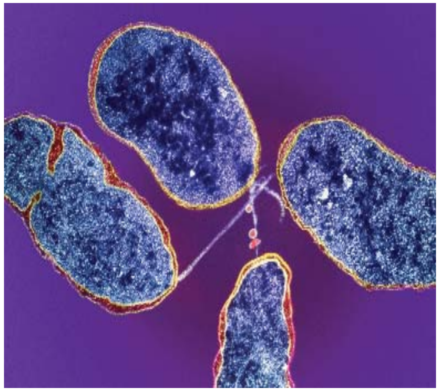
Treangen, Todd J. and Eduardo P. C. Rocha. 2011. Horizontal Transfer, Not Duplication, Drives the Expansion of Protein Families in Prokaryotes. PLoS Genetics 7:1, January.

"Vertically generated variation is necessarily highly restricted in character; it amounts to variations on a lineage's existing cellular themes.

Horizontal transfer, on the other hand, can call on the diversity of the entire biosphere, molecules and systems that have evolved under all manner of conditions, in a great variety of different cellular environments.

Thus, horizontally derived variation is the major, if not the sole, evolutionary source of true innovation."

Pattern: Horizontal Gene/Meme Transfer

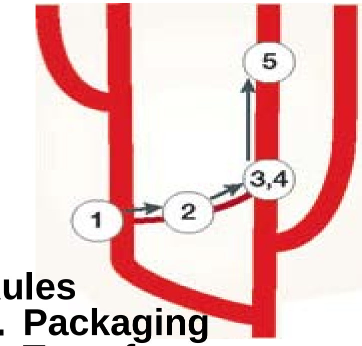


Available high variety cellular organisms

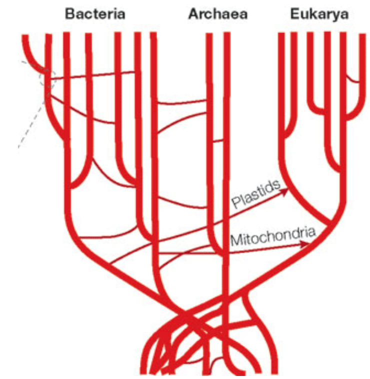
Intrachromsomal genes

Extrachromosomal genes

Two modular gene pools



Rules
1. Packaging
2. Transfer
3. Entry
4. Establishment
5. Inheritance



Innovative adaptation and evolution

Horizontal gene transfer speeds up innovative adaptation and evolution

Context: When conditions deteriorate, it makes a lot of sense to try to scavenge DNA from your neighbors. Horizontal gene transfer facilitates a fast microbial adaptation to stress. Higher-than-suspected transfer rates among microbes living in nutrient-poor environments, where sharing genes may be key to survival, has been observed. Evidence indicates that organisms limit gene exchange to microbes on nearby branches of the family tree, probably because their chromosomes share certain characteristics. Genes appear to be exchanged between species with similar chromosomal structures (Pennise 2011).

Problem: Situational or environmental changes that threaten fitness or survival.

Forces: Short-term adaptability vs. long-term-evolvability, horizontal gene transfer speeds the development of new traits by a factor of 10,000 (Woese 2000, Pennise 2011).

Solution: Incorporate appropriate material from other domains that have developed compatible and useful situational fitness. Mobile concepts don't just help a community survive, they also provide the grist for evolutionary innovations.

Pattern: Horizontal Gene/Meme Transfer

- **Self organization** is driven by natural selection of useful gene/meme transfer.
- **Adaptation** to immediate needs as latent genes/memes get expressed/employed.
- **Reactive resilience** occurs with sufficient gene/meme mix to meet needs.
- **Evolution** occurs in gene/meme mix with persistent expression and inheritance.
- **Proactive innovation** occurs with speculative acquisitions and assemblies.
- **Harmony** is maintained with a robust Highly-Optimized-Tolerance (Carlson & Doyle 2002) assembly repertoire.

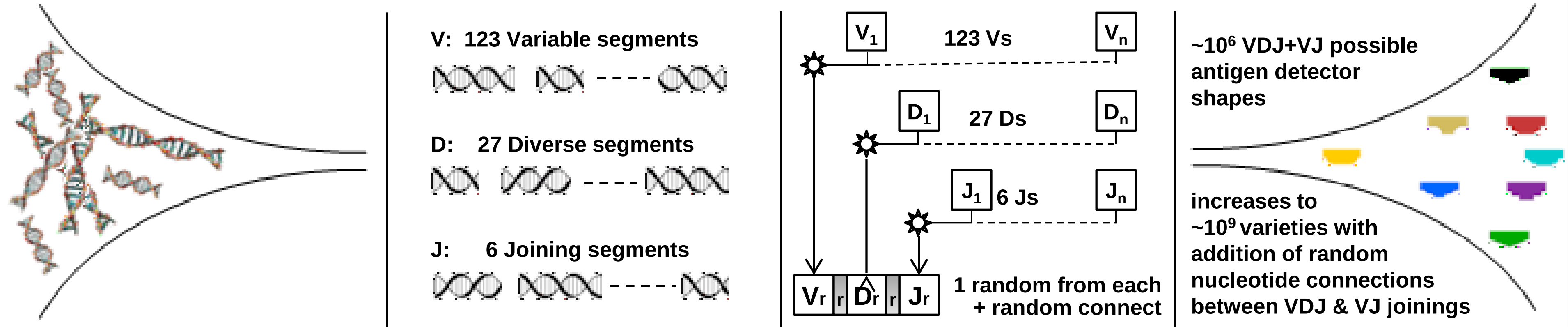
Examples:

- **Horizontal gene transfer and evolution (Woese 2000) & (Smets 2005),** www.ncbi.nlm.nih.gov/pmc/articles/PMC26958/pdf/pq008392.pdf www.nature.com/nrmicro/journal/v3/n9/pdf/nrmicro1253.pdf
- **Cross-domain user-behavior-channeling pattern-catalog (Lockton 2009, 2010),** http://bura.brunel.ac.uk/bitstream/2438/3664/1/Lockton_SI_paper_disclaimer_added.pdf, http://danlockton.com/dwi/Download_the_cards
- **Cross-domain dynamic-system process-pattern project (Troncale 1978, 2006),** www.allbookstores.com/author/International_Conference_On_Applied_General_Systems_Research_State_Uni.html, www3.interscience.wiley.com/journal/112635373/abstract?CRETRY=1&SRETRY=0.
- **Universal patterns in human activity and insurgent events (Bohorquez 2009),** www.nature.com/nature/journal/v462/n7275/full/nature08631.html.
- **Patterns in behavioral ecology and anti-predator behavior (Blumstein 2010)** www.eeb.ucla.edu/Faculty/Blumstein/pdf%20reprints/Blumstein_2010_BE.pdf .
- **Robustness and fragility tradeoffs in evolving complex systems (Carlson & Doyle 2000),** www.pnas.org/content/99/suppl.1/2538.full.pdf+htm.

Some Inspirational Patterns

from natural systems that effectively process noisy sensory input from uncertain and changing environments

Pattern: Bow Tie Processor (assembler/generator/mediator)



Available high variety genetic DNA input

Evolve three fixed V-D-J gene-segment libraries

Fixed-rule VDJ assembly with random interconnects

Random high variety output with VDJ + VJ assemblies

Millions of random infection detectors generated continuously by fixed rules and modules in the "knot"

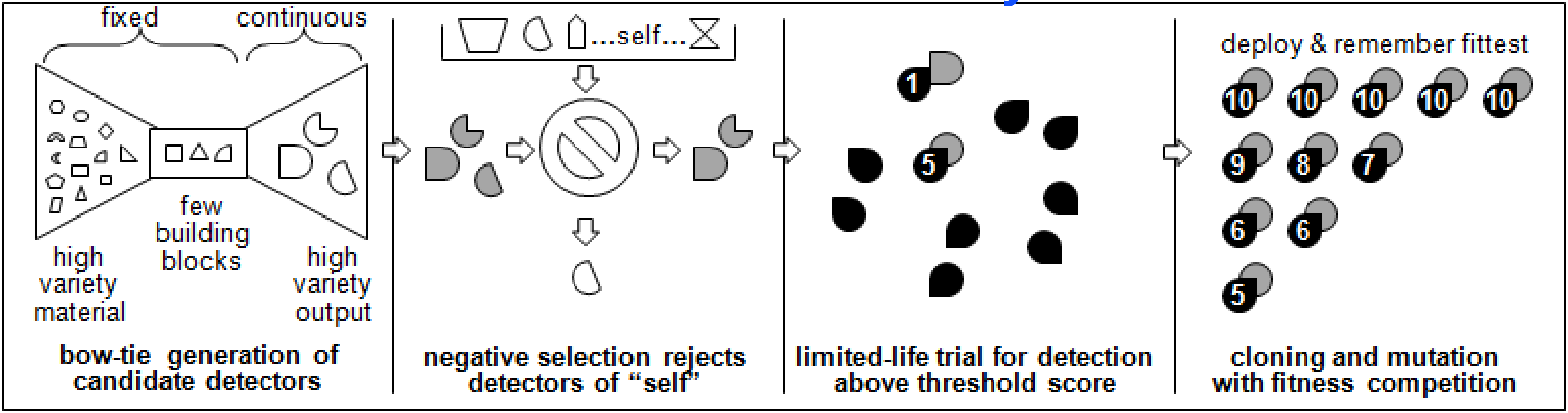
Context: Complex system with many diverse inputs and many diverse outputs, where outputs need to respond to many needs or innovate for many or unknown opportunities, and it is not practical to build unique one-to-one connections between inputs and outputs. Appropriate examples include common financial currencies that mediate between producers and consumers, the adaptable biological immune system that produces proactive infection detectors from a wealth of genetic material, and the Internet protocol stack that connects diverse message sources to diverse message sinks.

Problem: Too many connection possibilities between available inputs and useful outputs to build unique robust, evolving satisfaction-processes between each.

Forces: Large knot short-term-flexibility vs small knot short-term-controllability and long-term-evolvability (Csete 2004); robustness to known vs fragility to unknown (Carlson 2002).

Solution: Construct relatively small "knot" of fixed modules from selected inputs, that can be assembled into outputs as needed according to a fixed protocol. A proactive example is the adaptable immune system that constructs large quantities of random detectors (antigens) for unknown attacks and infections. A reactive example is a manufacturing line that constructs products for customers demanding custom capabilities.

Pattern: Proactive Anomaly Search



Speculative detector generation/mutation finds new attacks in biological immune system

Context: A complex system or system-of-systems subject to attack and infection, with low tolerance for attack success and no tolerance for catastrophic infection success; with resilient remedial action capability when infection is detected. Appropriate examples include biological organisms, and cyber networks for military tactical operations, national critical infrastructure, and commercial economic competition.

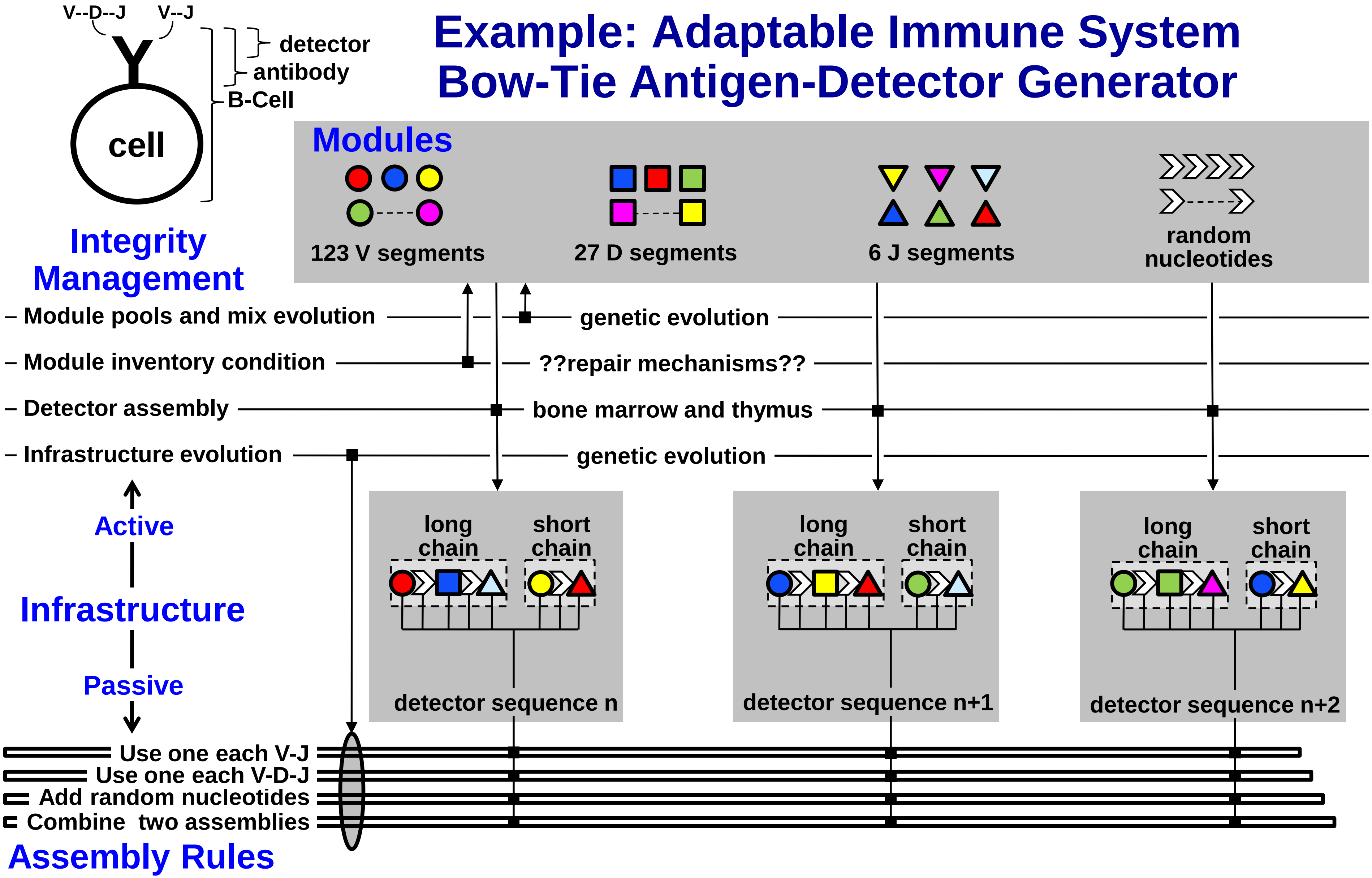
Problem: Directed attack and infection types that constantly evolve in new innovative ways to circumvent in-place attack and infection detectors.

Forces: False positive tradeoffs with false negatives, system functionality vs functionality impairing detection measures, detectors for anything possible vs added costs of comprehensive detection, comprehensive detection of attack vs cost of false self detection.

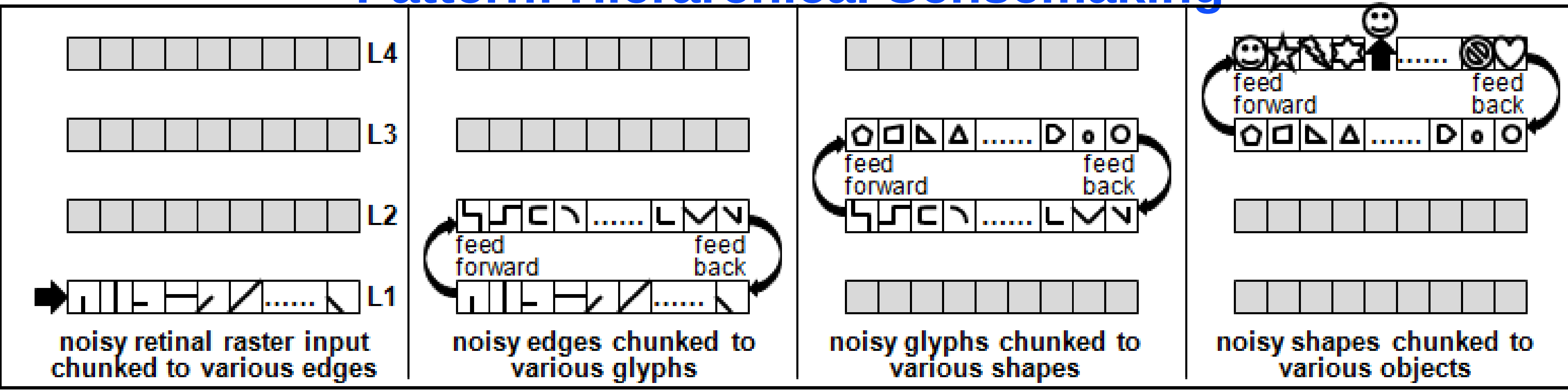
Solution: A high fidelity model of biological immune system antibody (detection) processes that generate high quantity and variety of anticipatory speculative detectors in advance of attack and during infection, and evolve a growing memory of successful detectors specific to the nature of the system-of-interest.

Pattern: Drag-and-Drop Framework and Modules

Example: Adaptable Immune System
Bow-Tie Antigen-Detector Generator



Pattern: Hierarchical Sensemaking



Four level feed forward/backward sense-making hierarchy modeled on visual cortex

Context: A decision maker in need of accurate situational awareness in a critical dynamic environment. Examples include a network system administrator in monitoring mode and under attack, a military tactical commander in battle, and the NASA launch control room.

Problem: A very large amount of low-level noisy sensory data overwhelms attempts to examine and conclude what relevance may be present, most especially if time is important or if sensory data is dynamic.

Forces: amount of data to be examined vs time to reach a conclusion, number of ways data can be combined vs number of conclusions data can indicate, static sensory data vs dynamic sensory data, noise tolerated in sensory data vs cost of low noise sensory data.

Solution: Using a bow-tie process, each level looks for a specific finite set of data patterns among the infinite possibilities of its input combinations, aggregating its input data into specific chunks of information. These chunks are fed-forward to the next higher level, that treats them in turn as data further aggregated into higher forms of information chunks. Through feedback, a higher level may bias a lower level to favor certain chunks over others, predicting what is expected now or next according to an emerging pattern at the higher level. Each level is only interested in a small number of an infinite set of data-combination possibilities, but as aggregation proceeds through multiple levels, complex data abstractions and recognitions are enabled.

How to Recognize a Pattern

Bob Blakley, Craig Heath, and members of The Open Group Security Forum,
Technical Guide* – Security Design Patterns, *The Open Group, 2004

[page 6] You might be wondering how you recognize a pattern if someone else hasn't already written it down. Jim Coplien recommends asking whether a solution to a problem has the following properties (if it has, it might be a pattern!):

- **Is it a solution to a problem in a context?**
- **Can you tell the problem solver what to do in order to solve the problem?**
- **Is it a mature, proven solution? In this context, ''proven'' means it has been used multiple times by architects and designers who are familiar with proper use of design patterns and on all occasions has not been found to be flawed in any way.**
- **Is it something you did not invent yourself?**
- **Does the solution build on the insight of the problem solver, and can it be implemented many times without ever being the same twice?**
- **Can the solution be formalized or automated? If it can be formalized or automated, then do that instead of writing it as a pattern.**
- **Does it have a dense set of interacting forces that are independent of the forces in other patterns?**
- **Is writing it down hard work? If it is easy to write, it may not be a pattern, or it is likely that you have not thought hard enough about the forces that bear down on the situation.**

To Start: Mirror the Enemy

Agile system security, as a minimum, must mirror the agile characteristics exhibited by the system attack community:

[S] Self-organizing – with humans embedded in the loop, or with systemic mechanisms.

[A] Adapting to unpredictable situations – with reconfigurable, readily employed resources.

[R] Reactively resilient – able to continue, perhaps with reduced functionality, while recovering.

[E] Evolving in concert with a changing environment – driven by vigilant awareness and fitness evaluation.

[P] Proactively innovative – acting preemptively, perhaps unpredictably, to gain advantage.

[H] Harmonious with system purpose – aiding rather than degrading system and user productivity.

www.parshift.com/Files/PsiDocs/Pap100226-AgileSecuritySelfOrganizingCoEvolution-ExtAbst.pdf

Our Pattern Form

Name:	Descriptive name for the pattern.
Context:	Situation that the pattern applies to.
Problem:	Description of the problem.
Forces:	Tradeoffs, value contradictions, constraints, key dynamics of tension & balance.
Solution:	Description of the solution.
Graphic:	A depiction of response dynamics.
Examples:	Referenced cases where the pattern is employed.
Agility:	Evidence of SAREPH characteristics that qualify the pattern as agile.
References:	Literature access to examples.

www.parshift.com/Files/PsiDocs/Pap100317Cser-OnDiscoveryAndDisplayOfAgileSecurityPatterns.pdf

Grounding the Pattern Investigation
with a
Hierarchical Sense-Making IDS Application

SornS
(self-organizing resilient network sensing)

a maturing platform for many
work-in-process self-organizing security patterns

Chosen Because

New associative memory pattern processor opens new opportunities:

- **parallel: any number of simultaneous patterns-in-process**
- **capacity: one chip = millions of multi-feature patterns**
- **complexity: FSM patterns of any length and structure**
- **scalable: unbounded multiple chips**
- **speed: data stream rates (but not optical…yet)**
- **switching: instant save and restore for interleaved flow work-in-process**
- **Wattage: low**
- **cost: low**

Timely irresistible challenge:

Human expertise is pattern based, not reasoning based

Expertise appears to require 200,000 to 1 million patterns

Recognition needs to be instant

Learning can take some time

Knowledge must improve in time

Cortical pattern recognition appears to be hierarchical temporal memory

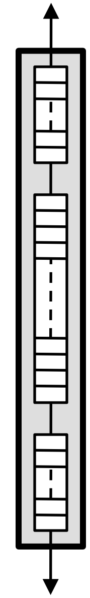
Patterns appear to consist of sparse (minimal overlap) anomalies (relatively rare occurrences)

Reconfigurable Associative-Memory Pattern Processor

Reusable Cells Reconfigurable in a Scalable Architecture

Independent multi-feature detector (MFD) content addressable by current input byte

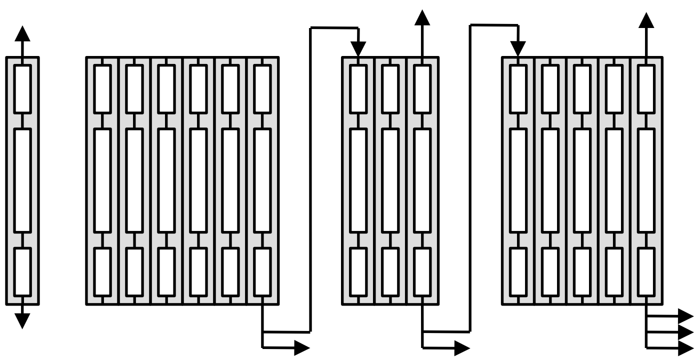
If active, and satisfied with current byte, can activate other designated MFDs including itself



Pattern-satisfaction output pointers

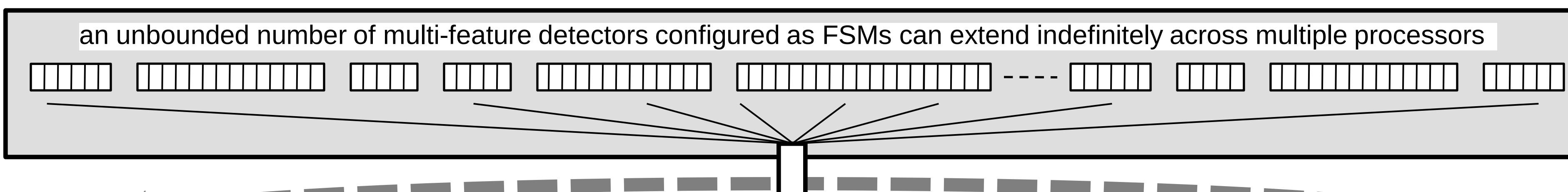
Up to 256 possible features can be "satisfied" by any/all "feature indicators" that are set

Feature-satisfaction activation pointers



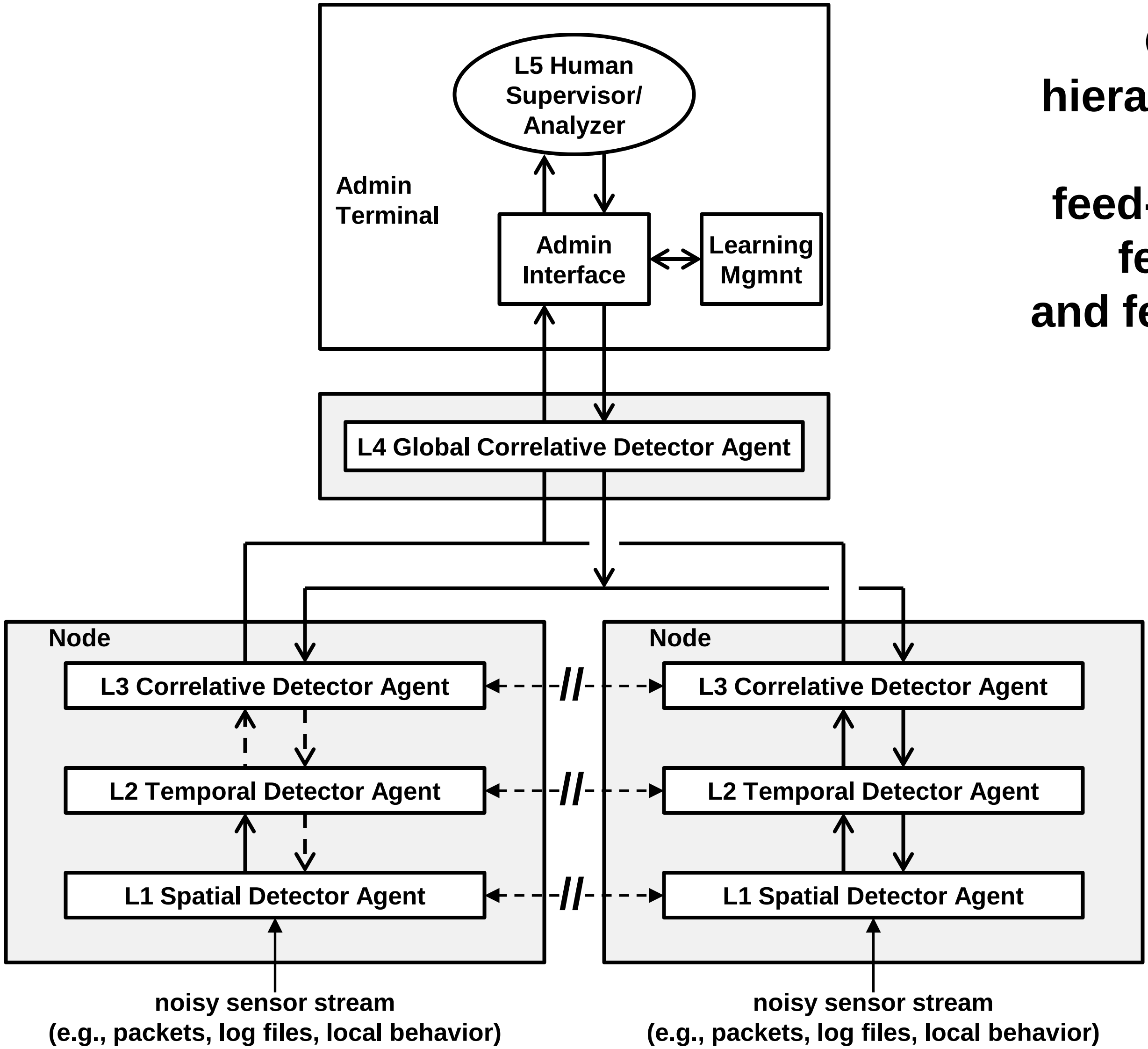
Individual MFDs are configured into finite state machines *(FSMs)* by linking activation pointers

- **Unbounded pattern capacity**
- **Detection at full data stream speed**
- **Simultaneous detection of all active patterns**



All <u>active</u> MFDs have simultaneous access to current data-stream byte

Grounded Application Target
Self Organizing Resilient Network Sensing (SornS)



General purpose hierarchical sense-making with feed-forward recognition, feed-back learning, and feed-side collaboration

Correlative: n events in any order.

Temporal: n events in order.

Spatial: n events in contiguous order.

Micron Announces Development of New Parallel Processing Architecture

Automata Processor Addresses Application Domains at the Frontier of High-Performance Computing

DENVER, Nov. 18, 2013 (GLOBE NEWSWIRE) -- Supercomputing 2013 — Micron Technology, Inc. (Nasdaq:MU), one of the world's leading providers of advanced semiconductor solutions, today announced the development of a fundamentally new computing architecture capable of performing high-speed, comprehensive search and analysis of complex, unstructured data streams. Micron's Automata Processor (AP) is an accelerator that leverages the intrinsic parallelism of memory and aims to dramatically advance computing capabilities in areas such as bioinformatics, video/image analytics, and network security which pose challenges for conventional processor architectures because of the amount of complex, unstructured data.

Automata processing exploits the natural parallelism found in Micron's semiconductor devices to deliver a level of parallelism previously unattainable with legacy architectures. Unlike conventional CPUs, the AP is a computing fabric comprised of tens of thousands to millions of processing elements interconnected to create a task-specific processing engine capable of solving problems with unprecedented performance. "Micron has an intense focus on the development of innovative and advanced silicon solutions that help our customers solve their most challenging computing problems," said Brian Shirley, vice president of Micron's DRAM Solutions Group. "This announcement is a huge step forward for Micron and has the potential to unleash unprecedented levels of computing power."

"The Automata Processor is a breakthrough technology that is designed to use advanced memory-based processing to solve complex computing challenges that existing solutions are not able to tackle effectively," said Chirag Dekate, IDC Research Manager, HPC. "This technology has the potential to solve some of the world's most complicated data-intensive challenges, including real-time security that could dramatically affect anti-terrorism efforts, or the highly efficient analysis of complex plant genomes that could allow scientists to rapidly advance their research agendas beyond what is possible today."

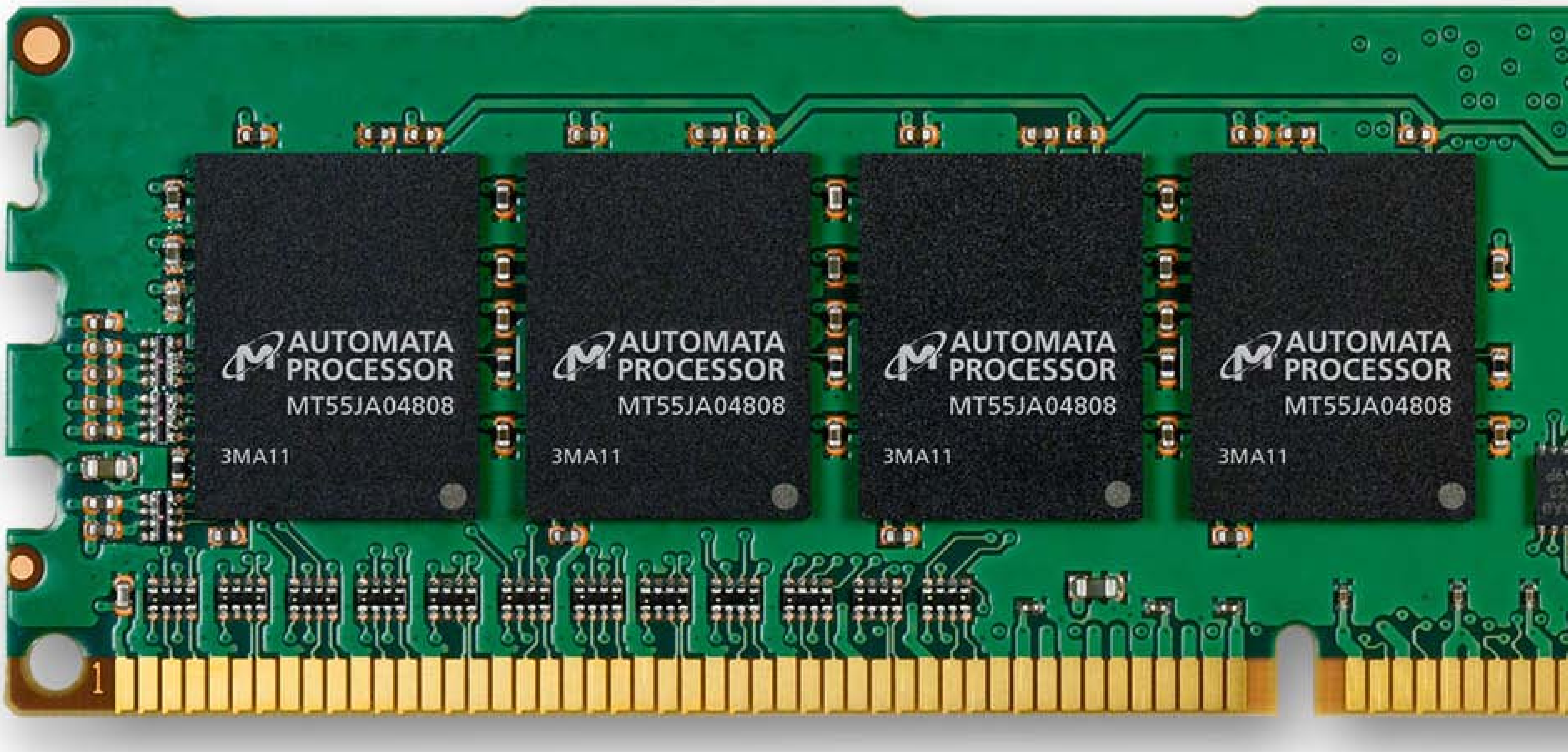
Micron is working closely with ecosystem partners and research institutions to grow awareness and engagement for this new technology. Srinivas Aluru, professor of computational science and engineering at Georgia Institute of Technology and a leader in the field of high-performance, computational biology, has been deeply involved in early research efforts using the AP to solve problems associated with applications in bioinformatics. "Micron's Automata Processor offers a refreshingly new way of solving problems that is very different from all other accelerator technologies," said Aluru. "By deploying this in interesting ways, we have been able to solve a much larger instance of the NP-hard biological motif-finding problem than was previously reported, using the resources within a single Automata Processor board."

Michela Becchi, an assistant professor in the Department of Electrical and Computer Engineering with joint appointments in Computer Science and the Informatics Institute of the University of Missouri, has focused her efforts on using the AP to address the challenges involved with implementing high-speed regular expression matching engines. "Micron's Automata Processor combines the advantages of NFA-based designs and memory-based solutions to achieve regular expression matching," said Becchi. "Specifically, it efficiently supports large sets of complex regular expressions, while providing worst-case processing guarantees and support for multiple input streams. In addition, it comes with a user-friendly programming tool-chain that can be easily integrated with existing tools for regular expression processing." Additionally, Micron and the University of Virginia today announced an agreement to establish the Center for Automata Computing at the University of Virginia.

Availability – Graphic design and simulation tools, along with a software development kit (SDK), to enable developers to design, compile, test, and deploy their own applications using the AP will be available in 2014.

Micron's Automata Processor www.youtube.com/watch?feature=player_embedded&v=5guIAaOrMhk


AUTOMATA
PROCESSOR
MT55JA04808
3MA11


AUTOMATA
PROCESSOR
MT55JA04808
3MA11

Micron Announces

Here's links to some of the press pickup – the paper given at the Supercomputer Conference is attached.

The Room 5 announcement is interesting – they have built a graphical development environment.

Also the ZDNet reference to: Micron and the University of Virginia have announced an agreement to establish the center for Automata computing at the University of Virginia.

And…I love the pun: "Non von Neumann Architecture"

http://www.forbes.com/sites/jimhandy/2013/11/18/is-micron-gunning-for-intels-processor-business/

http://www.zdnet.com/micron-develops-new-parallel-processing-architecture-7000023321/

www.wnct.com/story/24022205/room-5-develops-apworkbench-for-micron-technologys-new-automata-processor

http://www.eetimes.com/document.asp?doc_id=1320124

http://www.theregister.co.uk/2013/11/19/micron_automata_processor_launch/

http://www.linkedin.com/jobs2/view/9752899?trk=jobs_search_public_seo_page

www.xbitlabs.com/news/cpu/display/20131118124133_Micron_Announces_Development_of_New_Parallel_Processing_Architecture.html

www.pcworld.com/article/2065560/micron-challenges-conventional-computer-architecture-with-new-chip.html

www.v3.co.uk/v3-uk/news/2307830/micron-unveils-parallel-computing-architecture-with-automata-processor

PCWorld

Nov 20, 2013

The Register®

PROCESSORS

Micron challenges conventional computer architecture with new chip

Agam Shah, IDG News

Micron is challenging conventional computer architectures conceived decades ago with Automata, a highly parallel processor that can change its behavior to process the task at hand.

www.pcworld.com/article/2065560/micron-challenges-conventional-computer-architecture-with-new-chip.html

www.theregister.co.uk/2013/11/19/micron_automata_processor_launch/

Micron takes on Intel with 'breakthrough' processor for streaming data

Automata processor cuts through NP-hard problems like they're butter

By Jack Clark, 19th November 2013

Potential applications of the chip include genomic analysis and various security tasks such as the classification of online video footage.

A Massively Parallel Computing Solution

http://www.micron.com/about/innovations/automata-processing

The Challenge of Complex, Unstructured Data

Many of today's most challenging computer science problems involve very large data structures, unstructured data, random access, or real-time data analysis. These computationally intensive problems are not well aligned with traditional CPU and memory system architectures; they require a fundamentally new approach to computing. Micron's Automata Processor is a massively parallel computer architecture that provides dramatic processing efficiencies.

Related Resources

An Efficient and Scalable Semiconductor Architecture for Parallel Automata Processing
A technical paper presenting the design and development of the Automata Processor, to appear in *IEEE Transactions on Parallel and Distributed Systems.* Download

Appendix: Supplementary material, including additional details about the Automata Processor architecture and an extensive review of relevant literature. Download

Supplementary Video Simulations: Video captures from runs of the Micron AP Workbench, a visual editor for creating, visualizing and simulating Micron's Automata Processor. View the videos

Patterns Influencing SornS Development

1. **Dynamic Phalanx Shield**
   www.parshift.com/Files/PsiDocs/Pap100317Cser-OnDiscoveryAndDisplayOfAgileSecurityPatterns.pdf
2. **Peer-Peer Behavior Monitoring**
   www.parshift.com/Files/PsiDocs/Pap100317Cser-OnDiscoveryAndDisplayOfAgileSecurityPatterns.pdf
3. **Swarming Threat Sensors**
   ww.parshift.com/Files/PsiDocs/Pap100317Cser-OnDiscoveryAndDisplayOfAgileSecurityPatterns.pdf
4. **Drag-and-Drop Modules and Framework**
   www.parshift.com/Files/PsiDocs/PatternQualificationsForAgileSecurity.pdf
5. **Hierarchical Sense Making**
   www.parshift.com/s/110411PatternsForSORNS.pdf
6. **Proactive Anomaly Search**
   www.parshift.com/s/110411PatternsForSORNS.pdf
7. **Bow Tie Processor (assembler/generator/mediator)**
   www.parshift.com/Files/PsiDocs/PatternQualificationsForAgileSecurity.pdf
8. **Crowd Sourced Incident Reporting**
   www.parshift.com/s/110620ArchitecturalPatternsForSOSoS.pdf
9. **Swarm Discovery and Cooperation**
   www.parshift.com/s/110620ArchitecturalPatternsForSOSoS.pdf
10. **Collaborative Learning**
    www.parshift.com/s/110620ArchitecturalPatternsForSOSoS.pdf
11. **Stigmergic Interaction (4 sub-types)**
    www.parshift.com/s/110620AdversarialStigmergyPatterns.pdf
12. **Horizontal Gene/Meme Transfer**
    webinar: www.parshift.com/s/TowardsSystemicWillToLive.pdf
13. **Genetic Algorithm, ICCST 2012**
    www.parshift.com/s/121015ICCST-GeneticAlgorithm.pdf
14. **Combined Genetic-Algorithm-Neural-Network, ICCST 2012**
    www.parshift.com/s/121015ICCST-GANN.pdf
15. **Quorum Sensing, ICCST 2012**
    www.parshift.com/s/121015ICCST-QuorumSensing.pdf