Webinar

Agile Systems and Processes 103: Fleshing out Architecture with Design Principles, Activities, and Closure

10 September 2014

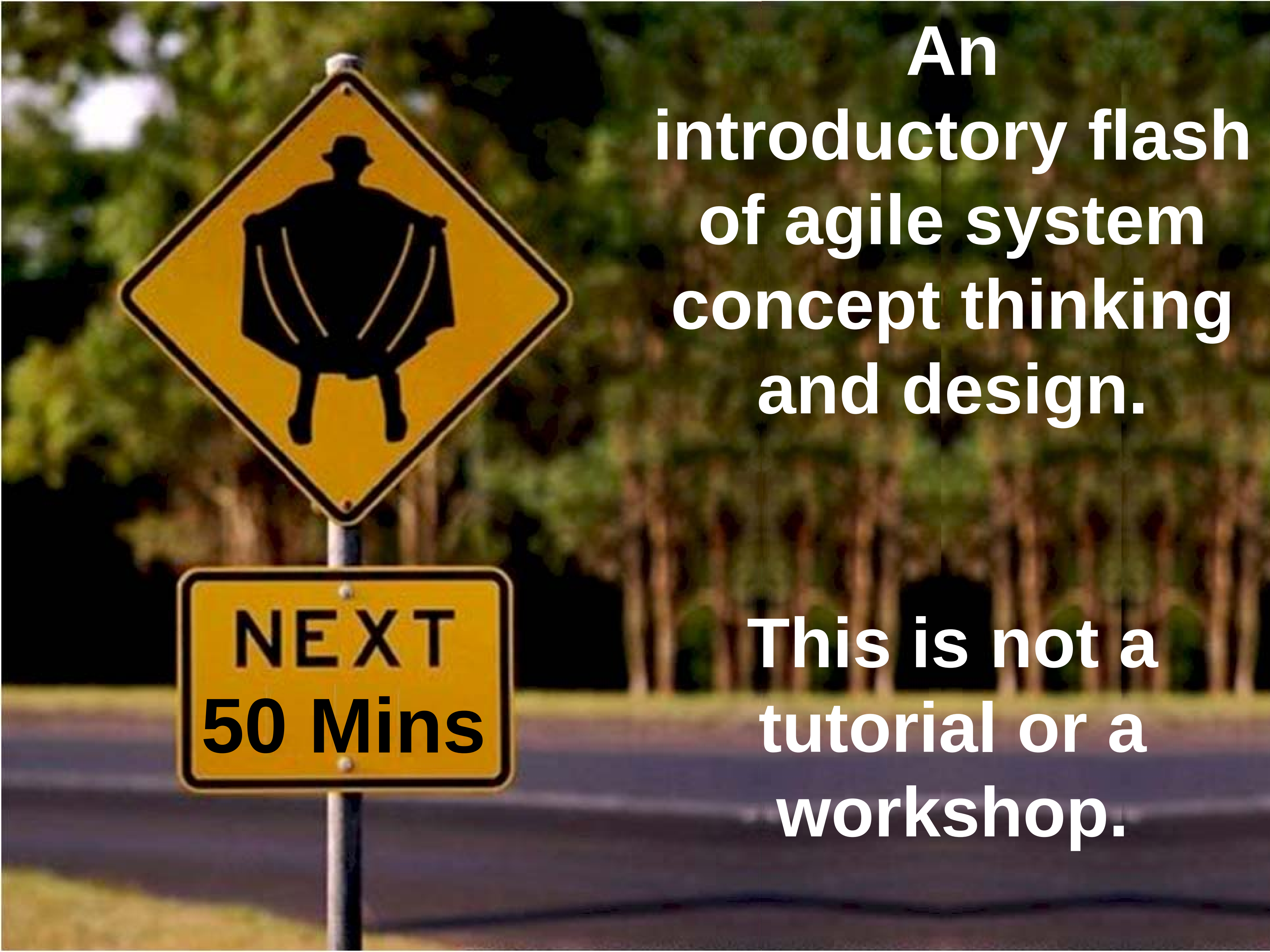
Rick Dove
Taos County, New Mexico, dove@parshift.com, 575-586-1536
CEO/CTO, Paradigm Shift International
Adjunct Professor, Stevens Institute of Technology
Chair: INCOSE WG on Agile Systems & Systems Engineering

Download 103 webinar slides: www.parshift.com/s/AgileSystems-103.pdf
Download 102 webinar slides: www.parshift.com/s/AgileSystems-102.pdf
Download 101 webinar slides: www.parshift.com/s/AgileSystems-101.pdf
(updated asynchronously from time-to-time)

Agile Systems & Processes 103:
Fleshing out Architecture with Design Principles, Activities, and Closure

Abstract: Agility is enabled and maintained by a fundamentally necessary and sufficient common structural architecture in systems of all kinds; from agile development and deployment processes, to the agile systems and products that are deployed. Architecture is viewed as structure and strategy. This webinar will focus on the strategy of fleshing out an agile structural architecture, reviewing fundamental design principles that drive system functional processes, and a method that brings closure to the basic design concepts. Examples will be drawn from agile systems and from agile engineering processes in a variety of domains. The presentation will lead off with a quick review of agile 101 (architecture fundamentals) and agile 102 (requirements fundamentals).

Bio: Rick Dove, INCOSE Fellow, was co-PI on the original work which identified Agility as the next competitive differentiator, funded by the US Office of the Secretary of Defense through the Navy in 1991 at Lehigh University. He went on to organize and lead the US DARPA-funded industry collaborative research at Lehigh University's Agility Forum, developing fundamental understandings of what enables and characterizes system's agility. He authored Response Ability – The Language, Structure, and Culture of the Agile Enterprise (Wiley, 2001). He has employed these agile concepts in both architecture and program management for large enterprise IT systems, for rapid manufacturing systems and services, and for self-organizing security strategies. Through Stevens Institute of Technology he teaches two 40-hour graduate courses in basic and advanced agile-systems and agile systems-engineering, at client sites. Through Paradigm Shift he provides training workshops and strategy development services. He chairs the INCOSE working groups on Agile Systems and Systems engineering, and on Systems Security Engineering.



An introductory flash of agile system concept thinking and design.

This is not a tutorial or a workshop.

Agile-Systems Research Focus – 1991+

Problem:

- Technology and markets are changing faster than the ability to employ/accommodate

- Life cycle requirements are uncertain and unpredictable

- Flexible system approaches inadequate when requirements change

- New approach needed that could extend usefulness/life of systems

Solution Search:

- Examined 100s of systems of various types

- Looked for systems that responded *effectively*

- Looked for metrics that defined *effectively*

- Looked for categories of response types

- Looked for principles that enabled response

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

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

Agility - Fundamentally

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

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

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

You can think of Agility as Requisite Variety.

You can think of Agility as proactive Risk Management.

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

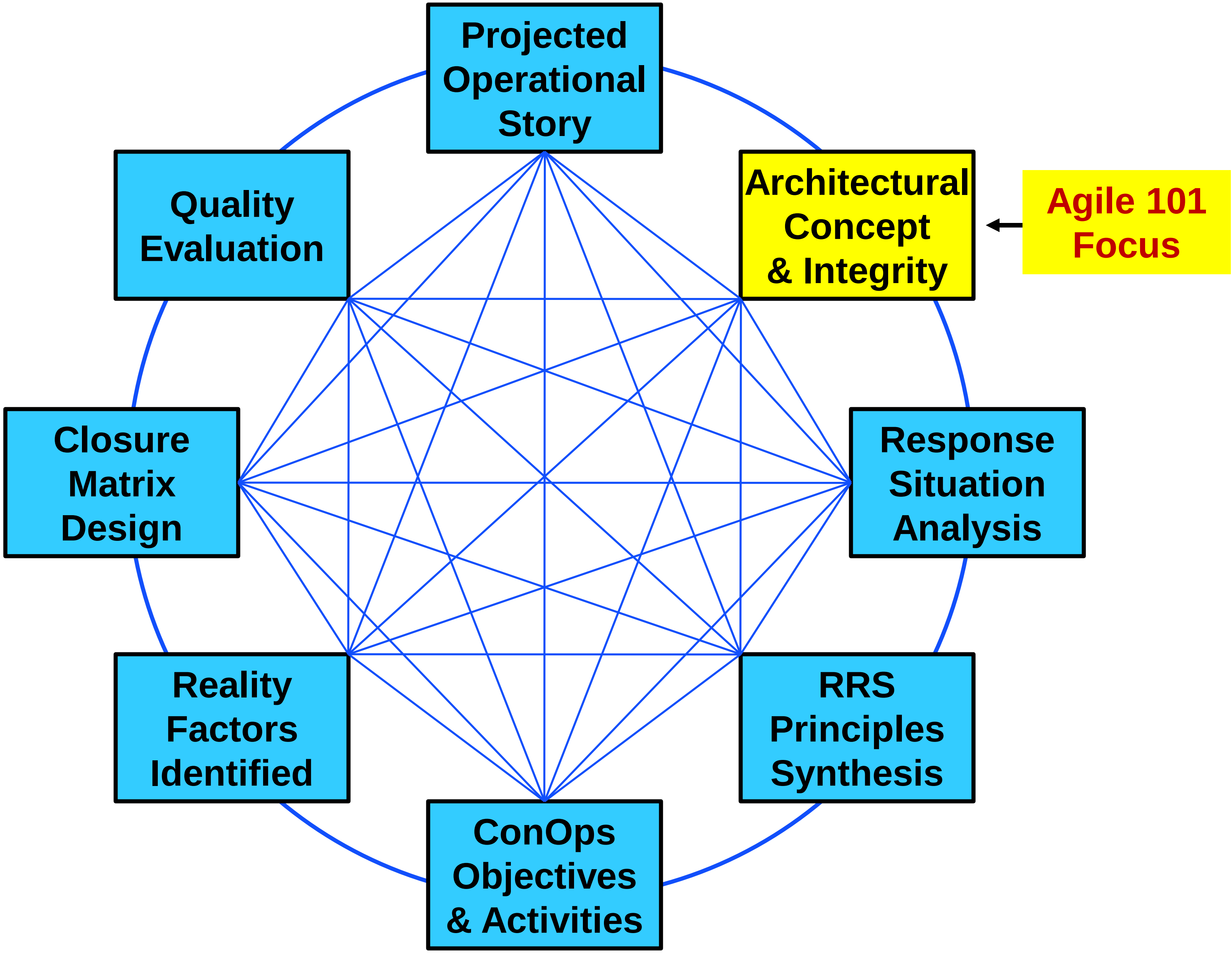
You can think of Agility as Life Cycle Extension.

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

Domain Independent

rick.dove@parshift.com, attributed copies permitted

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



Projected Operational Story

Architectural Concept & Integrity

Agile 101 Focus

Response Situation Analysis

RRS Principles Synthesis

ConOps Objectives & Activities

Reality Factors Identified

Closure Matrix Design

Quality Evaluation

rick.dove@parshift.com, attributed copies permitted

Reviewing 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/

rick.dove@parshift.com, attributed copies permitted

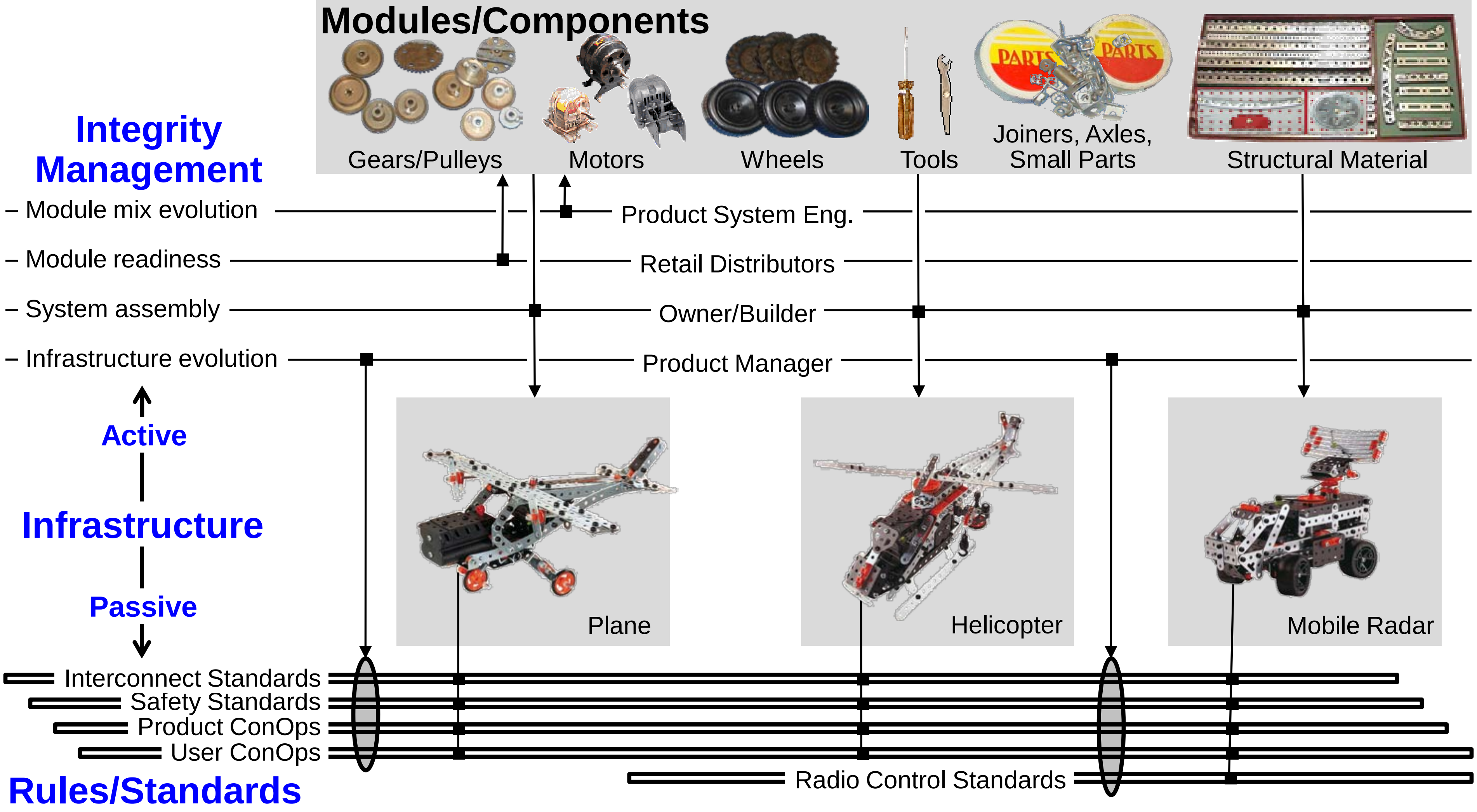
Here’s a Box of Bones

http://www.vintagetoys.com/toys/classified/962

www.girdersandgears.com/erector-1958-10052.html

Here is a System-Construction-Kit System

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



Modules/Components

Gears/Pulleys

Motors

Wheels

Tools

Joiners, Axles, Small Parts

Structural Material

Integrity Management

- Module mix evolution
- Module readiness
- System assembly
- Infrastructure evolution

Product System Eng.

Retail Distributors

Owner/Builder

Product Manager

Active

Infrastructure

Passive

Plane

Helicopter

Mobile Radar

Interconnect Standards

Safety Standards

Product ConOps

User ConOps

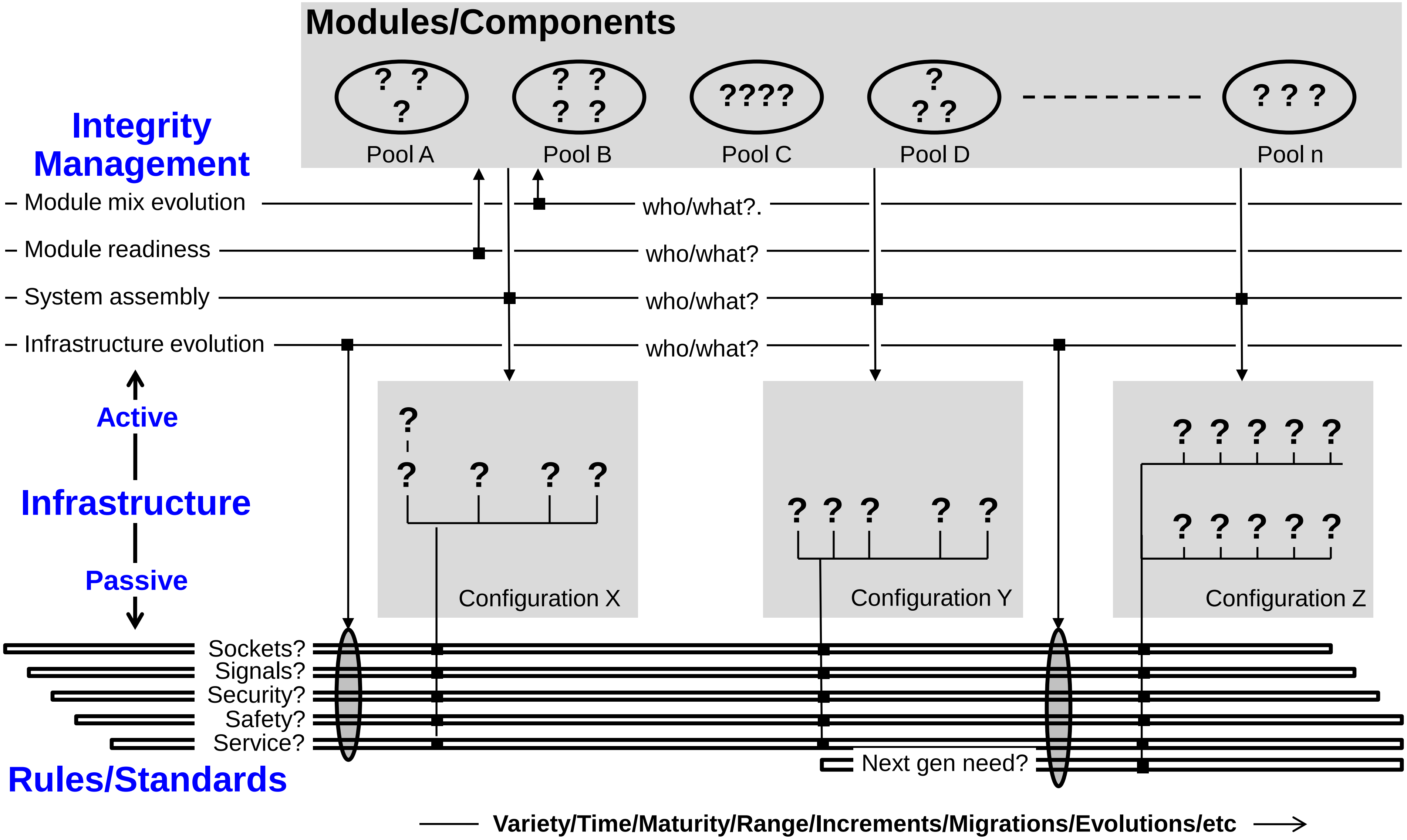
Radio Control Standards

Rules/Standards

rick.dove@parshift.com, attributed copies permitted

Developing the System-Construction-Kit System Architecture

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



Modules/Components

? ? ?
Pool A

? ? ? ?
Pool B

????
Pool C

? ? ?
Pool D

? ? ?
Pool n

Integrity Management

- Module mix evolution — who/what?.
- Module readiness — who/what?
- System assembly — who/what?
- Infrastructure evolution — who/what?

Active

Infrastructure

Passive

? ? ? ? ?
Configuration X

? ? ? ? ?
Configuration Y

? ? ? ? ? ? ? ? ? ?
Configuration Z

Sockets?
Signals?
Security?
Safety?
Service?
Next gen need?

Rules/Standards

Variety/Time/Maturity/Range/Increments/Migrations/Evolutions/etc

rick.dove@parshift.com, attributed copies permitted

System ______________________________

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

Module Mix Evolution:

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

Module Readiness :

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

System Assembly/Reconfiguration:

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

Infrastructure Evolution:

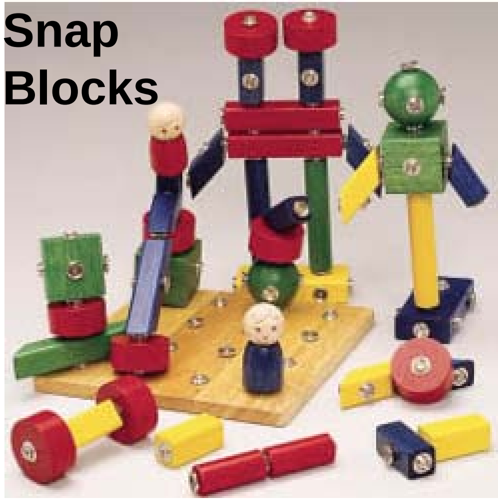
- Who (or what process) is responsible for evolving the passive and active infrastructures as new rules and standards become appropriate to enable next generation capability.

Bendables

Straws and Connectors

Marble Run

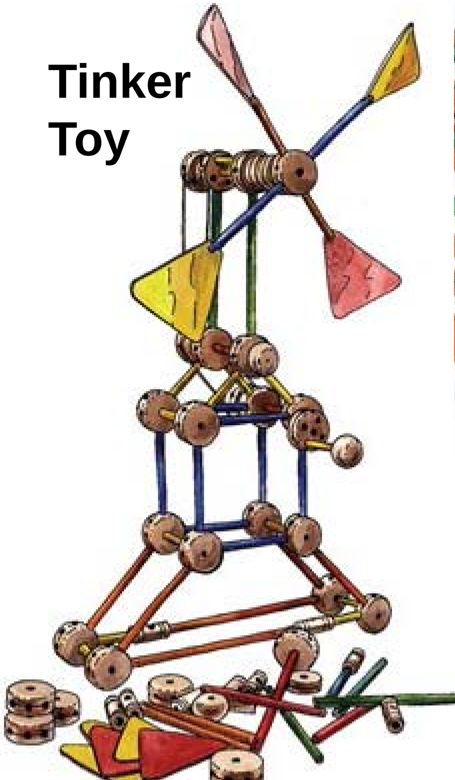
Snap Blocks



Woodbuilders

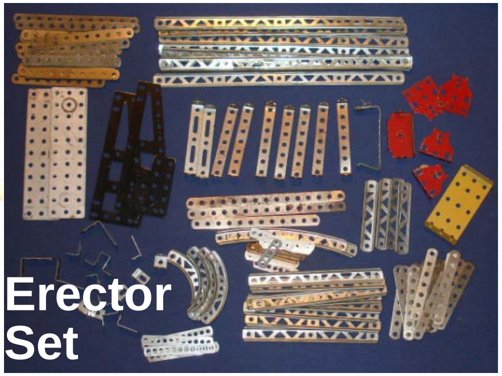
Design the Architecture of Your Construction Set

Tinker Toy



Log Builder

Lego

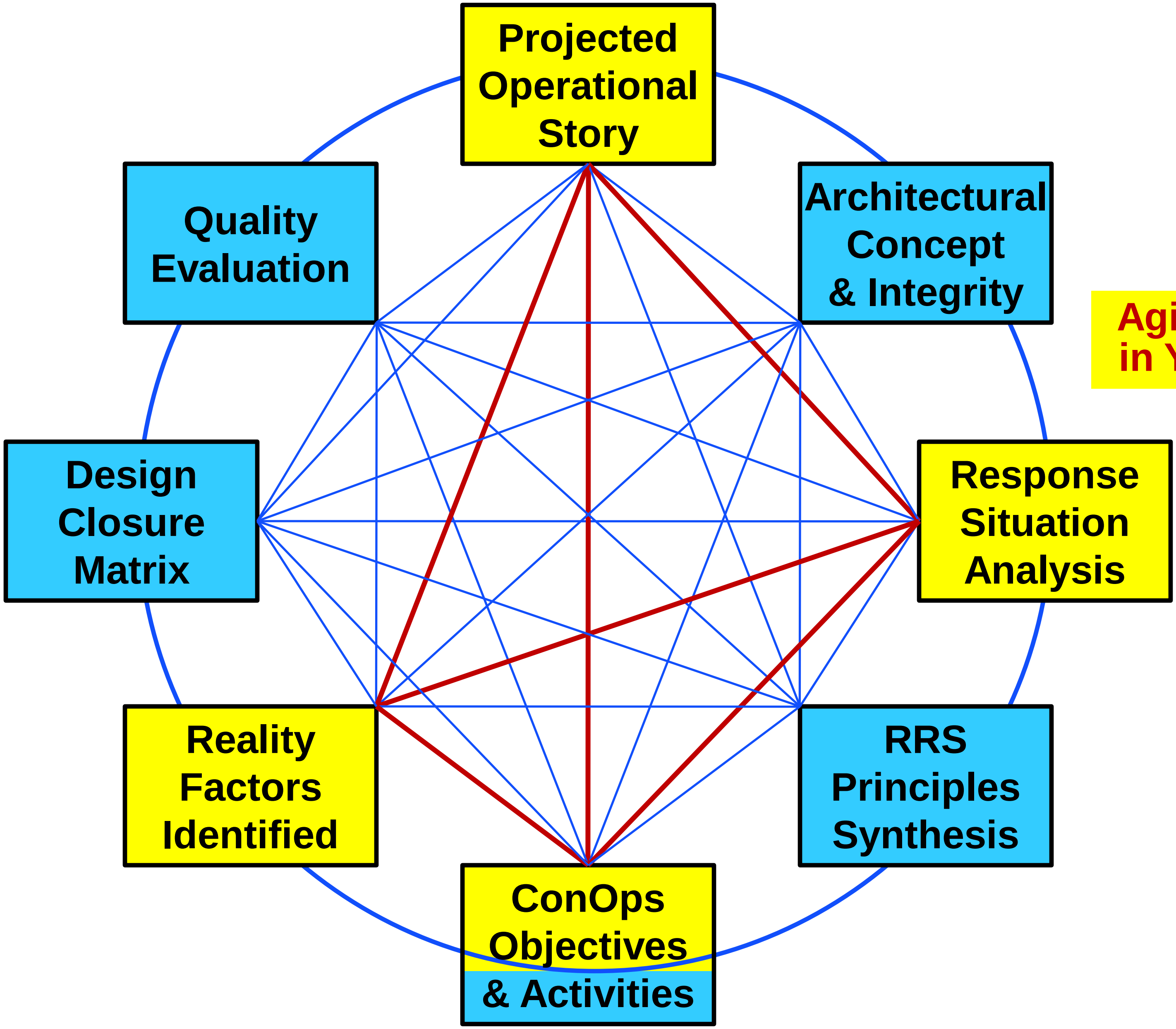


Erector Set

Bristle Blocks

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

Tools for developing *agile-response requirements & ConOps*



Projected Operational Story

Architectural Concept & Integrity

Response Situation Analysis

RRS Principles Synthesis

ConOps Objectives & Activities

Reality Factors Identified

Design Closure Matrix

Quality Evaluation

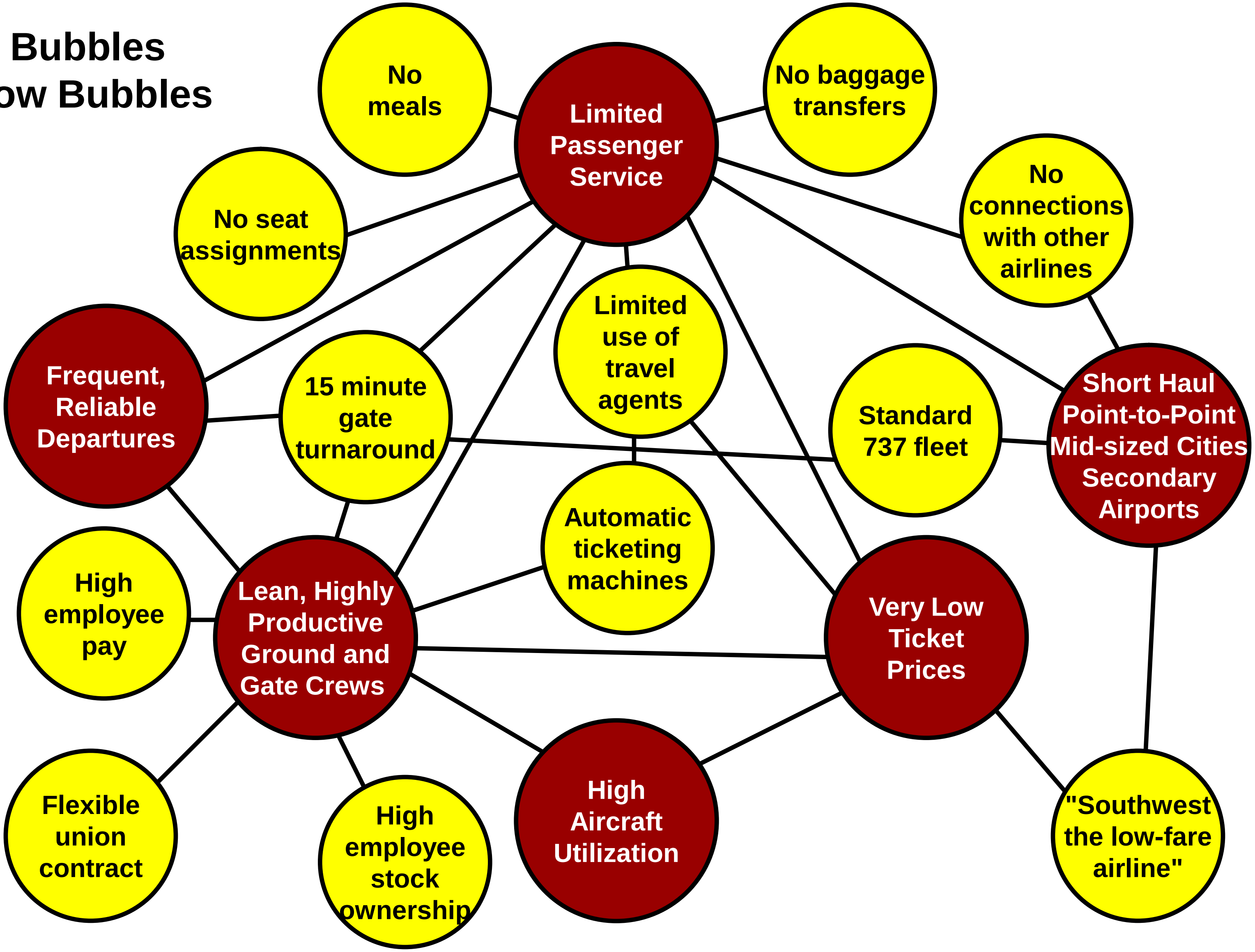
Agile 102 in Yellow

rick.dove@parshift.com, attributed copies permitted

Southwest Airlines
(Concept of Operations)

🟥 Reputation Perceptions (Objectives)
🟨 Causal Enablers (Activities)

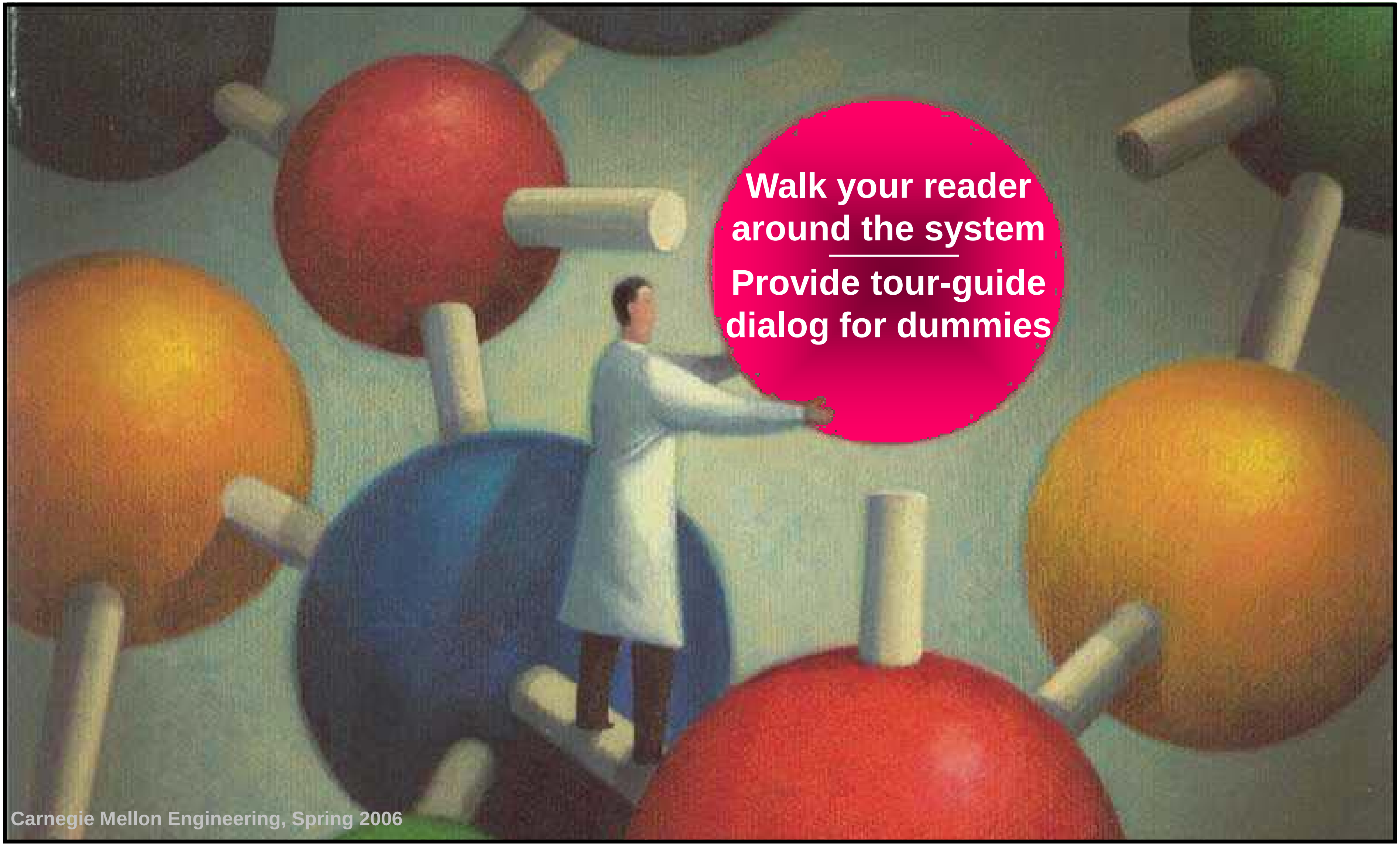
102: Red Bubbles
103: Yellow Bubbles



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

rick.dove@parshift.com, attributed copies permitted

2-Page Operational Story: Imagine yourself ASSEMBLING the drag-and-drop responses to a variety of “situations” in real time … tell us what you are doing



Walk your reader around the system

Provide tour-guide dialog for dummies

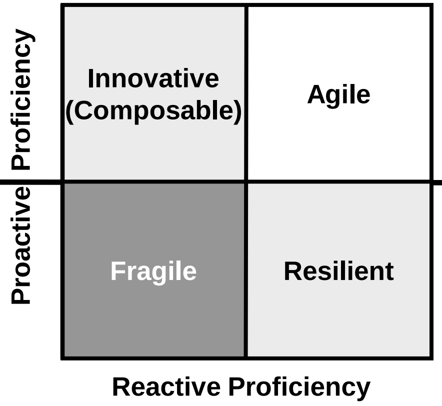
Carnegie Mellon Engineering, Spring 2006

rick.dove@parshift.com, attributed copies permitted

8 Requirement Domains

for Response Situation Analysis (RSA)

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	



rick.dove@parshift.com, attributed copies permitted

Environmental Reality Analysis

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

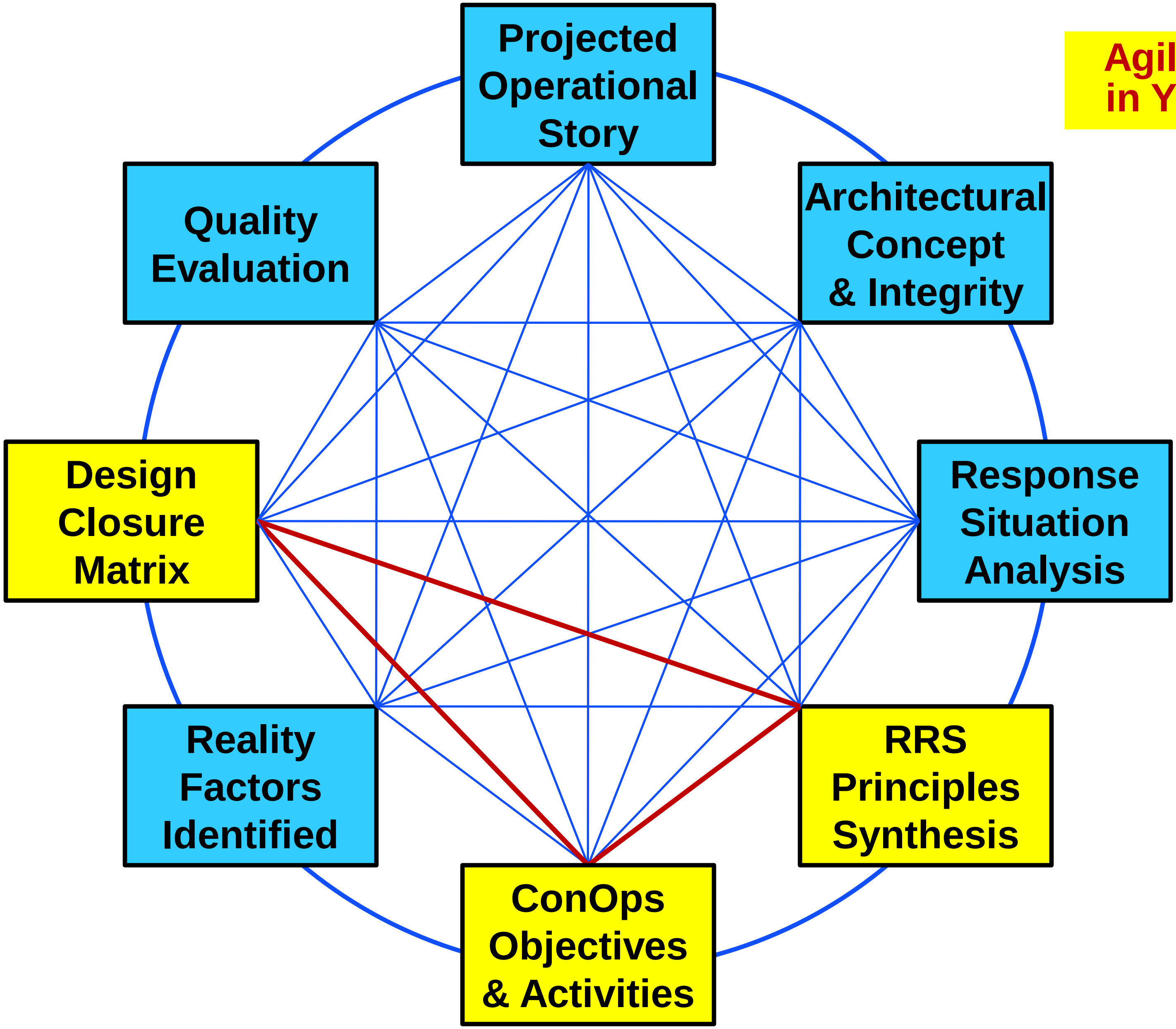
Reality Factors
Human (including customer) Behavior Reality – Human error, whimsy, expediency, arrogance...
Organizational Behavior Reality – Survival rules rule, nobody's in absolute control...
Technology Pace Reality – Accelerating process vulnerability-introductions, new agile SE methods and knowledge...
System Complexity Reality – Incomprehensible, unintended consequences...
Globalization Reality – Partners with different ethics, values, infrastructures...
Partially-Agile Enterprise Reality (Faddish Pratices) – Outsourcing, cots policies and affects, process and progress transparency...
Agile Customers/Competitors/Adversaries – Distributed, collaborative, self organizing, proactive, impatient, innovative…

rick.dove@parshift.com, attributed copies permitted

Agile 103 – Design Fundamentals

Fleshing out Architecture with Design Principles, Activities, and Closure

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



Agile 103 in Yellow

Projected Operational Story

Quality Evaluation

Architectural Concept & Integrity

Design Closure Matrix

Response Situation Analysis

Reality Factors Identified

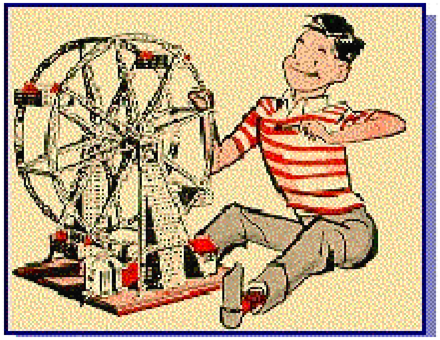
RRS Principles Synthesis

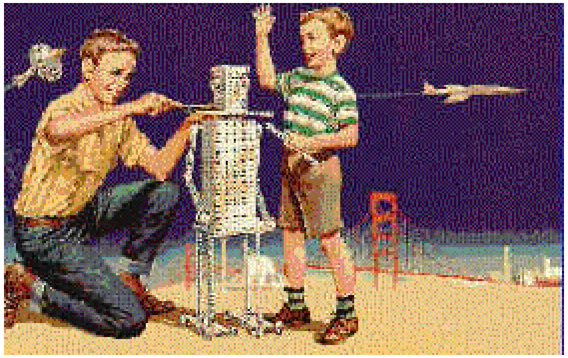
ConOps Objectives & Activities

rick.dove@parshift.com, attributed copies permitted

ERECTOR/MECCANO
A Modular Construction System

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





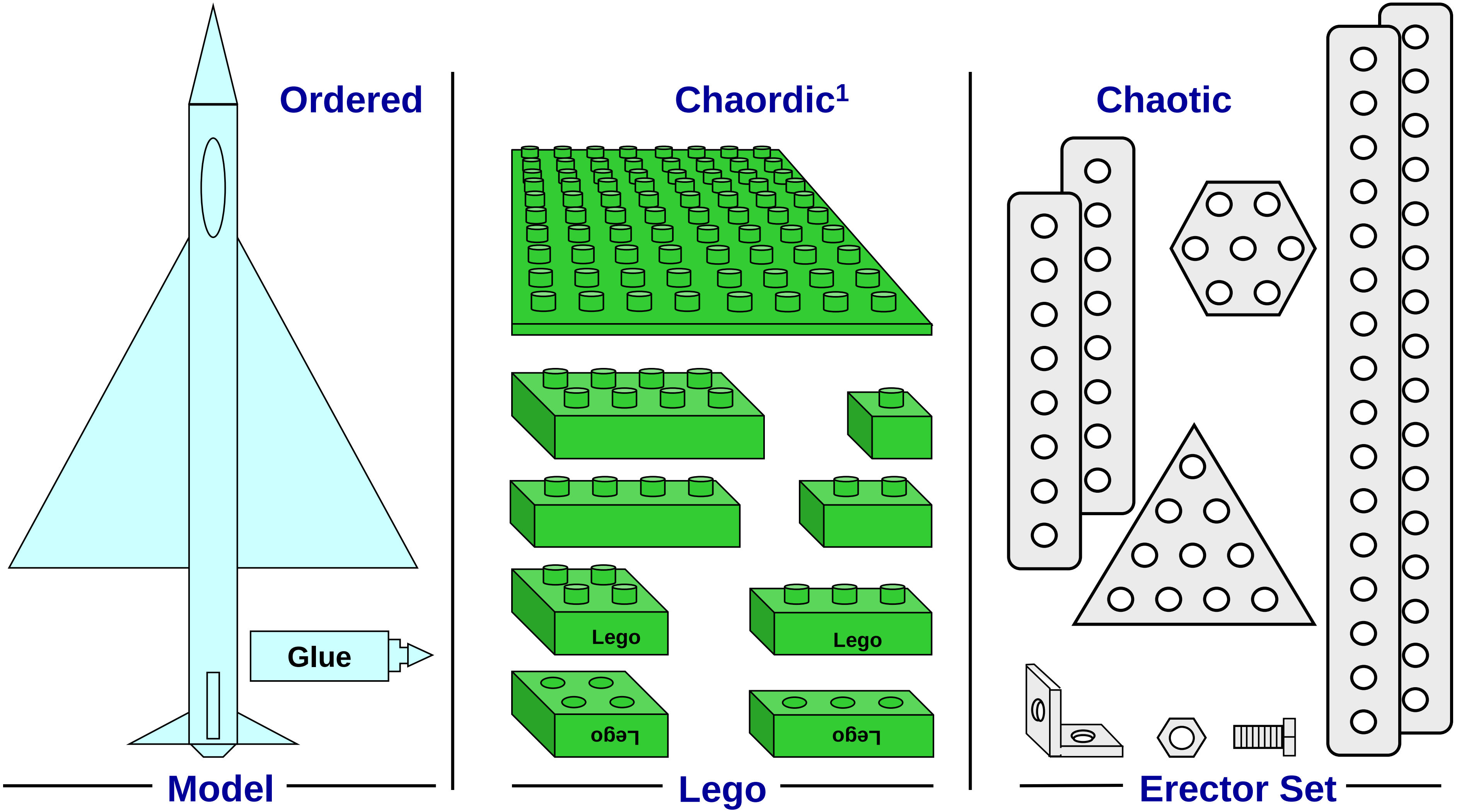
Marion Designs
8-1/2 Restoration

Lego Toy
A Modular Construction System

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

rick.dove@parshift.com, attributed copies permitted

Three construction system types



Ordered

Chaordic[1]

Chaotic

Model

Lego

Erector Set

---

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

Comparing

Erector Set

Features	Effects (plus/minus values)
• open system	• freedom to insert unintended pieces
• needs tools	• certain skills required
• plan needed	• anticipatory and reasoned thinking
• 2-piece screw/bolt connector pieces	• connection requires matched-pair pieces
• training required	• employs procedure skills
• committed connectivity	• not quickly reconfigurable
• creative constraints (by module shapes)	• “wire-mesh” like build up

Lego

Features	Effects (plus/minus values)
• closed system	• constrained to use intended modules
• no tools needed	• low skill requirement
• plan emerges	• incremental and iterative thinking
• modules have integrated connectivity	• no connectivity parts to find or loose
• no training required	• employs intuitive skills
• quick connect/disconnect	• easy trial and error convergence on result
• creative freedom (by module types)	• solid build-up (Lego Man example)

rick.dove@parshift.com, attributed copies permitted

Response Able System Principles – RRS

Reconfigurable, Reusable, Scalable

Encapsulated Modules
Need: System assemblers want replacement and internal change without side effects.
Intent: Modules physically encompass a complete capability, and have no dependencies on how other modules deliver their capabilities.

Facilitated Interfacing (Plug Compatibility)
Need: System assemblers want effective interfacing that facilitates integration and replacement of modules.
Intent: Modules share minimal interface standards, and are readily inserted and removed.

Facilitated Reuse
Need: System assemblers want effective module selection and acquisition that facilitates reuse.
Intent: Available modules are identified by capability and requirements, and can be readily discovered and acquired for deployment.

Reusable

Scalable

Evolving Standards
Need: System assemblers want effective acquisition/deployment of new module capabilities.
Intent: Passive infrastructure is monitored for current relevance, and evolves to accommodate new/beneficial module types in anticipation of need.

Redundancy and Diversity
Need: System assemblers want effective resilience under quantitative/qualitative situational variance.
Intent: Duplicate or replicable modules provides capacity and fault tolerance options; diversity of similar modules provides situational fit options.

Elastic Capacity
Need: System assemblers want to incrementally match resource commitments to capacity needs.
Intent: Modules may be combined in unbounded quantities, where possible, to increase/decrease functional capacity within the current architecture.

Reconfigurable

Peer-Peer Interaction
Need: System assemblers want effective communication among modules.
Intent: Modules communicate directly on a peer-to-peer basis to avoid intermediary relay failure, content filtering, and time delay.

Deferred Commitment
Need: System assemblers want to maintain effective response ability.
Intent: Conserve the commitment and consumption of limited resources to the last responsible moment, anticipating unpredictable/uncertain response needs.

Distributed Control and Information
Need: System assemblers want effective information-based operational decisions.
Intent: Decisions are made where maximal situational knowledge exists, relevant information is maintained local to decision making modules while accessible globally.

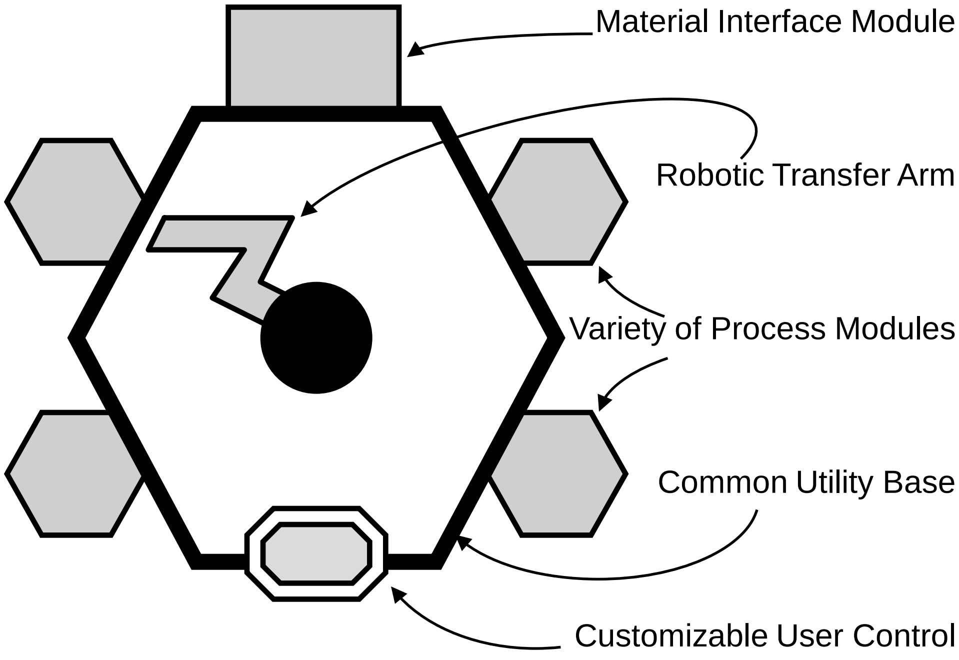
Self-Organization
Need: Systems assemblers want effective adaptation of interacting modules.
Intent: Module relationships are self-determined where possible, and module interactions are self-adjusting or self-negotiated.

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

rick.dove@parshift.com, attributed copies permitted

Semiconductor-Manufacturing 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.

Semiconductor-Manufacturing Cluster Machine

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

<table>
<tr><td>Encapsulated 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
Standardization focused on individual module interconnect only: mechanical coupling, communication protocols, and utility connections.</td></tr>
<tr><td>Facilitated Interfacing (Plug Compatibility)
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>Peer-Peer 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="3">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>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="3">Self-Organization
Wafer path within cluster determined in real-time according to the availability of appropriate process modules.</td></tr>
</table>

rick.dove@parshift.com, attributed copies permitted

Automotive ABS Production Cell

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

Reusable

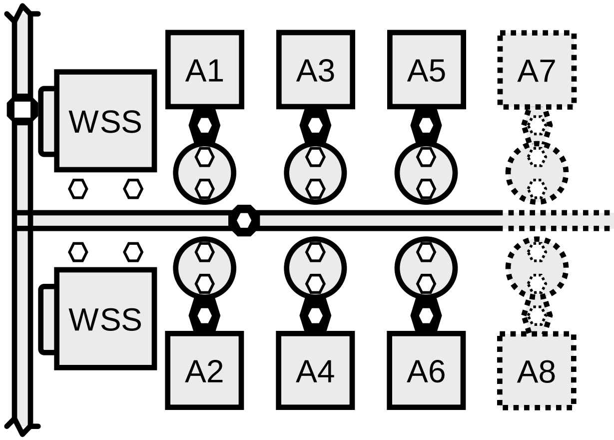
- Machines, work setting stations, pallet changers, fixtures are all standard, independent units.
- Common human, mechanical, electrical, and coolant framework.
- Machines do not require excavated pits or special foundations, and are relatively light and easy to move from one cell to another.

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.



Concept Based on LeBlond Makino A55 Cells at Kelsey-Hayes

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.

rick.dove@parshift.com, attributed copies permitted

Automotive ABS Production Cell

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

Encapsulated Modules
Flexible machines, guided vehicles, rail sections, work-setting stations, loader/unloaders, pallet changers

Facilitated Interfacing (Plug Compatibility)
Common human, mechanical, electrical, and coolant system interfaces. Common inter-module mechanical interfaces.

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.

Reusable

Scalable

Evolving Standards
General manager responsible for component commonality, and interconnect standards for mechanical coupling, communication protocols, and utility connections.

Redundancy and Diversity
Cells have multiples of each component, all cells made from same types of components, machines have full work functionality.

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.

Reconfigurable

Peer-Peer Interaction
Complete autonomous part machining, direct machine-repository program download negotiation.

Deferred Commitment
Machines and material scheduled in real-time, downloaded part programs serve individual work requirements.

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.

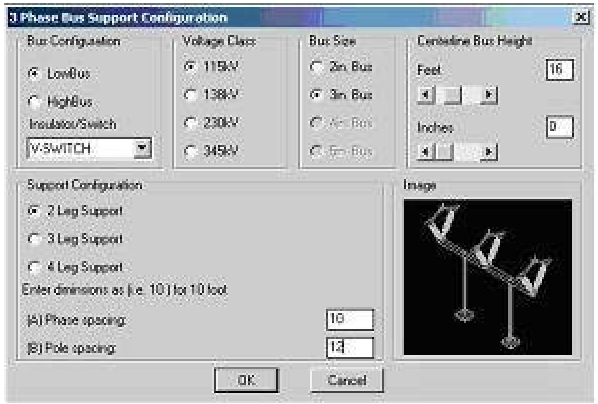
Self-Organization
Cell control software dynamically changes work routing for status changes and for new, removed, or down machines on the fly.

rick.dove@parshift.com, attributed copies permitted

Substation Designs in 6 Hours

(normally 6 months)

File



3 Phase Bus Support Configuration

Bus Configuration
- ● LowBus
- ○ HighBus

Insulator/Switch: V-SWITCH

Voltage Class
- ● 115kV
- ○ 138kV
- ○ 230kV
- ○ 345kV

Bus Size
- ○ 2in. Bus
- ● 3in. Bus
- ○ 4in. Bus
- ○ 5in. Bus

Centerline Bus Height
- Feet: 16
- Inches: 0

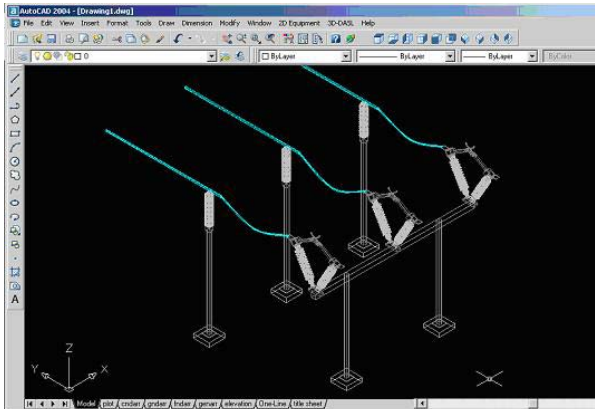
Support Configuration
- ● 2 Leg Support
- ○ 3 Leg Support
- ○ 4 Leg Support

Enter dimensions as (i.e. 10) for 10 foot

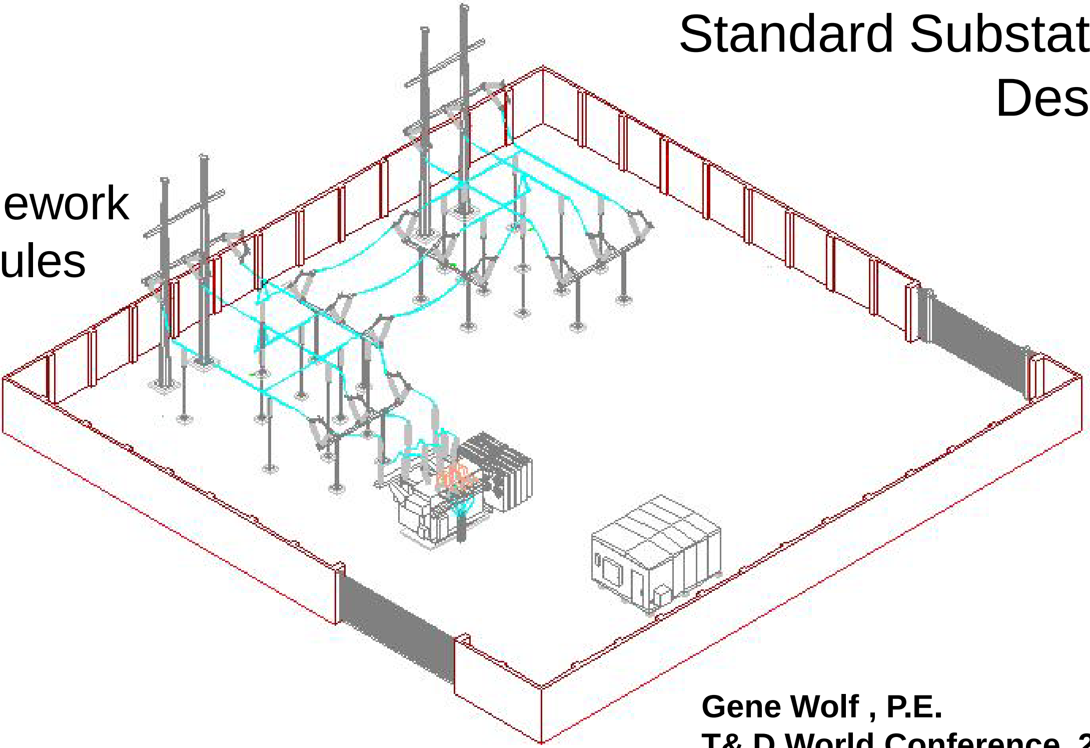
(A) Phase spacing: 10

(B) Pole spacing: 12

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

Encapsulated Modules

(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.**

rick.dove@parshift.com, attributed copies permitted

Evolving Standards

(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.**

rick.dove@parshift.com, attributed copies permitted

Facilitated Interfacing

(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)

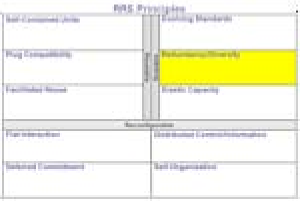
Reusability of modules is a paramount advantage of agile systems – but *facilitated* is the operable word.

***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.**

rick.dove@parshift.com, attributed copies permitted



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.**

rick.dove@parshift.com, attributed copies permitted

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.**

Peer-Peer 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.**

rick.dove@parshift.com, attributed copies permitted

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.**

rick.dove@parshift.com, attributed copies permitted

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.**

rick.dove@parshift.com, attributed copies permitted

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.**

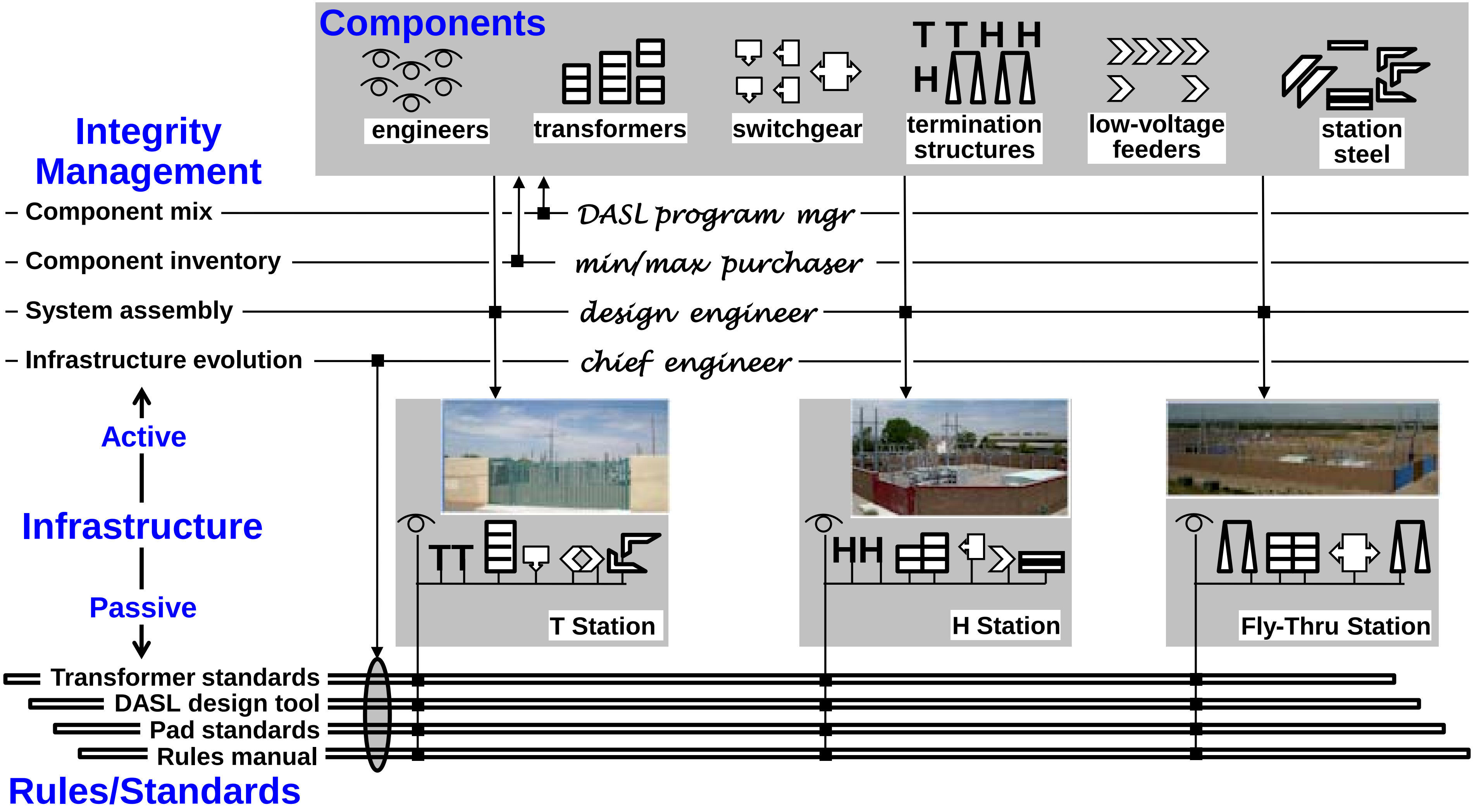
- **Collaborative improvement -- PNM's process is being tested at Long Island Power Authority and at Kansas City Power and Light, (December 2004). PNM's purpose for broadened usage is to develop a community of users, with new and diverse needs, that will collaborate in a self-organizing fashion toward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



Components

engineers

transformers

switchgear

termination structures

low-voltage feeders

station steel

Integrity Management

– Component mix

– Component inventory

– System assembly

– Infrastructure evolution

DASL program mgr

min/max purchaser

design engineer

chief engineer

Active

Infrastructure

Passive

T Station

H Station

Fly-Thru Station

Transformer standards

DASL design tool

Pad standards

Rules manual

Rules/Standards

rick.dove@parshift.com, attributed copies permitted

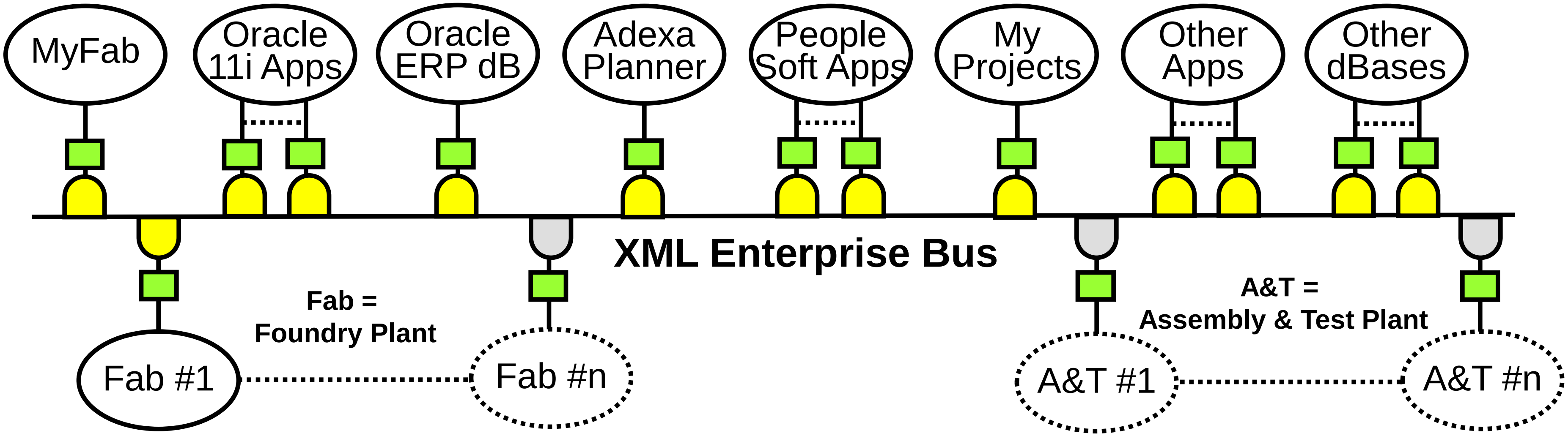
For PNM – Agility Costs Less

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 - 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 Interfacing - XML, message-data definitions, BIM spec, ETL-interface spec, rule on COTS.

Facilitated 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.

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

Peer-Peer 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.

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:

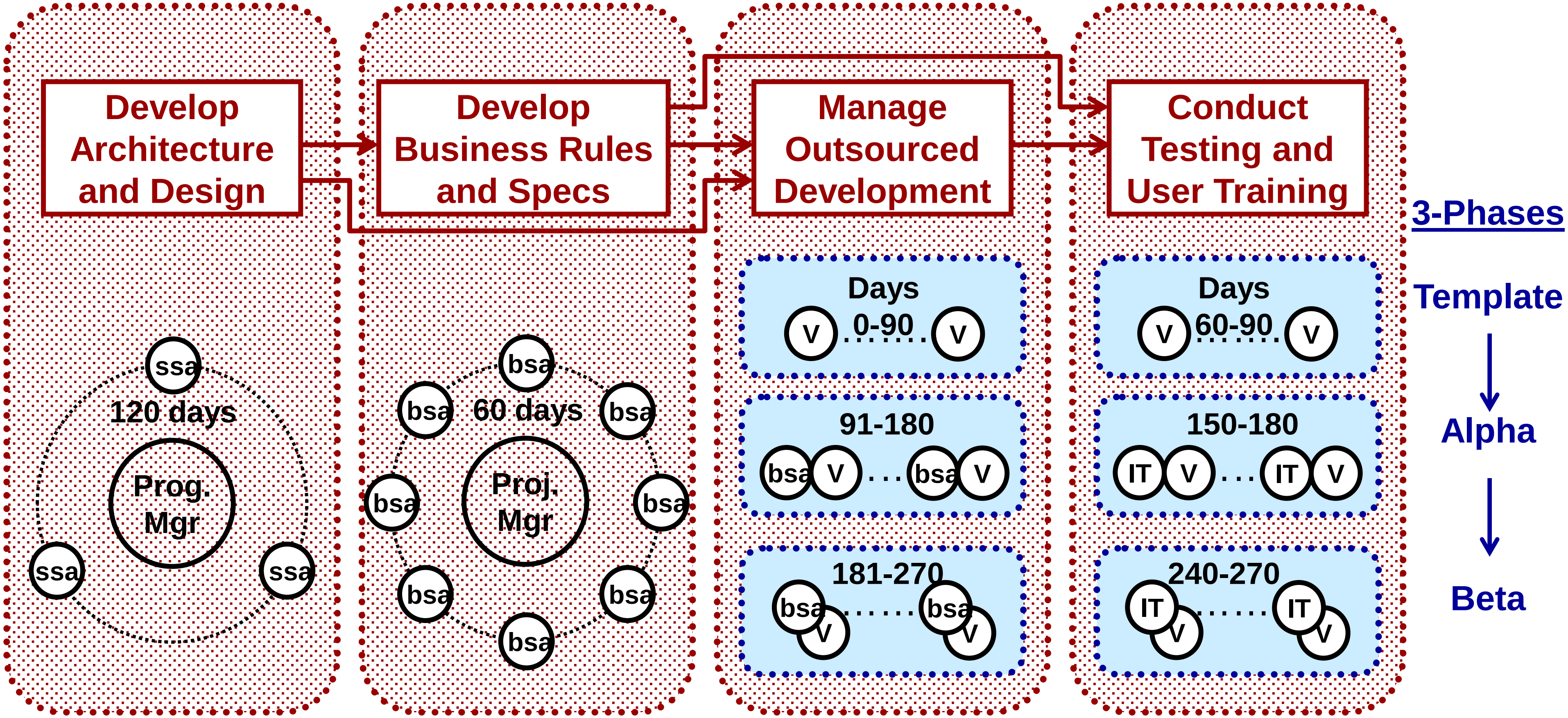
P1:	Out-of-box best practice from vendor – supporting the company Vendors configure the applications
P2:	BSA-developed business process rules Vendors + BSAs configure the applications
P3:	Refined business processes BSAs configure the applications

- 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

rick.dove@parshift.com, attributed copies permitted

RRS Principles Applied for Implementation Process

Evolving Standards – 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 Interfacing – vendor rules clear, agreed in advance, and managed.

Facilitated 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.

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

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

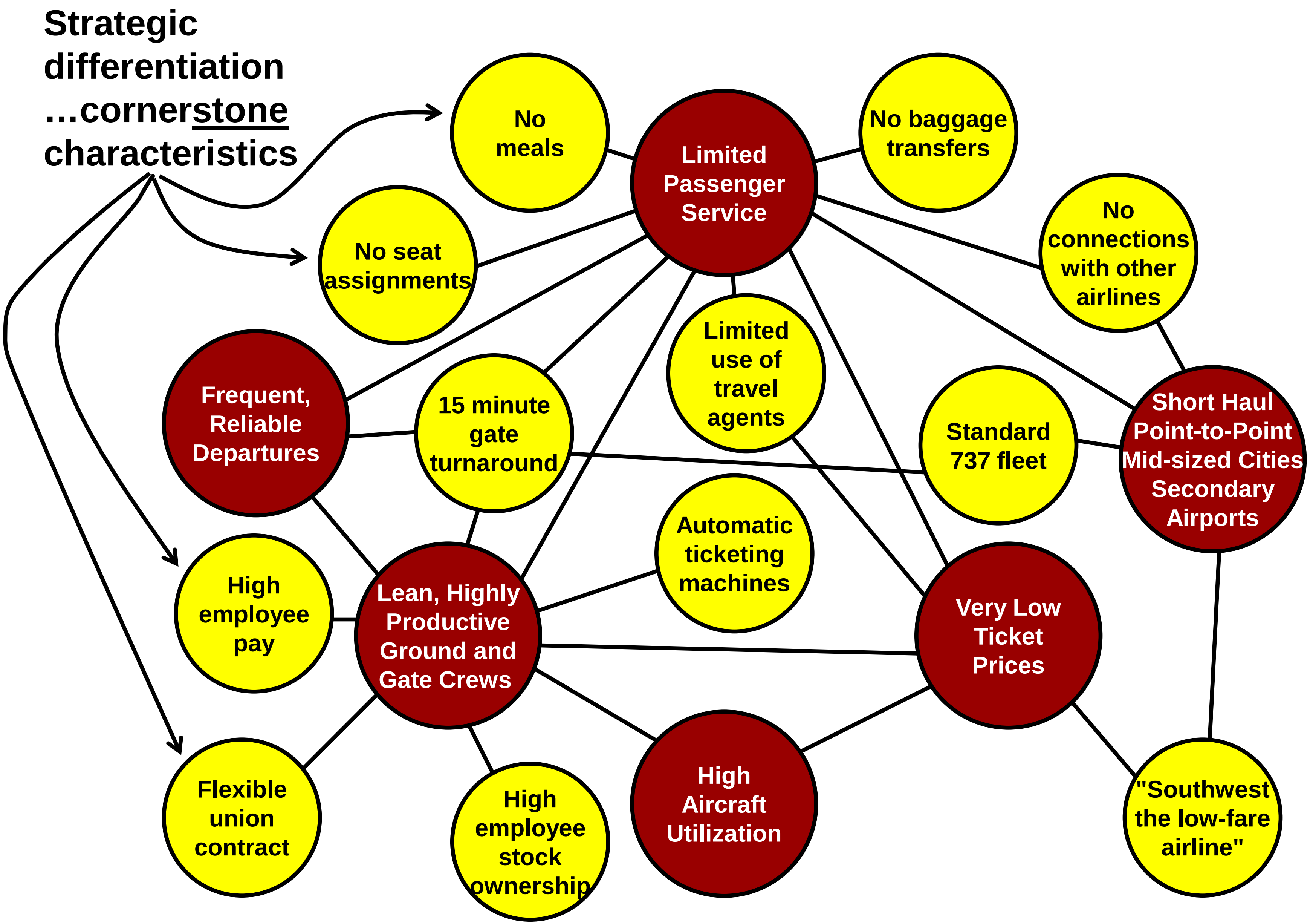
Self-Organization - BSA team relationships and assignments.

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)

Porter on Strategy

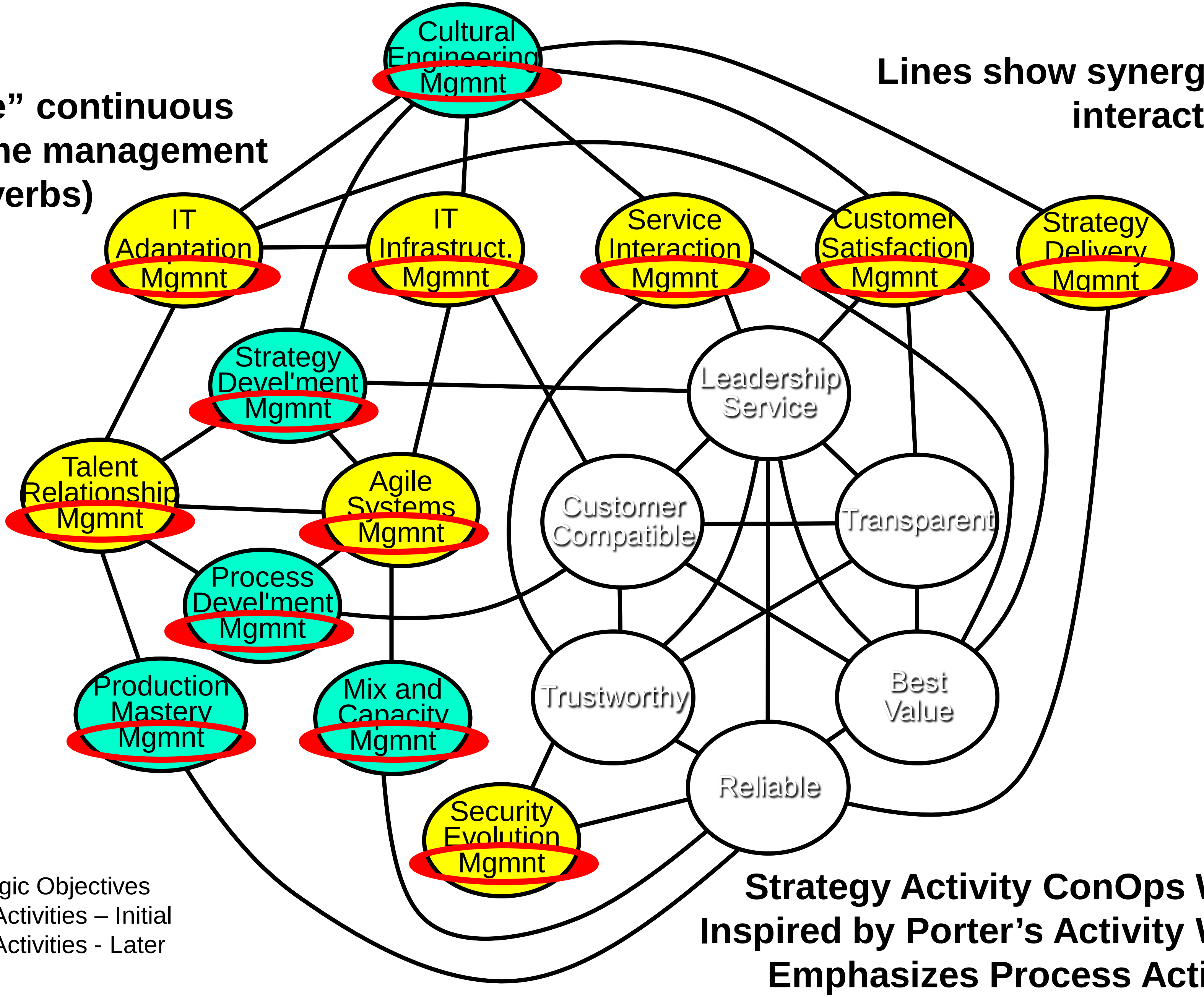


rick.dove@parshift.com, attributed copies permitted

Semiconductor Foundry

Lines show synergistic interactions

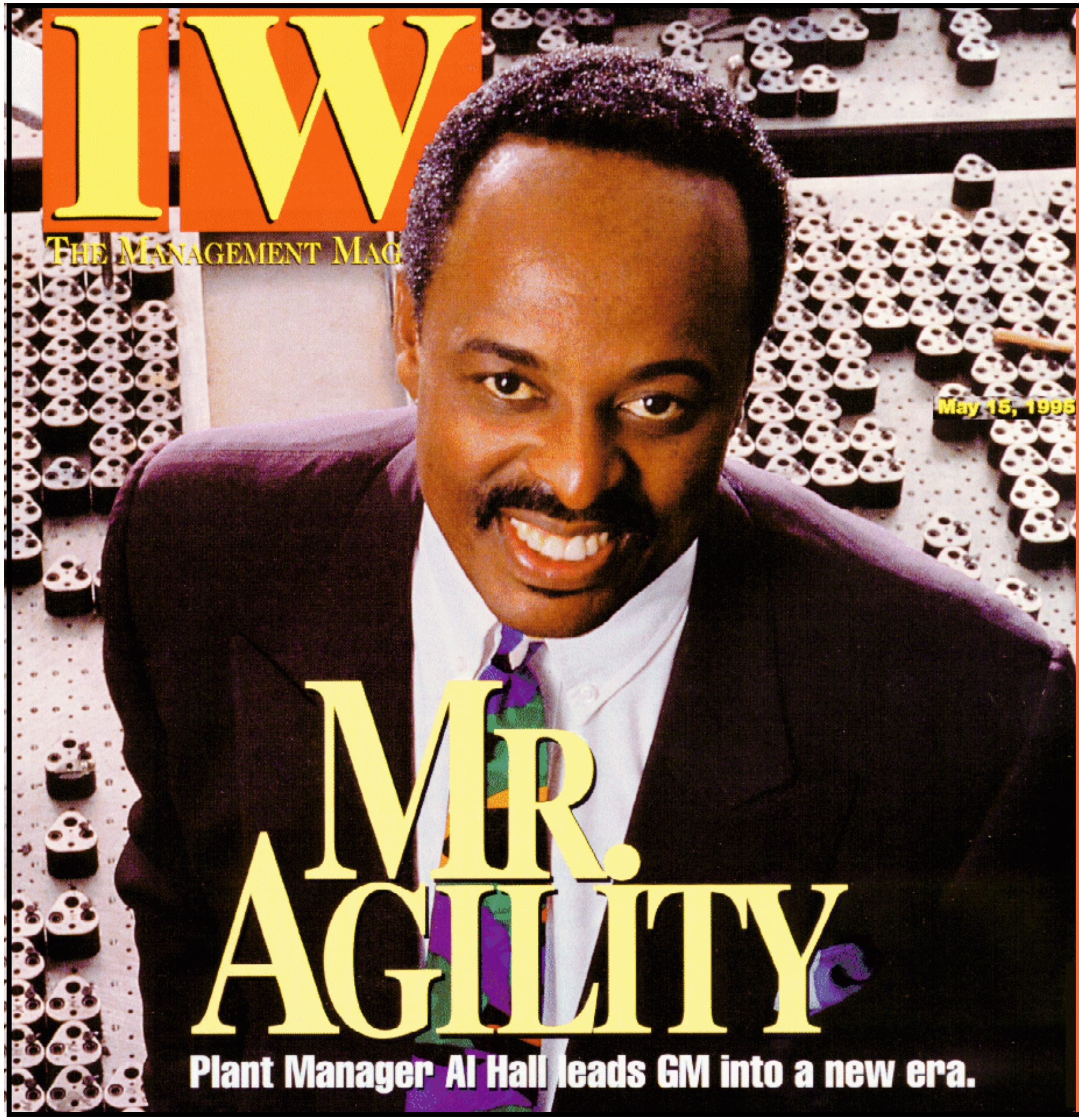
“Active” continuous outcome management (uses verbs)



🔴 - Strategic Objectives
🟡 - Agile Activities – Initial
🔵 - Agile Activities - Later

Strategy Activity ConOps Web
Inspired by Porter’s Activity Web
Emphasizes Process Activity

rick.dove@parshift.com, attributed copies permitted



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.

rick.dove@parshift.com, attributed copies permitted

General Motors Workshop

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

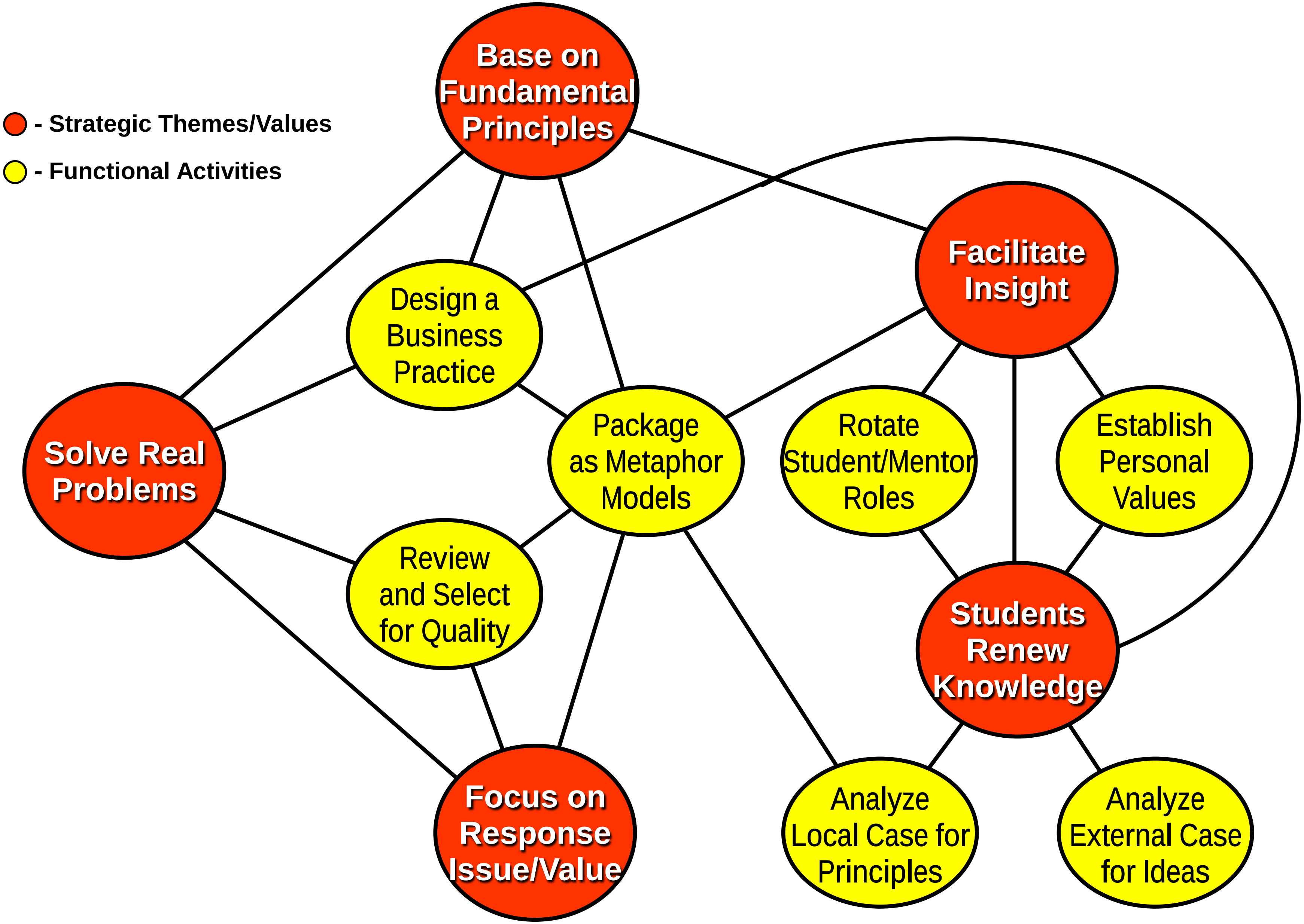
The workshop group included about 10 management and executive level participants from the plant. First they analyzed the two things above that they were very familiar with and respected highly – looking through the lens of the 10 RRS principles to identify how these were employed to enable agility. Then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

🟥 - Strategic Themes/Values

🟨 - Functional Activities



Base on Fundamental Principles

Facilitate Insight

Design a Business Practice

Solve Real Problems

Package as Metaphor Models

Rotate Student/Mentor Roles

Establish Personal Values

Review and Select for Quality

Students Renew Knowledge

Focus on Response Issue/Value

Analyze Local Case for Principles

Analyze External Case for Ideas

rick.dove@parshift.com, attributed copies permitted

Core Competency Insight Diffusion

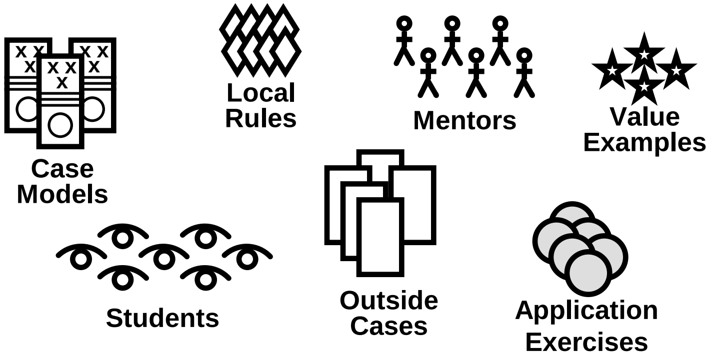
Change Proficiency

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:

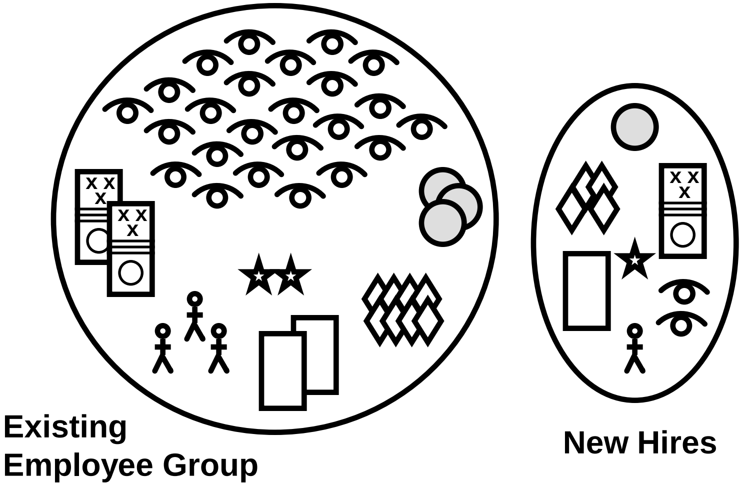
- Correction:
  - Incorrect Knowledge
  - Poor Value Knowledge
- Variation:
  - Flexible Student Schedule
- Expansion:
  - Any Size Group
- Reconfiguration:
  - Rules«–»Applications



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

rick.dove@parshift.com, attributed copies permitted

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

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

rick.dove@parshift.com, attributed copies permitted

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 (see *Response Ability* - Chapter 7)

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.

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.

rick.dove@parshift.com, attributed copies permitted

Example of the GM Activity Description

Analyze Local Case for Principles

Activity Description

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 …

Example of 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

rick.dove@parshift.com, attributed copies permitted

Issue: Capture Hidden Tacit Knowledge

(GM example of paragraph addressing how principles address an issue)

Employing the *peer-peer 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.

***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 an *encapsulated* 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

rick.dove@parshift.com, attributed copies permitted

Issue: Improving Knowledge Accuracy

(GM example of paragraph addressing how principles address an issue)

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

Self organization **of the sub-groups and allowing direct** ***peer-peer 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

(GM example of paragraph addressing how principles address an issue)

Chartering each sub-group as an ***encapsulated 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

(GM example of paragraph addressing how principles address an issue)

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

(GM example of paragraph addressing how principles address an issue)

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 *peer-peer 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

rick.dove@parshift.com, attributed copies permitted

Issue: Flexible Student Schedules

(GM example of paragraph addressing how principles address an issue)

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* 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

(GM example of paragraph addressing how principles address an issue)

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 *encapsulated units* that work to a common *facilitated-interfacing* process structure, and having them all work *redundantly* on the same objectives.

Issue: Reinterpret Rules for New Applications

(GM example of paragraph addressing how principles address an issue)

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

rick.dove@parshift.com, attributed copies permitted

Engaging the System Builder(s)

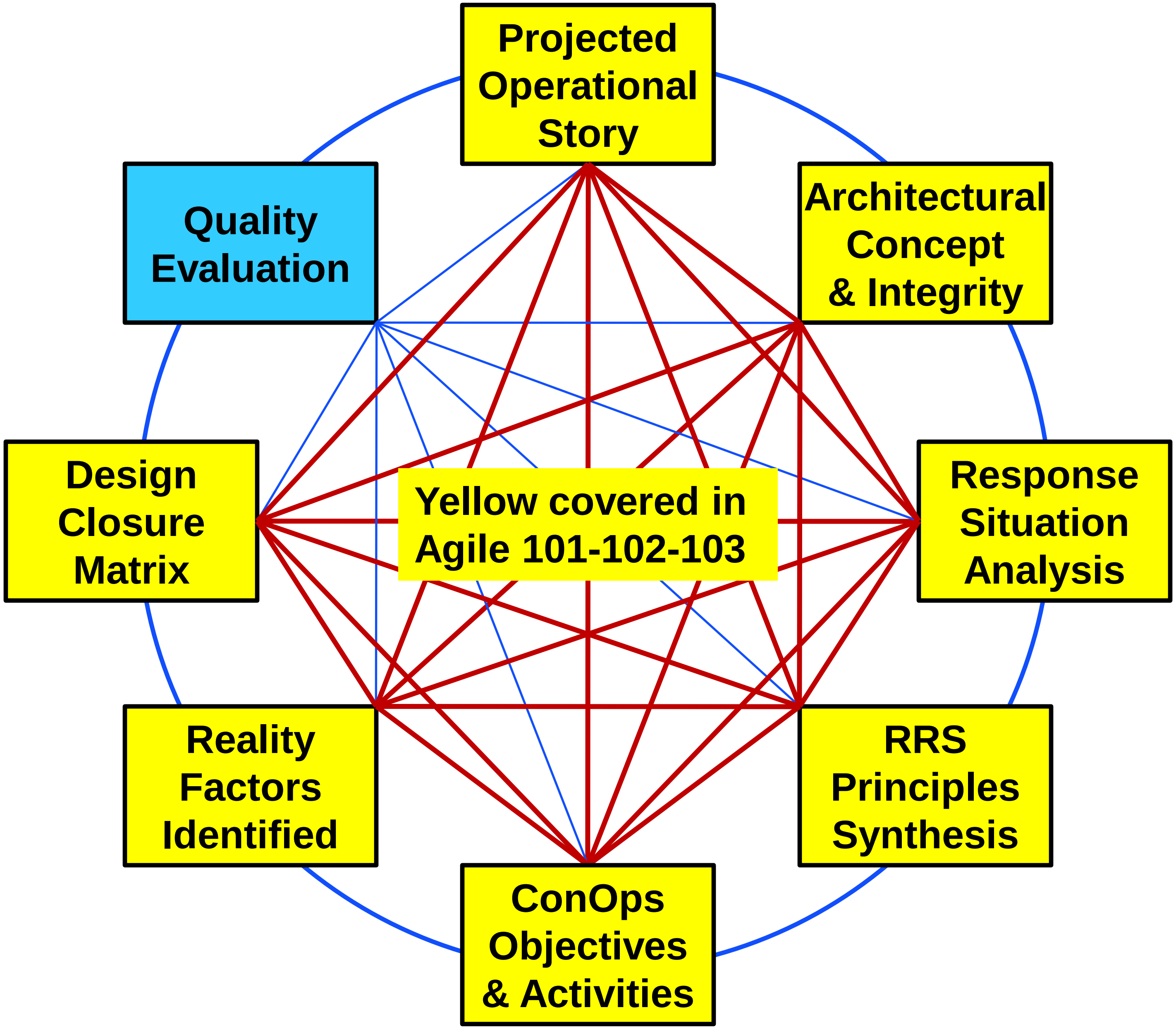
The closure matrix descriptive work
is where design pulls together
all prior conceptual thinking
and provides sufficient detail to engage a “builder”
who should not wander far from the design intent
if the descriptive material is sufficient, understandable, and employed.

The approaches outlined in Agile 101-102-103
and the material they produce
is the substance of a detailed Concept of Operations
for the achieving and sustaining System Agility

Wrapping it Up

rick.dove@parshift.com, attributed copies permitted

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



Projected Operational Story

Quality Evaluation

Architectural Concept & Integrity

Design Closure Matrix

Yellow covered in Agile 101-102-103

Response Situation Analysis

Reality Factors Identified

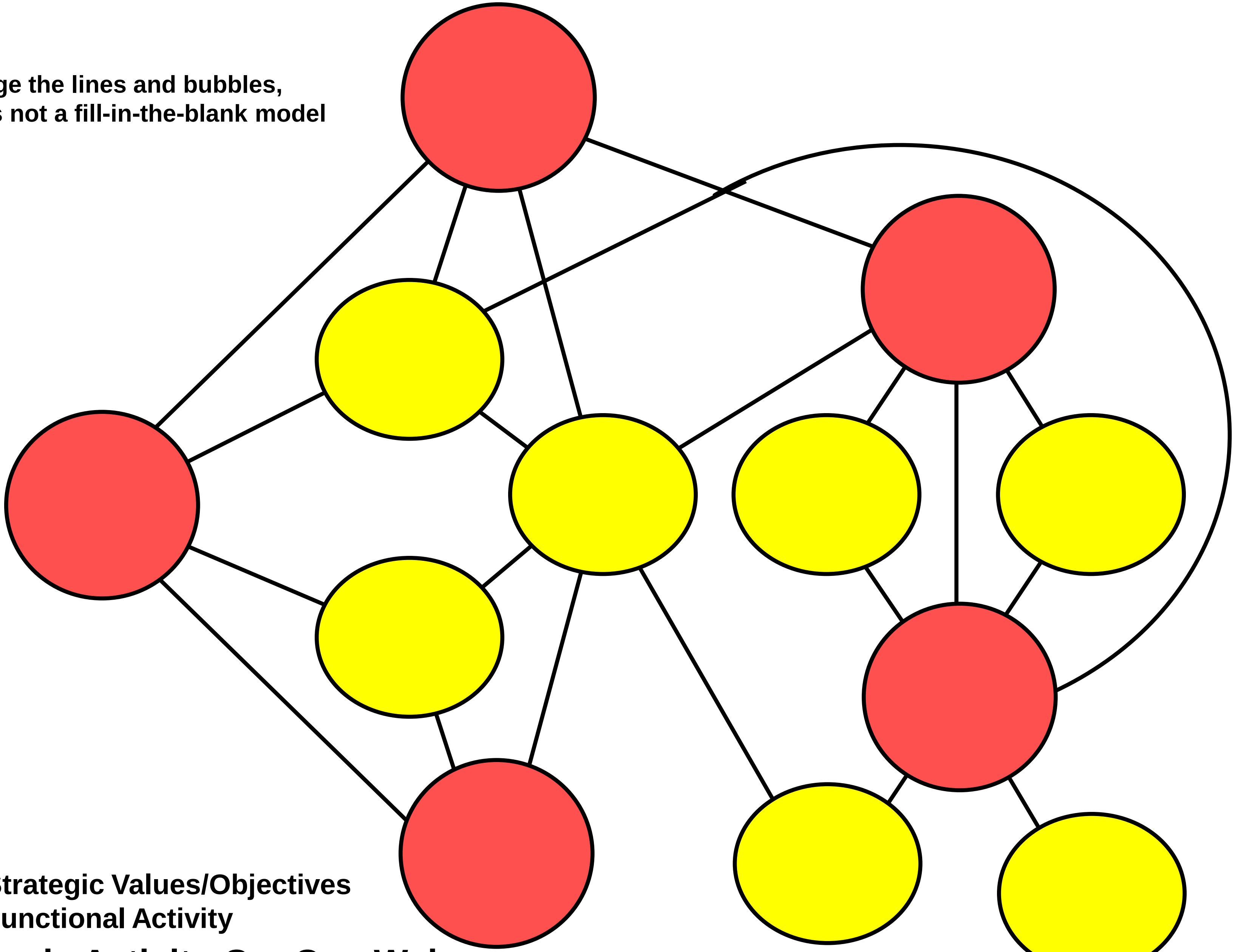
RRS Principles Synthesis

ConOps Objectives & Activities

rick.dove@parshift.com, attributed copies permitted

concept development-tool frameworks for Agile 101-102-103

System: ____________________________



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

🟥 **Strategic Values/Objectives**
🟨 **Functional Activity**

Strategic Activity ConOps Web

rick.dove@parshift.com, attributed copies permitted

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

rick.dove@parshift.com, attributed copies permitted

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...
- ?

Partially-Agile Enterprise Reality (Faddish Practices) – Outsourcing, web services, transparency, COTS, SOA...
- x

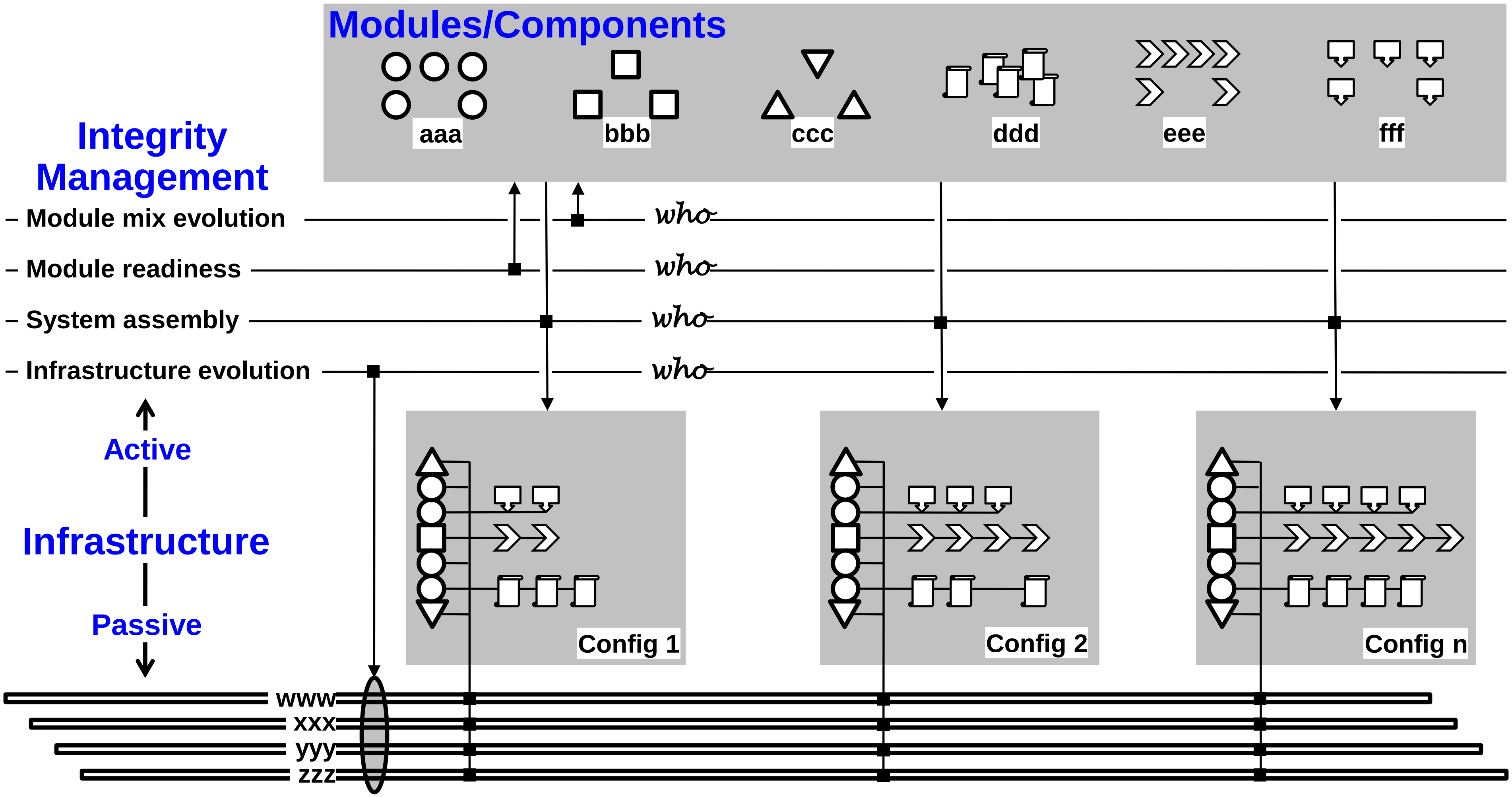
Agile Adversaries/Competitors/Customers – Distributed, collaborative, self organizing, proactive, impatient, innovative...
- ?

Other?
- ?

rick.dove@parshift.com, attributed copies permitted

Your System

Sample Graphics for your modification into your system needs



Modules/Components

aaa bbb ccc ddd eee fff

Integrity Management

- Module mix evolution — who
- Module readiness — who
- System assembly — who
- Infrastructure evolution — who

Active

Infrastructure

Passive

Config 1

Config 2

Config n

www

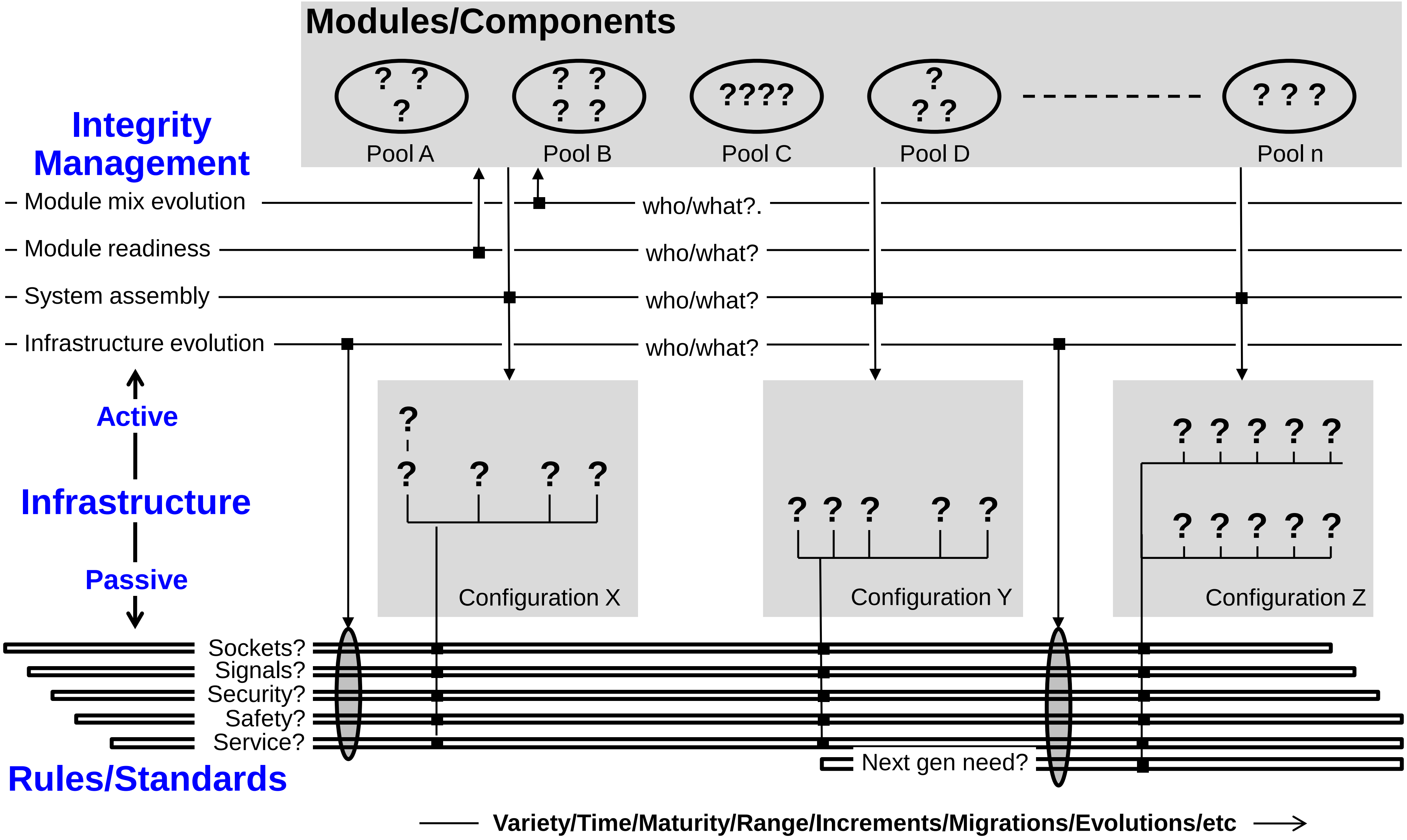
xxx

yyy

zzz

System ____________________

Graphic template for modification into your system architecture



Modules/Components

? ? ? Pool A

? ? ? ? Pool B

???? Pool C

? ? ? Pool D

? ? ? Pool n

Integrity Management

- Module mix evolution — who/what?.
- Module readiness — who/what?
- System assembly — who/what?
- Infrastructure evolution — who/what?

Active

Infrastructure

Passive

Configuration X

Configuration Y

Configuration Z

Sockets?

Signals?

Security?

Safety?

Service?

Next gen need?

Rules/Standards

Variety/Time/Maturity/Range/Increments/Migrations/Evolutions/etc →

RRS Principles for System: ______________________________

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

Encapsulated Modules) Modules are encapsulated independent units loosely coupled through the passive infrastructure.
?

Facilitated Interfacing (Plug Compatibility) 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.
?

Reusable

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

Peer-Peer 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.
?

rick.dove@parshift.com, attributed copies permitted

System ______________________________

Who Does What to Keep the Systems Sustainable and 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)**:

- **?**
- **?**
- **?**

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

- **?**
- **?**
- **?**

System Assembly/Reconfiguration **(who assembles/configures/reconfigures the systems):**

- **?**
- **?**
- **?**

Infrastructure Evolution **(who reviews and modifies the various infrastructure elements):**

- **?**
- **?**
- **?**

rick.dove@parshift.com, attributed copies permitted

<your system>

RRS Principles

Activities

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

Principles Employed and Issues Served

	Issues		Encapsulated Modules	Facilitated Interfacing	Facilitated Re-Use	Peer-Peer Interaction	Deferred Commitment	Distributed Control & Info	Self Organization	Elastic Capacity	Redundancy / Diversity	Evolving Standards
Proactive	??											
	??											
	??											
	??											
	??											
	??											
	??											
Reactive	??											
	??											
	??											
	??											
	??											

rick.dove@parshift.com, attributed copies permitted

References and Supportive Readings

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

(updated asynchronously from time-to-time)

fini

rick.dove@parshift.com, attributed copies permitted