Agile and Lean SE:
Complimentary and Contradictory Relationship
09:30 – 10:30

Lean and Agile Systems Engineering
The Gordon Center For Systems Engineering
Technion, Haifa, Israel
06-Jan-2014

Rick Dove

Paradigm Shift International and Stevens Institute of Technology

Before Relating Agile to Lean…

First, we need to know what agile means
…and doesn't mean

(probably not what you think)

- ❑ **History of agile knowledge**
- ❑ **What makes an agile system agile?**
  (agile SE must first be an agile system)
- ❑ **What does agile SE look like**
  …and not look like
- ❑ **So now – how does agile relate to lean?**

Agile ≠ Software Methods

For many people
agile is solely associated
with the family of
software development processes
that uses that label.

An older, more general, more precise
agile concept definition
is the root of the software usage
and employed in other domains.

Agility was identified in a 1991 study funded by US OSD* to determine the competitive operational strategy that would follow lean

Agility in domain independent systems was investigated by 250 organizations and 1000+ people in the 90's, funded by DARPA

In 2001 the software development community adopted the word and general concept, with a variety of branded practices that bury the core concept

Fundamental understandings of agile are critical for effective SE employment to deliver agile benefits

OSD: Office of the Secretary of Defense

Working Definition of Agility

The ability to thrive
in an environment of
continuous and unpredictable change.

The focal point here is change
- the ability to initiate it, and the ability to respond to it.

Thrive is a key word,
it implies long term success, as opposed to a lucky response, and it implies wielding agility as an offensive as well as a defensive capability.

Continuous and unpredictable underscores the new long-term picture and also distinguishes agility from mere flexibility, enabling successful change even when there is little advance notice and no prior expectation.

Dove, Rick. 1993. Lean and Agile: Synergy, Contrast, and Emerging Structure. Defense Manufacturing Conference '93, San Francisco, CA, USA, 29Nov-2Dec

More Precisely…

Agility is the
ability of a system to thrive
in an uncertain and unpredictably evolving environment;

deploying effective response
to both opportunity and threat, within mission.

Effective response has four metrics:

- **timely (fast enough to deliver value),**
- **affordable (cost can be repeated as often as necessary),**
- **predictable (can be counted on to meet the need), and**
- **comprehensive (anything/everything within mission).**

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

In a Word

Agile is Responsive

Lean is Efficient

Useful labels that say what they mean

Agile : Lean = Forgiving : Mean

Need for Agility

The pace of technology
is reducing the useful lifetime of deployed systems and increasing the risk in long development programs.

The pace of social collaboration on a global scale increases the pace of technological and social innovation.

Systems are becoming more complex,
yet are needed in shorter time frames.

Quick History

1980s: Japanese lean manufacturing concepts demonstrate scary superiority

1990s: General world-wide scramble to catch up in Lean manufacturing

US OSD/DARPA asked what's next for a head start?

Agile enterprise & systems concepts developed & socialized

Adoption

1993: Hewlett Packard was the first to initiate a program to educate its customers and bring to market IT support under the Agile Enterprise banner.

1996: DoD's Command and Control Research Program began an exploration of agile command and control that continues today.

2001: the Agile Manifesto for Software Development adopted the agile label as appropriately descriptive and fundamentally consistent with their concepts.

Somewhere in there SOA (Service Oriented Architecture) and Web Services defined agile IT systems to enable agile enterprise strategies.

2010ish: Agile Software Development accepted as useful and mainstream.

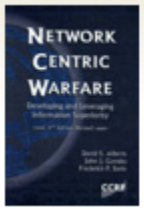
2012ish: Systems Engineering community gets interested … and confused.

Agile Command & Control (and Military enterprise)

www.dodccrp.org/html4/books_main.html

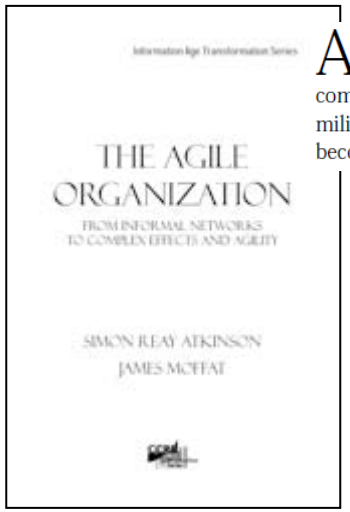
The Agility Advantage: A Survival Guide For Complex Enterprises and Endeavors
David S. Alberts | 2011
Download: PDF [9.6M]

Power to the Edge eBook
David S. Alberts and Richard E. Hayes | 2003
Download: PDF [3.2M] | ePub [3.3M] | Kindle [2.9M]



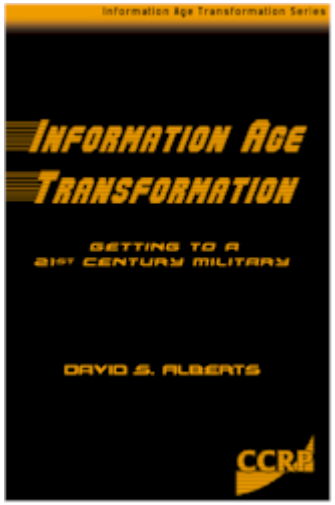
Network Centric Warfare eBook
David S. Alberts, John J. Garstka, and Frederick P. Stein | 1999
Download: PDF [1.8M] | ePub [4.8M] | Kindle [2.4M]

This book articulates the nature of the characteristics of Network and suggests a process for developing mission capability packag



Agility is the gold standard for Information Age militaries. Facing uncertain futures and new sets of threats in a complex, dynamic, and challenging security environment, militaries around the world are transforming themselves, becoming more information-enabled and network-centric.

And quite a few more.



the Agile path started in DoD with this 1995 publication.

Case: Home Entertainment Technology Migration

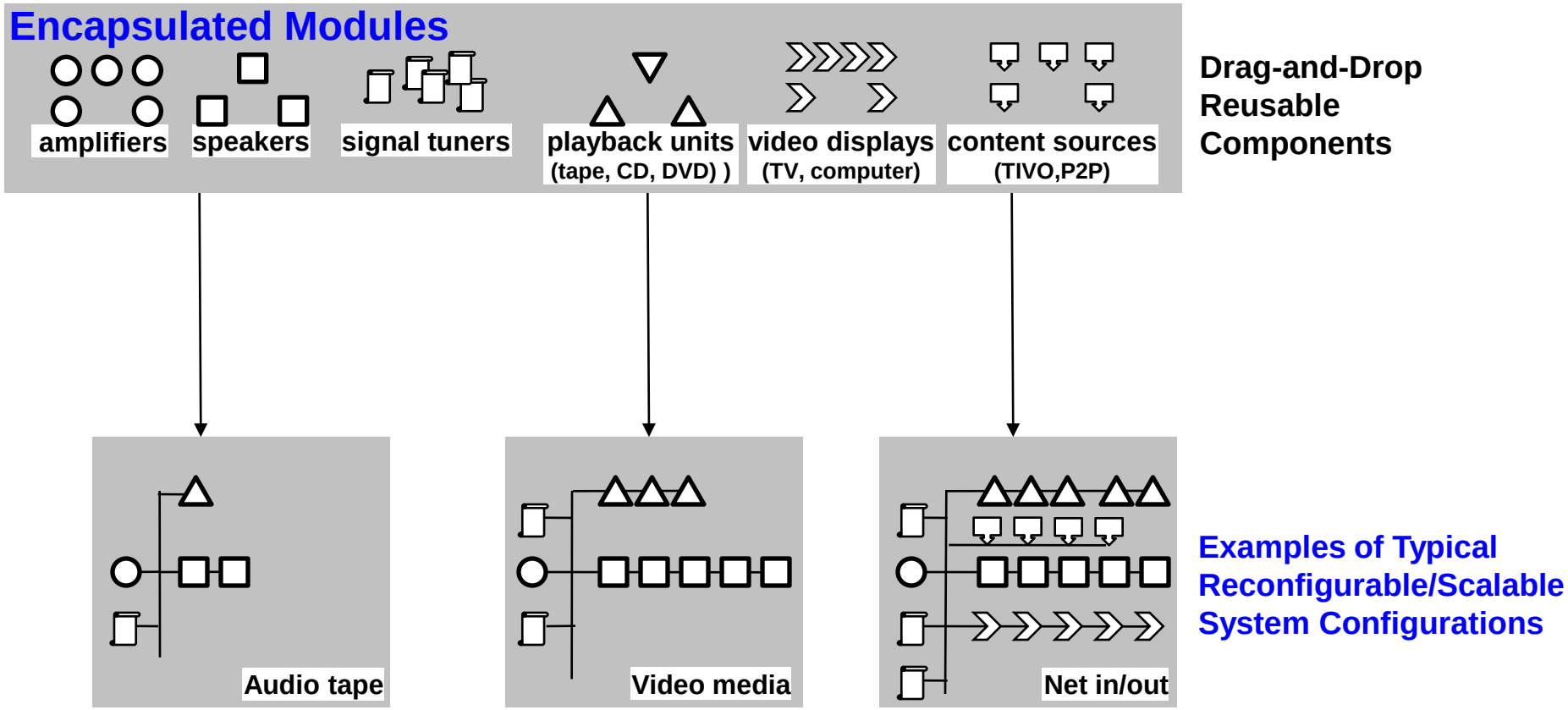
agile architecture pattern: drag-and-drop, plug-and-play

Encapsulated Modules

Drag-and-Drop Reusable Components

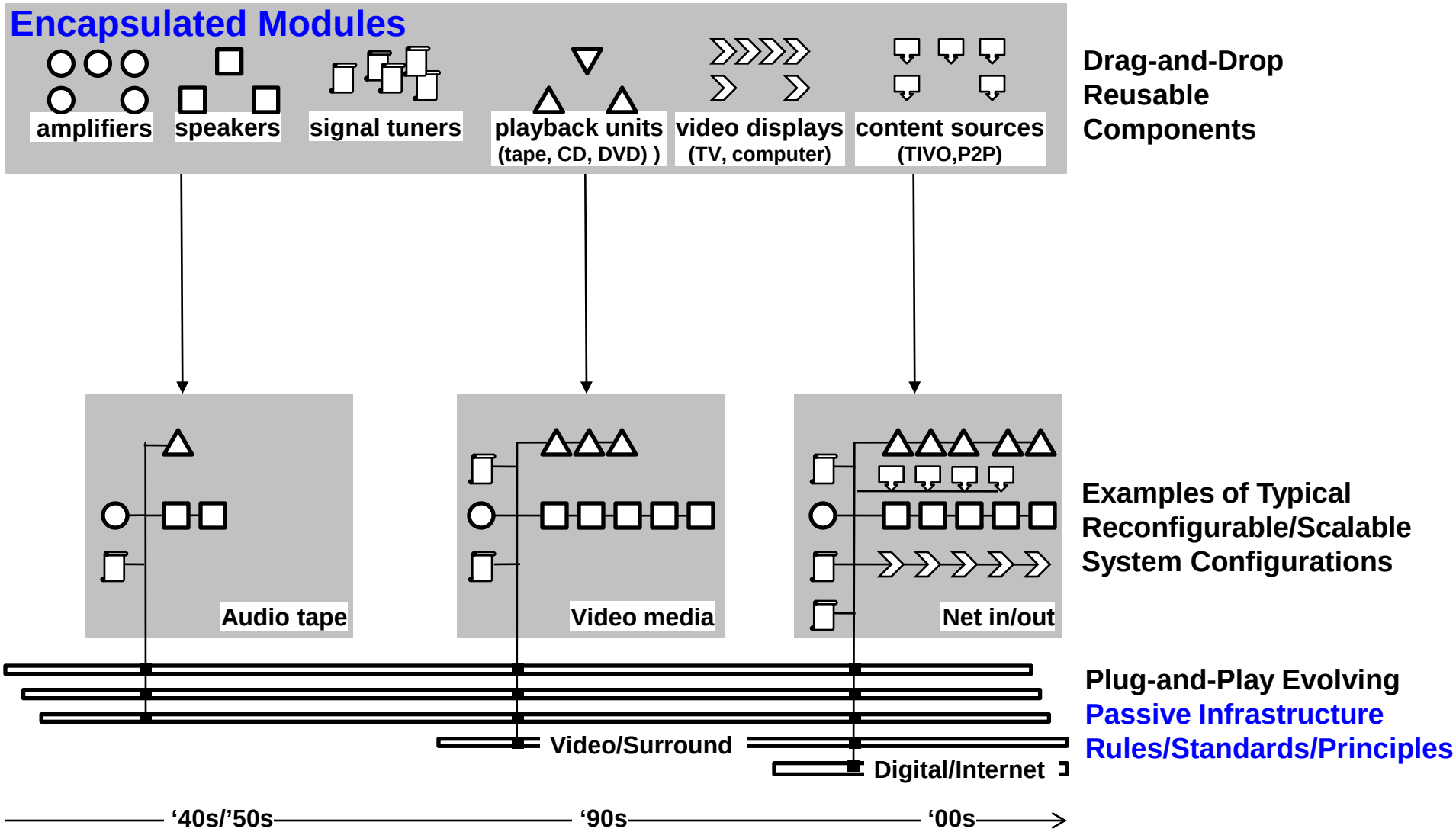
Case: Home Entertainment Technology Migration

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



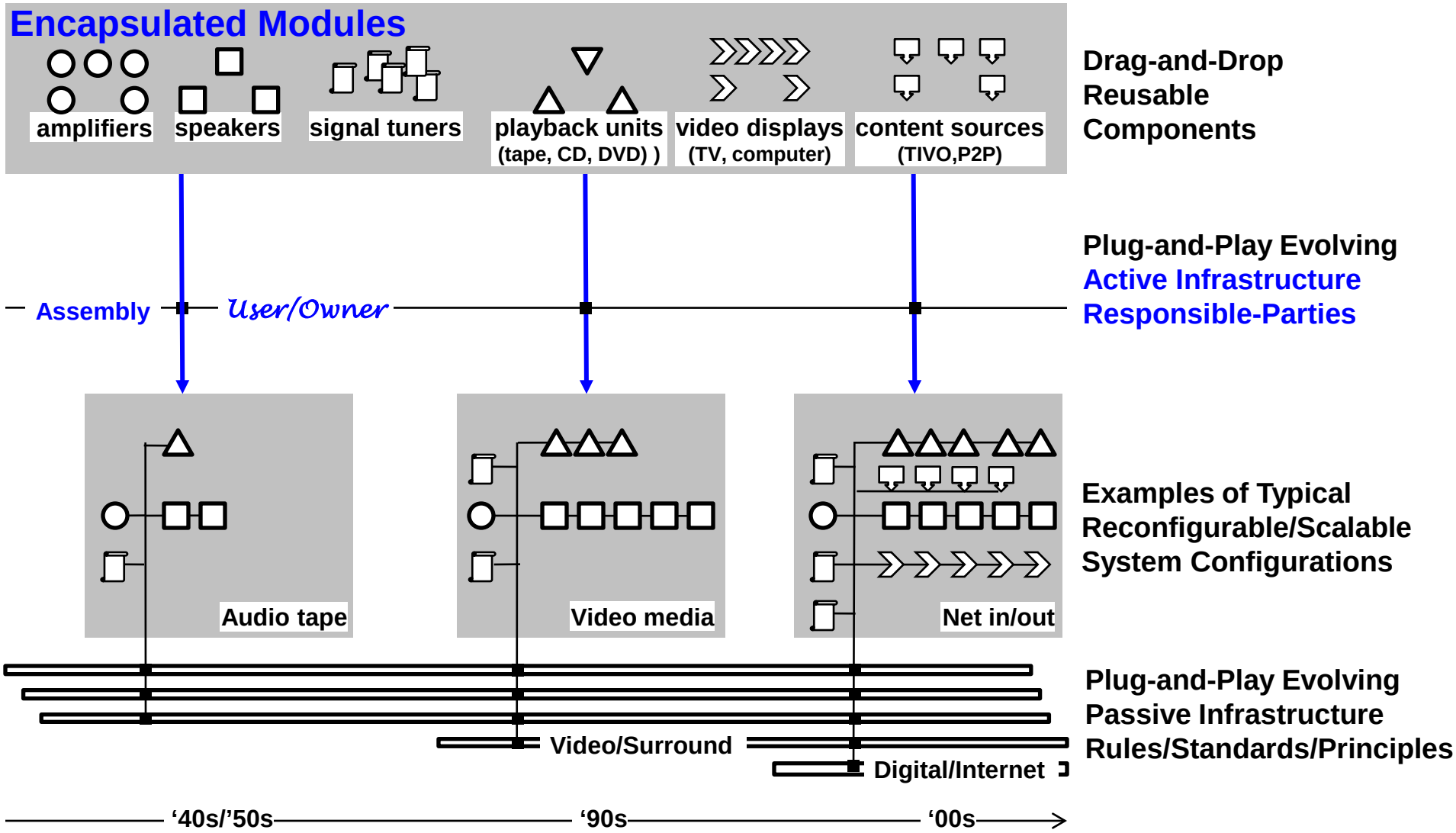
Case: Home Entertainment Technology Migration

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



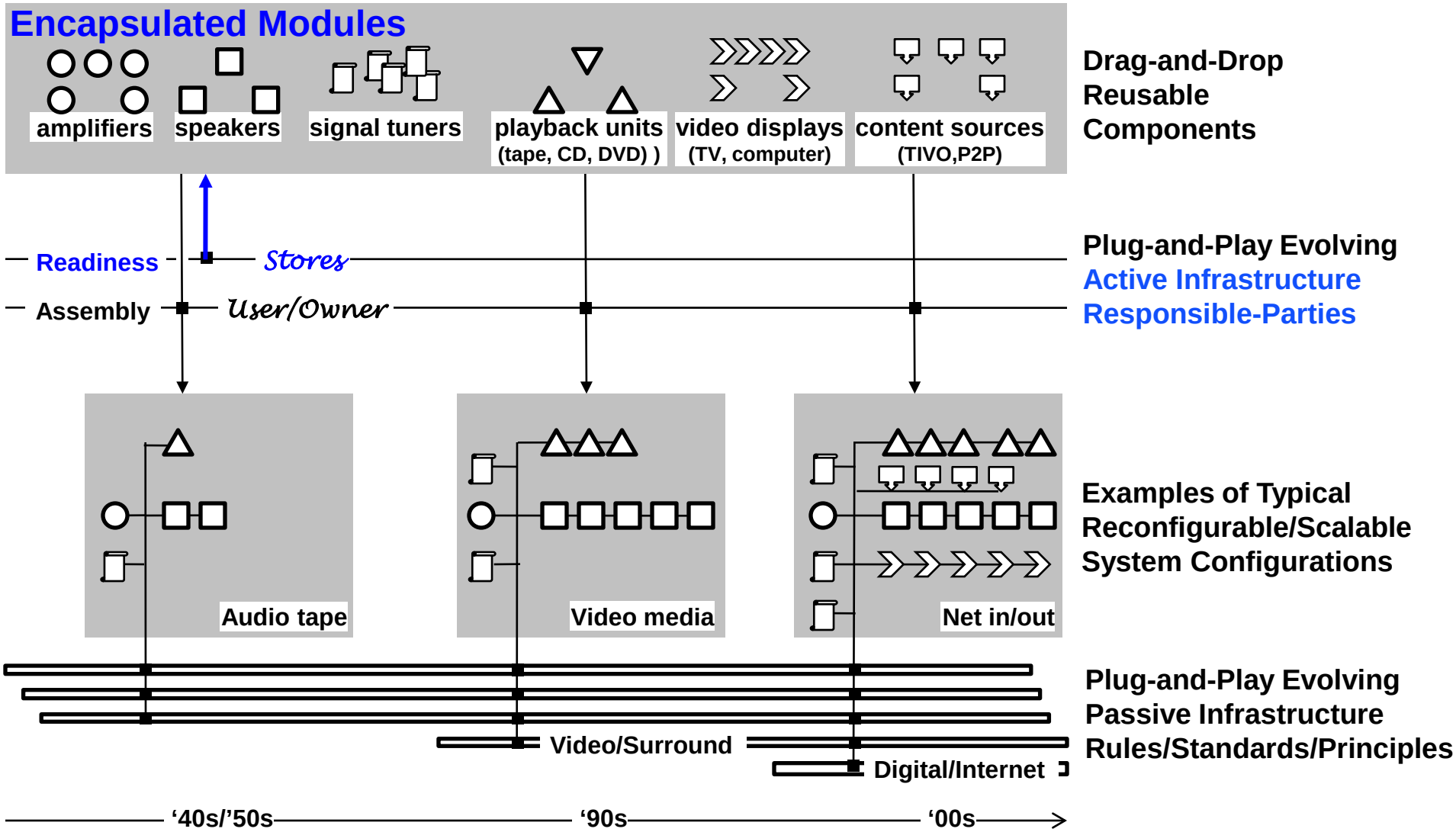
Case: Home Entertainment Technology Migration

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



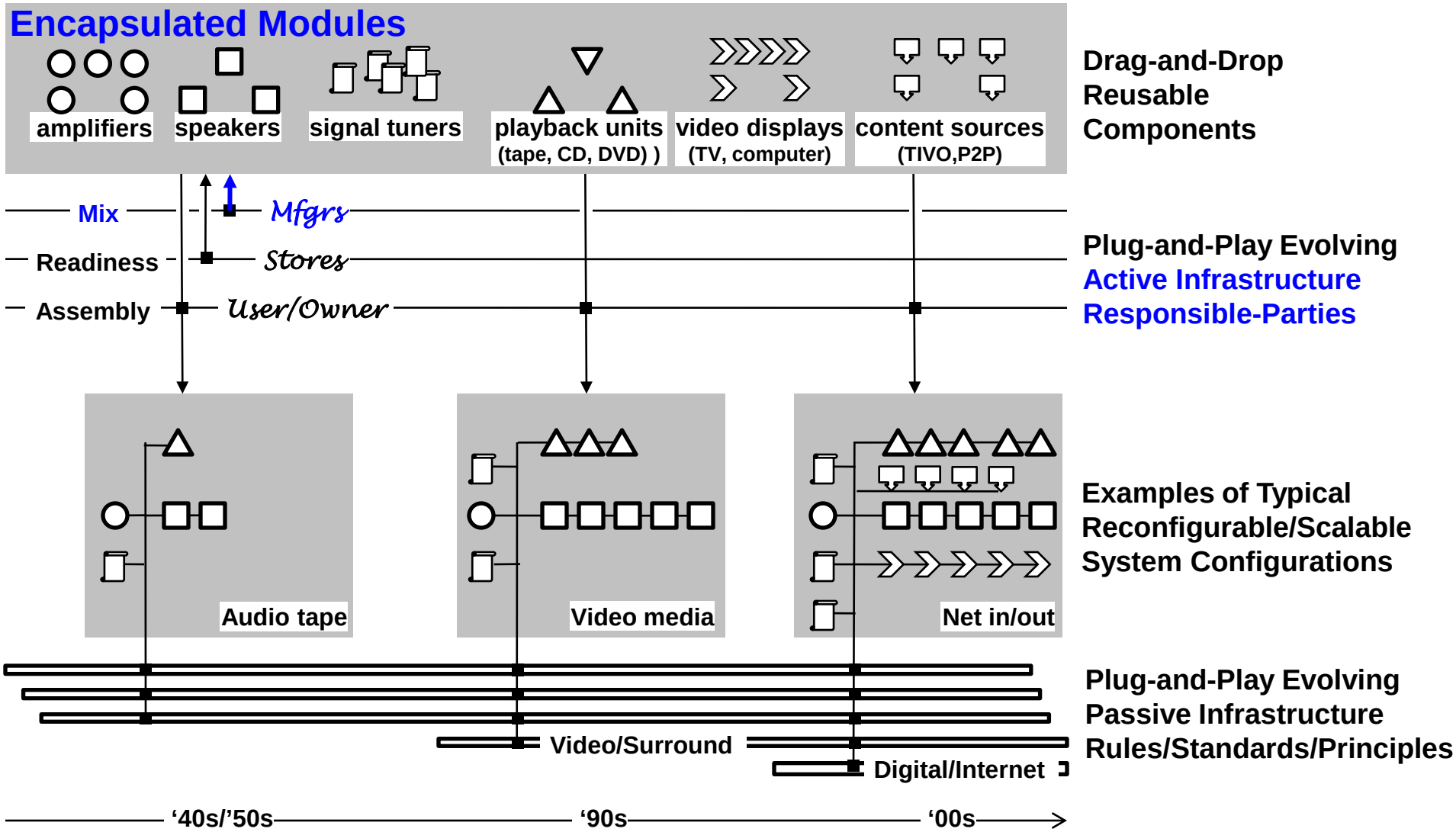
Case: Home Entertainment Technology Migration

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



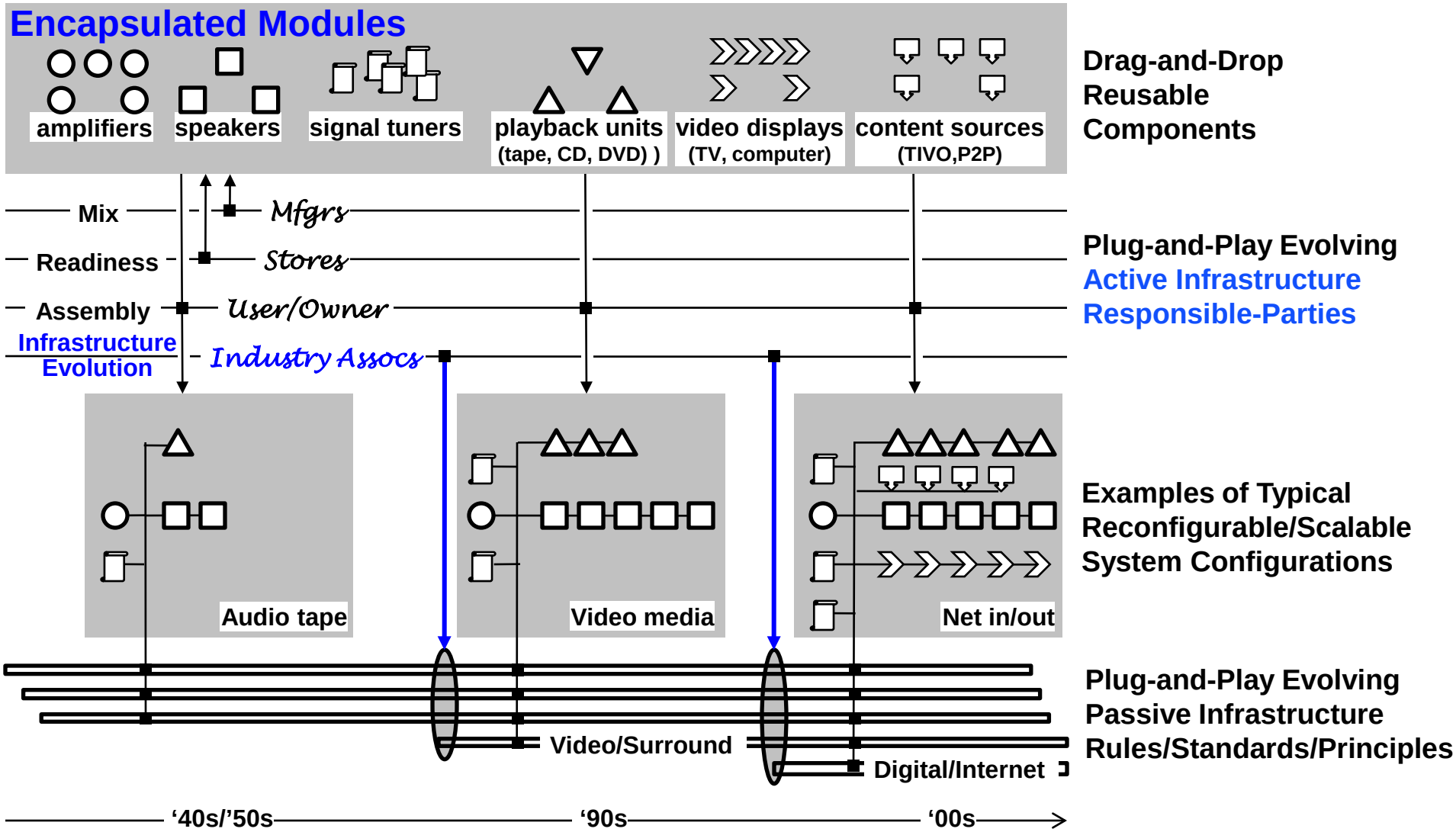
Case: Home Entertainment Technology Migration

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



Case: Home Entertainment Technology Migration

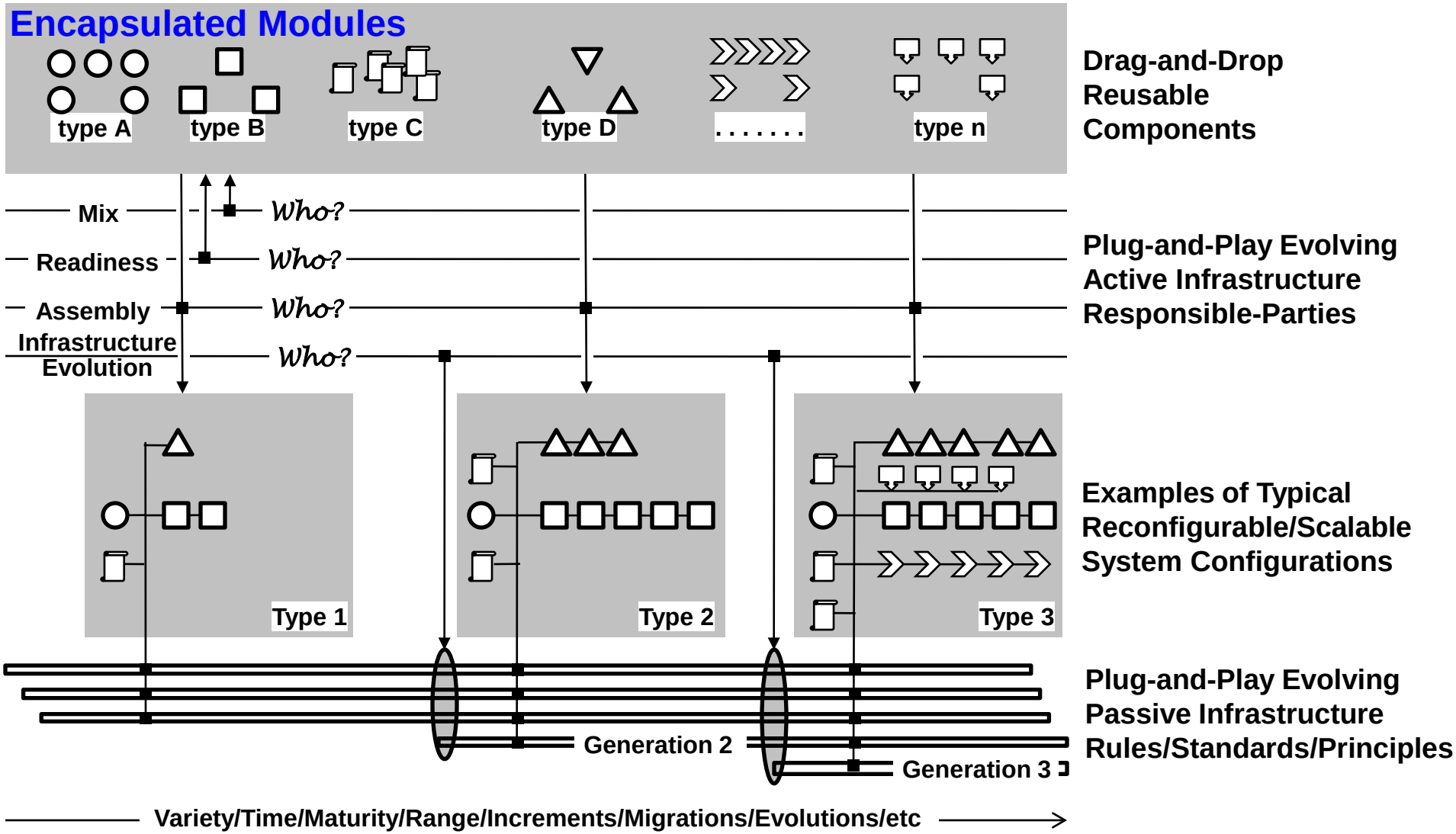
agile architecture pattern: drag-and-drop, plug-and-play



Fundamental Concept

Reusable modules Reconfigurable in a Scalable architecture (RRS)

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



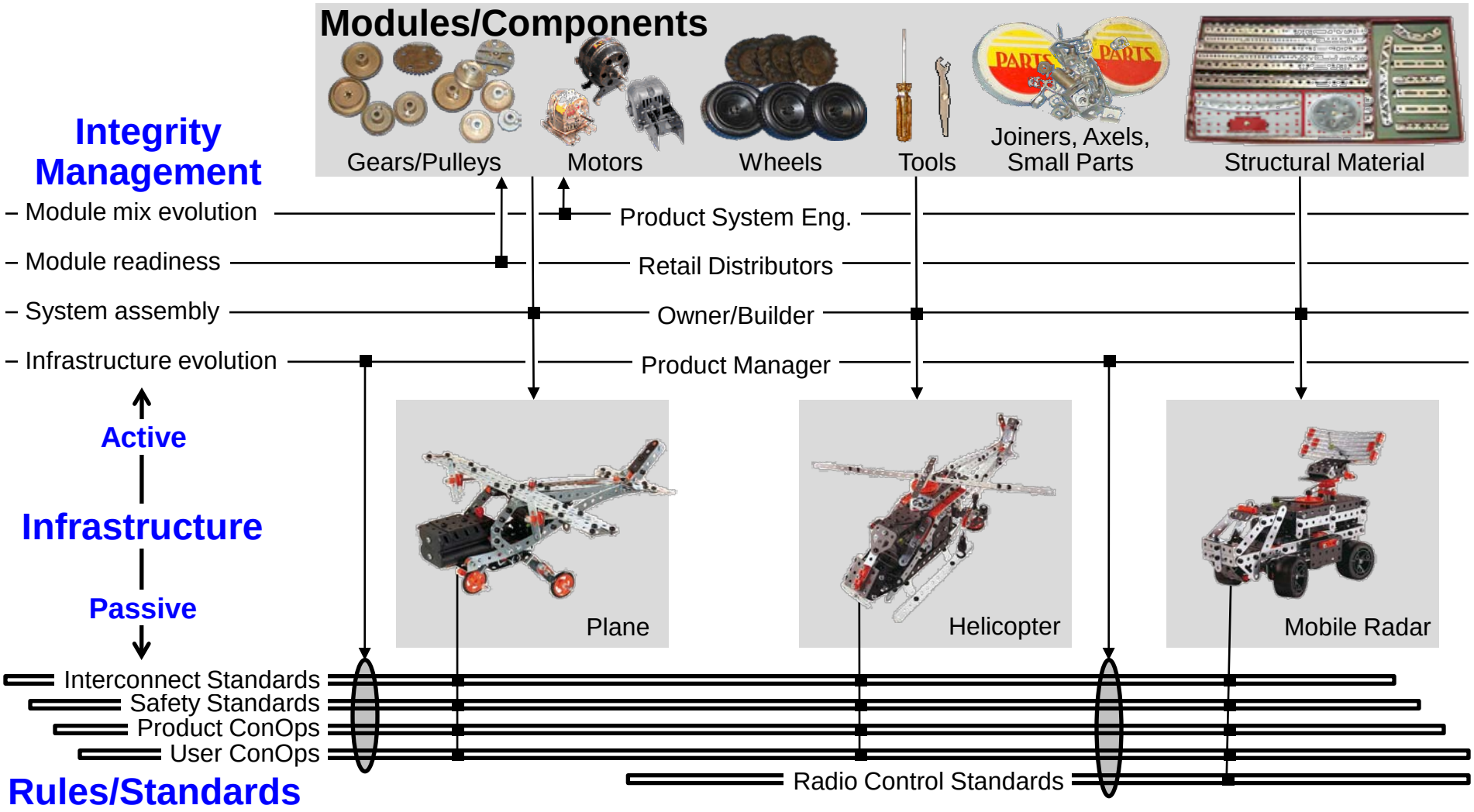
System X-Ray Vision
(revealing the bones)

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


http://www.vintagetoys.com/toys/classified/962
Here's a Box of Bones
Sensational New
ERECTOR
DEVELOPED AT THE GILBERT HALL OF SCIENCE
GILBERT
ERECTOR
No 10052
THE ROCKET LAUNCHER SET
Exciting Action
Rocket Launcher!
POWERFUL ELECTRIC MOTOR
PARTS
PARTS
www.girdersandgears.com/erector-1958-10052.html
ERECTOR=MECCANO

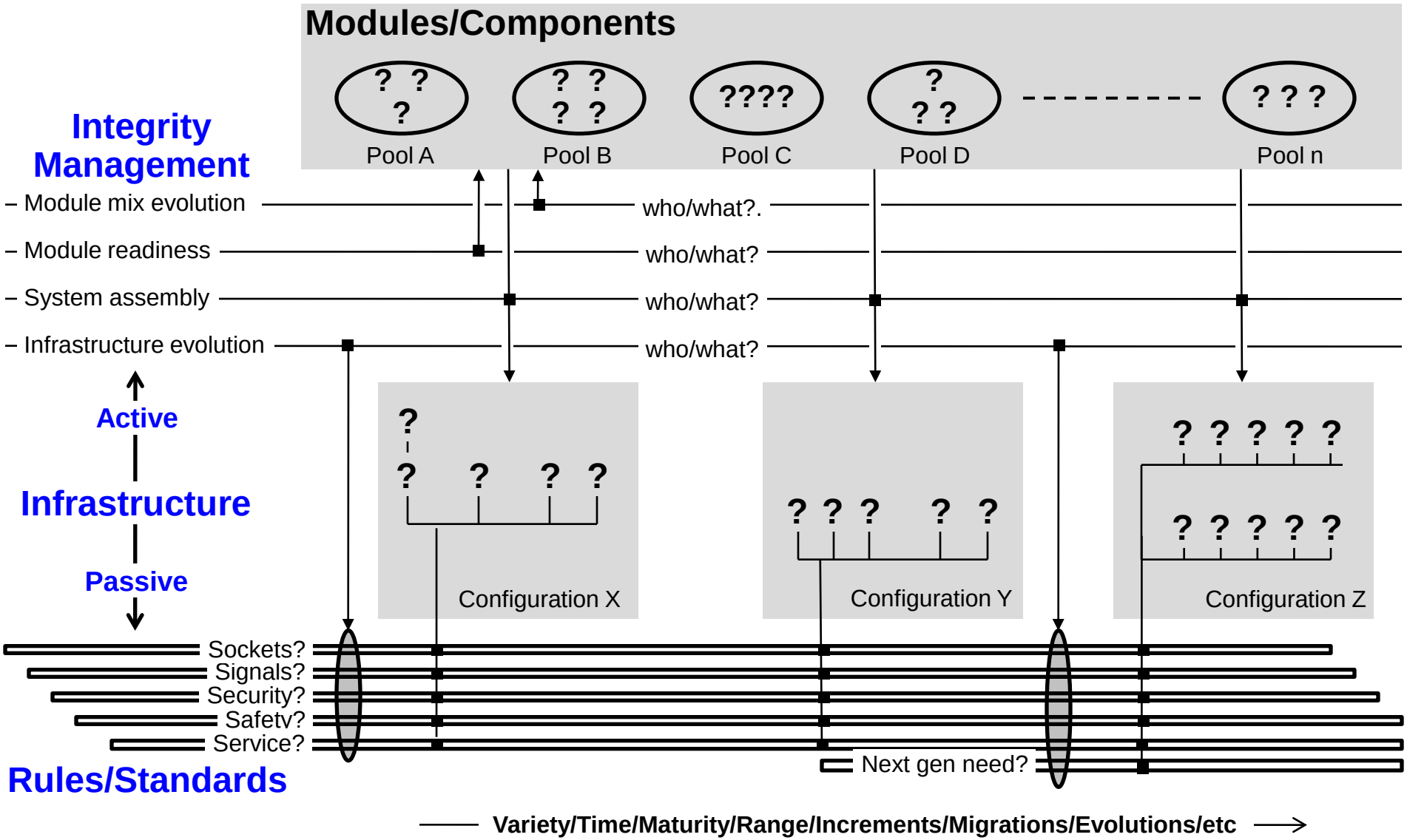
Here is a System-Construction-Kit System

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/Meccano 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?)

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 and 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**

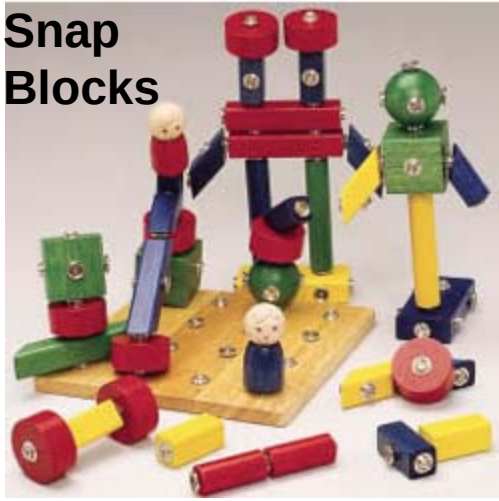
Design the Architecture of Your Construction Set

Bendables

Straws and Connectors

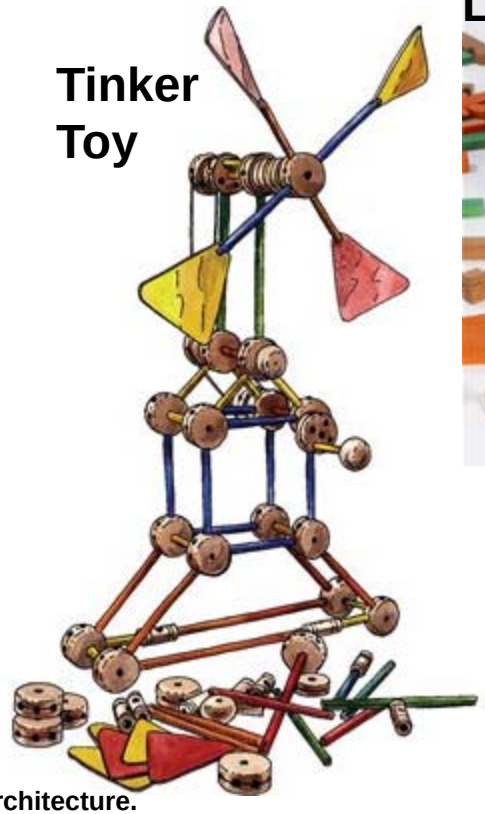
Marble Run

Snap Blocks



Woodbuilders

Tinker Toy



Log Builder

Lego

Bristle Blocks

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

rick.dove@parshift.com, attributed copies permitted

The UURV Environment Drives the Need

Agile systems are defined in counterpoint to their operating environments.

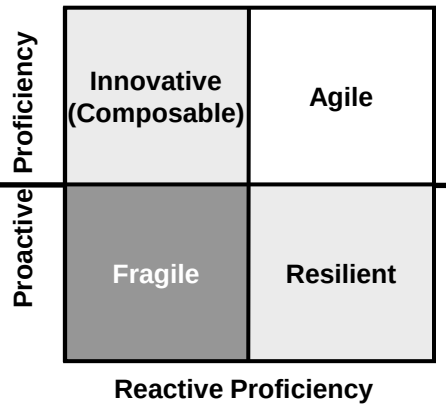
To design and develop a system that can deal effectively with changing environments it is useful to articulate the nature of changes that should be considered.

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.**

Response Situation Analysis (RSA)

	Change Domains	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	

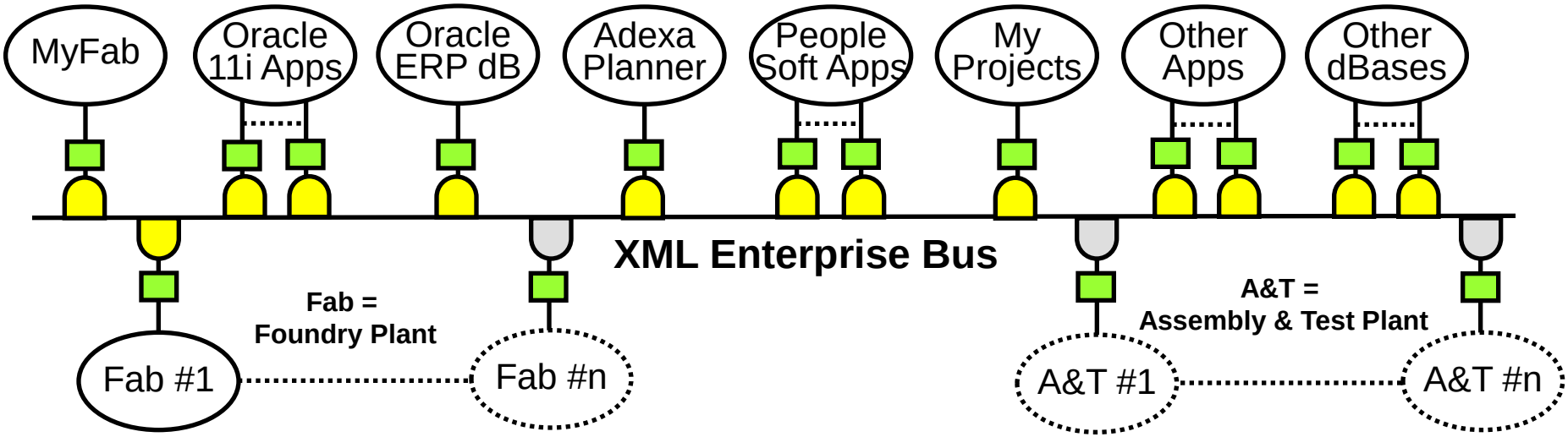


Pro Forma RSA for Agile SE Project Management

Response Metrics: t=time, cost, s=scope, p=predictability

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).

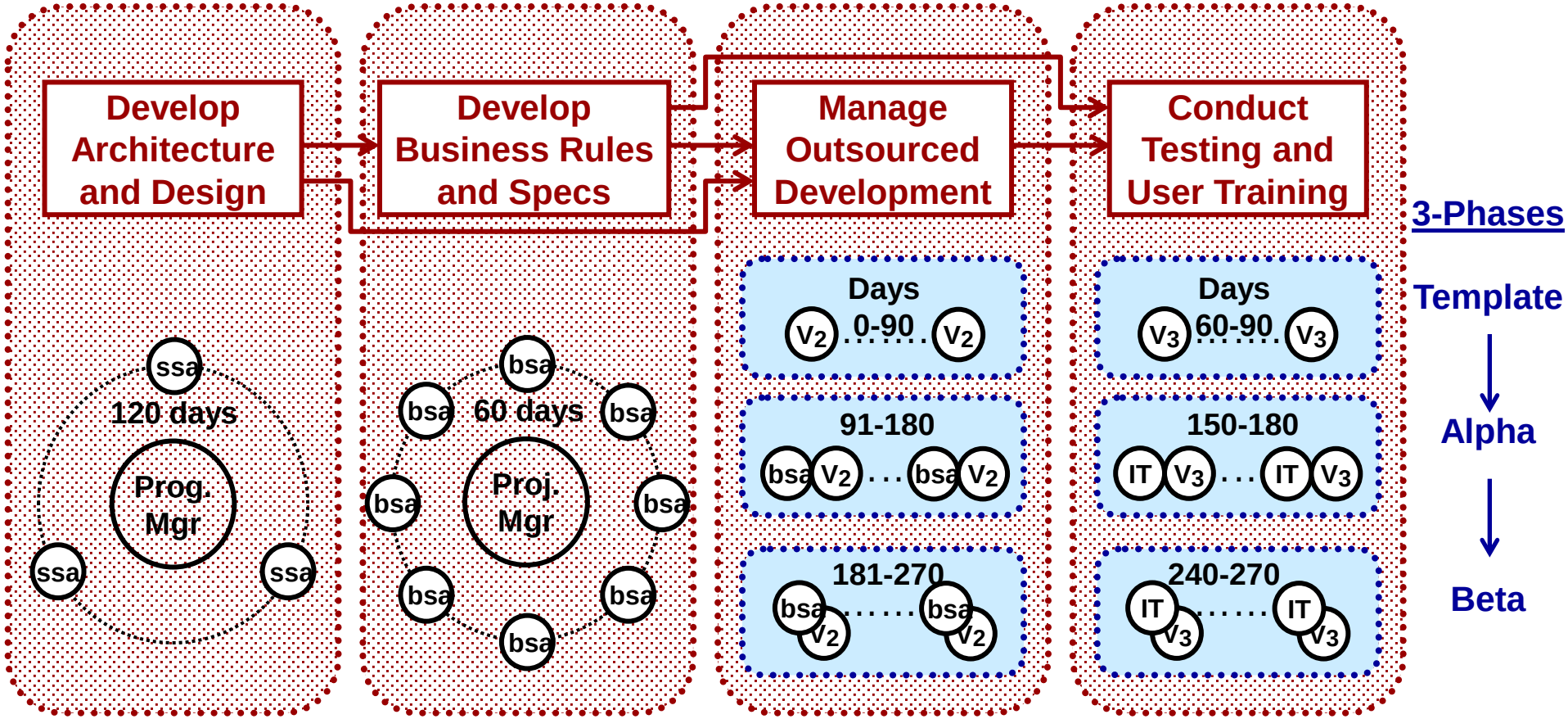
Enterprise IT-Infrastructure Design



- = 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

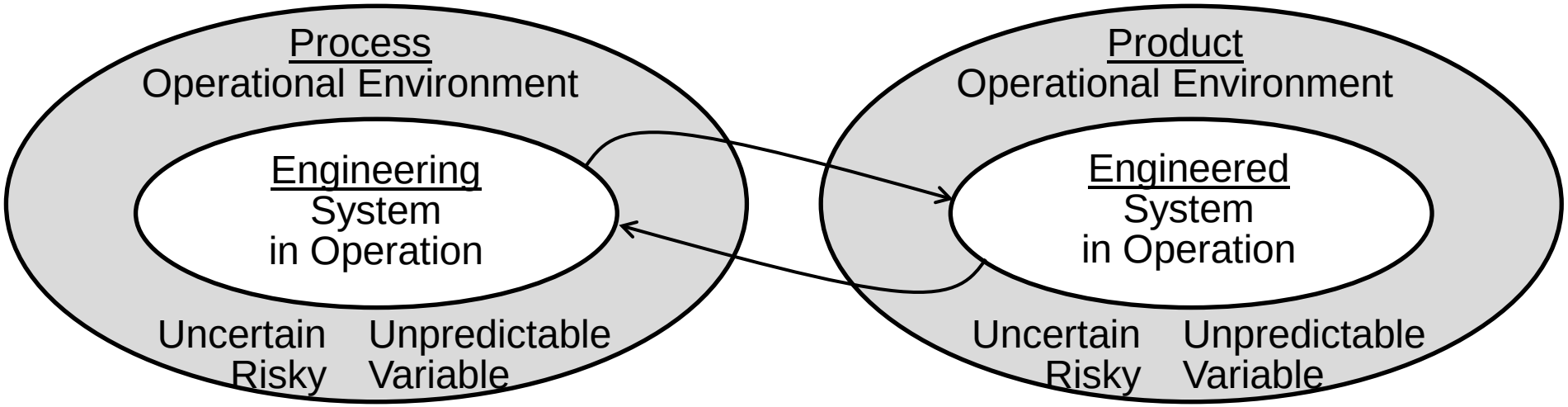
Encapsulated Development Process



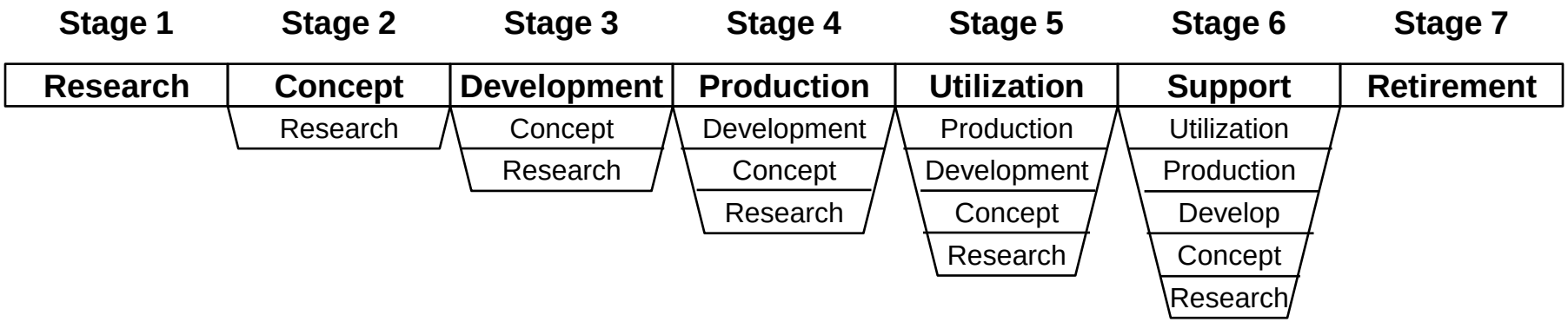
- Designed to Accommodate Requirements Evolution -

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

Agile Process and Agile Product



Agile SE Life Cycle Framework



“Classic” Scrum

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

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

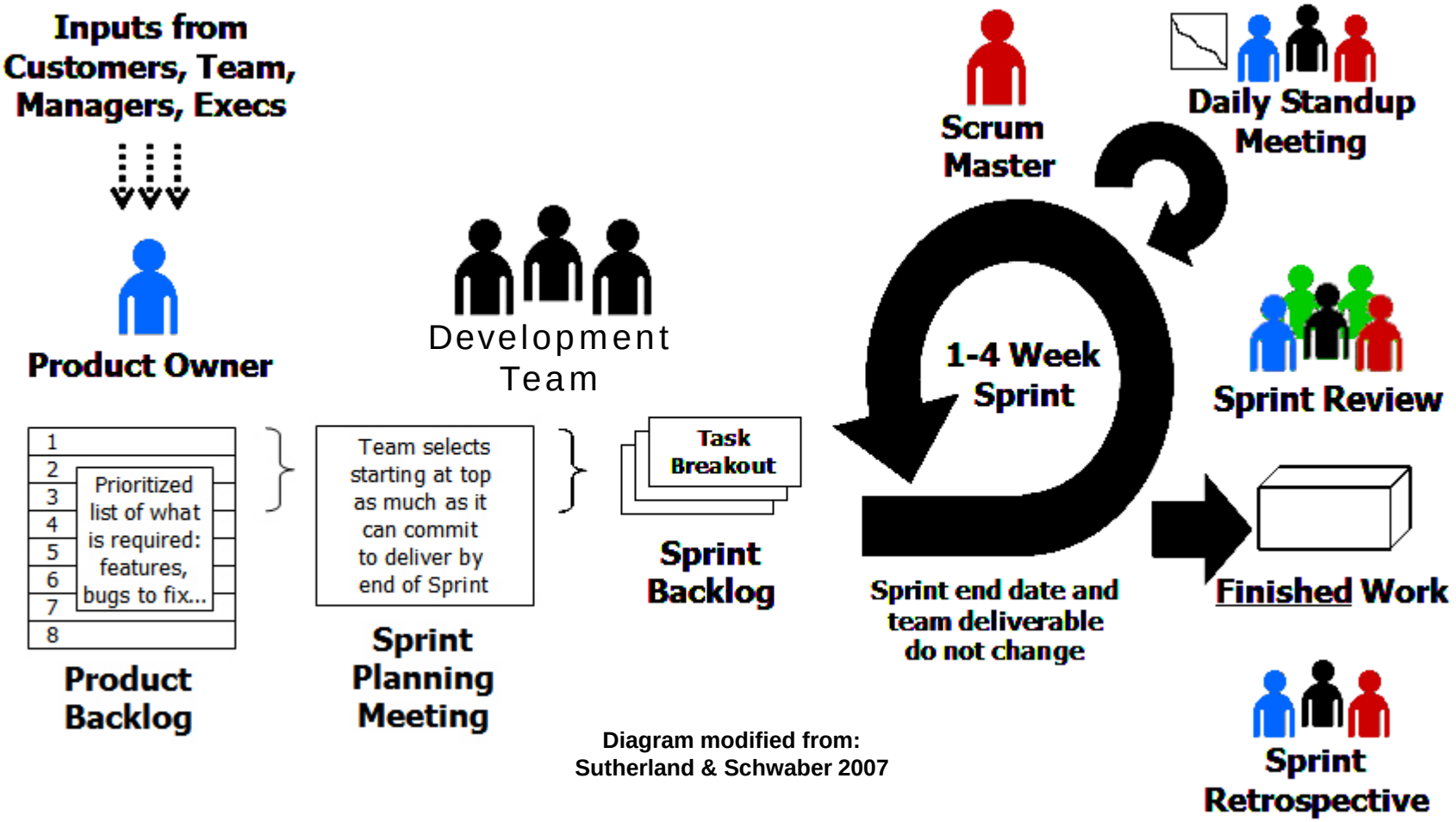


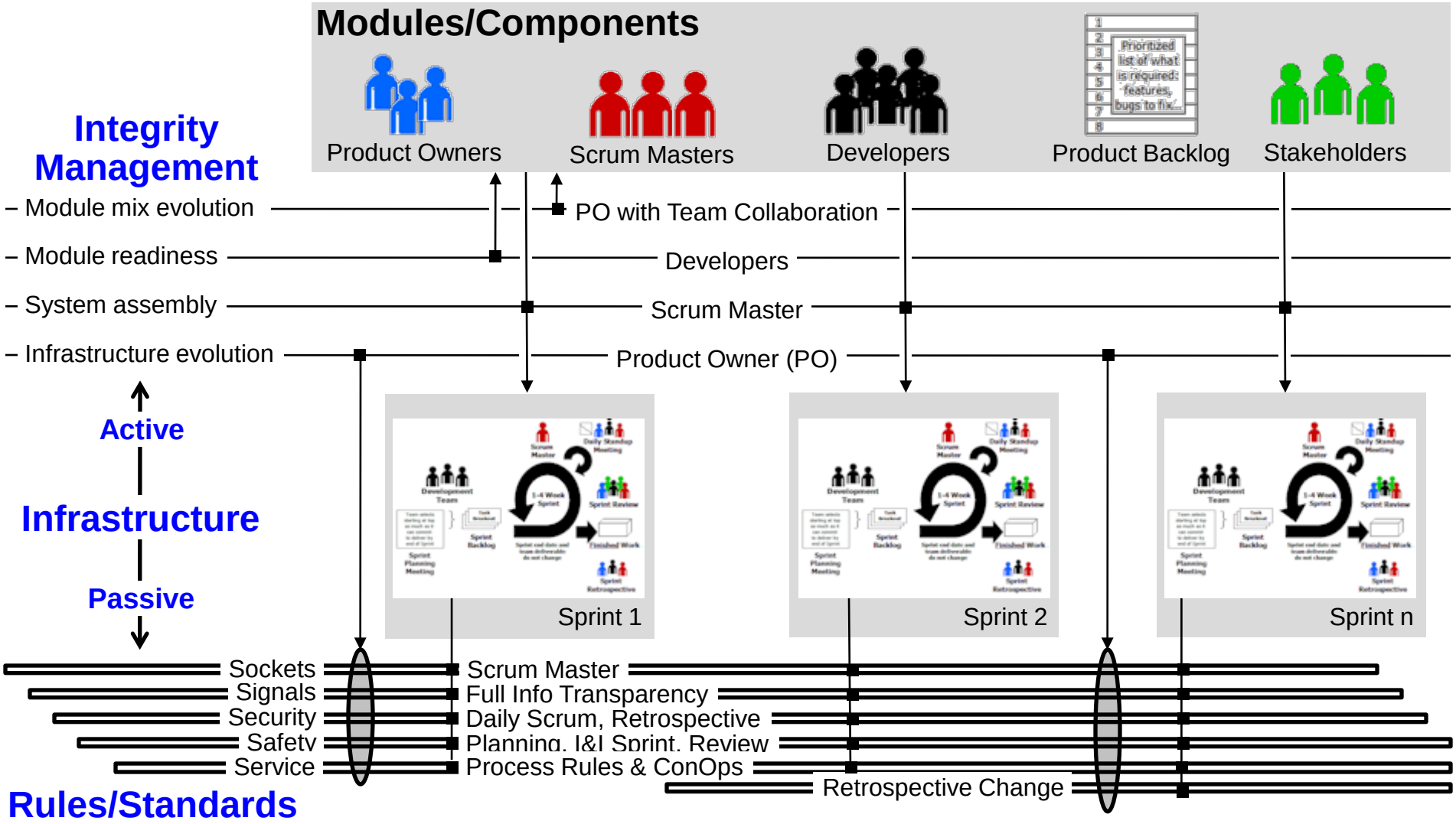
Diagram modified from:
Sutherland & Schwaber 2007

“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 …

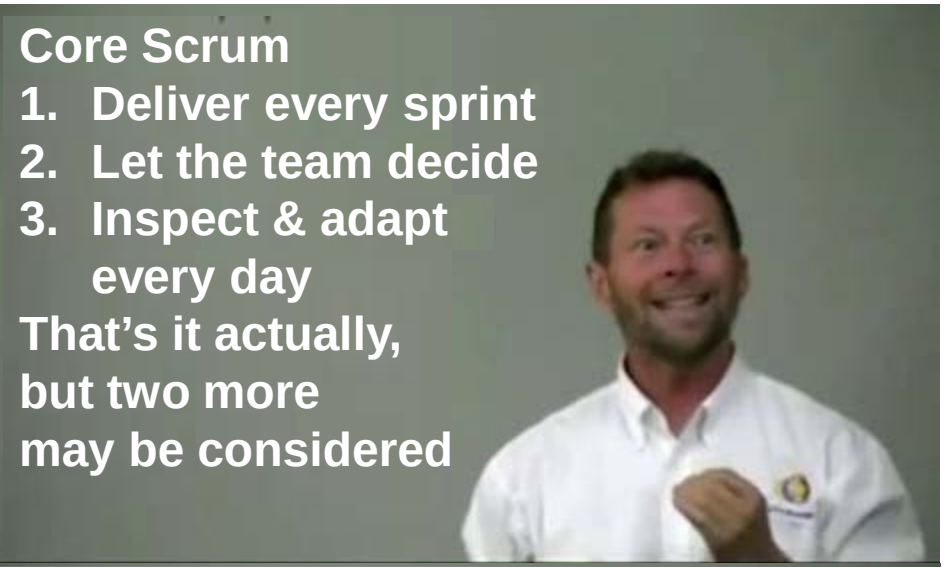


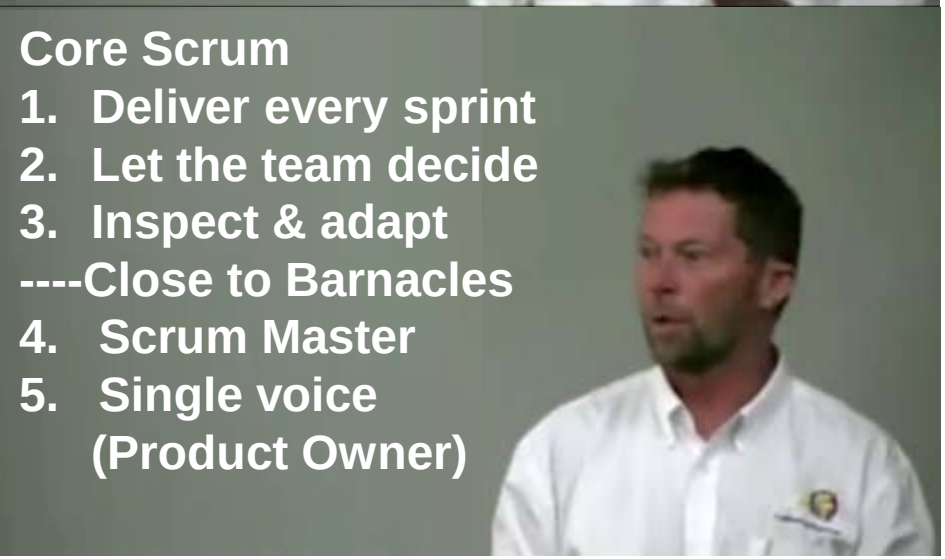
… because the RSA is different for an agile systems-engineering process, and the Scrum AAP *strategy* is inadequate for systems engineering

Alistair Cockburn on Scrum Dogma

File6.75

Core Scrum, Barnacles, Rumors & Hearsay





Everything falls into 3 categories: Core – Barnacles – Rumor/Hearsay

There's rules of the game and there's strategies for playing the game.

Burndown charts, for instance, might be a strategy, but are not rules, not core.

Planning poker is not part of scrum at any level – it's a cute idea. It's got zip-zero-nothing to do with Scrum.

Scrum Master is chief impediment remover. That's it.

Scrum helps you identify interesting questions but gives you no answers.

Video and text at: www.frequency.com/video/alistair-cockburn-core-scrum/129758869/-/5-7017272

Agile Software Development 2.0?

Lean and Agile Confusion and Complementarity

We see the phrase "Lean and Agile" appearing together as what might be Agile Software 2.0 complimenting emergence.

Joshua Kerievsky has a thought-starting presentation on "Lean Startup" at www.infoq.com/presentations/Lean-Startup.

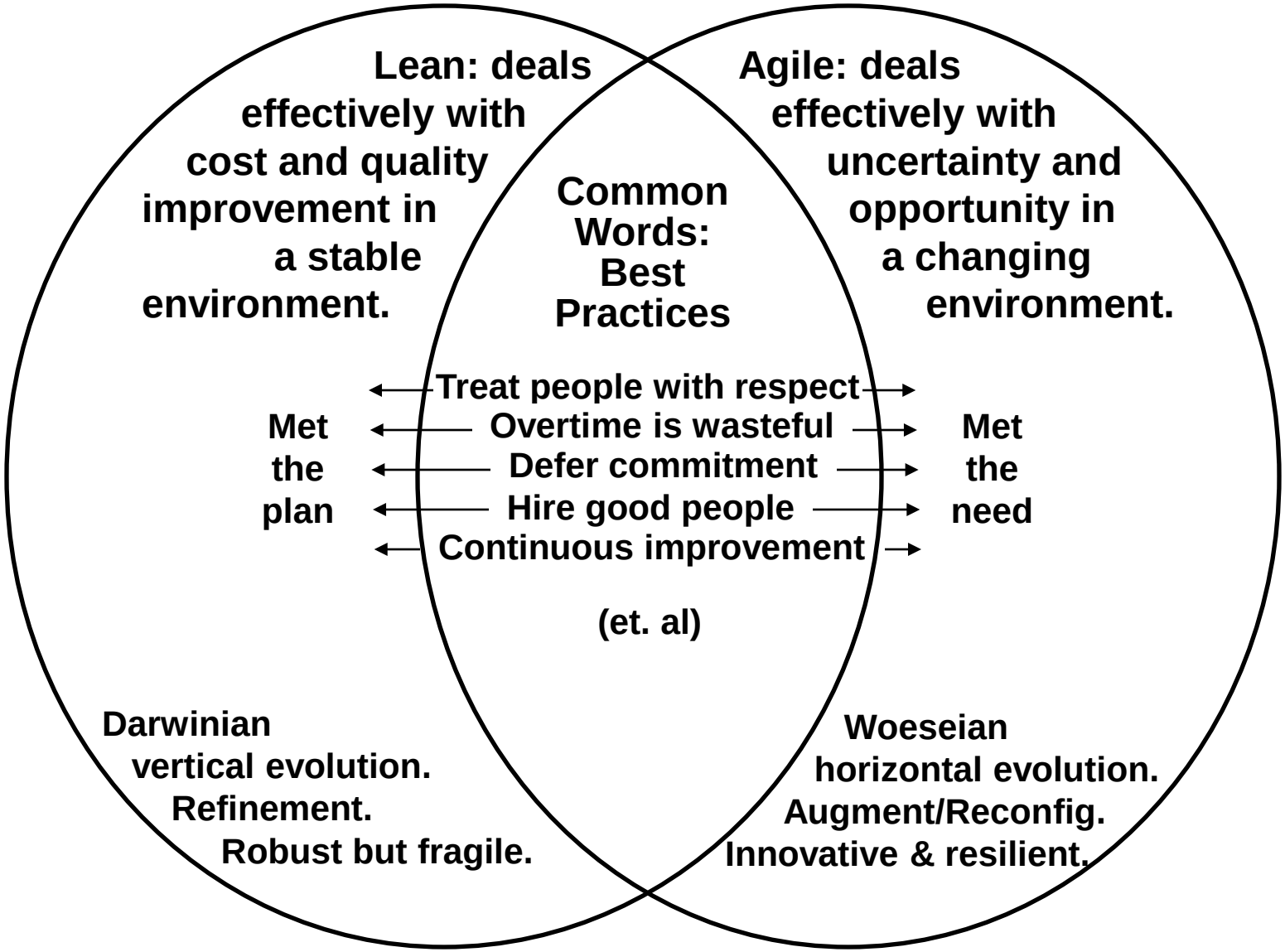
Kerievsky's presentation reveals the emergence of an Agile 2.0 understanding, but his choice of rhetoric polarizes this emergence as a lean dominated lead.

Fundamentally he is showing how a more mature understanding of agile software development is removing the wasted resource commitments in the first generation of agile software concepts - a timey rethinking and a good start.

But…The roots of agile and lean should be kept in mind.

Both Seek Customer Satisfaction

Each adopts and adapts practices that help achieve their definition of satisfaction. They are appropriate philosophies in relation to a context.



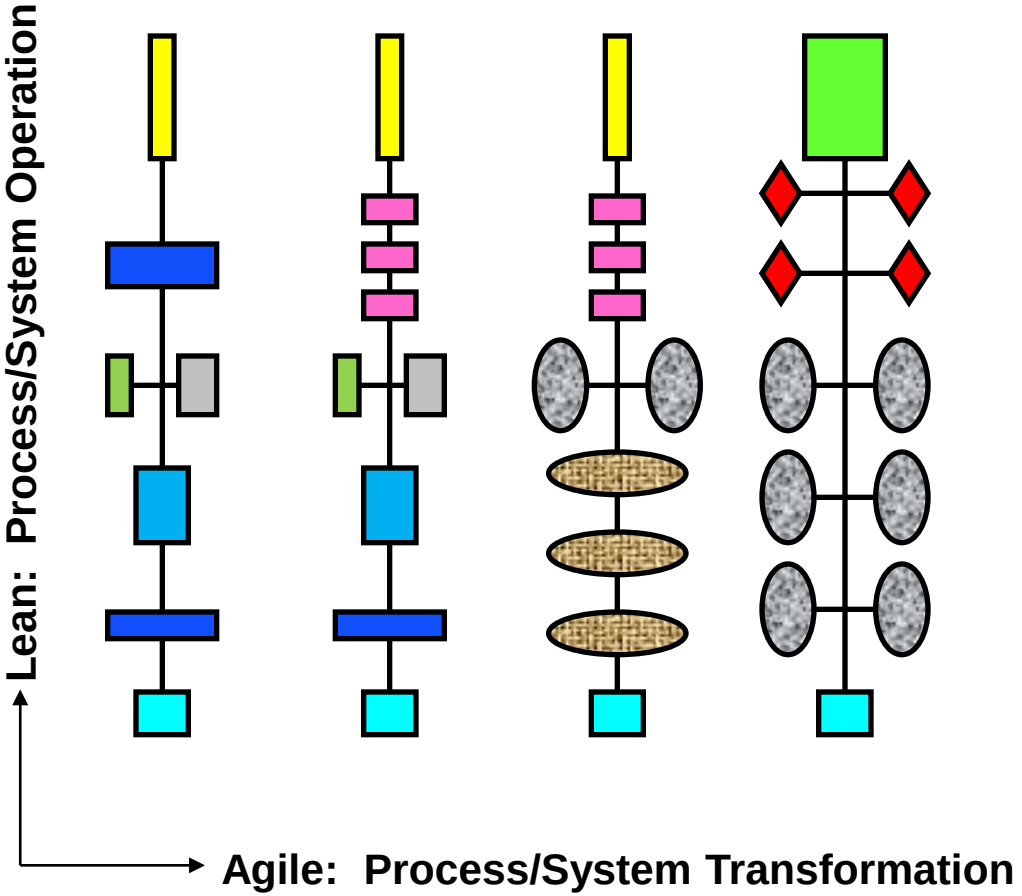
Lean & Agile: Orthogonal Focus

Agility deals with "design-for-transformation".

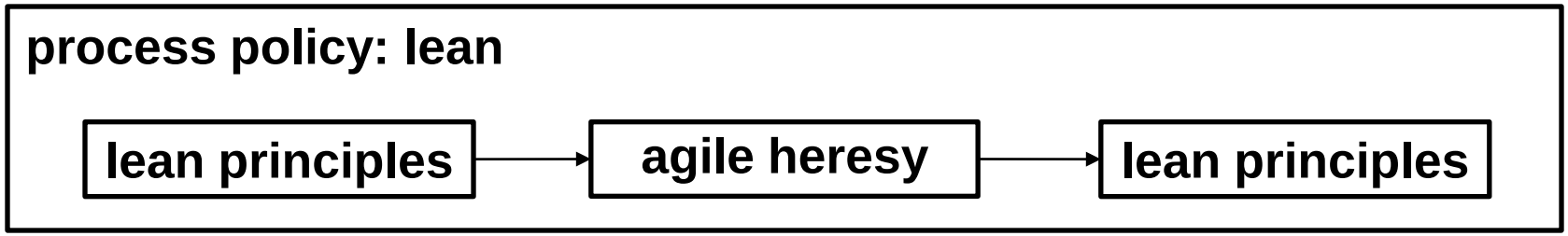
In a very general interpretation,

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

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

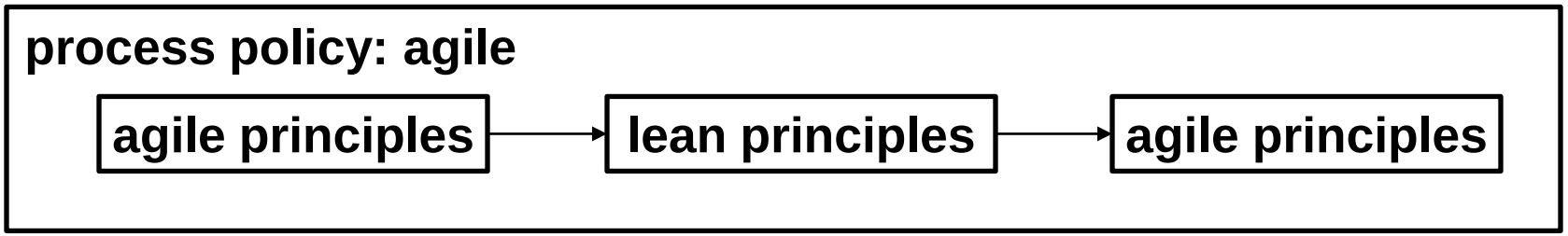


Both are concerned with operational effectiveness.
But the two have a different means for achieving different ends.



lean compromises its principles to incorporate agile principles

Agile can incorporate lean without compromise



agile improves with “selected” incorporation of lean principles

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

(Dove 2014) Rick Dove and Ralph LaBarge. Agile Systems Engineering – Part1 and Part 2. Submitted to INCOSE IS14. www.parshift.com/s/140721IS14-AgileSystemsEngineering-Part1.pdf, www.parshift.com/s/140721IS14-AgileSystemsEngineering-Part2.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

Epilog – System Security?

The Adversary is Very Agile

Architecture:

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

Behavior:

- Swarm intelligence
- Tight learning loops
- Fast evolution
- Innovative
- Disposable resources

Attacking a Lean security strategy

Planned vs Agile Security

Automated defenses can work against automated attacks. But the sophisticated attack is a manual operation, requiring mano-a-mano defense. An 80-20 distribution by Lockheed's thoughts, and the 20% are focused on high value results that justify high talent, large investment, and plenty of time.

"They're going to keep innovating, I need to out innovate them."

…Eric M. Hutchins, Lockheed

Lead author: Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaign and Intrusion Kill Chains

Minimum: 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.

Start at the Concept of Operations

Continuous evolution of system security is necessary to maintain parity with a continuously evolving threat environment: effective response under unpredictable and uncertain circumstances, as often as necessary.

The system ConOps should call out the ability to reconfigure and augment system security throughout the development and operational lifecycle of the system, and it should call out the need for rapid reconfiguration of security at the system level

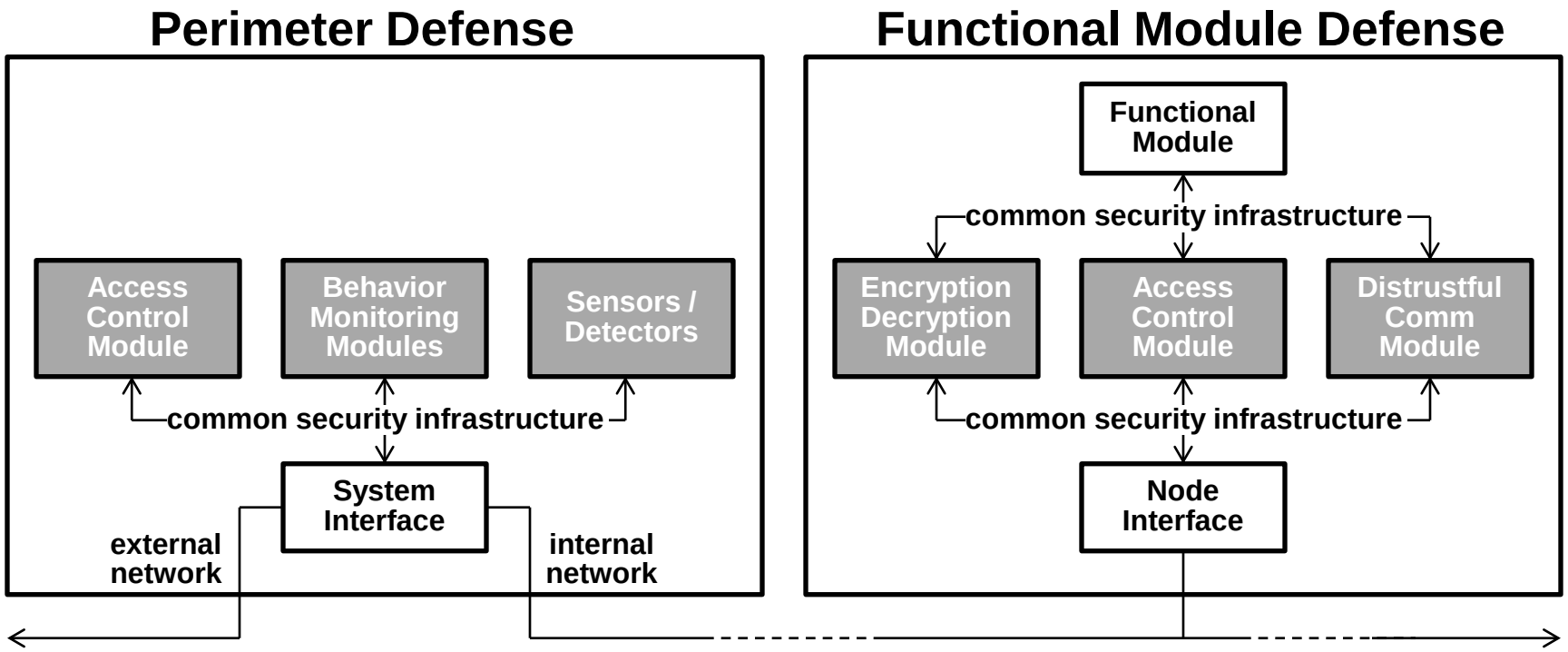
An agile-system concept of operations recognizes the need for effective asynchronous system-security change.

Effective response has four metrics: timely (fast enough to deliver value), affordable (at a cost that can be repeated as often as necessary), predictable (can be counted on to meet the need), and comprehensive (anything and everything within the mission boundary).

Value Proposition—Risk management in an evolving unpredictable environment is the value proposition for agile systems. An agile system is constructed to enable and facilitate augmentation, reconfiguration and scalability of reusable assets in response to unpredictable situations, and agility is sustained with active management of responsibilities that constantly evolve the agility-enabling capabilities.

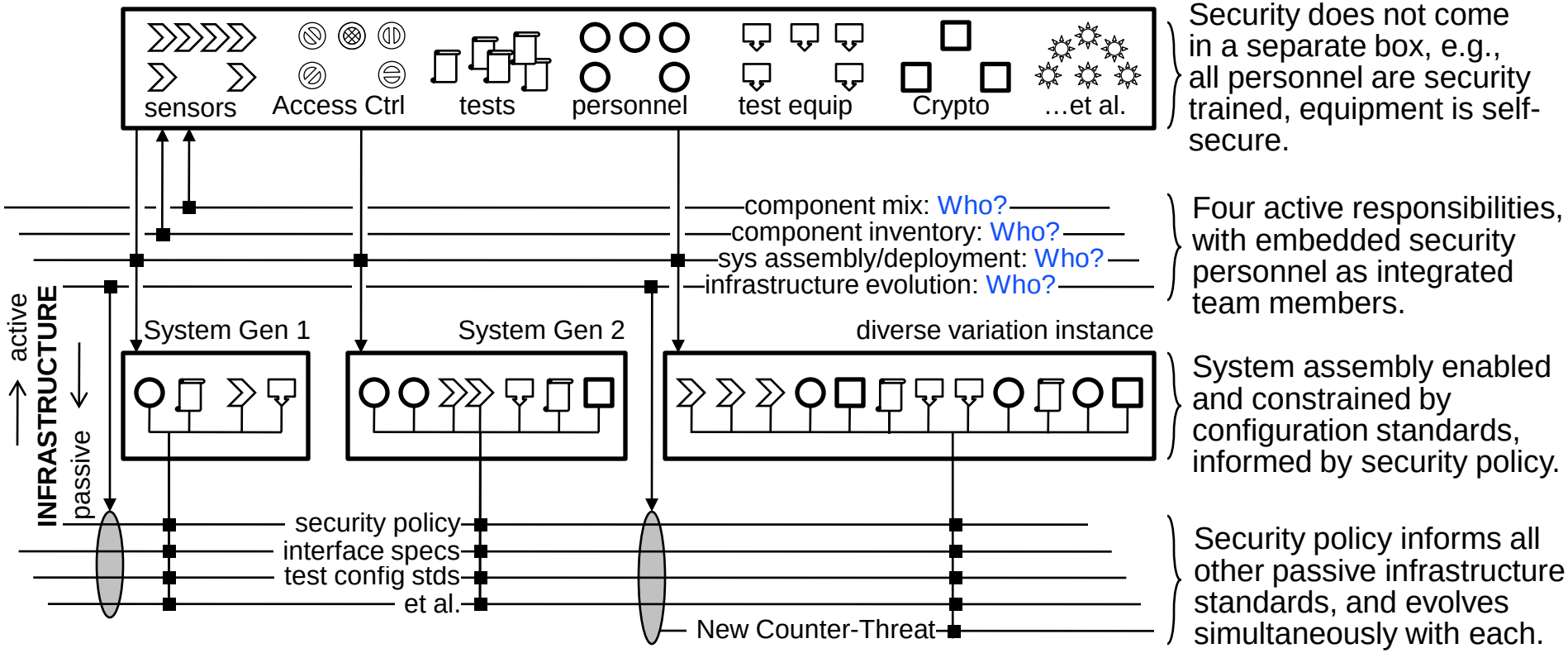
.

Reconfigurable/Scalable Notional Concept



Common security passive infrastructure enables rapid evolution, augmentation, reconfiguration. Other security modules included as desired, e.g., functional-module behavior monitoring, to name one only.

Agile Architecture for Security Reconfiguration/Augmentation/Evolution



But You Already Have Lean Security. Why Bother?

Panel: Agile and Lean SE
Are these two complementary or are they different and should be applied in specific environment
15:00 – 15:30

Lean and Agile Systems Engineering
The Gordon Center For Systems Engineering
Haifa, 06-Jan-2014

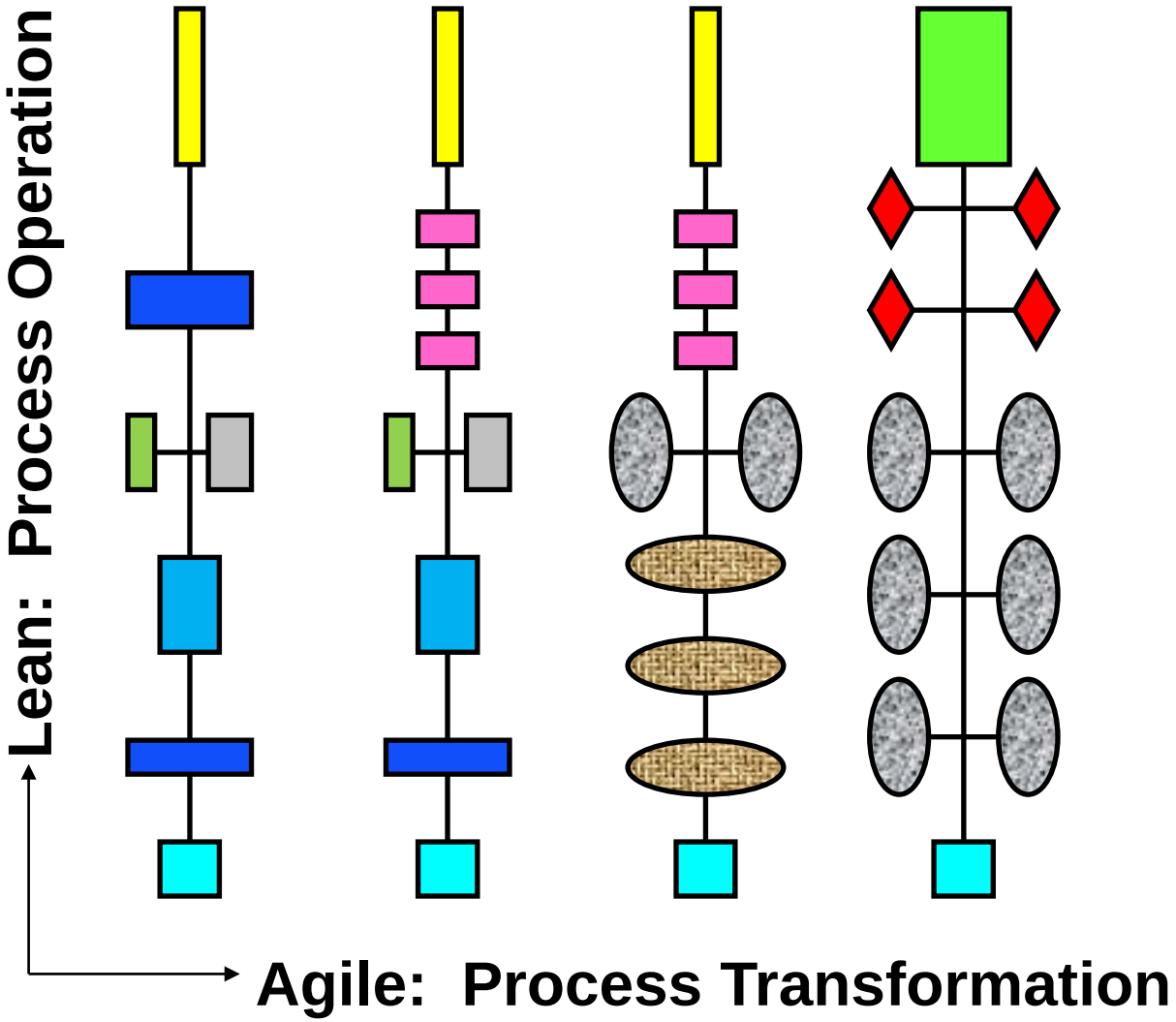
Position Statement

Rick Dove, Stevens Institute of Technology


Lean will straighten it out


Agile
will roll
with it.
AQUA SLALOM

Lean & Agile: Orthogonal Concerns



Brainwash & Blame Loss

Acquisition and contracting want to believe there are stable and predictable project execution paths. No surprises.

If things go wrong it is clear who's to blame.

If you want a cheap thinking result, pick either one and turn the crank.

- **I did what they said to do (blame loss).**
- **I kept it simple and followed the rules (brainwash).**

Lean was developed for stable environments:
manufacturing in a closed system, stable success metrics

Agile was developed for unstable environments:
enterprise in an open system, dynamic success metrics

Manufacturing knows what has to be done for every product.
Systems engineering has to figure that out differently for every product.

- **Can you standardize the SE process life cycle to eradicate surprises?**
- **Can you hire laborers or use robots to operate the process?**

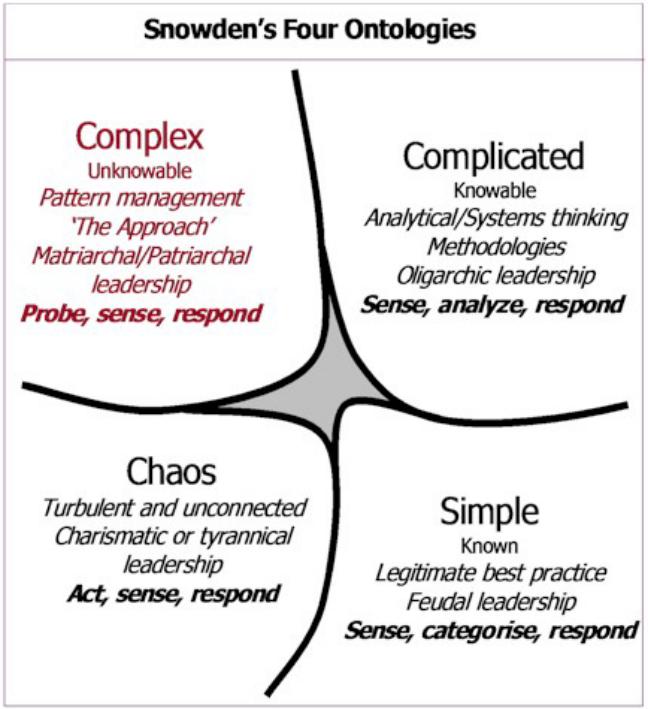
When you have a stable or repetitive Systems Engineering environment, Lean is a good guiding approach to take.

These two approaches, at core, are in conflict.

Q: Are these two approaches complementary or are they different and should be applied in specific environment?

Only two choices? Complimentary or exclusive?

Dave Snowden has a handle on it. Every project has its own operational personality at every point in time, and that personality is unlikely to be static. You have to be vigilantly aware, understand what is happening at every point in time, and re-fit your approach as the situation changes.



Project management cannot win with lean thinking.

But within agile project management, use lean as a tool of preference when applicable (things are under control & predictable).

David J. Snowden and Mary E. Boone. 2007. A Leader's Framework for Decision Making. Harvard Business Review.