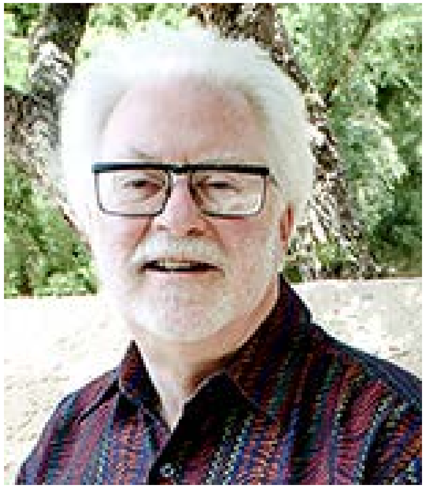
Enabling and Facilitating Agility in Systems Engineering and Hardware Development

North Texas Chapter, 14-March-2017

Enabling and Facilitating Agility in Systems Engineering and Hardware Development

Abstract: Initial findings for generic fundamental agile systems engineering life cycle patterns and principles arising from four analytical investigations will be reviewed: 1) Lockheed: Transition to a tailored SAFe-like process for 1200 engineers and executives; 2) Rockwell Collins: Market awareness-driven Product Line Engineering for mixed Hardware/Firmware/Software products; 3) Northrop Grumman: Continuous evolution of a security-critical systems-of-systems multi-database portal; 4) SpaWar System Center Pacific: Evolutionary development of innovative technology with mission-engaged subcontractors. A special focus will review supporting infrastructures for agile hardware development.



Bio: Rick Dove is a leading researcher, practitioner, and educator of fundamental principles for agile enterprise, agile systems, and agile development processes. In 1991 he initiated the global interest in agility as co-PI on the seminal 21st Century Manufacturing Enterprise Strategy project at Lehigh University. Subsequently he organized and led collaborative research at the DARPA-funded Agility Forum, involving 250 organizations and 1000 participants in workshop discovery of fundamental enabling principles for agile systems and processes of any kind. He is CEO of Paradigm Shift International, specializing in agile systems research, engineering, and education; and is an adjunct professor at Stevens Institute of Technology teaching graduate courses in agile and self-organizing systems.

He chairs the INCOSE working groups for Agile Systems and Systems Engineering, and for Systems Security Engineering, and is the leader of the current INCOSE Agile Systems Engineering Life Cycle Model Discovery Project. He is an INCOSE Fellow, and the author of *Response Ability, the Language, Structure, and Culture of the Agile Enterprise*.

Context

In The ‘90s we analyzed hundreds of real-world systems that exhibited agility, asking how they did that, and converged on fundamental structural patterns that fit facts.

We are now analyzing real-world processes that exhibit agility, asking how they do that, and converging on fundamental behavior patterns that fit facts.

No conjecture, no kinda good idea, no opinion.

An INCOSE Technical Product project:
Agile Systems Engineering Life Cycle Model (ASELCM)
(Project details at: www.parshift.com/ASELCM/Home.html)

Agenda

Overview: Systems Agility Fundamentals

Case: SSC-PaC – Wave process

Case: Northrop Grumman – Scrum/Wave process

Case: Rockwell Collins – Product Line process

Case: Lockheed Martin – Tailored SAFe transition process

Overview: ASELCM-Project findings

Epilog: Agile Hardware-Development Infrastructures

Note: Many slides will be shown for graphic context w/o detailed discussion in the time permitted. Case study papers are reference-linked for details.

Why Agility Matters

CURVE

Internal and external environmental forces that impact project/process/product as systems

Capriciousness: unanticipated system-environment change

Uncertainty: kinetic and potential forces present in the system

Risk: relevance of current system-dynamics understanding

Variation: temporal excursions on existing behavior attractor

Evolution: experimentation and natural selection at work

(CURVE: formerly known as UURVE, Capriciousness = Unpredictability)

SE-Process CURVE Environment

Agile systems have <u>effective</u> situational response under (some examples):

- **Caprice (Unpredictability): unknowable situations**
  - **Urgent need occurs**
  - **Un-availability of key personnel and/or subcontractor**
- **Uncertainty: randomness with unknowable probabilities**
  - **Feasibility of solution design**
  - **Contracting issues, funding gaps, and budget shortfalls**
- **Risk: randomness with knowable probabilities**
  - **Performance of sub-contractor**
  - **Meeting necessary schedules and/or performance measures**
- **Variation: knowable variables and variance range**
  - **Availability of critical test/demo facility/personnel**
  - **Performance differences in multiple COTS-sources**
- **Evolution: successive external developments**
  - **Change in targeted operating environment**
  - **Availability of superior technology matures**

Sustaining Agility Requires …

- Proactive awareness of situations needing responses
- Effective options appropriate for responses
- Assembly of timely responses

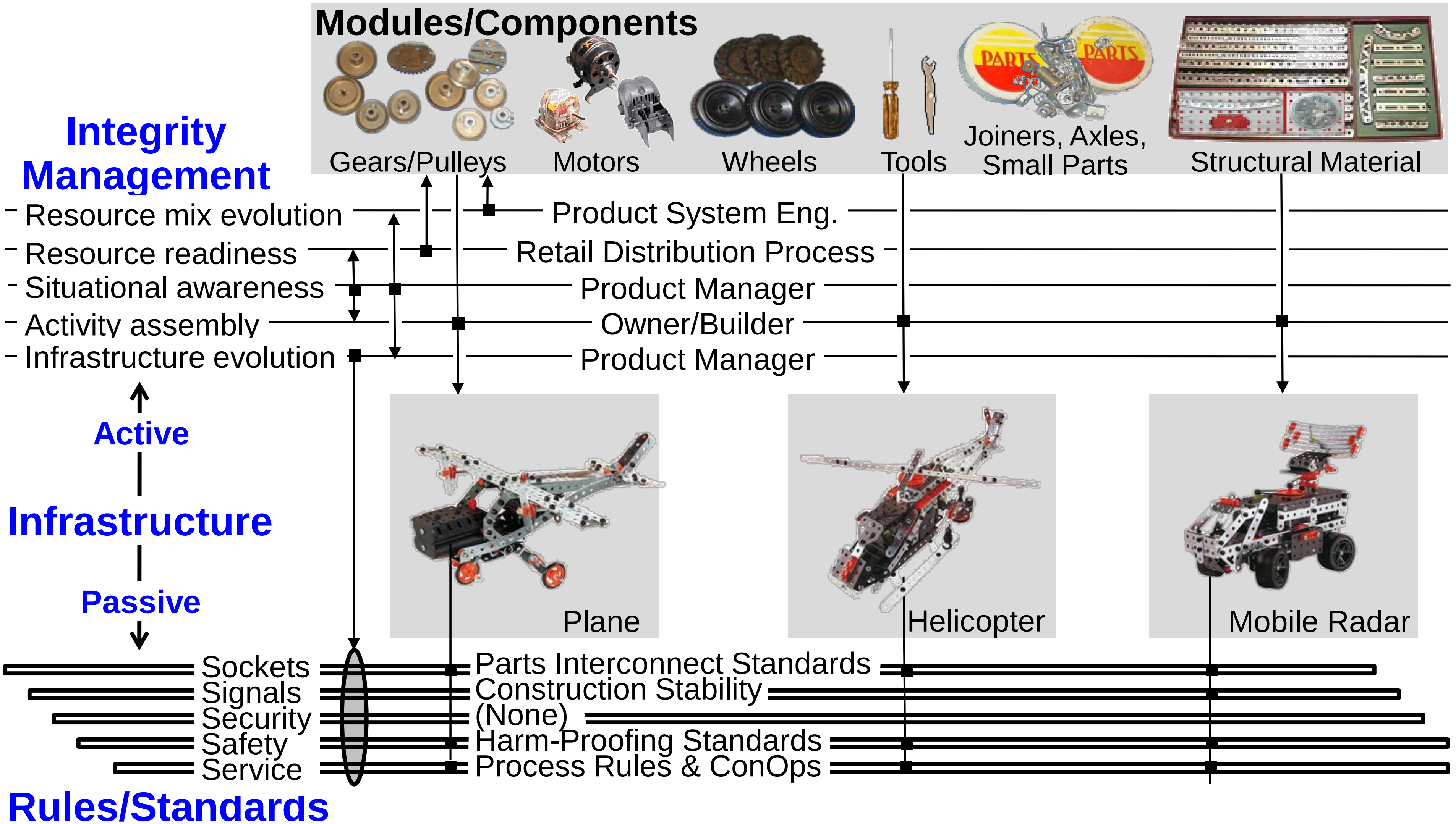
Five Agility-Sustaining Responsibilities:

1. Resource Mix Evolution – Who (or what process) is responsible for capabilities of resources appropriate for needs?
2. Resource Readiness – Who (or what process) is responsible for conditions of resources deployable rapidly?
3. Situational Awareness: Who (or what process) is responsible for monitoring, evaluating, and anticipating the operational environment?
4. Activity Assembly – Who (or what process) is responsible for assembling new response configurations as situations require?
5. Infrastructure Evolution – Who (or what process) is responsible for evolving the passive and active infrastructures?

Agile-System Architecture Pattern (AAP)

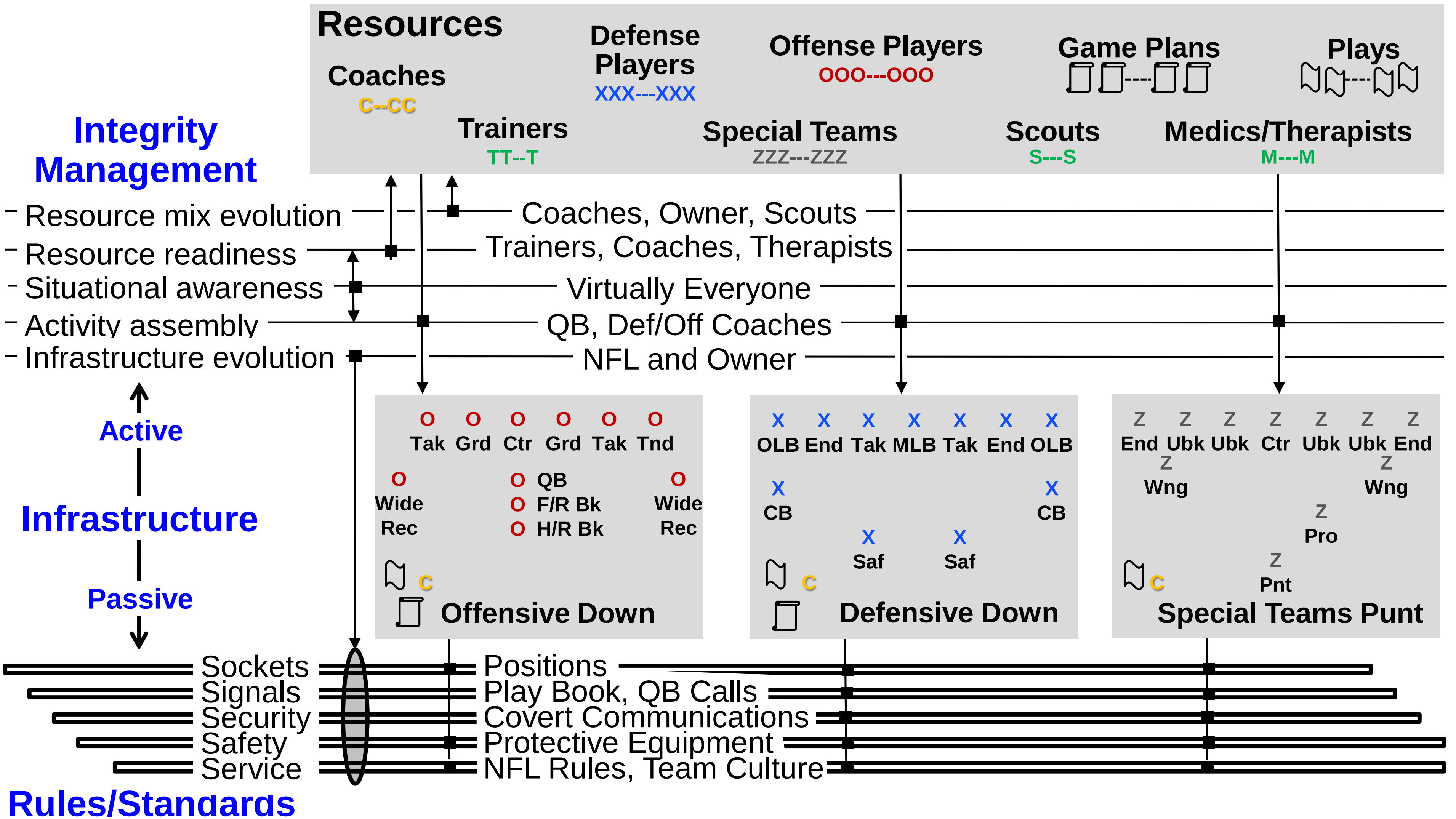
System Response-Construction Kit

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



Agile-Process AAP for USA Football

Drag-and-drop resources in a plug-and-play infrastructure



(a concept example, not exhaustive)

Agility-Enabling Design Principles

Prior Work: see INCOSE Webinar, www.parshift.com/s/AgileSystems-103.pdf

Reusable

- **Encapsulated resources (loosely coupled black-box units)**
- **Facilitated interfacing (easy resource insertion/removal)**
- **Facilitated re-use (support for finding/deploying appropriate resources)**

Reconfigurable

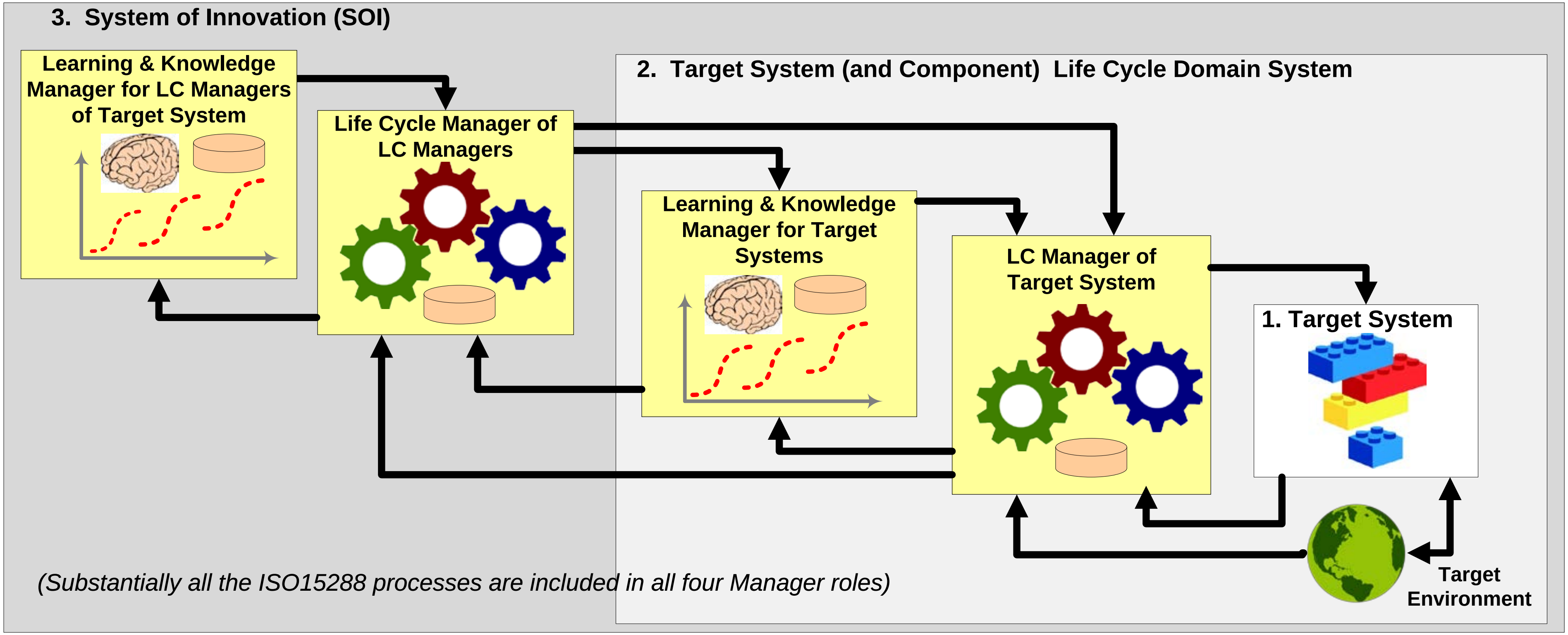
- **Peer-peer interaction (direct communication w/o intermediaries)**
- **Deferred commitment (decisions & fixed bindings at last-responsible-moment)**
- **Distributed control and information (decisions at point of maximum knowledge)**
- **Self organization (relationships and interactions negotiable)**

Scalable

- **Evolving infrastructure standards (resource interface and interaction change)**
- **Redundancy and diversity (duplicate and diverse resource populations)**
- **Elastic capacity (resource populations and functional capacity is variable)**

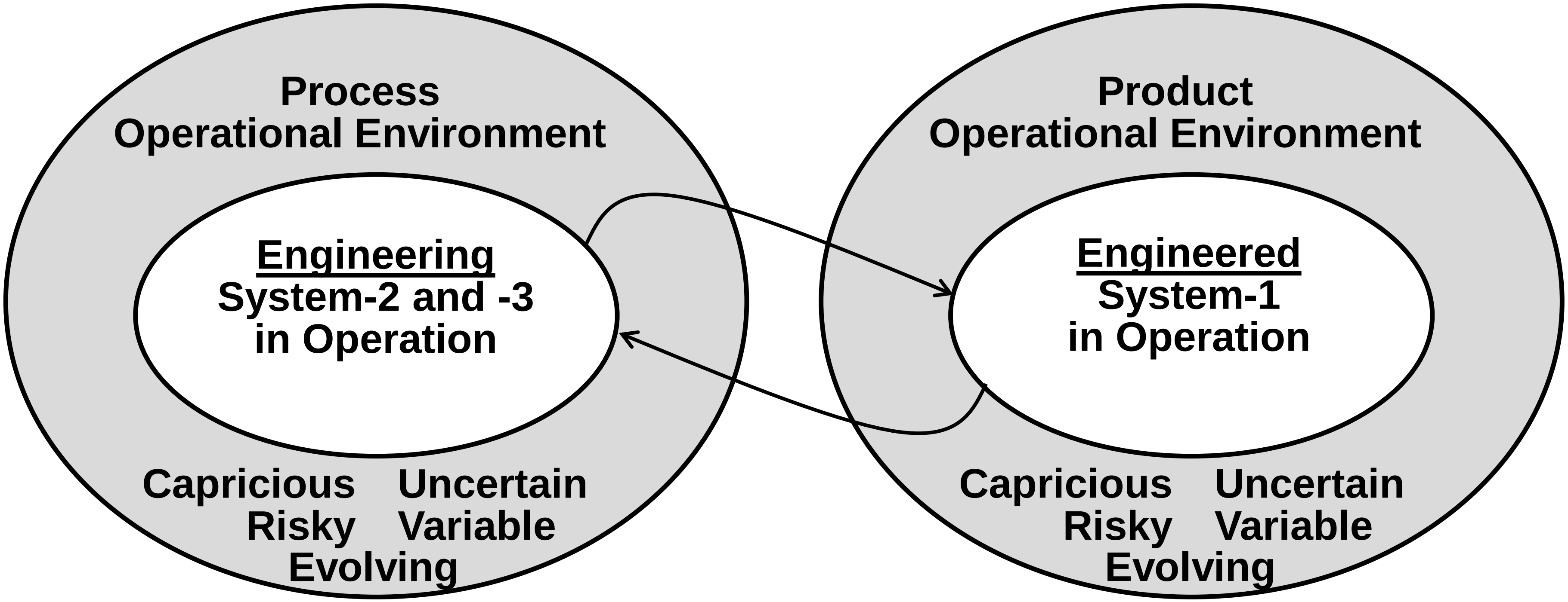
Agile Systems Engineering Life Cycle Pattern

Encompassing Systems 1, 2, and 3



- **System-1 is the target system under development.**
- **System-2 includes the basic systems engineering development and maintenance processes, and their operational domain that produces System-1.**
- **System-3 is the process improvement system, called the system of innovation that learns, configures, and matures System-2.**

Two different operational environments defining necessary agile counterpoint for the systems they encompass



It is counterproductive to have an agile development process if you don't have an agile product architecture

CURVE Environment

multi-customer autonomous off-road-vehicle robotic military technology

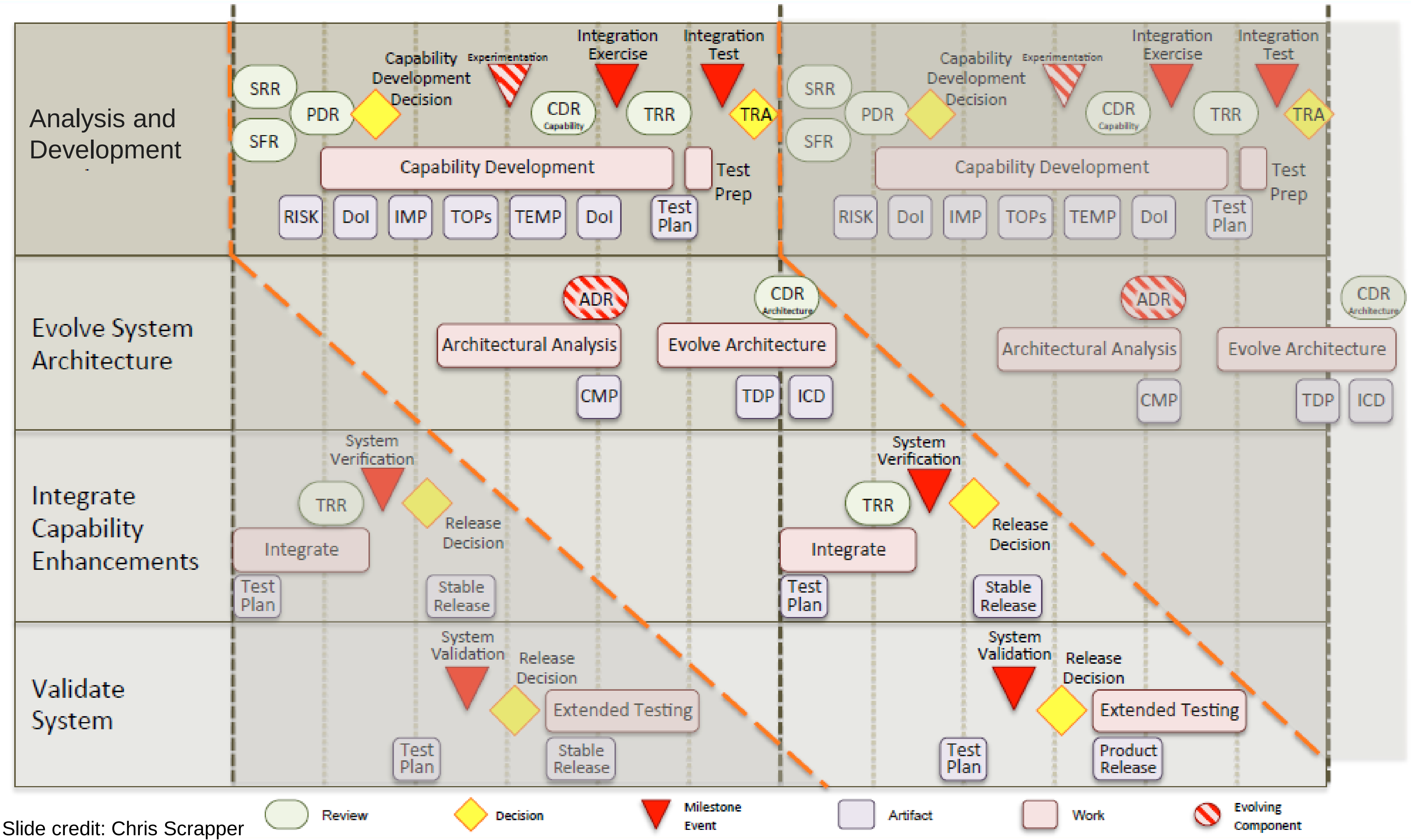
- **Caprice:**
  - **Strategic realignment by sponsor**
  - **Engagement and/or availability of personnel & contractors**
- **Uncertainty:**
  - **Feasibility of technical approach and initial designs**
  - **Contracting issues, funding gaps, and budget short falls**
- **Risk:**
  - **Failure to meet technical performance measures**
  - **Maturation and integration of required component technologies**
- **Variation:**
  - **Availability of test ranges and test support, and obtaining approvals**
  - **RAM* of vehicle test-beds (vehicle, sensors, computing HW, cables…)**
- **Evolution:**
  - **Technical landscape and insertion of emerging technology**
  - **Programmatic objectives and stakeholder's scope creep**

*RAM: Reliability, Availability, Maintainability

CDR: Critical Design Review
DoI: Declaration of Intent
PDR: Preliminary Design Review
SDR: System Design Review
SFR: System Functional Review
SRR: System Requirements Review
TEMP: Test and Experimentation Master Plan
TOP: Test Operating Procedures
TRR: Test Readiness Review

Integrated Strategy Chart

SPAWAR Systems Center PACIFIC



Slide credit: Chris Scrapper

Multi-Project SE Process

for evolving autonomous off-road-vehicle robotic military technology

SPAWAR Systems Center PACIFIC

SE-Process Reusable/Reconfigurable Resources

- (IL) Integration Leads
- (FL) Functional Leads
- (TL) Technical Leads
- (CP) Contract Performers
- (WF) Users (War Fighters)
- (RC) Reusable Components
- (CD) CIE Data
- (TM) Test Methods

Integrity Management

- Resource mix evolution — PM+CIT.
- Resource readiness — PM+CIT (Core Integration Team)
- Situational awareness — PM+CIT+Leads
- Activity assembly — Leads
- Infrastructure evolution — PM (Process Manager)

Active Facilitating

Infrastructure

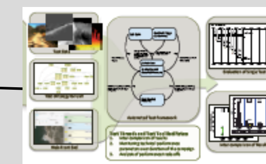
Passive Enabling

RaDER Integration: IL, FL, TL, CP, RC

EV1 Integration: IL, FL, TL, CP, RC

IPT Working-Group: IL, FL, TL, CD, CP, TM

Validation Testing: TM, IL, FL, TL, CP, WF



Sockets
Signals
Security
Safety
Service

Rules/Standards

Sockets: CIE, System-1 modular architecture, roles, culture, test threads
Signals: Vision, Declarations of Intent, Config Mgmnt Plan, Integration Strategy, CIE data, decisions, engaged team feedback
Security: User agreement/NDA, Config Mgmnt Plan, CIE access controls
Safety: Open-process visibility, open communication, protected communication
Service (SE ConOps): Vision, Culture, Consciousness(CIE), Conscience, Wave, Integration Strategy/TEMP, Sys-1 and Sys-2 AAP

Navy SpaWar System Center Pacific

Wave process with explicit product-line engineering for innovative HW/SW unmanned-vehicle technology development.

Some Notables:

- **Six-month delivery increments.**
- **Product line approach shares components across projects/sponsors.**
- **Wave approach decouples development from integration & test cycles.**
- **OSA agile-product infrastructure.**
- **Instrumented pre-integration testing of work-in-process.**
- **Suppliers do technology development, but…**
  - **Navy owns the architecture, in collaboration with suppliers.**
  - **Navy owns integration and test, with assistance of suppliers.**
  - **Navy cost outweighed by lowering project and product costs.**
- **Warfighter (user) workshops for requirements-reality correction.**
- **On-line, daily-updated, filtered status visibility (Home-grown CIE).**
  CIE: Continuous Integration Environment

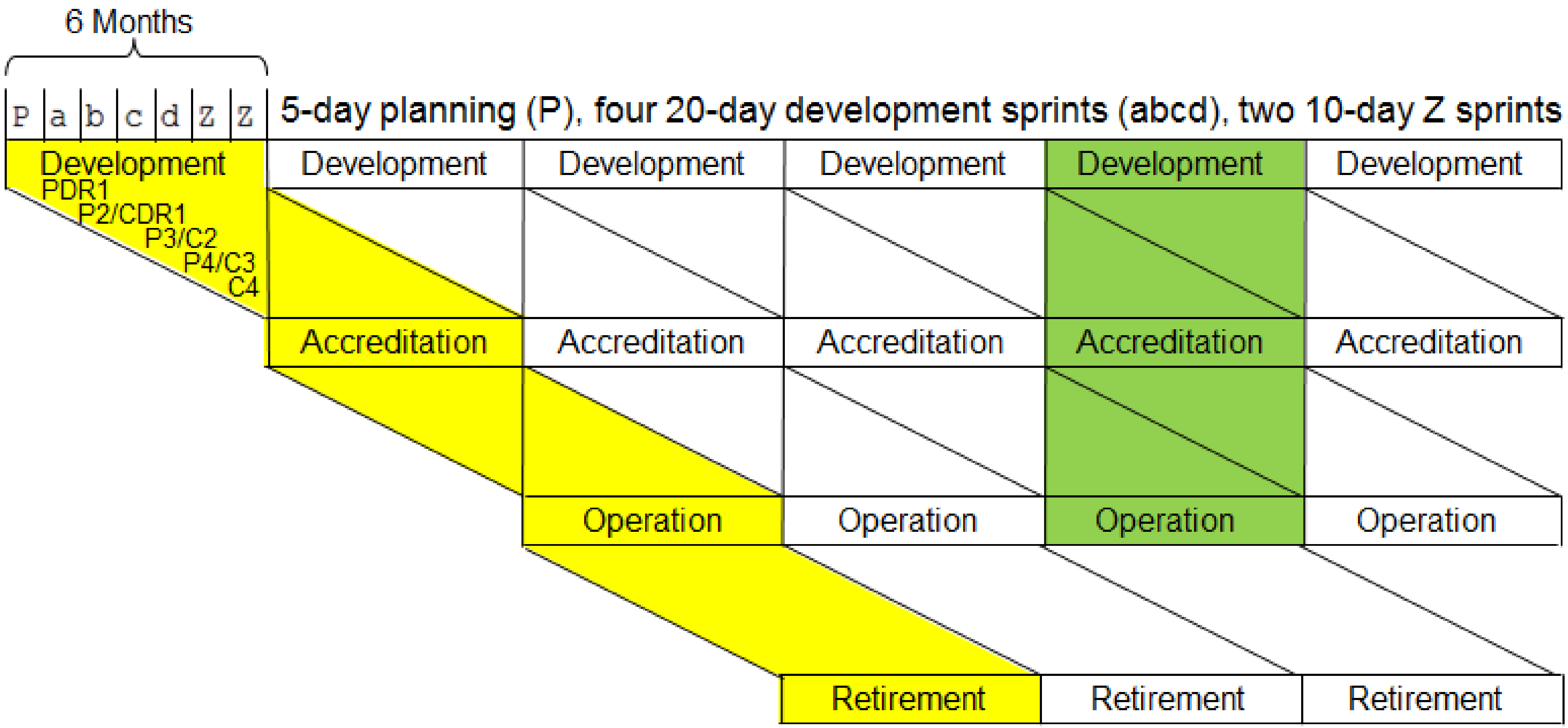
CURVE Environment

SoS web-hub: 12 customer groups accessing 22 independent data bases

- **Caprice:**
  - **External data sources change their services**
  - **Number of security vulnerabilities to address varies greatly weekly**
- **Uncertainty:**
  - **Software or Hardware may go end-of-life at any point**
- **Risk:**
  - **May not be able to meet 15-day schedule for delivery of security fixes**
- **Variation:**
  - **COTS upgrades deprecate existing interfaces**
- **Evolution:**
  - **The program must port existing capability to new technology**

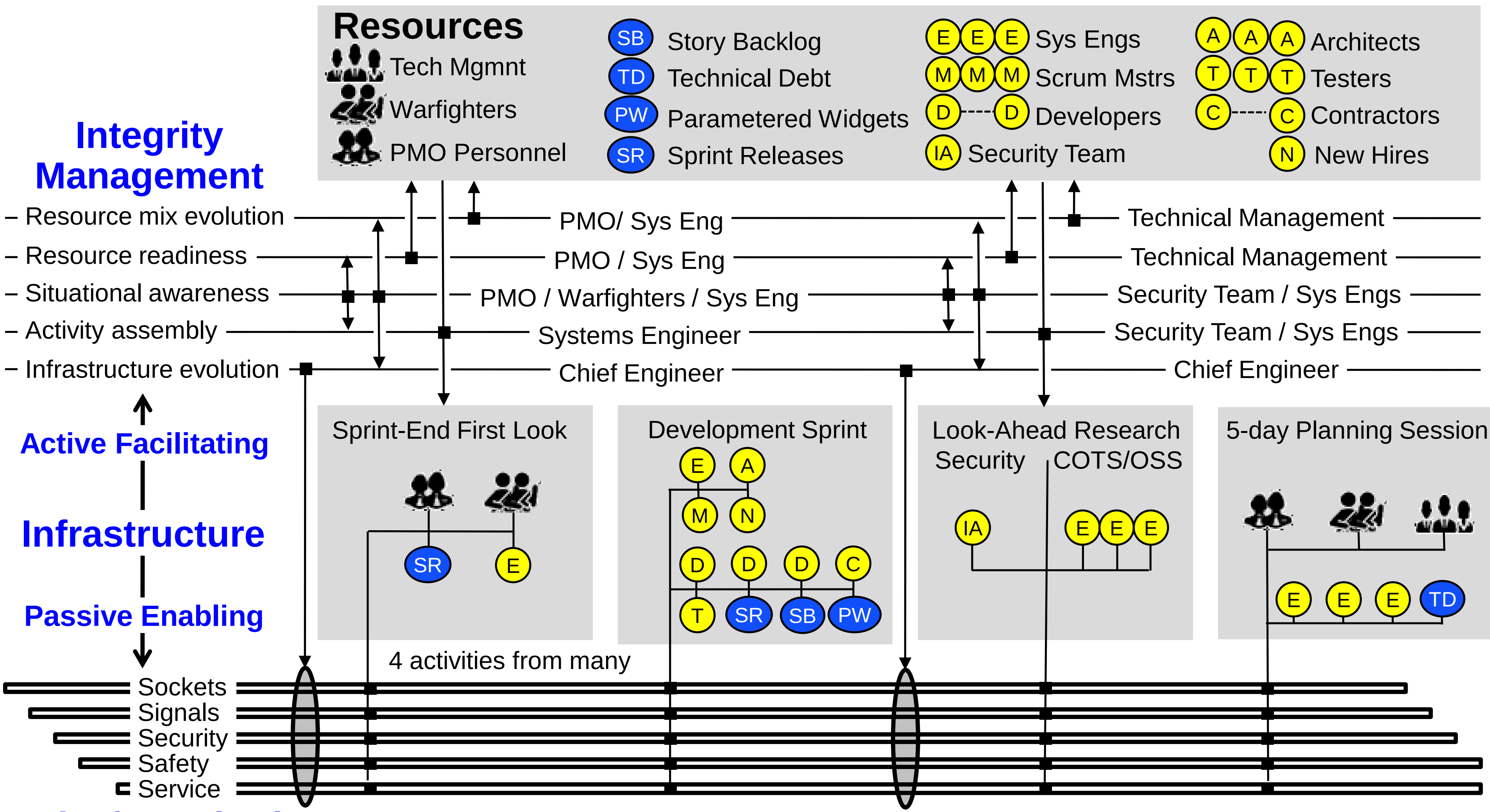
Content: Mark Kenny, Northrop Grumman

Scrum-Based Software Development Process in Decoupled Wave-Like Waterfall



SoS Web-Portal Evolution Process

NORTHROP GRUMMAN



Rules/Standards

Sockets: Meeting formats, Sys-1 modular architecture, Automated build environment, User story acceptance criteria, Roles, Culture
Signals: Vision/Intent, Release themes, Spikes, User stories, Wireframes, Code, SCR, Process status/metrics, Deliverables, Behavior
Security: Governance, Leadership, Cultural oversight, QA, Metrics, CMMI level 5 oversight, Configuration management
Safety: Open-process visibility, Open no-penalty communication, On-boarding, Team user-story estimation, 40-hour work load
Service Documented accessible ConOps, Embedded environment awareness, Continuous DevOps integration, AAP for Systems 1&2

Northrop Grumman

Wave/Scrum-like software process leveraging product-line concept for SoS web-hub access to 22 independently-evolving data bases.

Some Notables:

- **Six month delivery increments.**
- **Product line approach parameterizes components for multiple uses.**
- **Wave approach decouples development from accreditation & operation.**
- **Look ahead for likely high-priority security bulletins.**
- **Look ahead for pending COTS/OSS obsolescence (1000+ components).**
- **SoS re-stabilization after unannounced independent system changes**
- **Customer establishes/re-prioritizes sprint tasks constantly (principally for surprise security issues).**
- **Customer & user first-look testing of sprint results.**
- **On-line, daily-updated, filtered status visibility:
Rally (progress status), Jenkins (build/deploy pipeline),
internal wiki (program & lessons learned) .**

CURVE Environment

Rockwell Collins

product-line engineering for HW/FW/SW military radios

Caprice

- Markets have long/volatile acquisition cycles

Uncertainty

- Subjective feature requirements, not clearly defined
- Ever-moving competitive landscape
- Unknown and Emerging Stakeholders/Users/CONOPs

Risk

- Firmware/Hardware architecture may not be adaptable for future requirements
- Customer expectations exceed technology envelope
- Significant investment with no guarantee for return

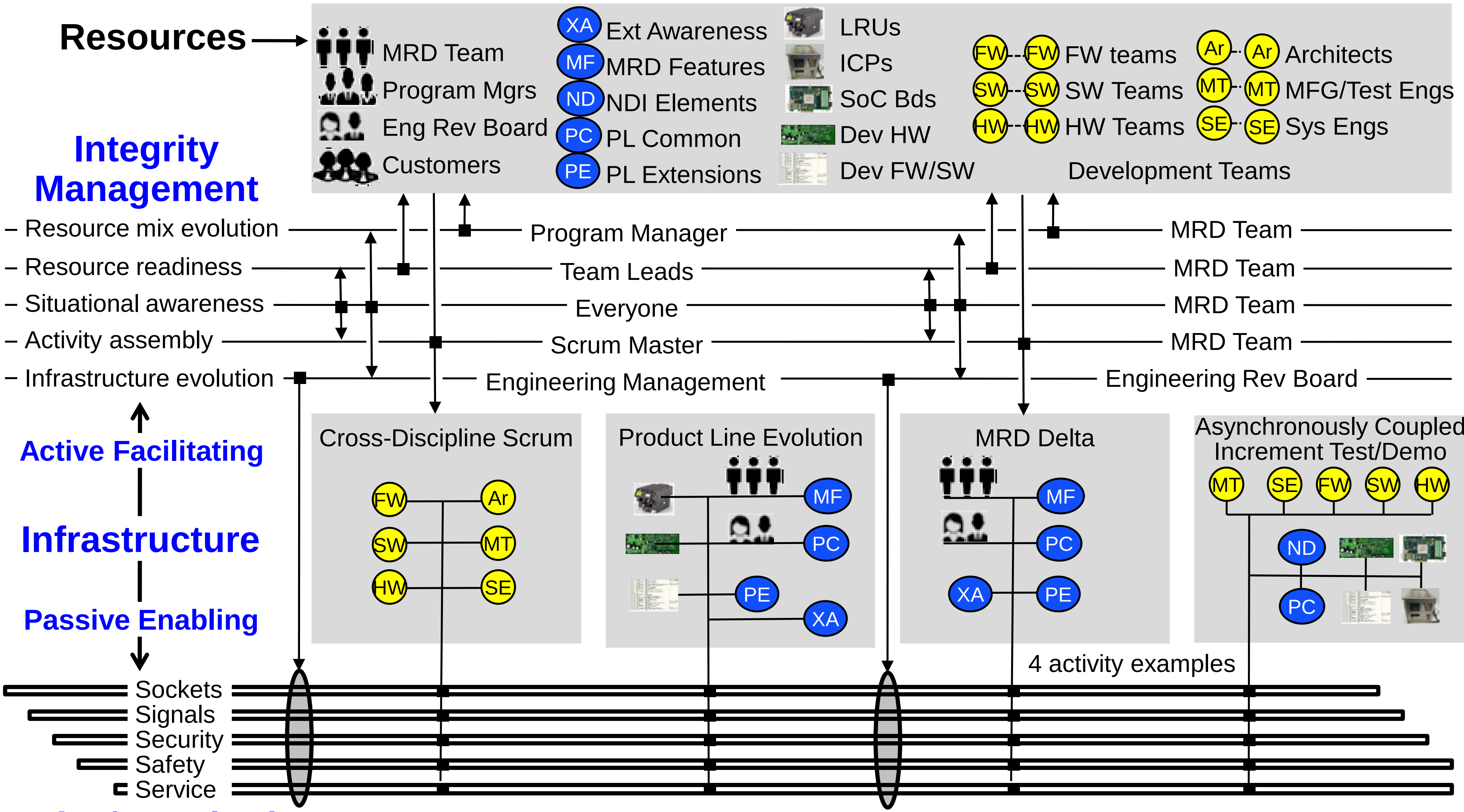
Variation

- Market-Based approach tied to evolving industry needs

Evolution

- Customer expectations and functionality

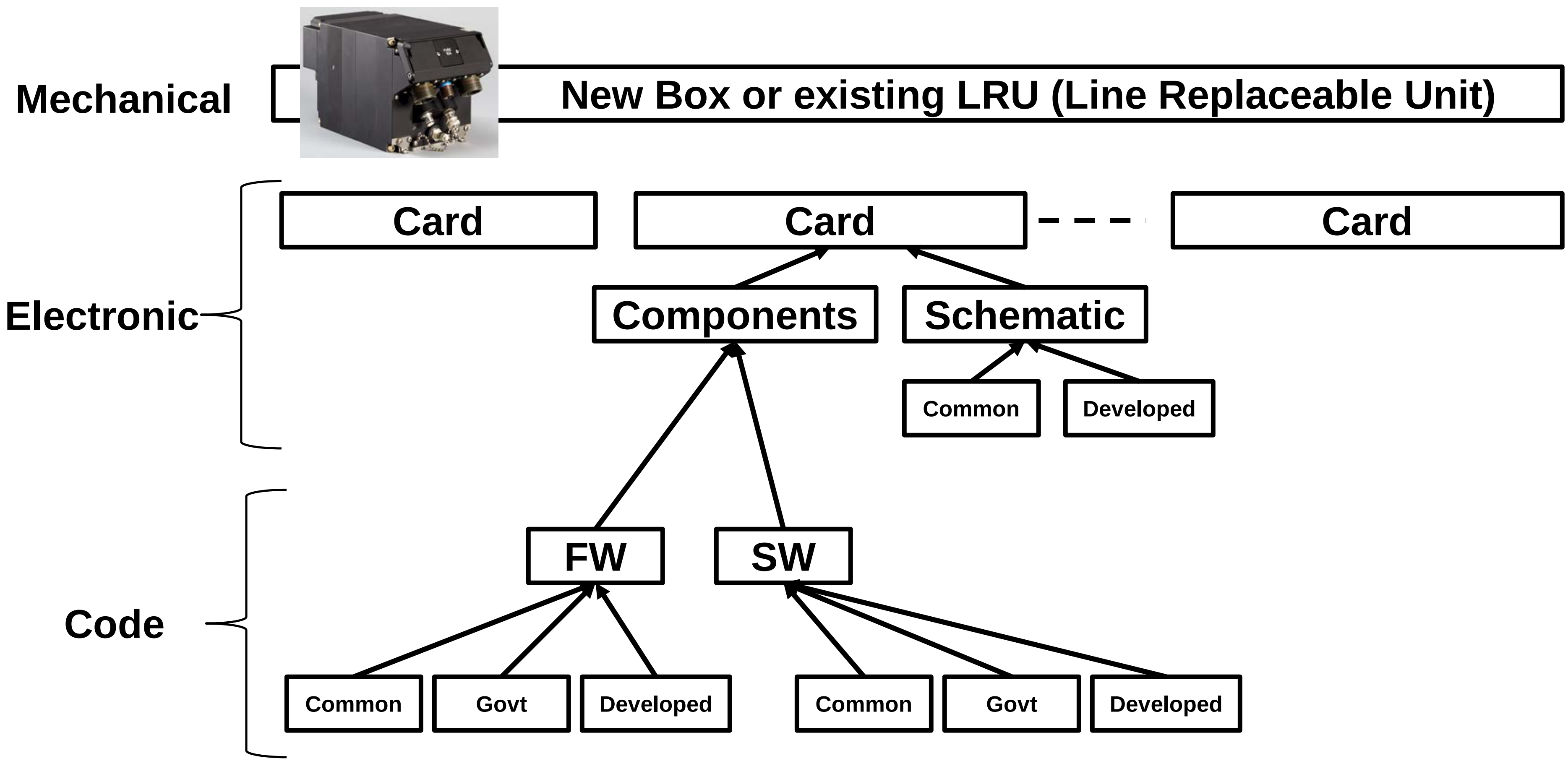
Rockwell Collins System 2 Product-Line AAP



Rules/Standards

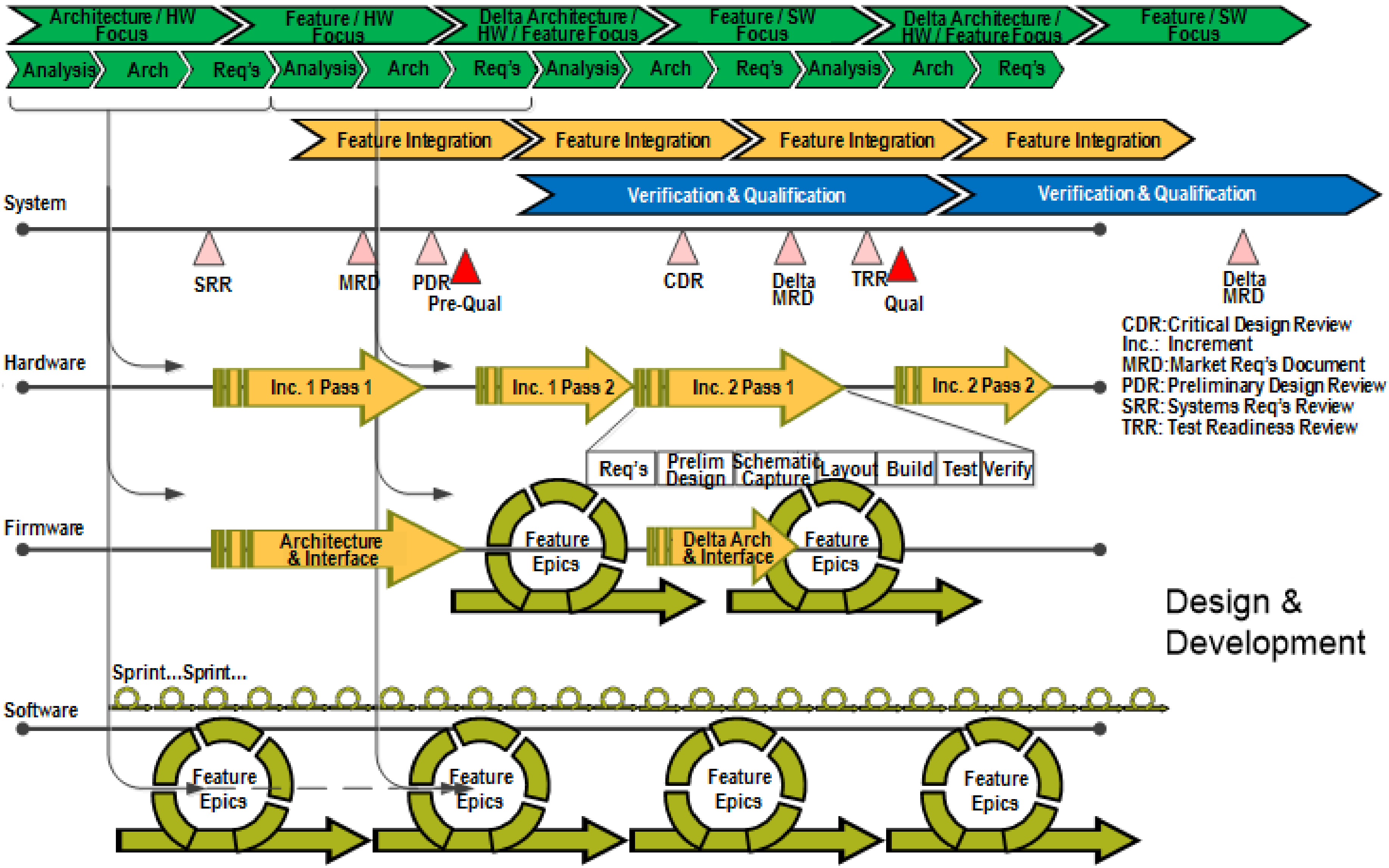
Sockets: PL component-interface standards, Scrums, Collaboration space
Signals: MRD, Epics, Stories, Specifications, Requirements, IMS, JIRA issues, Confluence data
Security: Program reviews, Retrospectives, Scrum ceremonies
Safety: Training, Scrum Ceremonies
Service: RC Agile process ConOps, Market requirements document, Confluence, HW development platforms

Multi-Discipline Engineering



Asynchronous Mixed-Discipline Increment Alignment

Rockwell Collins



Rockwell Collins

SAFe/Scrum-like process explicitly based on product-line engineering for domestic and international HW/FW/SW military radios.

Some Notables:

- **Agility-enabling concept as Product Line architecture and strategy.**
- **Agility-facilitating concept as active SE management of all relationships.**
- **Asynchronously-aligned increments across HW/FW/SW development.**
- **Active external awareness evolving the Product Line Market Requirements Document.**
- **Agile hardware-development platform infrastructure.**
- **Active opportunity management (as part of risk management).**
- **On-line, daily-updated, filtered status visibility (Confluence).**

CURVE Environment

Lockheed
IFG Avionics

Evolving HW/SW aircraft weapon system capability

Capriciousness:

- Urgent Operational Needs
- Diminishing Manufacturing Sources

Uncertainty:

- Funding (e.g. Sequestration)
- Solution Feasibility
- Regression Impacts

Risk:

- Competition Losses
- Attract/Keep Talent
- Systems Of Systems Requirements Changes
- Schedule/External Stakeholder Timelines (e.g. Certification)

Variation:

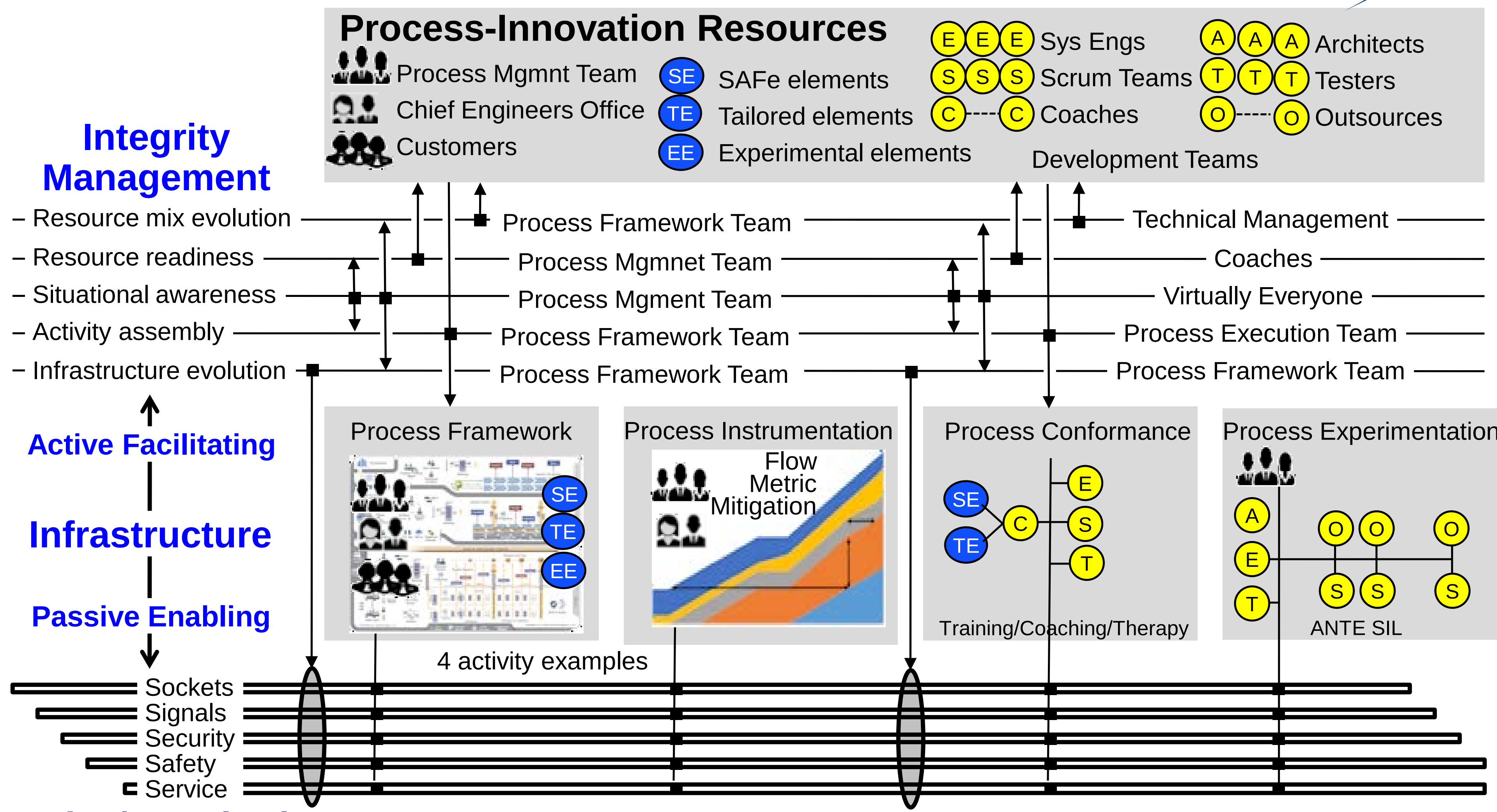
- Projects Competing For Bottlenecks (e.g. Ground/Flight Test)
- System Of Systems Integration

Evolution:

- Planned Modernization/Sustainment Increments
- Open Mission Systems Evolution

Agile-Transition System 3

Lockheed
IFG Avionics



Rules/Standards

Sockets: Process framework, Roles, Teams, Meeting formats, ANTE/Simulation frameworks
Signals: Flow, Info debt, Process conformance, Experiment results, Contract performance
Security: Executive commitment, Governance, Cultural consistency
Safety: Information radiators, No-penalty measurement, Flow monitoring/mitigation, Real-time status information, 2-3 PI look-ahead
Service (ConOps): Process framework, Cadence, Customer/User involvement, Optimal-process control, System 1-2-3 AAP

Lockheed Integrated Fighter Group

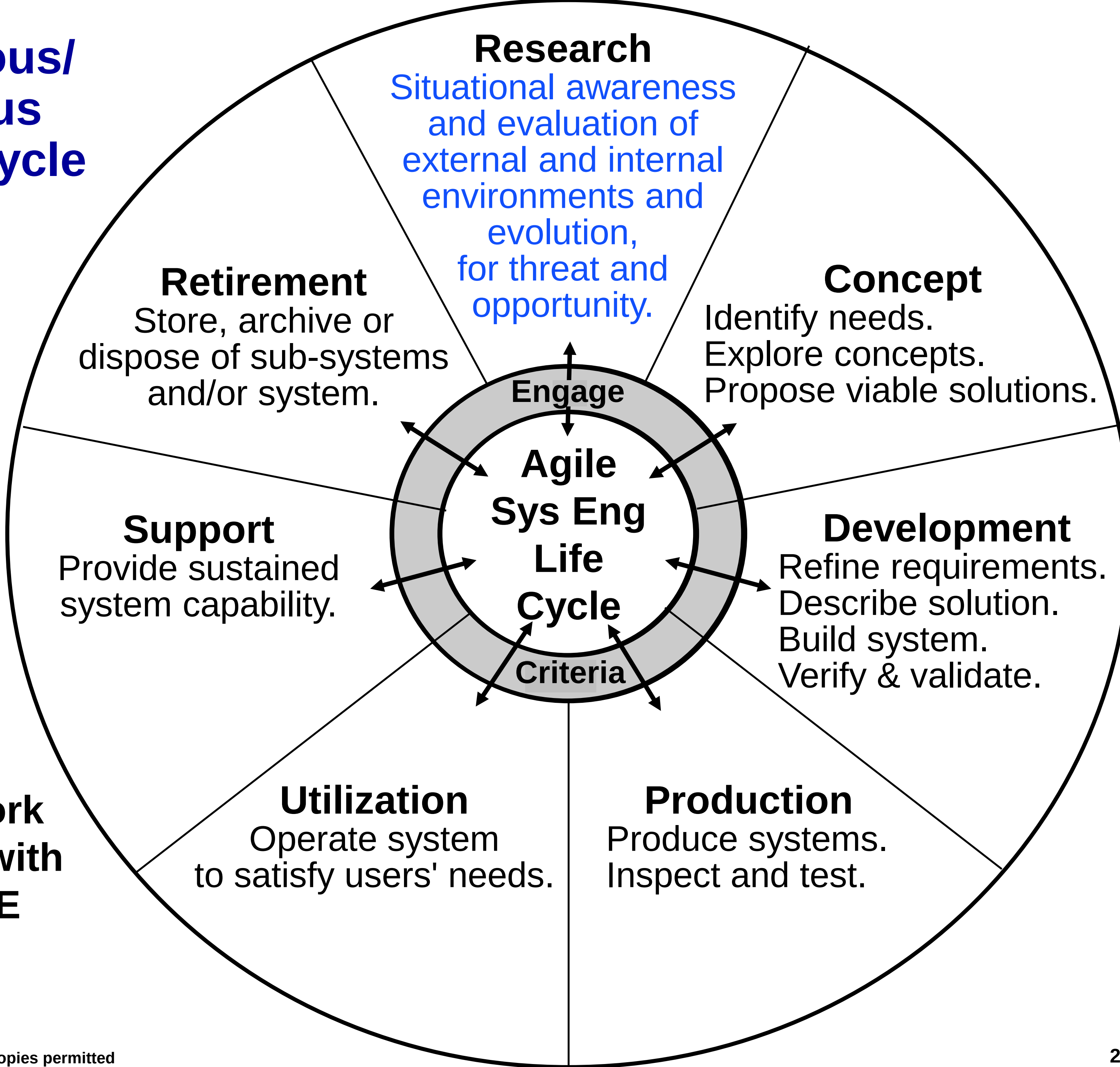
Lockheed
IFG Avionics

Tailored SAFe-like process with explicit product-line engineering for evolving HW/SW aircraft weapon system capability.

Some Notables:

- **System 3 focus: Process learning and evolution.**
- **Controlled proactive process experimentation.**
- **1200 people trained: executives, managers, engineers.**
- **Process instrumentation for optimal resource & task-allocation control.**
- **Preliminary SIL with low-fidelity COTS devices and evolving device simulations.**
- **Information-debt recognition: Documentation for depot service.**
- **Aircraft OSA agile-product infrastructure.**
- **Product Line cross-project component reuse.**
- **On-line, daily-updated, filtered status visibility (VersionOne).**

Asynchronous/ Simultaneous Agile Life-Cycle Framework



Observed in all workshops to date

This framework is consistent with ISO/IEC/IEEE standards

Emerging Fundamental Principles

All case studies enable and facilitate (in core, but different methods):

- **Project situational sensing and response.**
- **Team-members' engagement sensing and response.**
- **Development-issue sensing and response.**
- **Integration-issue sensing and response.**
- **Assimilated shared-culture and evolution.**
- **Process and procedure evolution.**
- **Product evolution.**

Three Categories of Fundamental Principles Emerge:

- **Sense/Monitor – awareness is the driver of agility**
- **Respond/Mitigate – action is the expression of agility**
- **Evolve – applied learning is the sustainer of agility**

Agility-Facilitating Operational Principles
Current work, based on analytical workshops in process

Monitoring (observe, orient)
- **External awareness (proactive alertness)**
- **Internal awareness (proactive alertness)**
- **Sense making (risk & opportunity analysis, trade space analysis)**

Mitigating (decide, act)
- **Decision making (timely, informed)**
- **Action making (invoke/configure process activity to address the situation)**
- **Action evaluation (V&V)**

Evolving (improve above with more knowledge and better capability)
- **Experimentation (variations on process ConOps)**
- **Evaluation (internal and external judgement)**
- **Memory (evolving process ConOps)**

In Summary

Initial Generic Findings

- Agility-facilitating operational principles discovered: MME
- AAP product architecture enables AAP SE process: confirmed
- Asynchronous simultaneous agile life-cycle framework: confirmed
- Addition of Research life-cycle stage: awareness drives agility
- Stage-engagement criteria: partial – engagement is Mitigation triggered by Monitoring

Methods Observed

- CURVE-driven decision for employing an agile approach
- Incremental and iterative development
- Asynchronous mixed-discipline increment alignment
- Product Line (equivalent) reusable components
- Decoupled Wave approach
- Proactive process experimentation and learning
- Managed, monitored, and enforced culture of engagement
- Supplier process-engagement
- Distributed/ubiquitous risk assessment and management
- Preliminary integration testing and instrumentation
- Active customer and user involvement
- SCRUM-like software development management
- Integrated active systems engineering involvement

Tools

- On-line, custom-filtered, daily status visibility (Confluence, VersionOne, …)
- Fixit tracking (Jira, …)
- Process performance-management instrumentation (VersionOne Flow, …)
- Development infrastructures (OO SW platform, PL library, FW/HW prototyping tools, …)
- Integration test infrastructures (preliminary SIL, WIP simulations, …)
- User incremental testing/feedback infrastructure (First Look, user workshops, …)

Epilog: Agile Hardware-Development Infrastructures

Product Agile Architecture Pattern (AAP) enables SE Process Agility.

Agile software development processes (silently) rely on product AAP.

- **Program code development employs an object-oriented AAP development platform (e.g., C++, Java, Eclipse).**
- **Web code development employs a loosely-coupled modular AAP inherent with hyperlinked web-pages.**

Agile hardware development doesn’t have off-the-shelf AAP tools.

- **Proprietary product-line-engineering employs AAP.**
- **Proprietary Open System Architecture (OSA) employs AAP.**
- **Proprietary Live-Virtual-Constructive platforms employ AAP.**

Infrastructure-Enabled Agile Hardware-System Development

RC electronic prototyping infrastructure tools
RC Product line component library (devices, boxes, electronics, software)
www.parshift.com/s/ASELCM-02RC.pdf

SSC-Pac OSA infrastructure
www.parshift.com/s/ASELCM-01SSCPac.pdf

LMC low fidelity COTS for incremental system test
LMC ANTE (agile non-target environment) SIL
www.parshift.com/s/ASELCM-04LMC.pdf

L3 aircraft HVACP infrastructure – great design example
L3 SIL – duplicates aircraft installation environment
www.parshift.com\AgileSysAndEnt\Cases\Case Agile Aircraft Installation Architecture.pdf

Team WikiSpeed CAD machining, mechanical stubbing, composite production
Many videos by Joe Justice: Google search: "Youtube Joe Justice"

Applied Materials semiconductor-fabrication-machine design
Book: Response Ability – the Language, Structure, and Culture of the Agile Enterprise

Note – Emphasis above is on hardware-system development, not hardware-item development.

3D Printing – Already Useful

http://asc.army.mil/web/news-alt-amj17-rambos-premiere/

Six month collaborative effort + four approaches = functional results.

3-D printed ammo fired out of a 3-D printed grenade launcher.

The first approach was to print the projectile body in aluminum as an alternative material. The problem with that approach is that aluminum is less dense than zinc; therefore, when fired, the projectile achieves higher speeds than system design specifications call for.Even though the barrel and projectile body were printed from the same aluminum material, because the printed barrel was hard-coat anodized, it allowed for proper rifling engagement with the softer untreated printed aluminum projectile body.

The second approach was to print the projectile body in steel, which better meets the weight requirements, and then mold a urethane obturating ring onto it. The obturating ring is required to ensure proper engagement and rifling in the aluminum barrel. We couldn't keep the obturating ring as steel, like we did with the first approach, because steel is a lot harder than aluminum, and even with the hard-coat anodization it would have destroyed the grenade launcher's barrel. So for this approach, the projectile body's design was modified to take advantage of design for AM. The original projectile body designs did not consider AM fabrication and processing. For this AM technology demonstrator, the design was modified to take advantage of AM design rules to reduce the amount of post-machining required. This approach also used 3-D printing to fabricate a "negative" mold and then create a silicone positive mold to produce an obturating ring onto the printed munition bodies.

The third approach also utilized a groove and obturating ring, but instead of overmolding, the plastic was printed directly onto the steel projectile body using a printer with a rotary axis.

The fourth approach used a wax printer to 3D-print projectile bodies. Using the lost-wax casting process, plaster was poured around the wax bodies and allowed to set. Once set, the hardened plaster mold was heated and the wax melted away. Molten zinc was then poured into the plaster mold to cast the zinc projectile bodies.

References

Dove, R., R. Labarge. 2014. Fundamentals of Agile Systems Engineering – Part 1 and Part 2. International Council on Systems Engineering, International Symposium, Las Vegas, NV, 30Jun-3Jul. www.parshift.com/s/140630IS14-AgileSystemsEngineering-Part1&2.pdf

Dove, R., W. Schindel. 2016. Agile Systems Engineering Process Features Collective Culture, Consciousness, and Conscience at SSC Pacific Unmanned Systems Group. INCOSE International Symposium (IS 2016), Edinburgh, Scotland, UK, 18-21 July. www.parshift.com/s/ASELCM-01SSCPac.pdf

Schindel W. and R. Dove. 2016. Introduction to the Agile Systems Engineering Life Cycle MBSE Pattern. INCOSE International Symposium (IS 2016), Edinburgh, Scotland, UK, 18-21 July. www.parshift.com/s/160718IS16-IntroductionToTheAgileSystemsEngineeringLifeCycleMBSEPattern.pdf

Scrapper, C., R. Halterman, J. Dahmann. 2016. An Implementers View of the Evolutionary Systems Engineering for Autonomous Unmanned Systems. IEEE SysCon, Orlando FL, 18-21 April.

Dove, R., W. Schindel. R. Hartney. 2017. Case Study: Agile Hardware/Firmware/Software Product Line Engineering at Rockwell Collins. IEEE SysCon, Montreal, Quebec, CA, 24-27 April. www.parshift.com/s/ASELCM-02RC.pdf

Dove, R., W. Schindel. 2017. Case Study: Agile SE Process for Centralized SoS Sustainment at Northrop Grumman. INCOSE International Symposium (IS 2017), Adelaide, Australia, 15-20 July. www.parshift.com/s/ASELCM-03NGC.pdf

Dove, R., W. Schindel. 2017. Case Study: Transition to Scaled-Agile Systems Engineering at Lockheed Martin's Integrated Fighter Group. Unpublished working paper. www.parshift.com/s/ASELCM-04LMC.pdf

INCOSE Webinars:

Agile 101: Architecture Pattern,
www.parshift.com/s/AgileSystems-101.pdf, www.parshift.com/s/AgileSystems-101.wmv

Agile 102: Design Requirements,
www.parshift.com/s/AgileSystems-102.pdf, www.parshift.com/s/AgileSystems-102.swf

Agile 103: Design Principles,
www.parshift.com/s/AgileSystems-103.pdf, www.parshift.com/s/AgileSystems-103.mp4

Agile 104: Engagement Quality,
www.parshift.com/s/AgileSystems-104.pdf, www.parshift.com/s/AgileSystems-104.mp4

Agile 105: Operational Awareness,
www.parshift.com/s/AgileSystems-105.pdf, www.parshift.com/s/AgileSystems-105.mp4

ASELCM project and workshop Host information/details:
www.parshift.com/ASELCM/Home.html

fini

Download this file (for a few days only):
www.parshift.com/s/170314.pdf