Airborne Net-Centric Supporting Architectures

John Richardson – PMA-290, FoS Prototyping Chief Engineer, VP Savvee Consulting Inc.
Dr. Gordon Long – Chief Engineer, Naval Aviation Systems, The MITRE Corp

"No modern war has been won without air superiority, and no future war will be won without air, space and cyberspace superiority."

...Author Unknown

Savvee Consulting, Inc.

MITRE

Net-Centric -- By Definition is…

- An abstract definition…
  - *"Participating as a part of a continuously-evolving, complex community of people, devices, information and services interconnected by a communications network to achieve optimal benefit of resources and better synchronization of events and their consequences"* …Wikipedia
- A more robust definition…
  - *"Net-Centricity" is the insertion and application of evolving information and communication technologies as it relates to human activity within any given domain and their associated cross domain activities; by utilizing a common communication infrastructure in order to achieve increased awareness, control, and response through the exploitation of information."*
- There are many terms and abbreviations used to reference Net-Centric:
  - Net-Centric Warfare: NCW (US)
  - Net-Centric Operations: NCO: (US and NATO)
  - Network Enabled Capability: NEC (UK)
  - Network Based Defense: NBD (Sweden)

Goals of Net-Centric Solutions

- *"...The goal is to increase the Net-Centricity of warfighter, business, intelligence, DoD enterprise management, and enterprise information environment management operations by enabling increased reach among the GIG users, increased richness in the information and expertise that can be applied to supporting operational decisions, increased agility in rapidly adapting information and information technology to meet changing operational needs, and increased assurance that the right information and resources to do the task will be there when and where it is required."*

  ...DoD Acquisition Guide

- The value of this vision is characterized by four main points of topic:
  - An increased ability to share information.
  - Greatly expanded sources and forms of information, and related expertise to support rapid, collaborative decision making.
  - Highly flexible, dynamic, and interoperable communications, computing, and information infrastructures that are responsive to rapidly changing operational needs.
  - Assurance and trust that the right information to accomplish assigned tasks is available when and where needed, that the information is correct, and that the infrastructure is available and protected.

"Rapidly provide the Joint Warfighter an advanced Network Centric Warfare environment able to exploit the unprecedented velocity of technological change while outpacing threats."

...Maj. Jeff Guimarin USAF

Goals of Net-Centric Solutions

- This vision encompasses much more than just Enterprise Architectures, it represents a paradigm shift in:
  - Data dissemination and exchange,
  - The establishment of new data usage and doctrine, systems architecture and interoperability, mission orientation/CONOPS, mission effectiveness models, business process management.
  - Multiple abstraction layers between applications and the underlying computing and network infrastructure and hardware.
  - Solidifies the Systems of Systems (SoS) approach into a Family of Systems (FoS) approach through the aggregation of Net-Centric Domains.

"The true success of Net-Centric military systems is the rapid application of information technology applied within the context of human activities and processes."

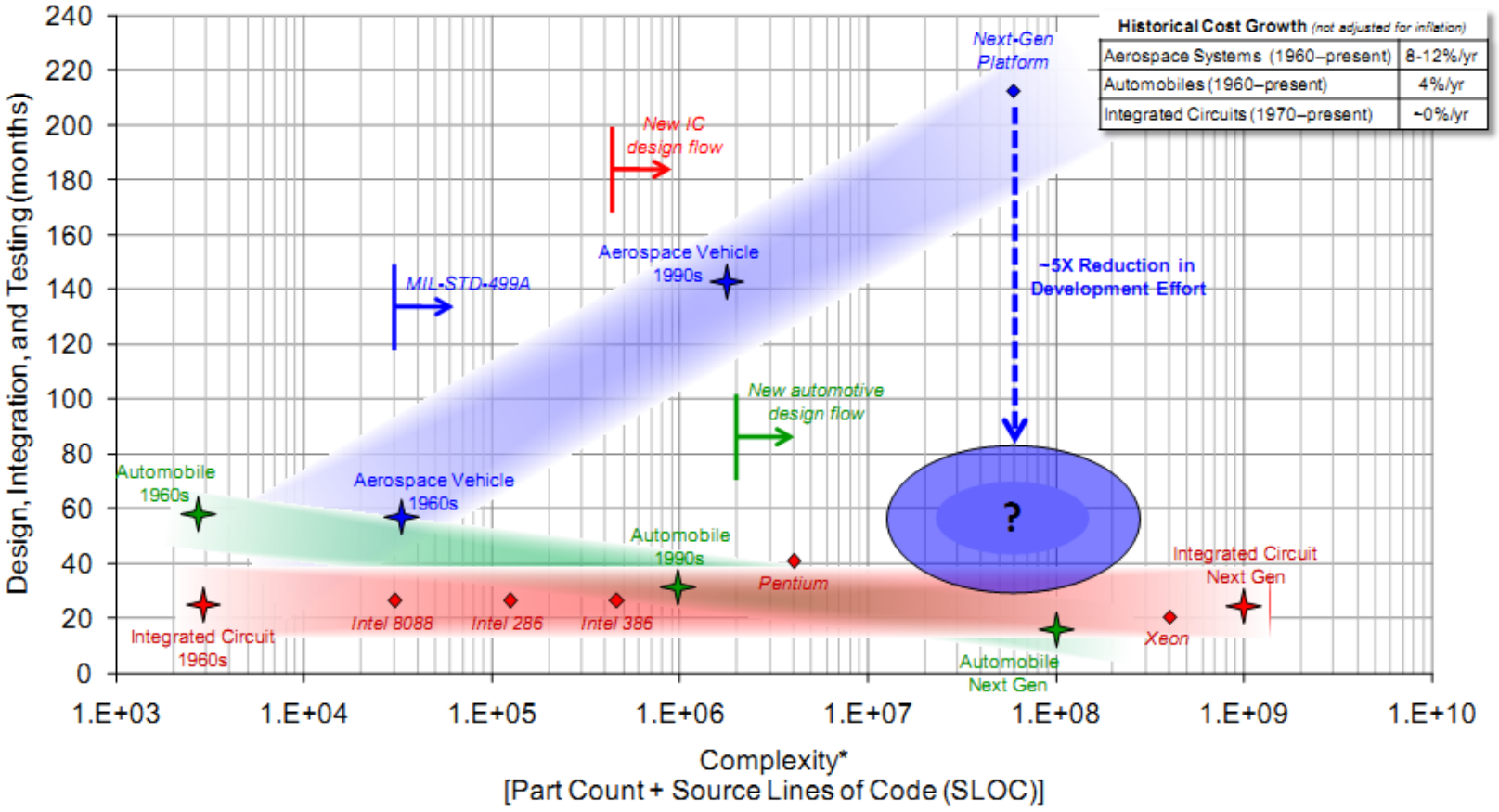
Supporting Platform Architectures

5

Goals for Platform Supporting Architectures

- Goals of Net-Centric supporting architectures
  - Provide an Open Architecture (OA) approach based on accepted commercial industry standards and best practices.
  - Provide software integration boundaries defined by abstract service boundaries.
  - Provide highly portability application environments
  - Provide increased modularization and componentization
  - Provide variable data exchange pathways
  - Protect and maintain existing system integrity
  - Provide for system capability migration
  - Provide for resource scalability (processing, memory etc.)
  - Enable data exposure and provide for managed information exchange
  - Enable business process automation utilizing commercial Business Process Management (BPM) solutions.

Accelerated Development by Reducing Complexity

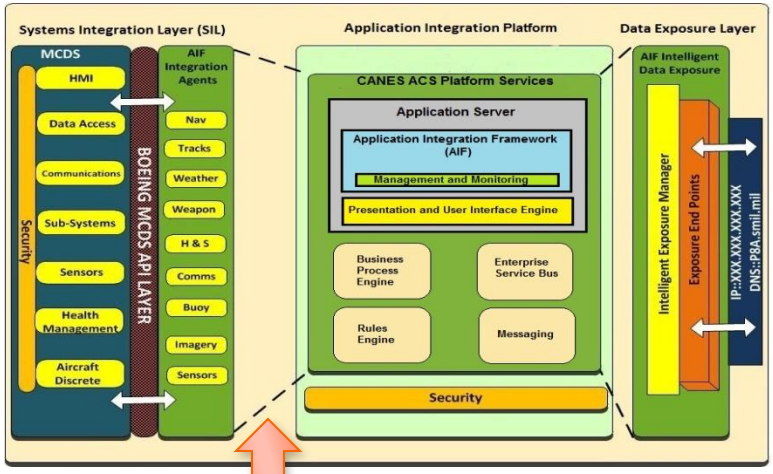


Managing Complexity with Scalable Architectures is Key to Managing Cost and Cycle Times

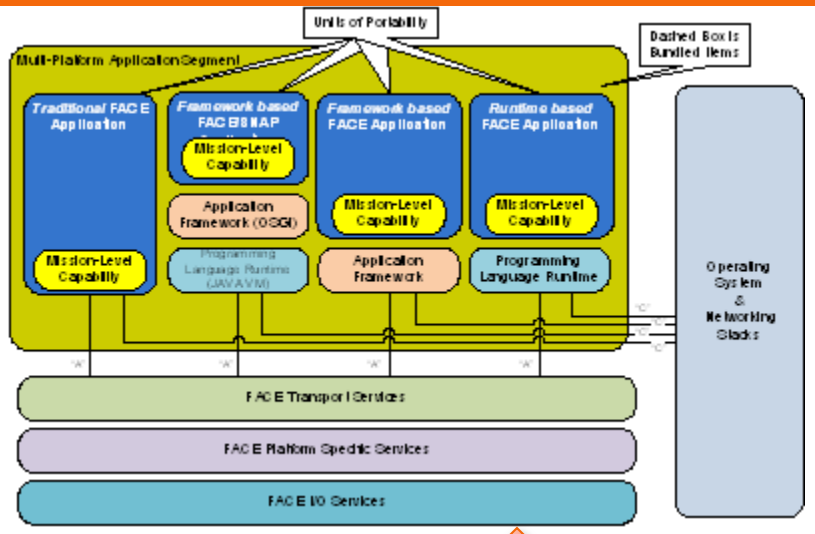
Net-Centric Architecture Development Approach

- "Learn by doing"
  - Grow infrastructure incrementally
  - Develop real applications intended for deployment
- Choose meaningful class of applications
  - Develop for Tactical Edge
    - Unstable underlying network topology
    - Size, weight, and power (SWaP) constraints
  - Cognitive software / mission-oriented applications
- Use modern techniques and technology
  - Agile software development approach (SCRUM)
  - Leverage existing tools and approaches

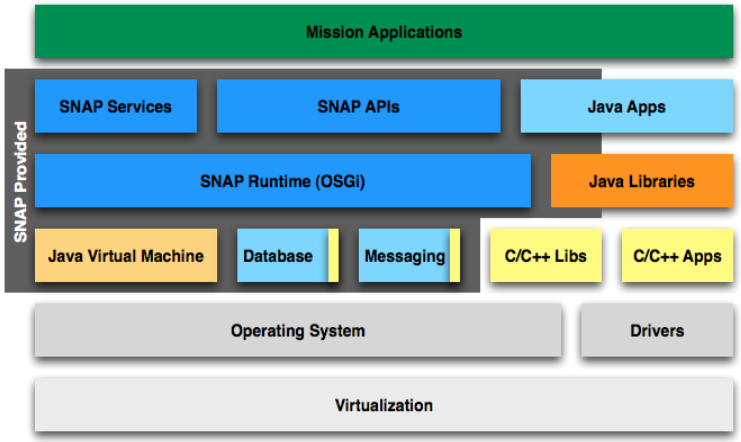
Airborne Supporting Architectures



P-8A Modular Open Systems Architecture (MOSA)



Future Airborne Capability Environment (FACE)



Scalable Network Application Proxy (SNAP) Architecture

Supporting Architecture Abstracts

- P-8A MOSA
  - Provides an open architecture reference
  - Provides a standardized, component-oriented environment
    - For both development and deployment
  - Employs Information Technology (IT) commercial industry best practices and standards.
  - Utilizes open-source technologies and software components.
  - Is focused on scalability and cross-platform application reuse
  - Leverages modern architectural concepts
    - In particular, a Services Oriented Architecture (SOA) and OSGi
  - Focuses on abstraction within the OSI Model layers 5 through 7, with framework integration at the application layer

- SNAP
  - Provides an open architecture reference
  - Provides a standardized, component-oriented environment
    - For both development and deployment
  - Consists of a Government-managed set of open-source software components
  - Is focused on scalability and cross-platform application reuse
  - Leverages modern architectural concepts
    - In particular, a Services Oriented Architecture (SOA) and OSGi
  - Focuses on abstraction within the OSI Model layers 5 through 7, with framework integration at the application layer

- FACE
  - Provides an open architecture reference
  - Developed by a government/industry consortium
  - Supports integration through well-defined interface standards.
  - Focuses on abstraction within the OSI Model layers 1 through 4, with low level abstraction at the POSIX interface
  - Can host other architectures and application frameworks.

Many commonalities, with some differences on approach. However, all focused on the same set of goals

SNAP and P-8A Architecture Approach

- Application Level Abstraction
  - Abstraction: ***allows the system to be defined similar to its actual meaning (semantic intent) but hides the actual implementation details.***
  - OSGi: the standard provides an environment for the modularization of applications into smaller bundles.
    - ***Bundles or Deployable Applications***: In the world of the OSGi model, these are typically "JAR" files with "manifest" information attached, but in essence these translate to standardized deployment components which represent applications; the deployment is constrained by the framework through the use of standardized xml schema(s). These schemas and/or manifests define how the framework will deploy and associate the application across the enterprise space.
    - ***Services***: The services layer dynamically connects bundles by way of offering a "publish-find-bind" for the packaged application and its object space; this includes breadth and depth, depth being any exposed shared resources and breadth as application or service reusability across the application space.
    - ***Service Registry***: A key component to Management Services, this exposes a well defined API or service front for common interface, resource sharing, as well as shared applications themselves; these concepts are articulated as part of the "publish-find-bind" mentioned above.
    - ***Life-Cycle***: The key component for Life Cycle Management (LCM), it exposes a well defined set of API(s) or services for the management of application end to end life cycle to include install, uninstall, start, stop, and pause.
    - ***Modules or Dependency Management***: This layer defines the encapsulation and dependency exposure for internal and cross application dependency that will be managed by the framework (how applications expose/export and consume/import code).

Defined Standard
• Exposes infrastructure as an abstraction
• Allows for consistency from development through deployment

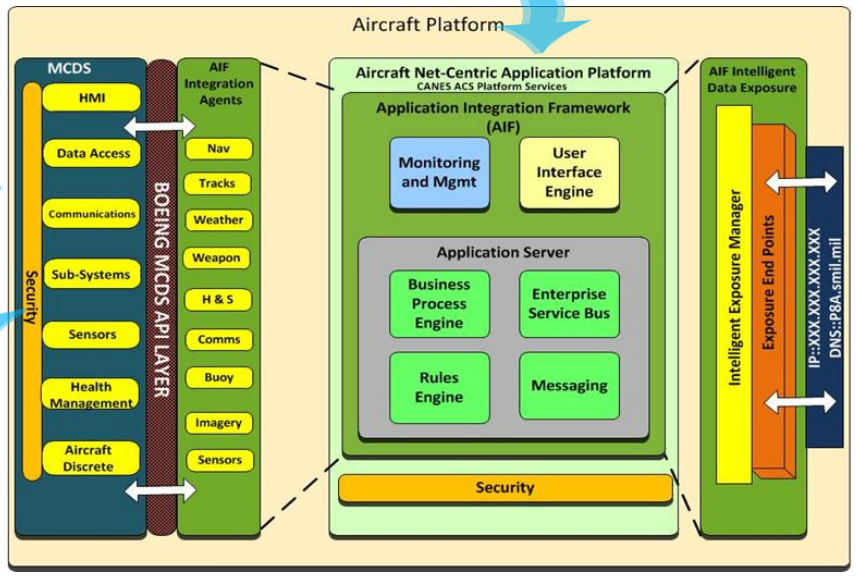
P-8A MOSA Architecture

Adaptable Architecture Supports:

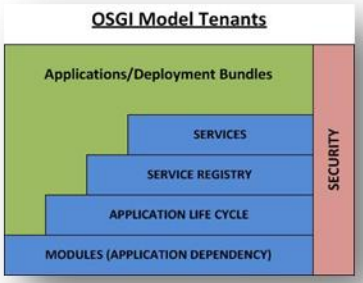
- Service Oriented Infrastructure
- Open Standardized Development
- OSGi Model
- Loosely Coupled Applications, Data, and Services
- Rapid Capability Insertion
- Net Centric Enablement
- Reduced complexity for integration with an existing legacy computing environment

Service Oriented Infrastructure Core

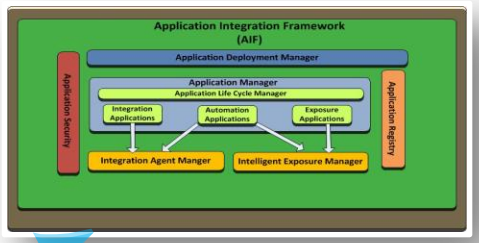
Existing Computing Environment

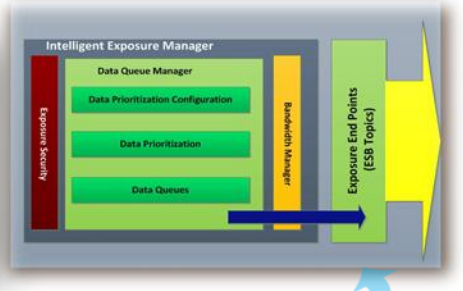


Integration Layer



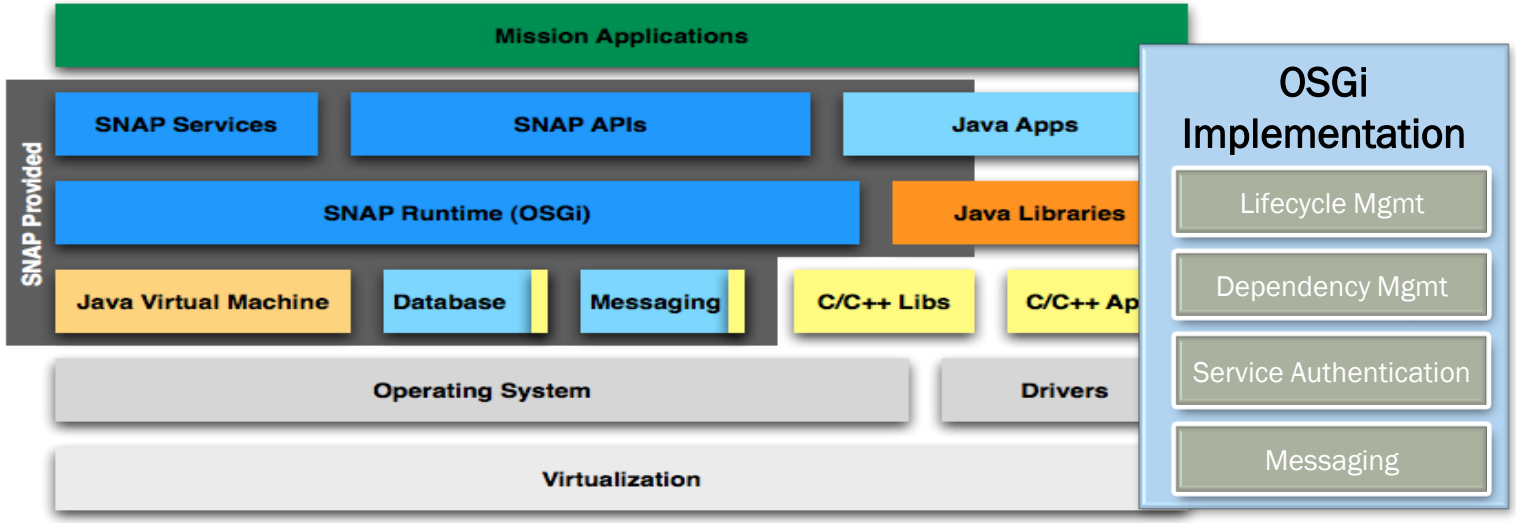
Enables process automation and rapid capability insertion and exudes the tenants of the OSGi Model





Intelligent, Managed Data exchange

SNAP Architecture

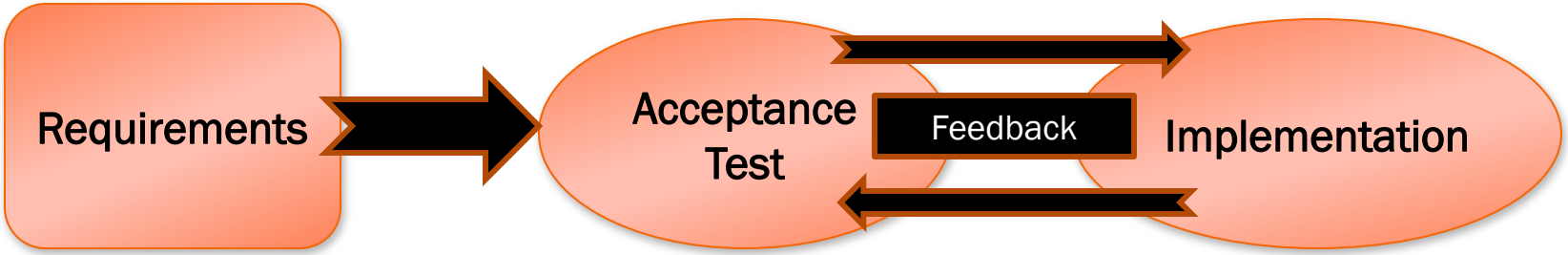


- SNAP Services, APIs and run-time infrastructure define the execution environment
  - Strong reuse of existing third party components dramatically reduces cost and effort
- Use of Java provides OS-independence and enables cross-platform reuse
  - Well supported by a wide array of solid development and test tools (e.g., Eclipse, Hudson)
  - Interface mechanisms to external software accelerate re-hosting to meet near-term objectives

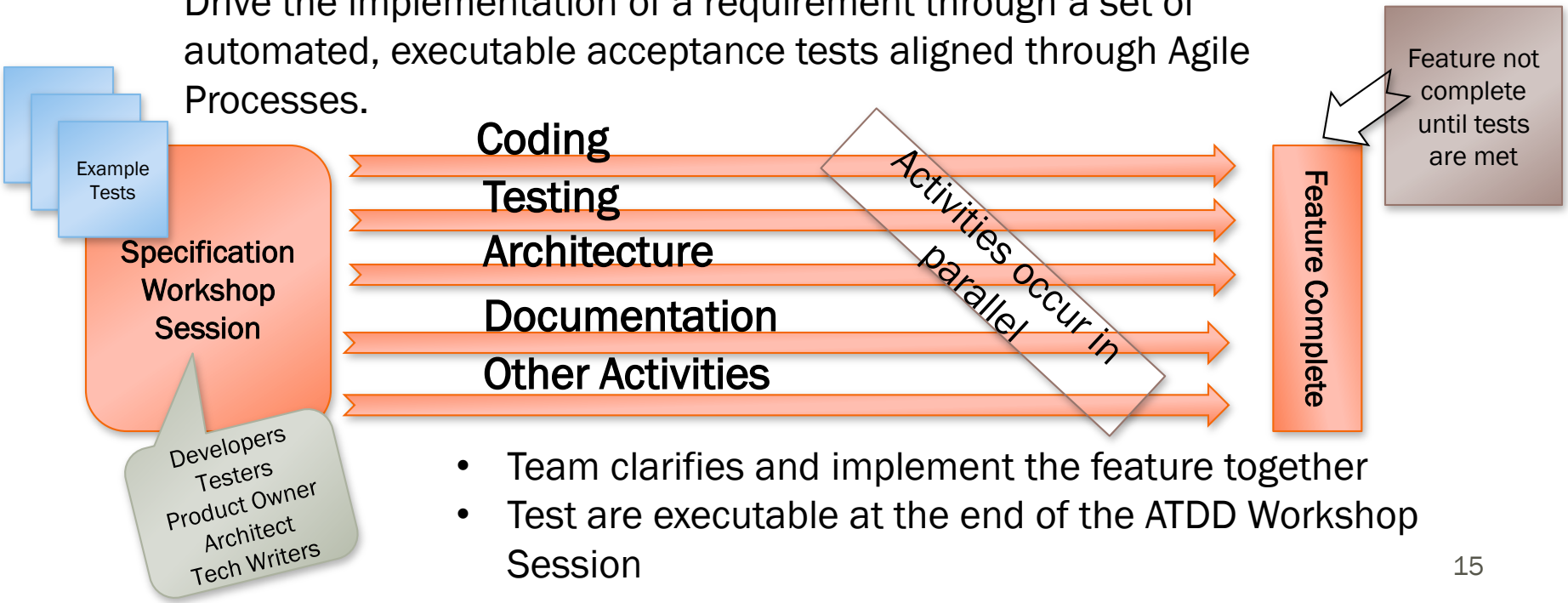
Supporting Technology and Tools

- Java / JDK
- Runtime Services & Support
  - OSGi Framework / Karaf
  - Apache ServiceMix
  - Camel messaging
  - JBOSS – SOA-P
- Testing
  - JUnit – automated unit testing
  - Hudson – continuous integration testing
- Other
  - Maven (dependency management)
  - ANT
  - Nexus (artifact repository)
  - Subversion (version control)
  - Bugzilla (defect tracking)
  - Eclipse IDE
  - NASA World Wind (GUI development, mapping software)

Acceptance Test Driven Development (ATDD)



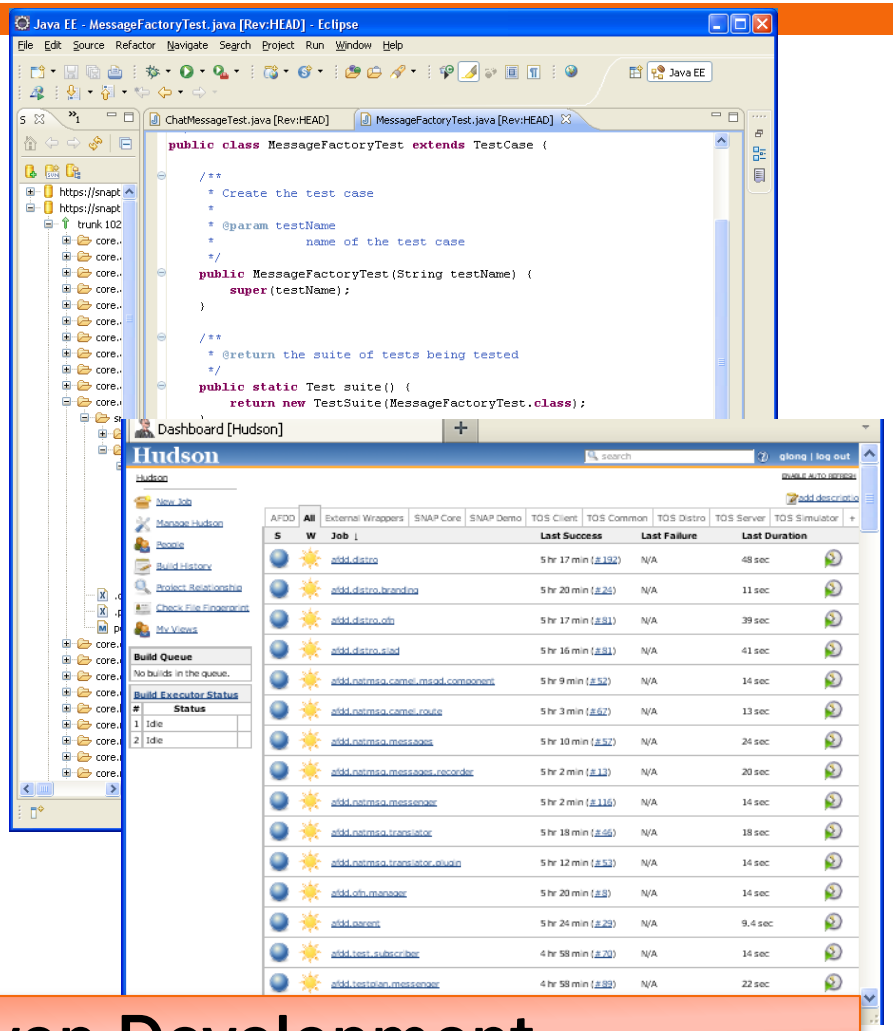
Drive the implementation of a requirement through a set of automated, executable acceptance tests aligned through Agile Processes.



- Team clarifies and implement the feature together
- Test are executable at the end of the ATDD Workshop Session

Testing Approaches

- Unit Tests
  - JUnit: automated unit testing
  - Experimenting with test-driven development (write JUnit tests first, then write the code)
- Integration Tests
  - Hudson: continuous integration
  - Runs whenever new code is checked in



Starts with Test Driven Development

Integration / System Testing

- Testing should be continuous
  - Need representative hardware and simulated operational environment
  - These environments should be available to all developers during development
  - Requires networked simulation and test facilities

Not yet available for SNAP development

PMA-290 working towards an Integrated Prototyping Lab to facilitate simulation based testing

P-8A currently utilizes system in the loop testing with Boeing MOLE System

Benefits of ATDD

- Done is Done!
  - Comprehensible examples over complex formulas
  - Close Collaboration with development , testing, and quality assurance.
  - Well defined definition of “Done”
  - Trust and commitment
  - Testing not only at the unit level, but at the systems level too!

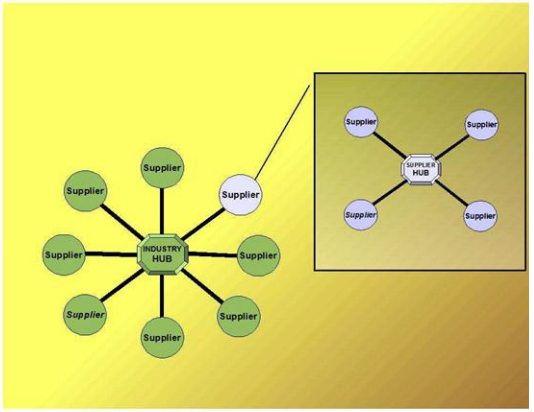
Questions ?

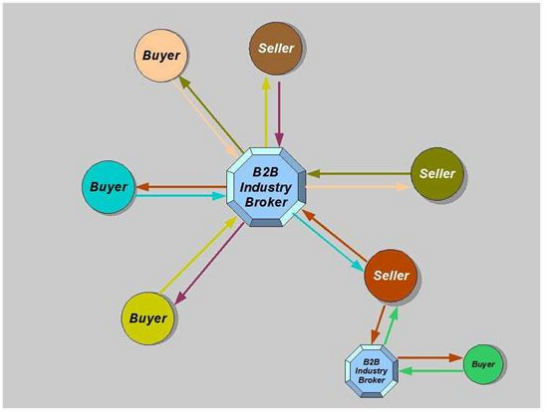
Back-Up Slides

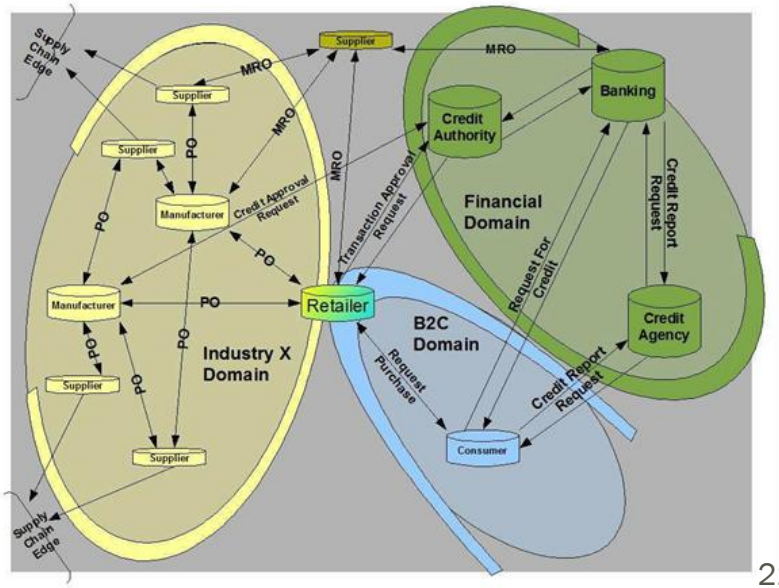
20

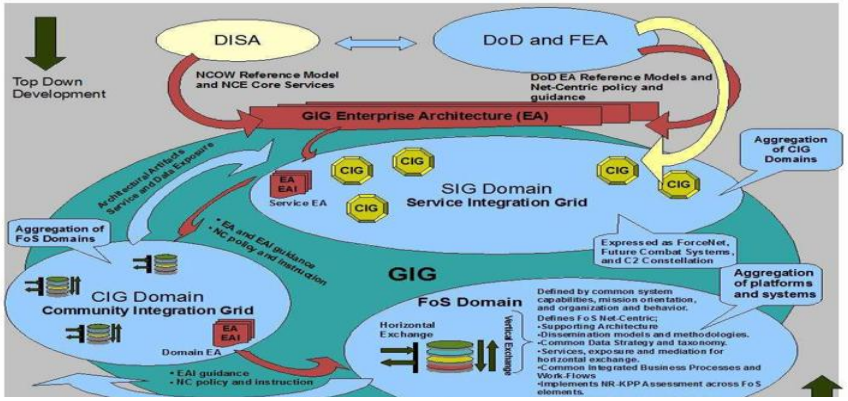
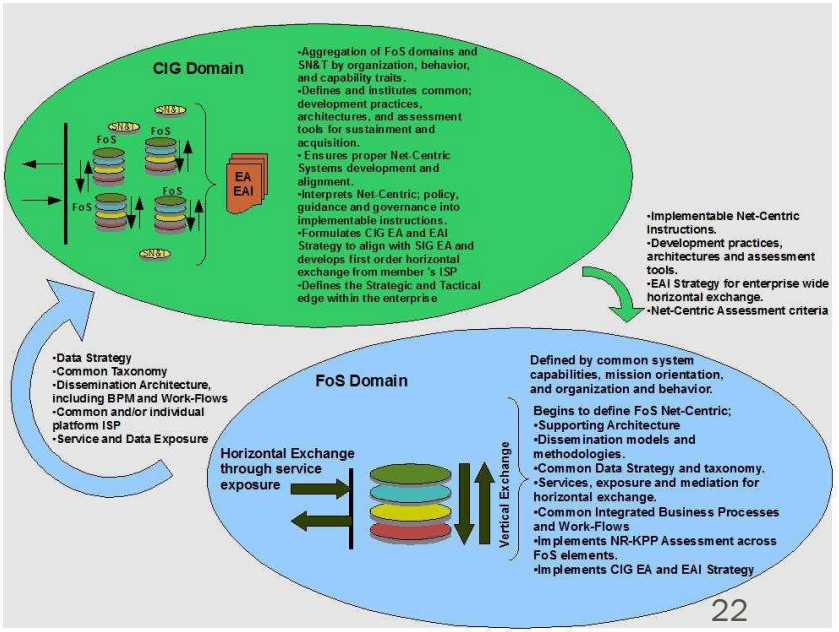
Evolution of Net-Centric Solutions

- Vertical Domain Exchange
  - Driven by cost reduction needs
  - Introduced B2B Exchange paradigm
  - Produced proprietary solutions
- Horizontal Domain Exchange
  - Provided further cost reduction
  - Refined the B2B Exchange Paradigm
  - Initiates cross domain exchange.
- Hybrid Exchange Model
  - Adoption of community driven standards
  - Exposure of information and services from disparate sources
  - Middleware standards spur the development of Message Oriented Middleware (MOM).
  - SOA emerges as a primary supporting architecture.
  - Not just for B2B, but introduces the producer-consumer relationship







DoD Net-Centric Exchange Domains

- **The GIG is represented as a heterogeneous aggregation of:**
- Family of Systems Domains (FoS)
  - Vertical exchange domain
  - Common systems, mission, organization, business structures, and behavior models
  - Promotes supporting architectures, dissemination models, data strategies, and taxonomies.
- Community Integration Grids (CIG)
  - Self-forming, heterogeneous, Horizontal Exchange Domain.
  - Aggregation of FoS Domains and other Nodes.
  - Promotes cross domain interoperability and supporting standards.
  - Manages multi-domain IT infrastructure
- Service Integration Grids (SIG)
  - Aggregation of CIG Domains
- Singular Nodes and Transients
  - Nodes or assets that freely operate across CIG, SIG boundaries.