Mission Thread Workshop

Lessons Learned

STSC 2013

Bill Wood, Mike Gagliardi, Tim Morrow

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM-0000208

Outline

Problem

Background

Mission Threads

Lessons Learned

Challenge Themes

Next Steps and Conclusion

Problem

Integration and operational problems arise due to inconsistencies, ambiguities, and omissions in addressing quality attributes between system and software architectures. This is further exacerbated in an SoS.

Example quality attributes: predictability in performance, security, availability/reliability, usability, testability, safety, interoperability, maintainability, force modularity, spectrum management.

Functionality and capability are critically important, but the architecture must be driven by the quality attributes. Specifying and addressing quality attributes early and evaluating the architecture to identify risks is key to success.

The Need for Augmented End-to-End Mission Threads in DoD SoS Architecture Definition

DoDAF provides a good set of architectural views for an SoS architecture. However, it inadequately addresses cross-cutting quality attribute considerations.

System use cases focus on a functional slice of the system.

More than DoDAF and system use cases are needed to ensure that the SoS architecture satisfies its cross-cutting quality attribute needs.

SoS end-to-end mission threads augmented with quality attribute considerations are needed to help define the SoS Architecture and then later evaluate the SoS architecture and constituent system/software architectures.

Background-1

SEI has successfully used ATAMs to evaluate software architecture over the past 10 years

- Quality attribute scenarios are the “test cases” to evaluate an architecture qualitatively during a workshop with stakeholders
- The Quality Attribute Workshop (QAW) develops scenarios
- This has been expanded to “system architecture” and works equally well

We had concerns with using scenarios for evaluating SoS architectures

- We determined to use Mission Threads as a basis for evaluation
- We created the MTW to mimic the QAW
- Focus on highlighting activities across multiple nodes

Background-2

Types of Mission Threads

- Operational, acquisition, developmental, sustainment
- Most of our experience is with *Operational*

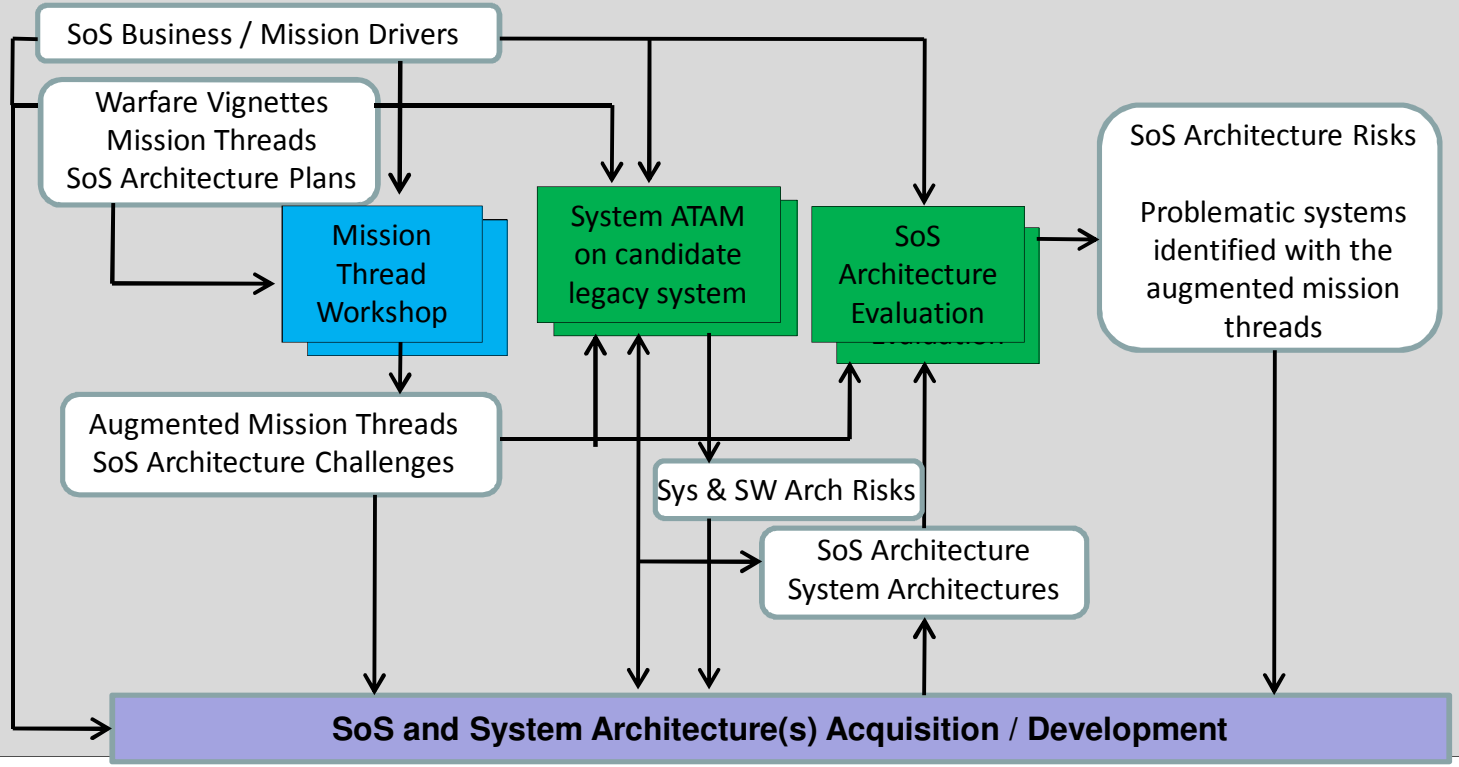
Choosing critical *Operational* Mission Threads

- New threats and missions, doctrinal changes
- Manpower reduction – automation
- Underlying technology upgrades
- Planning for operations
- Sufficient to cover UNTLS
- Oftentimes good vignettes can be found in existing AoA, CONOPS, etc

Army prefers to call them “User Stories” and commerce “Workflows”

SoS Architecture Quality Attribute Specification and Evaluation Approach

- Early elicitation of quality attribute considerations
- Early candidate legacy system architecture evaluation
- Early identification and mitigation of architectural risks



Mission Thread Workshop - Goal

To augment a set of end-to-end System of Systems (SoS) mission threads with quality attribute and engineering considerations with the stakeholders.

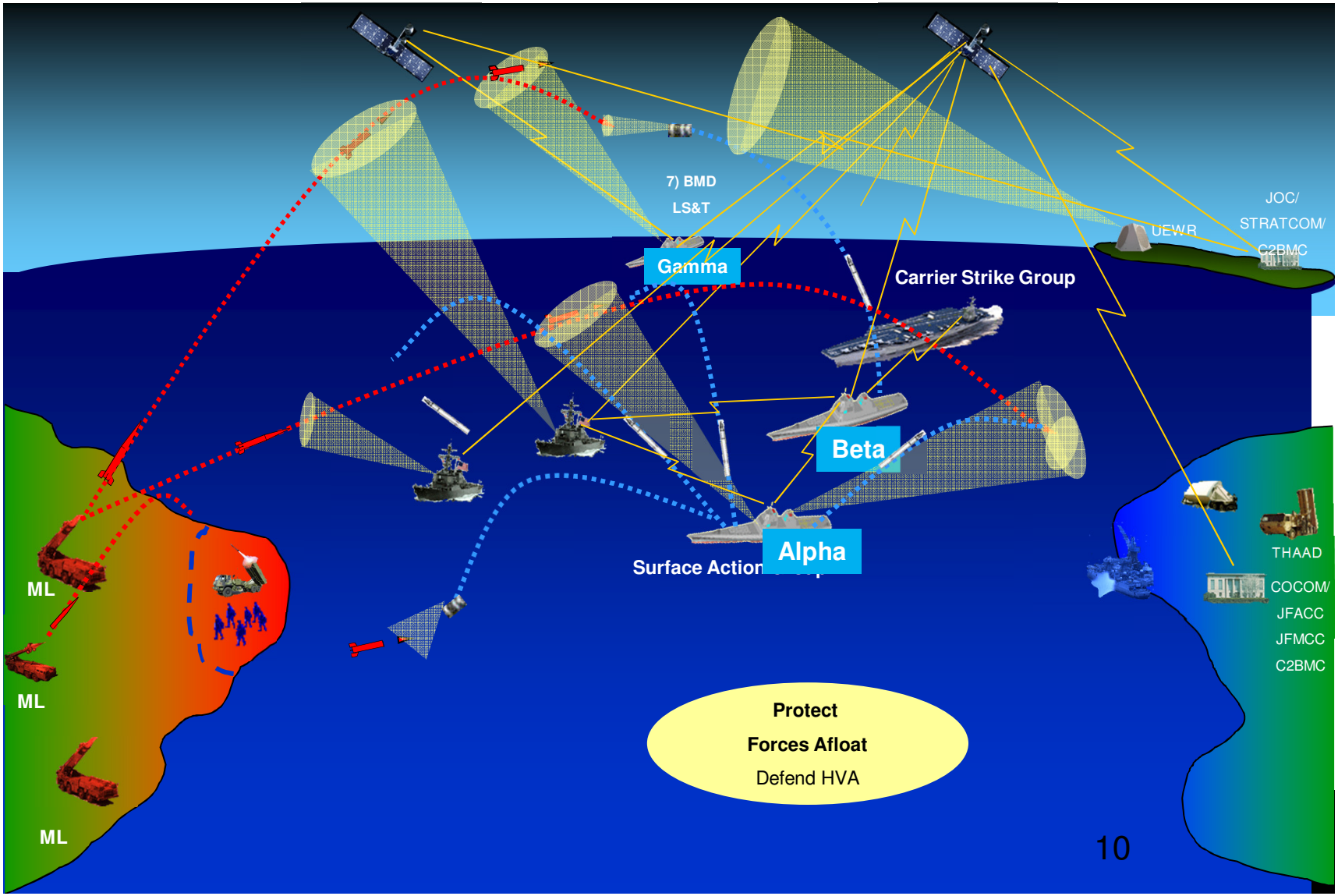
To capture at each step of the mission thread AND each SoS quality attribute

- the engineering considerations from diverse stakeholders
- identify gaps, overlaps in capabilities
- the quality attribute concerns associated with the mission thread
- the applicable use cases for the different nodes and/or systems

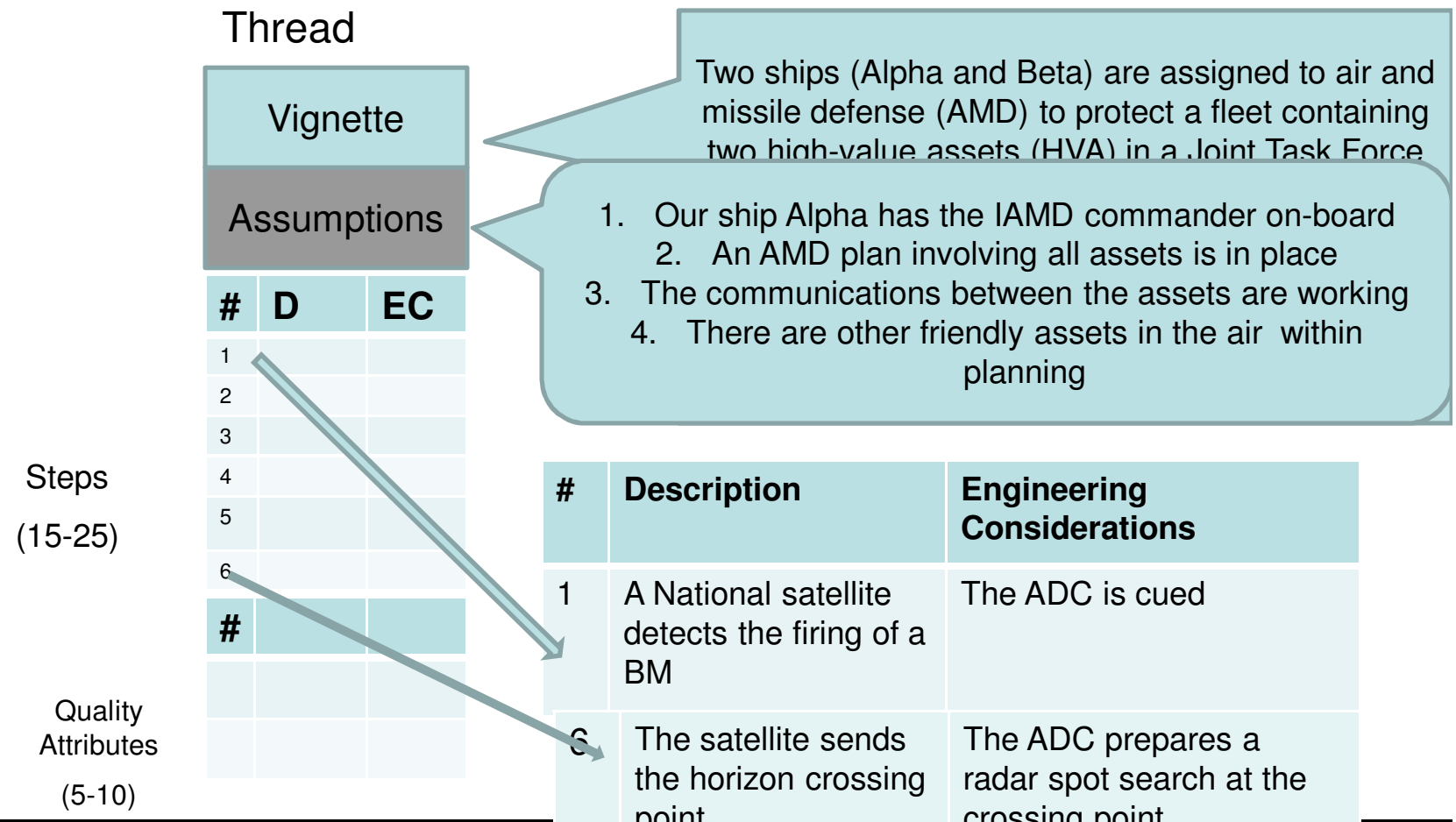
To develop technical challenges associated with the threads, and to aggregate the challenges over a number of MTWs

Outputs will inform and drive SoS Architecture Decisions.

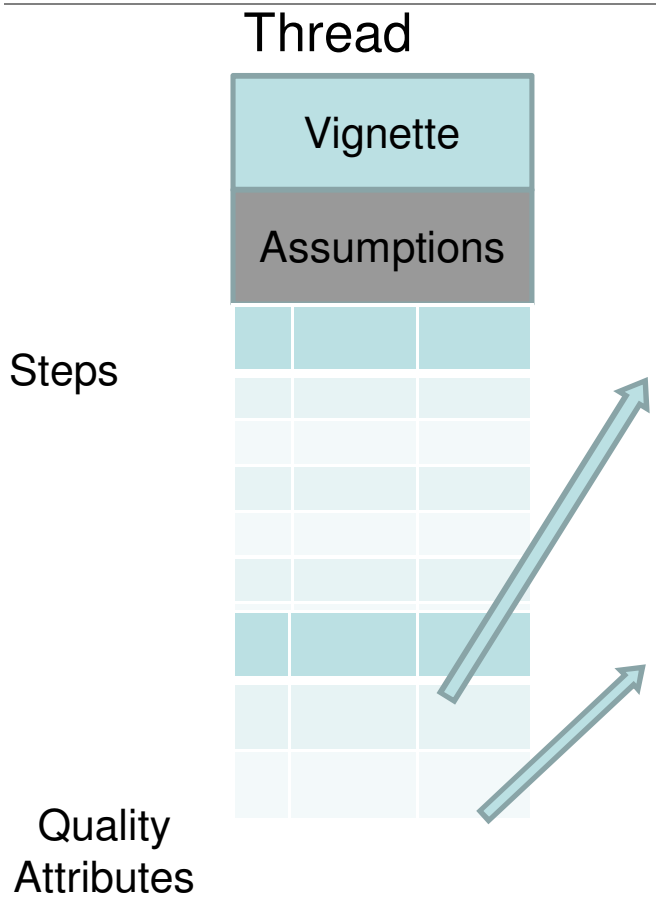
Air and Missile Defense (AMD) OV-1 Example



Mission Thread (Template)

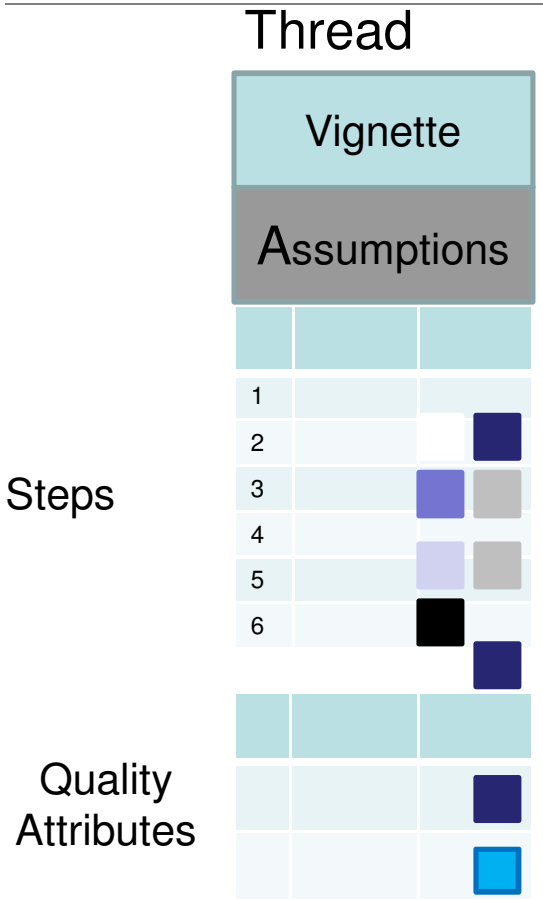


Mission Thread (details)



Quality	Aspect	Comments
Performance	Bandwidth	During high tempo periods prioritization of usage must be imposed
Availability	Recovery	This capability must recover from a single point of failure within .x sec.

Use Case Pointer



Use Cases (Built as follow-on)

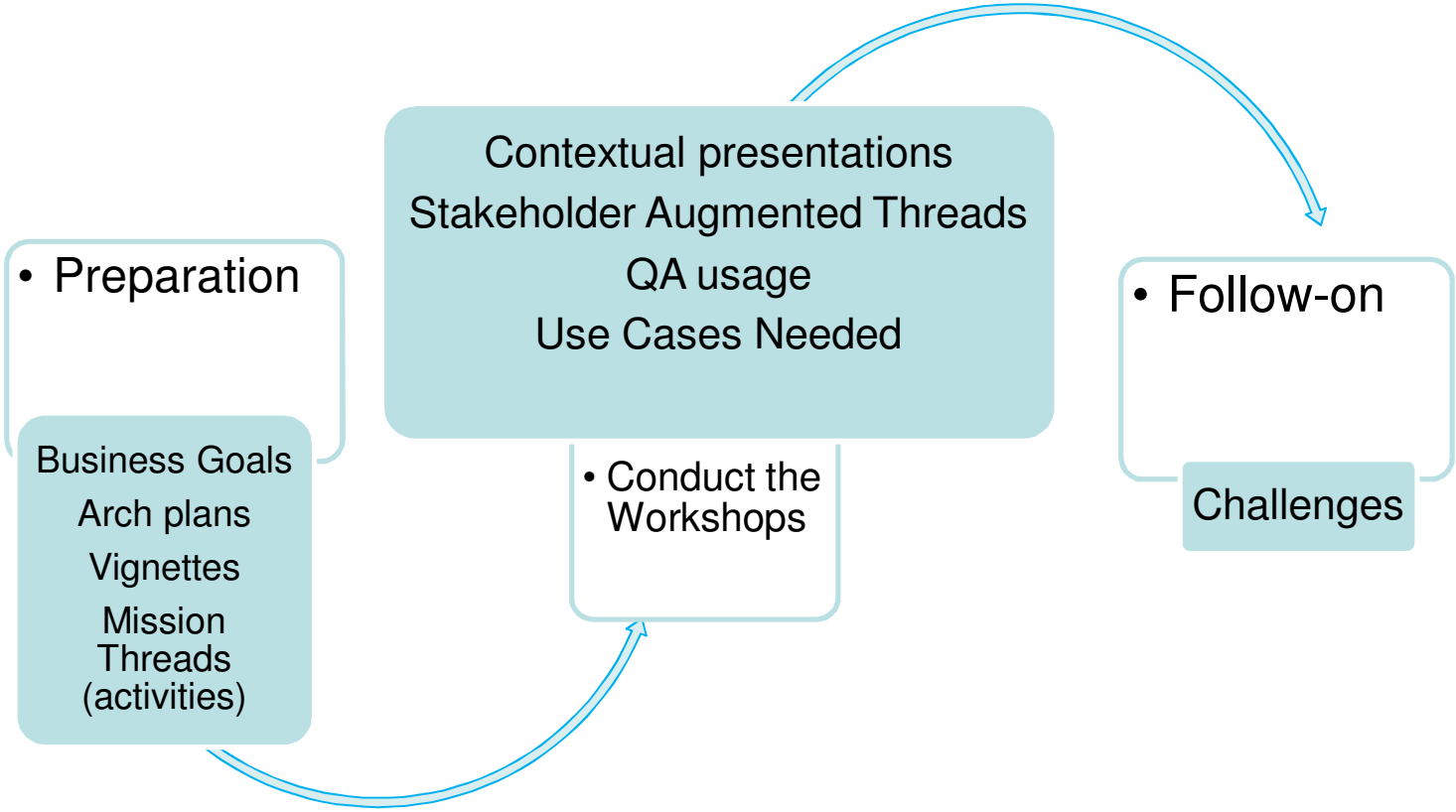
Node 1

Node 2

Node 3

MTW Process

Stakeholder Inputs are Key



Critical Threads - Developmental

New Technology and shared resources

- Configuring software/hardware mapping
- Certifying multiple configurations
- Automated components

Reusing and re-engineering existing components

Architectural Framework

- Applications: Levels, Tiers, Services
- Security: Single login, multi-level
- Infrastructure Services: Directory, communications, load balancing
- External Services: weather, geo-spatial, planning

Critical Threads - Sustainment

Reduced Manpower-automation

Recovery from failure

- On board, off board, in foreign port, home base
- Spare parts and test equipment and trained personnel

Fuel and supplies and port visits

Training: on/off board, JiT, simulation

Ship/Test Facilities integration

- Collecting and exporting and filtering and re-playing data records
- Version upgrade strategy

Mission Thread Workshops - Experiences

Client	Description	MTWs	Vignettes	Mission Threads	Stakeholders
A	IRAD New platform/capability	1	1	2	8
B	New Naval Ship	13	17	37	>200
C	Battle Command	6	3	4	>100
D	Maritime Detection	2	4	4	30
E	NSF	1	3	3	15
F	Air Force Program	1	1	1	10
G	Other Govt Agency	1	4	4	12

- Identifies SoS architecture gaps, overlaps and challenges
- Identifies issues for constituent legacy system/software architectures
- Overcomes organizational stovepipes and facilitates stakeholder communication

Lessons Learned

Phases

- Preparation
- Conduct
- Follow On

SoS Challenges

Preparation Activities

- Scope the series of MTWs to satisfy operational coverage needs
- Develop OV-1 diagrams and vignettes for the operational capabilities
- Develop step-by-step description of activities (threads) in response to a set of stimuli for the vignettes
- Develop a set of architectural quality attributes for the vignettes
- Determine the stakeholders to attend each MTW
- Identify the planned use of legacy systems

Preparation Lessons Learned

- OV-1 (or a user story) is crucial
  - AoA and User Story documents are a good source
  - MTWs served to normalize the different OV-1's capabilities
- Assumptions are a key part of the template
- Focus is on SoS capabilities, activities, and QAs
  - Software is critical, but implicit
- Initial coaching and oversight needed to build the threads
  - Leads for later workshops attended earlier workshops and developed VERY good vignettes/threads
  - Threads should be well vetted prior to workshop
- 15 to 30 steps are typical for each mission thread
- Operational thread often needs associated planning thread
- Time period of a thread can be from minutes to days

Conducting Workshop

Activities

- Briefings on the operational challenges and the workshop intent and description
- Augment the thread template for engineering considerations /QAs/Use Cases with each step
- Augment the QA template adding over-arching considerations

Conducting Workshop – Lessons Learned

If there was no planning thread, planning assumptions and perhaps a step 1 or a new thread will have to be added

Don't mix operational, developmental and sustainment threads

First thread takes 3 to 4 hours, following threads take less time

Only a few added steps were needed typically (for a well vetted thread)

- Some poorly vetted threads required more changes to the steps
- Listen to the *warfighters*, engineers can get the thread wrong

Work initially with a small group then work to get confidence (pilot)

Strong third party facilitation allowed operational principles to discuss rather than defend

Diverse operational experiences eliminate stovepipe mentality

Dialogue between stakeholders was illuminating to all

Overlaps in capabilities are more problematic than gaps

Follow-on Activities

Facilitation team

- Form a table of challenges (5 to 7) with pointers to MTW steps/QA/assumptions
- Build a briefing, one page per challenge
  - Description, evidence, impact and recommendations
  - Keep the pointers and put the major points in the Notes Page
- Vet and update each challenge with the clients and the leads

Lessons Learned:

- As many capability / engineering gaps and challenges as architectural
  - Clients corrected domain specific misunderstandings
- Avoid rolling up too much, it can become meaningless
- Need actionable recommendations for challenges.

SoS Quality Attributes

Quality Attributes of interest depend on vignette/thread type

- Operational： performance, availability, security, interoperability
- Developmental: legacy reuse, extensibility, openness, integrability
- Sustainment : maintainability, training, deployability, upgradeability

New consideration examples

- Survivability: Machinery MT on how to contain compartmental flooding in a critical compartment resulted in discussion on using new pump technologies to avoid flooding.
- Availability: Machinery MT on failure of a generator has a massive impact on all ship operations and mission
- Availability: Degraded operation on a failure needs to be defined across echelons, and mitigation alternatives defined
- Reduced Manning/Automation

Challenge Rollup Across SoS Clients

#	Name	# Clients
1	Usability/Automation	3
2	Capability Gaps	4
3	Resource Management / Disaster Recovery	4
4	Training	3
5	Legacy Migration	3
6	Collaboration	4

Recommendations not rolled up for this presentation.

Usability/Automation

Each system has its own “Look and Feel” and a common “look and Feel” must be developed using a common toolkit, graphics and icons.

There is a lack of “grunt-work” automated support and tool integration for many critical processes used by the warfighters

Human Factors

- The cognitive burden on the warfighters must not overwhelm them
- In order to support “reduced manning” we need more automation
  - Both operational and sustainment (field service engineers)
- Alert management requires root cause analysis

Capability Gaps

Omissions

- Aircraft as communication relay, as well as sensors
- Data collaboration to reduce classification time

Situational Awareness

- Engagements can last for hours, the warfighters need 360° Awareness

Multi-Mission Planning

- Distributed/collaborative planning - overlapping time periods

Demonstration Omissions

- Effectiveness called into question because of missing critical capabilities

End-to-End Modeling and Simulation was under-played

Resource Management / Disaster Recovery

Individual systems had

- Low operational reliability
- Have to re-build Situational Awareness state after recovery from failure

Disconnected operations poorly defined and managed

Degraded modes of operation inconsistently defined within SoS

- Impact of loss of FCQ track

Distributed Resource Manager could not map from large scale failure to impact on current missions to suggested recovery strategies

Training

Training system has capability gaps

Operator proficiency degrades between assignments, but no re-training

Need lightweight simulations on-board for embedded training and mission rehearsal

New “Look and Feel” will cause extensive re-training

Maintenance and training considerations are not sufficiently well defined for the support systems to be well architected

Migration of Legacy Systems

Current stovepiped systems will have trouble migrating to a COE, and both FMS and weapons safety certification further complicates this effort.

Each stovepipe has its own data architecture for: data-at-rest, data-in-transit, and external interfaces. The Architecture Team will have to determine commonality (and differences) between the information being used, and formulate common data structures.

Each stovepipe use different development environments and tools, have different CCBs, integration and test environments, development processes and different backward compatibility strategies.

Collaboration

There is little automated support for geographically distributed, cross-echelon efforts to classify tracked objects

Mapping the external interoperations semantically to the missions being planned or conducted is inadequate

Cutoff between manual and automated management of the fight involving many incoming missiles is not defined

The strategy to move currently stovepiped systems to a COE, and to deploy across to multiple echelon TOCs and platforms

SoS Architecture Development Process - 1

Each MTW prepared a good OV1 diagram to support the vignette. The set of OV1's became "normalized" over time.

Diagrams developed in PowerPoint or Visio were more than sufficient to support MTW effort, but use of a modeling tool (i.e., System Architect) is probably needed to support development of artifacts in later architectural processes

The MTs developed were the basis for building OV 2,3,5 and SV1, 2 DoDAF diagrams, but additional guidance was needed in architecting process to provide a clear transition

Stakeholders were uncomfortable developing vignettes/mission threads without a CDD-like requirements document

SoS Architecture Development Process - 2

Development of a Mission Thread Description Document (MTDD) is a good way to capture architecture decisions. The MTDD contains the artifacts developed to support the MTWs, outputs of the MTWs (capability gaps, quality attribute augmented mission threads and architectural challenges) and ties to high level use cases.

SoS Architectural Guidelines and Principles document is needed to provide consistent guidance through the architecting process.

Summary

Can augment end-to-end threads with QA considerations

Identifies SoS challenges early (very good risk predictors)

Cross-discipline stakeholders can agree on thread steps

- Reduce “rice-bowls”, identify “long poles”

Good facilitation is necessary

- Enough patience to hear things through
- Enough control to move things along

Approach can be easily tailored and has been used for an Enterprise Service context

A core team for MTW facilitation and SoS stakeholders provided consistency

Contact Information

Mike Gagliardi
Software Engineering Institute
MJG@sei.cmu,edu
412-268-7738

Bill Wood
Software Engineering Institute
WGW@sei.cmu.edu
412-268-7723