

# Mission Thread Workshop

## Lessons Learned

SATURN 2012

Mike Gagliardi, Bill Wood



Software Engineering Institute

Carnegie Mellon

SEI  
Gagliardi  
© 2010 Carnegie Mellon University

## Copyright 2012 Carnegie Mellon University

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.

### NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

This material has been approved for public release and unlimited distribution except as restricted below.

The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013 and 252.227-7013 Alternate I.

Internal use: \* Permission to reproduce this material and to prepare derivative works from this material for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.

External use: \* This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other external and/or commercial use. Requests for permission should be directed to the Software Engineering Institute at [permission@sei.cmu.edu](mailto:permission@sei.cmu.edu).

\* These restrictions do not apply to U.S. government entities.



Software Engineering Institute

Carnegie Mellon

SEI  
Gagliardi  
© 2010 Carnegie Mellon University

# Outline

---

Overview of MTW

Background

Lessons Learned

Challenge Themes

Next Steps and Conclusion



# Mission Thread Workshop - Goal

---

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

Capture at each step of the mission thread

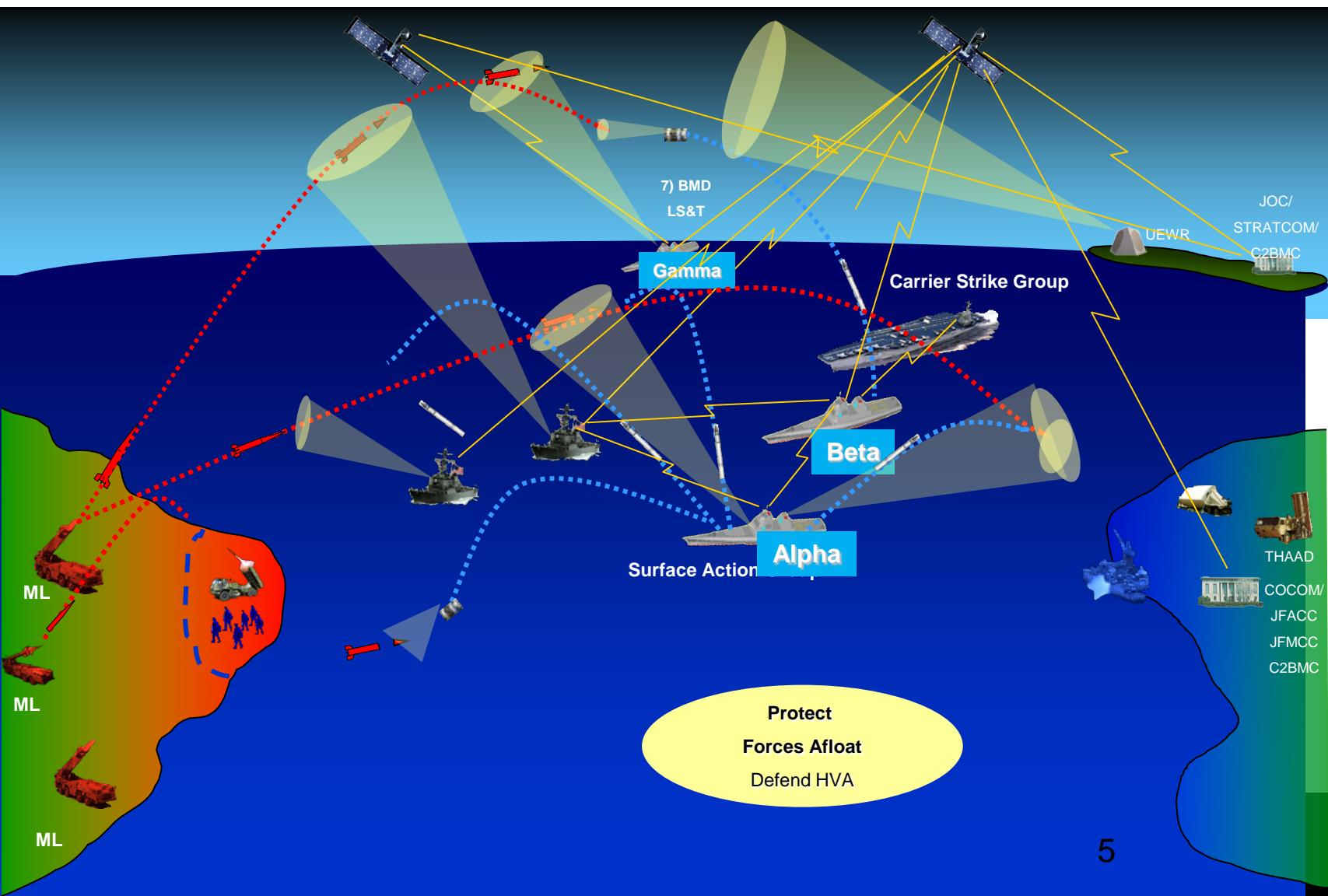
- the engineering considerations from diverse stakeholders
- the quality attribute concerns associated with the mission thread
- the applicable use cases for the constituent systems

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

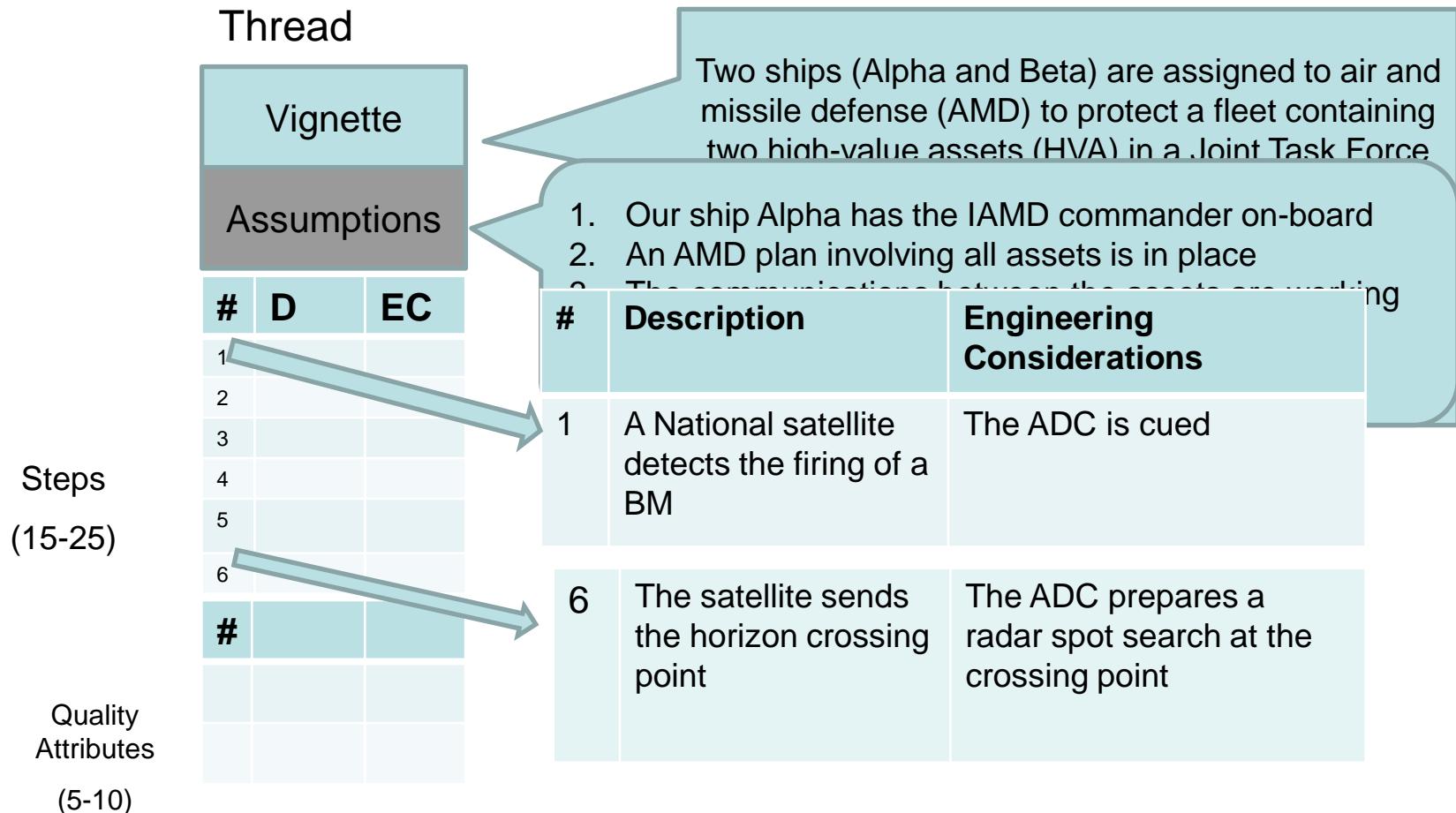
Outputs will drive SoS and System/Software Architecture Decisions.



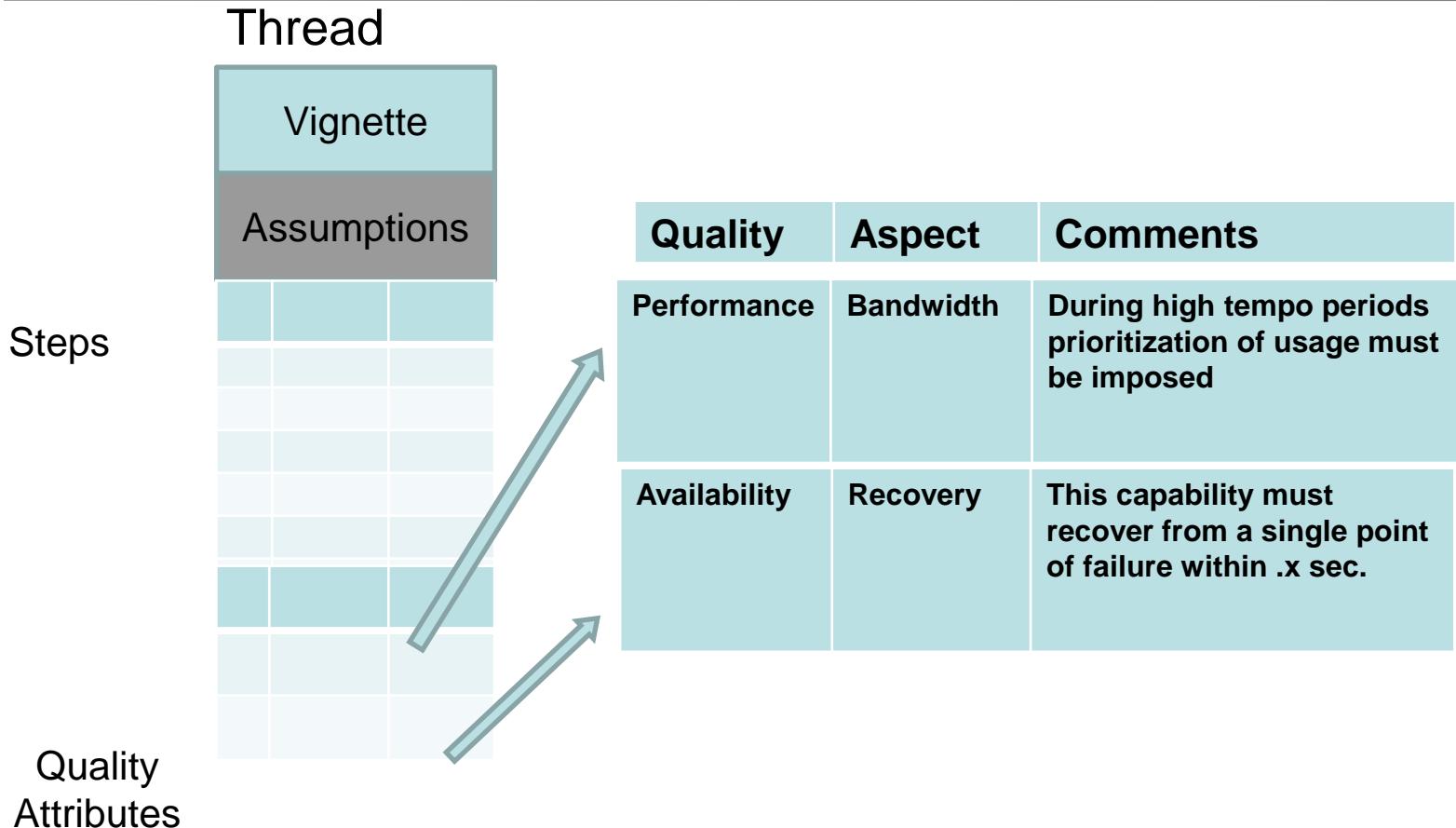
# Air and Missile Defense (AMD) OV-1 Example



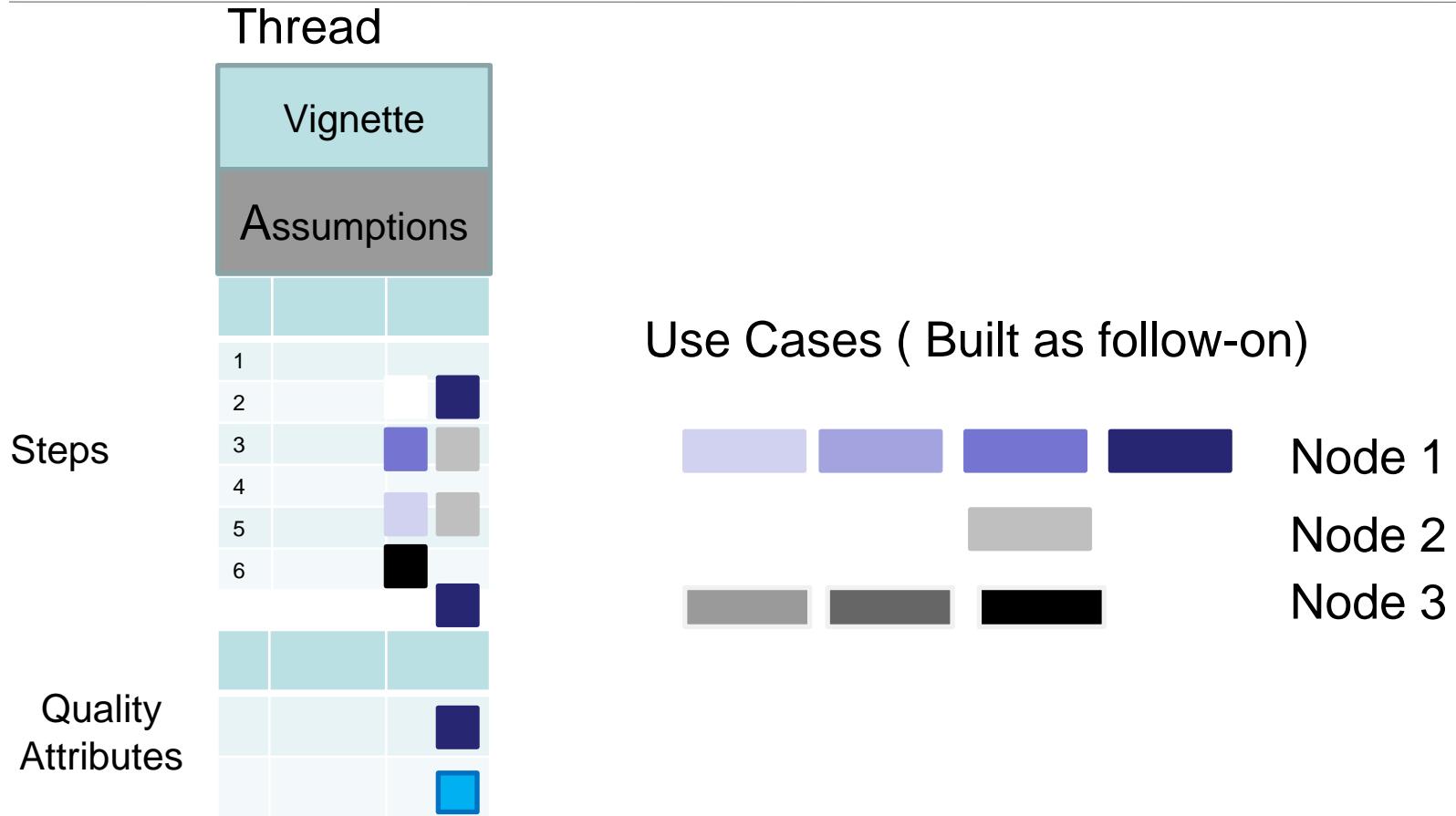
# Mission Thread (Template)



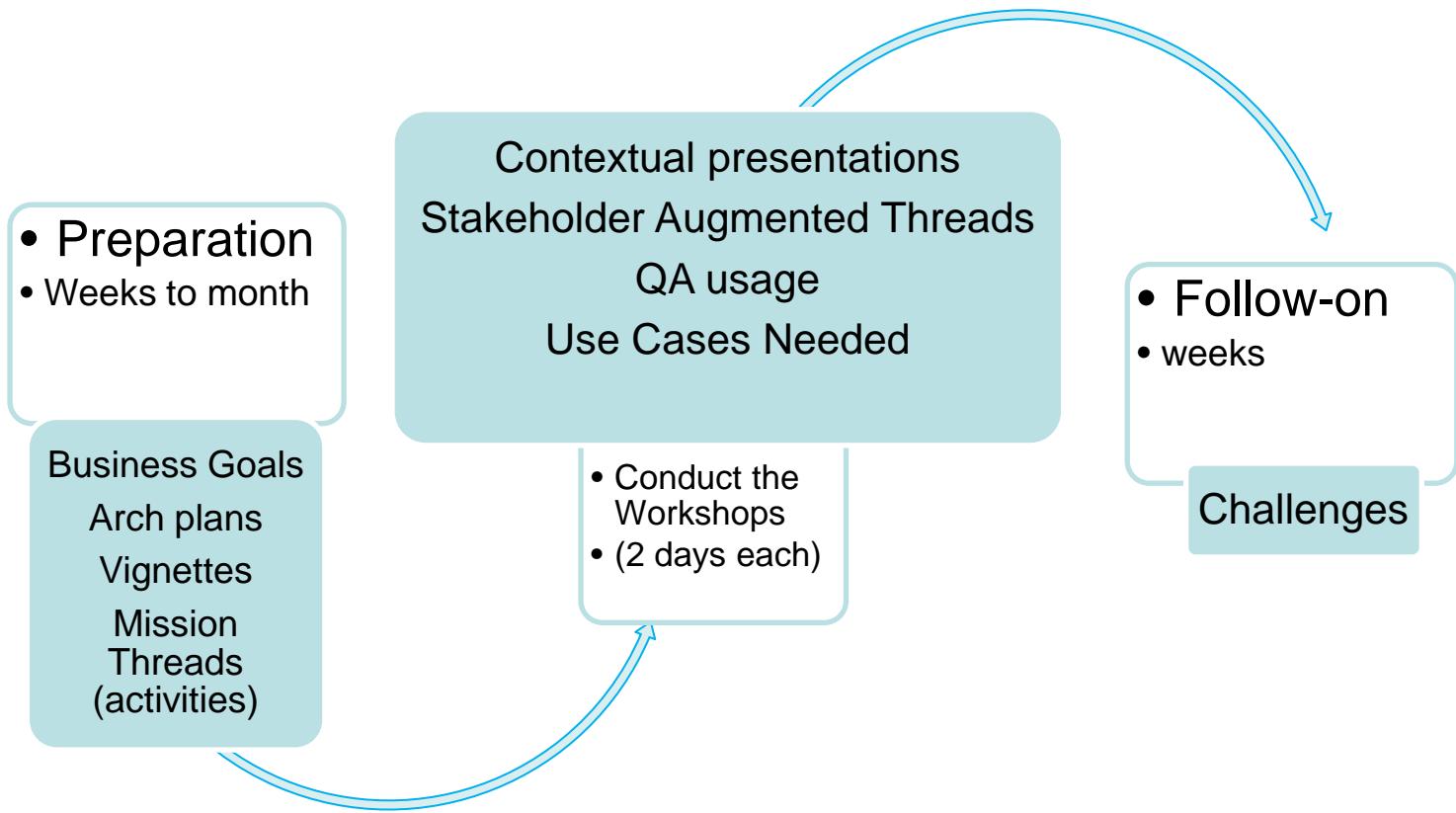
# Mission Thread (details)



# Use Case Pointer



# Process



# Numbers to Date

---

Client	Description	# MTWs	# Vignettes	# Mission Threads	# of 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	DHS	1	2	2	12



# Lessons Learned

---

## MTW Phases

- Preparation
- Execution
- 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



# 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	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

---

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



# Next Steps

---

Acquisition Strategy

Developmental threads

Courses to support training needs



# 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

