

# Introduction to the Mission Thread Workshop

Mike Gagliardi  
Bill Wood  
Tim Morrow

**October 2013**

**TECHNICAL REPORT**  
CMU/SEI-2013-TR-003  
ESC-TR-2013-003

**Software Solutions Division**

<http://www.sei.cmu.edu>



Copyright 2013 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.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

This report was prepared for the  
SEI Administrative Agent  
AFLCMC/PZM  
20 Schilling Circle, Bldg 1305, 3rd floor  
Hanscom AFB, MA 01731-2125

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.

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.

Architecture Tradeoff Analysis Method®, ATAM® and Carnegie Mellon® are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

DM-0000602

---

# Table of Contents

<b>Acknowledgments</b>	<b>vii</b>
<b>Abstract</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background	2
1.2 Systems of Systems	3
1.3 Vignettes	3
1.4 Mission Threads	4
1.5 Quality Attributes	6
<b>2 A Brief Introduction to the Mission Thread Workshop</b>	<b>7</b>
<b>3 The Phases of the Mission Thread Workshop</b>	<b>8</b>
3.1 Preparation Phase	8
3.2 Execution Phase	10
3.3 Follow-on Phase	11
<b>4 The Steps of the Execution Phase of the Mission Thread Workshop</b>	<b>13</b>
4.1 Step 1: Present the Mission Thread Workshop	13
4.2 Step 2: Present the Business and Mission Drivers	13
4.3 Step 3: Present the Architectural Plan	14
4.4 Step 4: Review the Mission Thread Header	14
4.5 Step 5: Augment the Mission Thread Steps	15
4.6 Step 6: Consider Extensions to the Mission Thread	15
4.7 Step 7: Discuss Overarching Quality Attribute Considerations	16
4.8 Step 8: Analyze Remaining Mission Threads	16
<b>5 Experience with the Mission Thread Workshop</b>	<b>17</b>
5.1 Mission Thread Workshops as a Series of Workshops	17
5.2 Additional Value Beyond Quality Attributes	17
5.3 Architectural Challenges	18
5.4 Considering Constituent Legacy Systems for an SoS	19
5.5 Applicability to Enterprise Architectures	19
<b>6 Observations and Conclusions</b>	<b>21</b>
<b>Appendix A Mission Thread Template – Augmentation Example</b>	<b>22</b>
<b>Appendix B Glossary</b>	<b>27</b>
<b>References</b>	<b>29</b>



---

## List of Figures

Figure 1:	Operational View of the Example Vignette for Ballistic Missile Defense	4
Figure 2:	Conceptual Flow of the Mission Thread Workshop	7



---

## List of Tables

Table 1:	Consequences of Failing to Address Quality Attributes Early	1
Table 2:	Operational Mission Threat for Ballistic Missile Defense	5
Table 3:	Examples of Quality Attribute, Capability, and Engineering Considerations	11





---

## Acknowledgments

Linda Northrop, James Ivers, Tamara Marshall-Keim, Phil Bianco, Rod Nord, and John Robert provided invaluable review for this document and contributed to making this a much-improved report from its initial draft.



---

## Abstract

In Department of Defense programs, a system of systems (SoS) is integrated to accomplish a number of missions that involve cooperation among individual systems. Understanding the activities conducted within each system and how they interoperate to accomplish the missions of the SoS is of vital importance. A mission thread is a sequence of end-to-end activities and events, given as a series of steps, that accomplish the execution of one or more capabilities that the SoS supports. However, listing the steps and describing them do not reveal all the important concerns associated with cooperation among the systems to accomplish the mission; understanding the architectural and engineering considerations associated with each mission thread is also essential. The Mission Thread Workshop (MTW) is a facilitated, stakeholder-centric workshop whose purpose is to elicit and refine end-to-end quality attribute, capability, and engineering considerations for SoS mission threads. This report introduces the MTW, describes the three phases of an MTW, and explains the steps of each phase in detail. It also describes the benefits that a program can expect from conducting an MTW, which include identifying quality attributes and architectural challenges but also reach beyond these goals to expose gaps in capability, functionality, documentation, and engineering.



---

# 1 Introduction

Just as they do in software architectures, quality attributes play a driving role in the architectures of both systems and systems of systems. Examples of quality attributes relevant to a system of systems (SoS) include performance, availability, reliability, security, usability, testability, safety, interoperability, maintainability, weight, power consumption, and hardware or logistic resource utilization. Failure to address quality attributes in the architecture early can lead to costly consequences, such as operational and developmental failures. Table 1 shows several examples of such failures.

Table 1: *Consequences of Failing to Address Quality Attributes Early*

<b>Operational Failures</b>	Communication bottlenecks under various load conditions throughout the SoS
	Systems within the SoS that hang up or crash; portions that need rebooting too often
	Difficulty synching up after periods of disconnect and resuming operations
	Judgment by users that a system within the SoS is unusable for a variety of reasons
	Database access that is sluggish and unpredictable
	Lack of stability in overload conditions
<b>Developmental Failures</b>	Integration schedule blown; difficulty identifying root causes of problems
	Proliferation of patches and workarounds during integration and test
	Integration of new capabilities taking longer than expected, triggering breaking points for various resources
	Significant operational problems ensuing despite passage of integration and test
	Anticipated reuse benefits not being realized

The Mission Thread Workshop (MTW) is a facilitated, stakeholder-centric workshop whose purpose is to elicit and refine end-to-end quality attribute, capability, and engineering considerations for SoS mission threads. The MTW also identifies architecturally significant SoS challenges, which are architecturally significant questions that are distilled from the architecture, engineering, and capability issues identified in a qualitative analysis of the augmented mission threads. The SoS challenges have the potential to turn into risks if they are not addressed during SoS architecture development. The augmented mission threads and SoS challenges serve as inputs to developing the SoS architecture, evaluating the SoS architecture and the constituent system and software architectures, and validating the SoS architecture against test cases. Examples of engineering considerations are unclear requirements, missing system-level use cases, hardware mismatches, or overlapping capabilities in different systems. The MTW is based on the principles of the Carnegie Mellon® Software Engineering Institute's (SEI) software architecture methods and practices, extended and scaled into the system and SoS domains. These principles include

- eliciting stakeholder inputs early in the life cycle
- articulating and addressing the quality attributes that drive the architecture early in the life cycle
- identifying challenges impacting architectural decisions early in the life cycle

Creating and developing a shared vision for how an SoS and its constituent systems are to function is a challenging task for the typically diverse group of stakeholders associated with the SoS. Obtaining their inputs and shaping them into a vision early in the life cycle will help to narrow the focus of the acquisition.

Once the stakeholders have a vision of the SoS, then prioritizing what quality attributes are important for the SoS and refining what they mean in the SoS's context will enhance the architects' understanding of what the SoS's architecture will need to support. With a vision for the SoS and its architecture, stakeholders can begin to consider and address tradeoffs (technical, funding, schedule, etc.) for the SoS as well as for its constituent systems. The tradeoff analysis efforts will lead to the identification of challenges that architects can investigate early in the life cycle before they potentially become risks associated with the SoS.

## 1.1 Background

In the early 2000s, the SEI developed a number of methods that when combined form the basis for a software architecture-centric engineering approach. At the software architecture level, it is well understood that design involves tradeoffs that key stakeholders need to discuss in a context that helps them understand the decisions they must make and enables them to provide inputs. The Quality Attribute Workshop (QAW) [Barbacci 2003] and the Architecture Tradeoff Analysis Method<sup>®</sup> (ATAM<sup>®</sup>) [Clements 2001] were two of the methods developed to facilitate the discussion among stakeholders at the software architecture level.

The QAW is a mechanism in which stakeholders develop quality attribute scenarios before a system's notional architecture has been developed, typically during the requirements phase. A quality attribute scenario is an extended use case that focuses on a quality attribute, such as performance, availability, security, maintainability, extensibility, or testability. It thus precisely defines a key requirement that will drive architectural decision making and the architecture's ability to support business and mission goals.

The focus of the ATAM is to identify risks associated with a software architecture as documented in a number of architecture views. It is usually conducted early in a system's life cycle. First, the evaluation team elicits quality attribute scenarios from the stakeholder for the system. Then participants prioritize the scenarios. The software architect walks the scenarios, one at a time, through the software architecture while the evaluation team and stakeholders ask probing questions about the architecture. The evaluation team identifies any risks, tradeoffs, and other considerations discovered during the evaluation.

When we considered extending this architecture-centric engineering approach to an SoS that could contain many geographically separated nodes, each of which could contain many systems, we realized that we needed to enhance the scenario-based model used for the QAW and ATAM in order to visualize and articulate an SoS's scale and context. We needed to do this in a way that enabled discussions among stakeholders from multiple engineering disciplines as well as the business side from an end-to-end perspective. We were familiar with a Department of Defense (DoD) program in which system engineers used the concept of mission threads, which are opera-

tional and technical descriptions of the end-to-end set of activities and systems that accomplish the execution of a mission [CJCSI 2009]. We conceived of extending this concept of a mission thread to consider quality attributes in the same manner as quality attribute scenarios had extended use cases. Based on this approach, we developed the Mission Thread Workshop (MTW). The MTW provides the same type of context setting for stakeholders at the system and SoS levels as the QAW provides at the software level. The MTW builds on the foundations created by the QAW.

## 1.2 Systems of Systems

SoS engineering is rapidly emerging as a discipline with which to address large-scale, high-complexity problems. The *Defense Acquisition Guidebook* defines an SoS as

*a set or arrangement of systems that results when independent and useful systems are integrated into a larger system that delivers unique capabilities. [DoD 2004]*

The individual systems are typically complex by themselves, so we need to take a more abstract view to consider an SoS. In this report, we use the term *node* to refer to systems as defined in the Department of Defense Architecture Framework (DoDAF) [DoD CIO 2007]. Some DoD examples of nodes, each of which contains many systems of varying technical age and maturity, are

- platforms of various ages and maturity: ships, aircraft, tanks, missile launchers, radars, trucks, and missiles
- command centers: air operations centers, tactical command centers, theater command centers, and command and control centers
- the communications systems connecting them: satellites, local area networks, wide area networks, and tactical line-of-sight radio communication

## 1.3 Vignettes

A *vignette* is a short story about the environment that provides the context in which the SoS exists. The purpose of a vignette is to describe those environmental factors that may be architecturally significant—that is, they impose a constraint on the architecture that would not exist in the absence of these environmental factors. This story can usually be told in a paragraph or two, with an accompanying context diagram showing the nodes containing the systems. Vignettes come in three flavors:

1. An *operational vignette* describes the geography, own force structure and mission, strategies and tactics, and the enemy forces and their attack strategies and tactics, including timing. For example, an operational vignette could describe a missile attack on a joint task force.
2. A *development vignette* describes the environment and assets involved in a development activity. For example, if software assets in the SoS are built as a software product line [SEI 2010a], a development vignette could describe how to use the software product line's core assets to develop software products suitable to serve in the SoS.
3. A *sustainment vignette* describes factors associated with a sustainment activity, the assets involved, and how the activity is conducted. For example, a sustainment vignette could de-

scribe the delivery of supplies to a ship at sea, delivery of fuel to a tank platoon, or a port visit by a ship to effect repairs.

Here is an example of an operational vignette for ballistic missile defense in a naval context:

*Two ships (Alpha and Beta) are assigned to air defense to protect a fleet containing two high-value assets. A surveillance aircraft and four unmanned aerial vehicles (UAVs; two pairs) are assigned to the fleet and controlled by the ships. A pair of UAVs flying as a constellation can provide fire-control quality tracks directly to the two ships. A two-pronged attack on the fleet occurs:*

- *five aircraft-launched missiles from the southeast*
  - *three minutes later, seven submarine-launched missiles from the southwest*
- The fleet is protected with no battle damage.*

Figure 1 shows the context diagram that accompanies this operational vignette.



Figure 1: Operational View of the Example Vignette for Ballistic Missile Defense

## 1.4 Mission Threads

A mission thread is a sequence of end-to-end activities and events that takes place to accomplish the execution of one or more of an SoS's capabilities. A mission thread takes place in the context defined by a vignette. By "end to end," we mean that the sequence starts with some input or stimulus from outside the SoS and concludes with the SoS producing the required response to it,



which may also be visible from outside the system. We identify three basic types of mission threads:

1. An *operational mission thread* describes how the SoS nodes (and perhaps the systems within the nodes) react to an operational stimulus. It is given as an end-to-end sequence of steps (external events, operator activities, and automated activities) that take place over a time period. For example, an operational mission thread for a DoD command and control system might begin with threat detection and then list a number of steps to determine the intent of the threat, make decisions to counter the threat, apply the counter measures, and finally document the commander’s assessment of damage after completion.
2. A *development mission thread* focuses on development activities, including adding new capabilities, technology refreshment, integration, test, certification, and release.
3. A *sustainment mission thread* focuses on deployment, installation, sustainment, or maintenance. A sustainment mission thread describes how the SoS nodes operate together to sustain the SoS.

Mission threads typically have 10 to 25 steps and include the major activities needed to execute the specific mission.

Table 2 provides an example mission thread supporting the ballistic missile defense vignette.

Table 2: *Operational Mission Thread for Ballistic Missile Defense*

Steps	Description
1	Alpha develops the air defense plan (ADP) and rules of engagement (ROE) and sends them to Beta. The plan assigns to Alpha the area of regard (AoR) to the west, and Beta the AoR to the east. Alpha configures surveillance and weapons systems to support eastern engagements.
2	The Situational Awareness (SA) aircraft detects that the five enemy aircraft have changed course and are heading toward the fleet at low altitude.
3	SA informs both Alpha and Beta of the change.
4	Alpha (and Beta) go to General Quarters.
5	SA detects that missiles have separated from the enemy aircraft and informs Alpha and Beta.
6	Alpha assigns its two UAVs to track the missiles.
7	The two Alpha-controlled UAVs send Fire Control Quality (FCQ) tracks for the five missiles to both Alpha and Beta.
8	Alpha assigns three missile engagements to itself and two to Beta.
9	Alpha receives FCQ tracks (continuously) for the missiles and initiates engagements for three missiles.
10	Both Alpha and Beta launch two AD missiles.
11	SA detects submarine launch of missiles from the southwest.
12	SA informs both Alpha and Beta.
13	Alpha commands Beta to use its two UAVs to track these submarine missiles.
14	Alpha launches a third AD missile against the western attack.
15	Alpha assigns itself three of the seven missiles and Beta the other four.
16	Alpha launches two missiles against the southern attack.
17	Analysis of UAV track data shows that one western missile was not killed.
18	Alpha picks up the missile on its own radar.

19	Alpha launches another missile at the "leaker."
20	Ships' radar confirms that the "leaker" was killed.

## 1.5 Quality Attributes

Quality attributes drive many important architectural decisions. This is true in software architecture as well as system and SoS architecture development. SoS architects need to address some quality attribute considerations on a per-step basis within the mission thread and others from an overarching perspective for the mission thread. The MTW allows facilitators to guide stakeholders through augmenting the mission thread with quality attribute considerations during the workshop, whether it is on a per-step basis or an overarching basis for each quality attribute.

For example, an SoS architect needs to consider that many measures of performance crosscut more than one step. If an incoming missile must be intercepted within  $x$  minutes, then each step of the mission thread must be augmented with performance information and the sum of the times at each step—including any manual decision making, the inherent communication delays, the computational times, and the flight time of intercept missiles—must take less than  $x$  minutes. If any of the involved systems within any of the nodes fail, then the recovery time must also take place within the  $x$  minutes. We refer to such crosscutting issues as *overarching quality attribute considerations* for the entire mission thread. Continuing the example, architects need to consider the number of ways that the system could become overloaded such that it cannot meet the  $x$ -minute response time. Manual operations could overload at  $k$  incoming missiles; the intercept planning mechanism could overload at  $l$  incoming missiles; the communication network could overload at  $m$  incoming missiles; and the missile tracking mechanism could overload at  $n$  incoming and intercept missiles.

The quality attributes for different types of vignettes and mission threads vary. Some examples are shown below:

- Operational: performance, availability, security, interoperability, usability, accuracy
- Development: reusability, openness, modifiability, testability
- Sustainment: maintainability, extensibility, deployability, instrumented to support training

Appendix A contains an example of a completed mission thread augmentation.

## 2 A Brief Introduction to the Mission Thread Workshop

Figure 2 depicts the conceptual flow of the MTW. SoS drivers and capabilities inform the development of vignettes and mission threads from which a set of SoS quality attributes are derived. Plans for SoS architecture development include an initial set of SoS architecture views that support the vignettes and mission threads. Constituent legacy systems are identified for consideration in the SoS architecture, based on the SoS capabilities and mission threads. The MTW team uses these inputs to augment the mission threads with quality attribute considerations, as well as engineering and capability considerations, with participation from SoS and legacy system stakeholders. The other outputs from the MTW are the SoS challenges derived from the issues identified in the qualitative analysis.

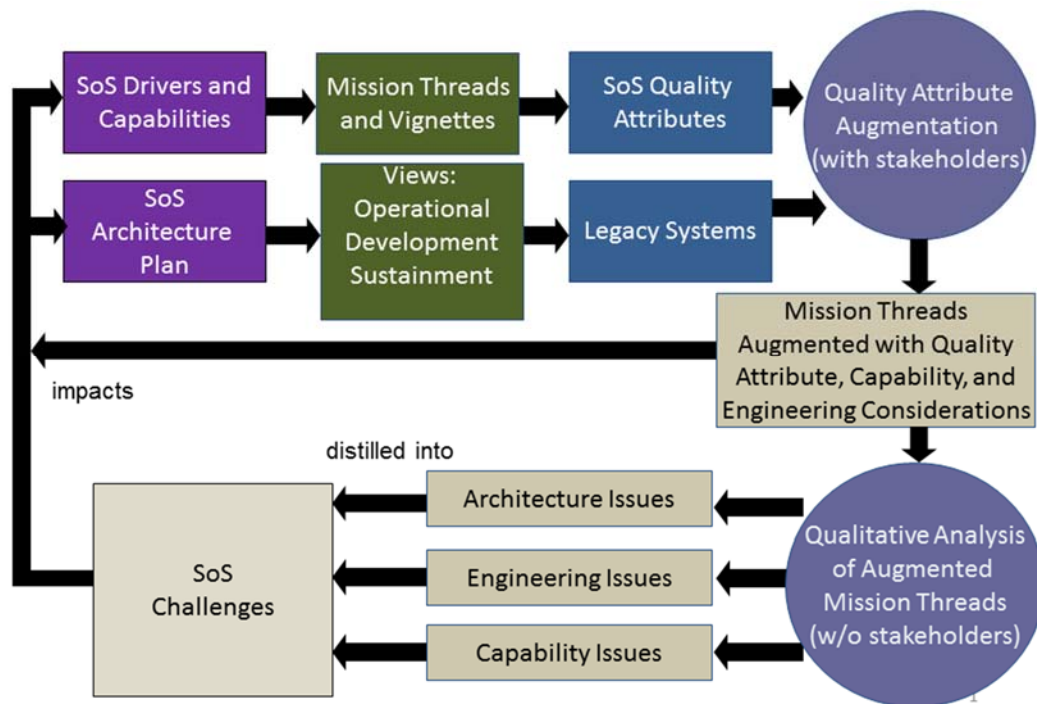


Figure 2: Conceptual Flow of the Mission Thread Workshop

Stakeholder participation is essential to the success of an MTW. To make the best use of stakeholders' time in the MTW (as detailed in Section 4, Steps 4–7), the preparation work for an MTW is critical to provide relevant context and meaningful mission threads that address the capabilities expected of the SoS.

---

## 3 The Phases of the Mission Thread Workshop

This section describes the three phases of the MTW and the time frame in which to do them to support successful engagements. The goal of the MTW is to elicit and capture quality attribute and engineering considerations for SoS mission threads from the SoS stakeholders and to identify SoS challenges (architecture, engineering and capability) early in the SoS development life cycle. The augmented mission threads and SoS challenges will drive the successful development of the SoS architecture and inform decisions relating to constituent legacy systems in the SoS.

The participation of key stakeholders is essential to the success of an MTW. These people are typically extremely busy due to their importance to the SoS being developed. To make the best use of their time in the MTW, the Preparation Phase is critical to provide relevant context and meaningful mission threads that address the capabilities expected of the SoS. To assure that the MTW uses stakeholders' time wisely, we developed the following engagement approach, which consists of three phases:

1. Preparation Phase
2. Execution Phase
3. Follow-On Phase

### 3.1 Preparation Phase

In the Preparation Phase, the MTW lead works with SoS representatives—program management, architect(s), and capability subject-matter experts (SMEs)—to develop and collect the artifacts identified in the following list of steps.

We have numbered the activities performed during the Preparation Phase, but the SoS representatives will work through many of these steps in parallel.

1. **Review the MTW process.** The MTW lead provides an overview of the process and examples of the artifacts developed to support an MTW to the SoS program management representative and SoS architect. The MTW lead works with the SoS representatives to create a timeline that identifies target dates for developing key artifacts that the MTW lead will provide to the stakeholders before the workshop.
2. **Develop SoS mission and business drivers.** The SoS program management representative develops the driver information, including key functional requirements and constraints, plan for development, and business and programmatic context, and provides it to the MTW lead for review and feedback. Based on the feedback, the SoS representatives will update the driver information and create two or three slides to present during the workshop. The presentation will give MTW participants an overview of the purpose of the SoS and the important business drivers and quality attributes that will shape its architecture.
3. **Develop the SoS architecture plan.** The SoS architect develops the SoS architecture plan, consisting of key technical requirements and constraints, existing context, and SoS diagrams

and descriptions, with feedback given by the MTW team. The SoS architect will then create two or three slides to use at the workshop. This presentation will share with the MTW participants the technical considerations, constraints, and drivers that affect the development of the SoS architecture. The information will focus on the vignette and mission thread capabilities that were selected in Step 4. It includes only those nodes and capabilities expected to be involved in the threads. For example, if the capability being explored is “defense of the fleet against land-launched ballistic missiles,” then nodes and capabilities involved with anti-submarine warfare would not be included, though participants could explore this as a second capability.

4. **Define the MTW scope.** The MTW team meets with the SoS representatives to scope the effort:
  - Determine the number of vignettes and mission threads to address in the MTW.
  - Develop a graphical representation and description of the vignettes. The graphical representation should show the nodes and forces involved and their relationships (for example, an OV-1 or context diagram). The description should outline the operational considerations, the nodes and actors involved, and the assumptions that participants have made about the context.
  - Develop the mission thread steps.
  - Identify the relevant quality attributes for the SoS. For example, an operational thread will often consist of performance, availability, security, scalability, and usability elements.

This step enables the method to scale to meet the SoS representatives’ priorities and funding. The vignettes, mission threads, context diagrams, and identified quality attributes will be the primary inputs to the next phase of the MTW, where stakeholders will augment them to the degree necessary to uncover architectural challenges.

5. **Identify participating stakeholders.** It is critical to identify key stakeholder roles whose participation is essential to this particular workshop. As in any stakeholder-focused activity, the experience of the stakeholders in attendance largely determines the quality of the results. A key outcome of the Preparation Phase is insuring the participation of key stakeholders. Examples of stakeholders whom we often seek to involve in an MTW include the following:
  - operational commanders and SMEs associated with the force structure and the capability that the MTW will explore
  - modeling and simulation experts who have explored the capability
  - integration and test facilities staff
  - Concept of Operations (CONOPS) operational analysts
  - SoS, system, and software architects, including architects of legacy systems incorporated into this SoS
  - logistics and sustainment staff
6. **Select the MTW team.** The MTW team consists of three or more people who perform the four roles of lead, facilitator, scribe, and analyst.

- **MTW lead:** The lead plans and executes the MTW; he or she may also take one of the other roles during the workshop.
- **MTW facilitator:** This person facilitates the augmentation of the mission threads and the related discussions during the Execution Phase based on the steps in Section 4.
- **MTW scribe:** The scribe documents the discussions that occur during the workshop and keeps track of how the time is spent during the Execution Phase.
- **MTW analyst:** One or more analysts listen to the discussion and interject to address quality attribute or engineering considerations that participants have not addressed or that could use further clarification.

We recommend that each member of an MTW team be an SEI-authorized ATAM Evaluator [SEI 2010b] and the MTW lead be an SEI-certified ATAM Leader [SEI 2012], but having experienced architects with good facilitation skills and quality attribute knowledge would be sufficient. In some instances, there will be no analysts who have appropriate experience in all the capabilities that stakeholders envision for the SoS. In these cases, the MTW lead should add SMEs (independent of the program) to the team to provide the missing expertise.

7. **Settle on logistics.** Where and when will the MTW be held? What clearances will be necessary? How will the participants enter the facility? Will the facility provide meals for them? Will only on-site stakeholders participate, or does the facility have a network or telecom to support remote personnel? The MTW team and SoS representative will work out these and other necessary details during the Preparation Phase.

The Preparation Phase is carried out via informal interactions between the MTW lead and the SoS representatives. They can handle these interactions by whatever combination of telephone, email, and face-to-face communication the parties find most effective and convenient.

At the completion of the Preparation Phase, the MTW team and SoS representatives have produced the following outputs:

- one or more vignettes, each with one or more mission threads
- SoS mission and business driver presentation
- SoS architecture plan presentation
- identified key stakeholders and their role and availability for the MTW
- MTW invitation letter
- selection of MTW team
- finalized workshop logistics
- package of MTW preparation material for MTW stakeholders

### 3.2 Execution Phase

The Execution Phase focuses on augmenting the mission threads with capability, engineering, and quality attribute considerations based on stakeholder inputs and the dialogue between the stakeholders and architects. The workshop will occur over 1 to 1.5 days, based on the number of mis-

sion threads that the program would like to augment. During each step, the facilitator guides the discussion, and the scribe documents the augmentations and any issues that arise.

The MTW team will use the mission thread template during this phase to capture all of the augmentations and issues that arise during the workshop for a particular mission thread. The template has three main parts:

1. Header: Contains the vignette description, nodes, actors and assumptions for this mission thread.
2. Steps: Contains the sequence of end-to-end activities and events that involve the realization of one or more capabilities of the SoS.<sup>1</sup> These steps also address activities that involve the interfaces with systems external to the SoS.
3. Overarching Quality Attributes: Contains the quality attributes that participants will consider from an all-encompassing point of view during discussion of the mission thread.

Each part of the template contains space to include quality attribute and engineering considerations and issues that arise during the workshop. Table 3 shows some sample considerations pulled from Appendix A: Mission Thread Template – Augmentation Example.

*Table 3: Examples of Quality Attribute, Capability, and Engineering Considerations*

Type of Consideration	Example
Further clarification of mission thread step and associated assumptions	“The enemy aircraft are within the AoR (area of responsibility) of the SA (situational awareness) sensors. The SA has been tracking these aircraft and sending tracks to the Alpha and Beta.”
Performance	“Within X seconds.”
Missing functional capability	“Need a use case on assigning the UAVs to track the aircraft at this point.”
Issue	“Current DES does not handle more than two engagements simultaneously for AD (air defense).”

The MTW facilitator and scribe will capture these considerations in the appropriate part of the template. In the next phase of the MTW, the team will use these augmentations to develop SoS architectural and engineering challenges.

Section 4 of this report describes the steps in the Execution Phase in more detail.

### 3.3 Follow-on Phase

The Follow-on Phase produces two outputs based on the efforts in the Execution Phase; the MTW team provides the augmented mission threads within days of the workshop completion and the list of SoS challenges within a couple of weeks.

The MTW team begins to analyze the data collected during the workshop after the completion of the Execution Phase. The first task is to “scrub” the augmented mission threads. The information that the scribe documents during the Execution Phase is raw information. After the MTW, the

---

<sup>1</sup> Each step in a mission thread identifies an action or operation that occurs at the SoS level. When the SoS architecture is broken down into constituent systems, components, and subsystems, a single step could significantly expand into a large number of actions and operations that affect one or more constituent systems.

scribe will expand that raw information into sentences and bullets that make sense and reflect the information discussed. After scrubbing the comments section in the template, the scribe will reference each remaining comment by a unique identifier. These identifiers support the next phase of the analysis by allowing the MTW team and stakeholders to reference the comments easily and specifically. The MTW team provides these updated versions of the augmented mission threads to the client stakeholders to enable them to start their own analysis efforts within days after the workshop.

The MTW team then analyzes the updated mission threads and begins to identify architecture, engineering, capability, and quality attribute issues. Then they group these issues into challenges. The MTW team documents each challenge by listing its source in a table that maps challenges to contributing steps of the augmented mission thread, the overarching quality attributes, and the vignette. To produce the list of challenges, the MTW team

- reviews the steps one by one and uses qualitative analysis to form a list of potential challenges, noting which steps contribute to which challenges in a cross-referencing table
- reviews the overarching quality attribute section at the bottom of the template and inserts any additional potential challenges
- reviews the potential challenges and combines similar challenges to reduce the number (typically to about five to seven), changing the cross-references to the contributing steps accordingly

We use the term *challenge* to signify potential problems with achieving a capability based on the proposed architecture and the quality attributes' impact on the architecture. These potential problems could become architectural risks if a program does not perform additional investigation to assess and correctly address them. These challenges are not usually considered risks at the time of the MTW, because the mission threads provide a vision for implementing a capability and developing the SoS's architecture. There is no actual problem because these decisions usually have not been made yet. If a program is participating in an MTW to gain a better understanding of and improve architecture documentation for a legacy SoS, then the challenges should be considered potential risks.

The MTW team then distills the remaining challenges into a briefing. This briefing will include the table of combined challenges, the description of each challenge, the impact of each challenge on mission and business goals, and a set of recommendations for addressing the challenge. The MTW team reviews the briefing with the SoS leadership, makes appropriate changes to clarify the issues and correct any misunderstandings of the MTW team, and then delivers a final briefing. The SoS representatives should use these challenges as inputs into their program's risk management process, and the program can decide if they accept any resulting risks and how they will mitigate them. A possible mitigation step could be to perform an architecture evaluation of the SoS or system.



---

## 4 The Steps of the Execution Phase of the Mission Thread Workshop

This section breaks down the process used by the MTW team during the Execution Phase into eight steps. These steps describe the activities and stakeholder interactions that occur during the execution phase of the workshop.

### 4.1 Step 1: Present the Mission Thread Workshop

An SoS program representative sets the context for staging the MTW. Then the MTW facilitator describes the MTW technique to the assembled stakeholders (typically system engineers, system architects, software architects, program managers, requirements engineers, etc.). The presentation should take 5 to 10 minutes and include

- the workshop process
- the use of vignettes and mission threads to elicit their participation in the discussions
- the MTW team and their roles during the MTW
- the outputs from the MTW (augmented mission threads and SoS challenges)

The facilitator uses this time to explain the technique, provide the stakeholders with an opportunity to ask questions about the technique, lay the ground rules, and set expectations for the workshop. The facilitator also informs the stakeholders that the MTW team will strongly facilitate and document the workshop, so that they can focus on the discussion.

### 4.2 Step 2: Present the Business and Mission Drivers

An SoS program representative presents the SoS business and mission drivers, including the business and programmatic context, high-level functional requirements, high-level constraints, high-level quality attribute requirements, and the plan for development. During the presentation, the facilitator gives the stakeholders and the MTW team opportunities to ask questions related to the business and mission drivers. The presentation could take up to 30 minutes, depending on the number of questions that arise, but the facilitator should ensure that it does not exceed this time.

Ideally, business and mission drivers already exist within the program, but we have experienced several cases in which the MTW team needed to develop these drivers because the program had not yet addressed them. In these cases, the MTW team provides examples from similar programs and works with the program to develop drivers that satisfy their acquisition and technical guidance. Example business and mission drivers include

1. affordable in design, production, and acquisition throughout the life cycle
2. capable and survivable from delivery to end of life
3. producible, maintainable, and flexible over the SoS's life cycle

4. componentized combat system architecture and common information standards to achieve a documented, open architecture vision
5. an established product line approach based on a common objective architecture

### **4.3 Step 3: Present the Architectural Plan**

The SoS architect presents the architecture development plan, including key business and programmatic requirements, key technical requirements and constraints that will drive architectural decisions, existing context diagrams, high-level SoS diagrams and descriptions, SoS quality attributes, development spirals, and the integration schedule. The goal of this presentation is to provide the architect's vision for the SoS based on currently available documentation. Stakeholders and the MTW team can ask questions dealing with the development plan. The presentation could take up to 30 minutes, but the facilitator should ensure that it does not exceed this time frame. Sometimes relevant architecture discussions cause this step to go beyond the recommended 30 minutes, and the facilitator determines whether to let this occur or to move to the next step.

### **4.4 Step 4: Review the Mission Thread Header**

The SoS architect or a representative from the SoS stakeholder community presents the header information, which consists of (1) the name of the mission thread, (2) the vignette, (3) the nodes and actors, and (4) the assumptions and then the first mission thread that participants will augment in the MTW. The vignette provides the context for the mission thread. The goal is to help the stakeholders understand the environment in which the mission thread steps will occur. In DoD programs, the vignette is based on the program's OV-1 views, which are high-level graphical and textual descriptions of the operational concept, and describes the main resources and organizations involved. Reviewing the vignette typically takes 5 to 10 minutes, and all participants have an opportunity to ask questions about the vignette and context diagram.

Due to the scale of an SoS, the MTW participants should narrow the vignette's context information with a set of assumptions that focus on the environment in which the mission will take place. These assumptions typically instigate a lot of participant discussion as everyone tries to attain a consistent vision for the mission thread. The MTW participants should document an initial set of assumptions for the mission thread before the MTW. For example, some mission threads have used the following assumptions:

- It is not a wartime situation; ships are at Battle Condition 3.
- Sea state is 3.
- Ships' readiness condition is YOKE.

The MTW facilitator starts by reviewing the assumptions, asking if the stakeholders have any questions, additions, or clarifications. The facilitator should allow 5 to 10 minutes for this effort, because during presentation of each mission thread step, the participants will have an opportunity to identify additional assumptions that help clarify the mission thread. The facilitator next leads a short discussion that addresses the nodes and actors for this mission thread, for participants' understanding and clarification. This is typically the first time that the facilitator will need to keep

the stakeholders focused and limit off-subject discussions as the participants work on “normalizing” what the nodes and actors are in their vision of the SoS.

The MTW facilitator will redirect detailed questions about the mission thread assumptions, steps, and possible extensions to Step 5.

#### **4.5 Step 5: Augment the Mission Thread Steps**

The facilitator proceeds to walk through and discuss each step of the mission thread with the stakeholders until they have covered all of its steps. The goal is to spend no more than two hours on each mission thread while performing Steps 5 through 7, but typically the first mission thread will take longer while the stakeholders learn the technique.

Each step in a mission thread identifies an action or operation that occurs at the SoS level. When workshop participants further break down the SoS architecture into constituent systems, components, and subsystems, a single step could significantly expand into a large number of actions and operations that affect one or more constituent systems. The participants would then apply the quality attribute considerations identified for the step as overarching quality attribute considerations for the actions and operations in the lower level constituent systems, components, and subsystems. Overarching quality attributes can also be used to refine or identify new use cases for the constituent systems, as well as to derive quality attribute scenarios that can be used to evaluate the constituent system and software architectures, in the context of the SoS architecture. Sometimes workshop participants will raise a consideration that relates to an overarching quality attribute in the mission thread; if so, the scribe should capture it in the overarching quality attribute section during this step. This enables the focus of the discussions to remain on the individual steps of the mission thread. When the workshop proceeds to Step 7, then stakeholders will have an opportunity to discuss the comment.

Typical discussions around a step include requirement elicitation and clarification; identification of architecture or engineering activities or constraints that concern the stakeholder, new use cases, and capability gaps; and tradeoffs among competing quality attributes and the impacts to the SoS. Stakeholders are encouraged to ask questions and raise issues for each step, while the facilitator keeps the discussions on track and in scope. An MTW scribe captures the relevant points from the discussions for each step and documents them in the MTW template. The scribe’s role is a challenging one because he or she must listen to the discussion and document it while the discussion continues.

Often, participants will uncover the same discussion topic or issue that they noted in a previous step. The MTW facilitator may reference the discussion issue in this step and move on for the sake of time.

#### **4.6 Step 6: Consider Extensions to the Mission Thread**

During the stakeholder discussion, the group may decide that a mission thread step needs extensions to enable future analysis of different aspects of the thread. The focus during the workshop is on discussing each step and not on spending a lot of time considering alternative paths. Typically,

we have found that these extensions address system use cases that the SoS representatives had not previously considered but that the SoS architecture team should investigate as a part of the remaining analysis for the SoS. The scribe documents the extensions for future use, and the facilitator returns the discussion to the mission thread steps.

#### **4.7 Step 7: Discuss Overarching Quality Attribute Considerations**

Once workshop participants have considered all the steps of the mission thread, the MTW facilitator leads the stakeholders in a discussion about each identified quality attribute associated with the mission thread from the perspective of the entire mission thread. Up to this point, quality attributes have been considered on a per step level, but it is also important to consider the quality attributes from an end-to-end perspective. This places the focus on the one or more capabilities described in the mission thread at a higher level than the systems and components that make up the SoS. The MTW facilitator ensures that stakeholders discuss each overarching quality attribute and that the MTW scribe captures all issues and concerns in the MTW template. The MTW facilitator elicits discussions from the stakeholders until they have covered the entire set of quality attributes. Sometimes the group will capture a consideration that relates to a specific step in the mission thread; it is fine to go back and update that thread augmentation at this time.

#### **4.8 Step 8: Analyze Remaining Mission Threads**

The MTW facilitator will repeat Steps 4–7 for the remaining vignettes and mission threads to be covered in the workshop. Typically, we have found that the process of analyzing the mission threads speeds up after the stakeholders become familiar with the MTW process and the context for the mission threads. In our experiences, we can typically augment four to five mission threads in a two-day MTW. Also, if major issues and challenges arise that were previously captured in the MTW, the scribe can reference them at the step in which they emerge in the mission thread while the participants choose to discuss only new considerations on that issue in further detail, in order to use the stakeholders' time effectively.

---

## 5 Experience with the Mission Thread Workshop

### 5.1 Mission Thread Workshops as a Series of Workshops

Due to the size and scale of most systems of systems, a wide variety of mission threads may require consideration, and each may traverse different paths through the constituent nodes and systems. Our experience has shown that up to five mission threads can be augmented in a single two-day MTW. This means that a program may need a series of MTWs to support the SoS. We advise against mixing operational, development, and sustainment threads in a single MTW, as each type of mission thread tends to require different stakeholders. Usually we recommend conducting a series of MTWs that focus on well-defined issues associated with the SoS. For example, operational issues in a DoD naval warfare domain could pertain to air defense, anti-submarine warfare, land attack warfare, and surface ship warfare. In this case, we recommend a separate MTW for each area. The MTW lead and SoS representatives must match stakeholders for each MTW to the mission threads involved.

When running a series of workshops, the MTW lead meets with the SoS architect and management to plan a series of MTWs to cover the necessary operational, development, and sustainment threads for each SoS area chosen for analysis. Preparatory to the series of workshops, the MTW lead works with each area lead to develop one or more vignettes that will apply to that area. The number of vignettes needed depends on the scope, scale, and schedule considerations of the SoS. Criteria for selection and development of vignettes include

- capability and new requirements coverage
- stressing the SoS and its constituent systems
- integration of existing capabilities to deliver a new capability

For a series of MTWs, some of the steps of the Preparation Phase described in Section 4.1 can be factored out of planning individual MTWs and dispatched for the whole workshop series. These steps include developing the vignettes, staffing the MTW team, identifying quality attributes of importance to the SoS, and planning for various items involved in logistics for the workshops.

### 5.2 Additional Value Beyond Quality Attributes

Although the emphasis in MTWs is firmly on quality attributes, the approach often exposes gaps and overlaps in capability, functionality, and engineering issues. The following are some examples from MTWs for DoD programs:

- A mission thread dealing with compartmental flooding in a critical compartment led us to discuss new pump technologies to avoid flooding.
- A mission thread dealing with failure of a generator led us to discuss its broad impact on all ship operations and missions and the need for increased automation.
- A mission thread dealing with aircraft tracking identified two systems with overlapping capabilities.

The augmented mission threads developed during the MTW can also be used to identify SoS capability requirements and quality attribute expectations for an undocumented SoS, as well as for capturing SoS capability upgrades and identifying architectural challenges for the upgrades.

In addition, we have found that mission threads allow stakeholders from different operational areas and segments to relate to each other's concerns in ways that benefit the SoS. MTWs encourage stakeholders from diverse engineering disciplines to actively participate and contribute collaboratively, and using this approach can break down organizational stovepipes. For example, in a DoD naval program, aviation was a segregated segment whose engineering and architectural considerations were not being addressed prior to the MTWs. But it became an integral part of combat systems considerations after the MTWs, due to the gaps identified in a number of mission threads.

### 5.3 Architectural Challenges

Architectural challenges resulting from the engineering considerations and quality attribute augmentations identified during MTWs are a key output of the MTW. The challenges provide useful input to downstream engineering and architecting activities. For example, the augmented mission threads establish a framework for the next level of architecture development and decomposition. The SoS architects can identify one or more system-level use cases from a mission thread step and its extensions. Architects may also need to consider and address in their development efforts a number of questions documented in the comments section of the mission threads. Developers can use the augmented mission threads to scope the SoS program's planning and testing efforts, and the mission threads provide one of the first reference points for these activities.

For each challenge, the MTW analysis includes a description of the challenge and pointers to the contributing mission thread steps, the SoS's impacted areas if the challenge is not addressed during the design phase, and recommendations for mitigating the challenge. Two examples follow:

#### 1. Training/Certification

Description: Providing the training and certification to support the mission flexibility needed for surface warfare

Impact areas: support, human systems integration, combat systems

Recommendation:

- Perform training/certification to support the integration of all available sensors to develop and maintain the tactical picture.
- Perform training/certification to support the use of aviation assets in surface warfare.
- Perform training to support weapons' readiness states for the envisioned engagements, with their short timelines.

#### 2. Manning/automation

Description: Analyzing the tradeoffs of manning versus automation to support the mission flexibility needs

Impact areas: support, human systems integration, combat systems, aviation

Recommendation:

- Analyze the ability to support sustained aviation operations based on the assumption that the ship will support one helicopter and two UAVs.
- Review the sensor and weapons coverage areas for potential automation efforts to address short engagement timelines.
- Study de-confliction in multi-mission scenarios to assess manning levels necessary to provide effective support.

#### **5.4 Considering Constituent Legacy Systems for an SoS**

The MTW analysis of an SoS's challenges may reveal that one or more constituent legacy systems pose potential problems in the SoS context and warrant further investigation. In some cases, the architecture documentation for the legacy systems does not exist or has not been kept current as the system has been updated over the years. The programs that support these legacy systems have experience operating and maintaining them, so they can use the mission threads to further explore the potential problems and help develop documentation about their system.

An effective approach is to start with the augmented SoS mission threads that involve the legacy system. Focus on the mission threads steps where the interactions occur, and drill down to the next level in these steps. MTW participants can enhance and expand the context and assumptions from the SoS mission threads to focus on the legacy system being investigated. Participants can break these mission thread steps into sub-steps that describe how they expect the legacy system to function and include information about interfaces, operations, indications, modes, and other details. This additional information could form the start of a set of use cases that describe these functions, and it will assist with developing architectural documentation to support the effort.

With the information gleaned from the MTW, the program may realize additional benefit from also executing a QAW or an ATAM on the legacy system. In cases where a constituent legacy system's documentation is not current or the program is not experienced with the operational features of the system, a QAW would help the program stakeholders develop quality attribute scenarios to understand what quality attributes are important in the legacy systems. An ATAM would be more effective in situations where stakeholders understand the constituent system well and have current documentation. The SoS mission threads provide a framework for performing these types of investigations to help identify potential risks to the SoS.

#### **5.5 Applicability to Enterprise Architectures**

The concepts of the MTW apply well outside of the SoS context in which they were developed. Enterprise architecture is a good example. An enterprise architecture is the architecture of a company's enterprise that focuses on the business processes, technologies, and information system. It includes the logical organization of business functions, business capabilities, business processes, people, information resources, business systems, software applications, computing capabilities, information exchange, and communications infrastructure within the enterprise. A company uses its enterprise architecture to discuss the different systems within the enterprise and how they are integrated together to support the business. The architecture provides a way to investigate how to

improve business workflows or process efficiencies. The enterprise architecture for a company, especially one that has grown through acquisitions, is similar in nature to an SoS.

The MTW method and its associated artifacts (vignette, context diagram, mission threads, SoS drivers and capabilities, and SoS architecture plan) are easily mapped to support these efforts for a company. Business workflows are like mission threads that stakeholders can discuss, using supporting context diagrams and vignettes that represent the company's business environment. The workflows are the missions of the enterprise, and stakeholders can augment them to address the quality attributes identified in the enterprise's business drivers and information technology plans.

Following the MTW method, stakeholders can use the company's primary business workflows to identify capability gaps in their current operations or to consider the implications of adding additional capabilities to their enterprise. The method enables the company's staff to identify, consider, and develop a roadmap for investigating the architectural tradeoffs documented during the MTW. The MTW method provides a way to reduce potential risks for a company in a lightweight, qualitative manner.



---

## 6 Observations and Conclusions

After performing over 30 MTWs, we have observed several trends. First, as in all stakeholder-focused methods, buy-in from the principals and the stakeholders is critical to the success of the approach. Sometimes stakeholders were initially skeptical about our qualifications in their domain, but as the MTW process unfolded and participants recognized that our primary role was to facilitate their contributions, that skepticism vanished. To encourage buy-in, the SEI's MTW teams often developed an initial thread in the stakeholders' domain for review and discussion and conducted a teleconference with the government lead to walk through the steps and make changes. In one case, the government convened a small team to meet and validate the thread, making changes as suggested. In all cases, we have been able to convince the government lead to proceed with the MTW.

Strong third-party facilitation has been critical to the success of the approach across MTWs. It helps squelch local biases dominating SoS-level concerns. Challenges identified by consensus under the auspices of independent third-party facilitation tend to benefit from higher group buy-in than do challenges identified by various factions among SoS stakeholders. These group decisions will more likely influence the architecture and development of the SoS. We also discovered that the architectural challenges identified in MTWs have resulted in updates to the SoS CONOPS and Mission Needs Statements.

Although third-party facilitation is important, motivated programs can adopt the MTW process for their own use. For example, after participating in a few MTWs facilitated by the SEI, a DoD naval program was able to successfully execute its own MTWs.

SoS architecture principles and guidelines were often absent when we began an engagement. Documenting the principles and guidelines for the SoS architecture is beneficial to drive and constrain the architecture development process for both SoS and constituent systems, including legacy modifications. We found that the MTW approach feeds the development of these principles and guidelines. With proper scoping and selection of vignettes and mission threads, programs easily can attain a high percentage of coverage of their SoS's envisioned capability, which can provide a focused starting point for the architecture development effort.

Overall, our experience in executing a number of MTWs for a variety of clients has proven to be very positive. The clients all gained insight into the desired behavior of their SoS and constituent systems; identified architectural challenges, engineering challenges, and capability gaps; and defined architecture guidelines and principles to guide their development. These outcomes can help a program avoid the costly consequences of development and operational failures and lead to more cost-effective and on-time delivery of new capabilities for the SoS.

## Appendix A Mission Thread Template – Augmentation Example

We present an example of a mission thread template filled in during an MTW to provide guidance on the size and level of detail in a mission thread. The information that is not in italics would be developed by the MTW team and SoS representatives in the Preparation Phase. The information in italics denotes information elicited from the stakeholders during the Execution Phase.

### Mission Thread Template: Header

<b>Name</b>	Protect Fleet Assets Against Cruise Missile Attacks
<b>Vignette (Summary Description)</b>	<p>Two ships (Alpha and Beta) are assigned to air defense (AD) to protect a fleet containing two high-value assets (HVA). A surveillance aircraft (SA) and 4 UAVs (2 pairs) are assigned to the fleet and controlled by the ships (Alpha and Beta). A pair of UAVs flying as a constellation can provide fire-control quality (FCQ) tracks directly to the two ships. A two-pronged attack on the fleet occurs:</p> <ul style="list-style-type: none"> <li>• 5 aircraft launch missiles from the southeast</li> <li>• 3 minutes later, 7 submarines launch missiles from the southwest</li> </ul> <p>The fleet is protected with no battle damage.</p>
<b>Nodes and Actors</b>	Two ships (Alpha and Beta), 4 UAVs, 2 HVAs, 1 SA, 5 enemy aircraft and their missiles, and 7 enemy submarines and their missiles
<b>Assumptions</b>	<p>Enemy aircraft are flying along a route normally used for training, and suddenly change direction and head for the fleet. They are being tracked.</p> <p>The submarines are undetectable until they fire their missiles.</p> <ul style="list-style-type: none"> <li>• <i>No sonobouys are deployed, but they could be in a new vignette.</i></li> </ul> <p>The vignette is not concerned with counterattacking the enemy aircraft or submarines. It is not a wartime situation.</p> <p>Sea State 3.</p> <p>Ships readiness condition is YOKE.</p> <p>Alpha controls 2 UAVs and Beta 2 other UAVs.</p> <ul style="list-style-type: none"> <li>• <i>Each ship has two organic UAVs.</i></li> </ul> <p>During normal operations, the UAVs have separate, non-overlapping areas of regard (AoRs).</p> <p><i>Alpha ship's helo is in the air.</i></p> <p><i>The SA has an area of regard that will detect both the launched missiles.</i></p> <p><i>The Air Defense Commander (ADC) is on board Alpha.</i></p> <p><i>Both ships are aware that a potentially hostile country has some fighter aircraft conducting training missions nearby.</i></p>

### Mission Thread Template: Steps

Steps	Description	Quality Attribute, Capability, and Engineering Considerations
1	Alpha develops the air defense plan (ADP) and Rules of Engagement (ROE) and sends them to Beta. The plan assigns to Alpha the AoR to the west, and Beta the AoR to the east. Alpha configures surveillance and weapons systems to support eastern engagements.	<ol style="list-style-type: none"> <li>1. <i>How much is predefined and how much is done manually?</i></li> <li>2. <i>ROE dictates a "Shoot-Look-Shoot" defense.</i></li> <li>3. <i>How is this communicated to Beta? Using the fleets NRTC: near-real-time communications.</i></li> </ol>
2	The SA aircraft detects that the 5 enemy air-	<ol style="list-style-type: none"> <li>1. <i>The enemy aircraft are within the AoR of the SA</i></li> </ol>

	craft have changed course and are heading toward the fleet at low altitude.	<p>sensors. The SA has been tracking these aircraft and sending tracks to Alpha and Beta.</p> <p>2. Need a "fleet" SA use case.</p>
3	SA informs both Alpha and Beta of the change.	<p>1. Within X seconds of detecting the change</p> <p>2. Using the GIG. Is the GIG usable for tactical near-real-time data? Probably not!</p> <p>3. Need a use case on assigning the UAVs to track the aircraft at this point.</p>
4	Alpha (and Beta) go to General Quarters.	<p>1. ADC informs the captain, who orders general quarters.</p> <p>2. Using Internal Comms.</p>
5	SA detects that missiles have separated from the enemy aircraft and informs Alpha and Beta.	<p>1. Within X seconds.</p>
6	Alpha assigns its 2 UAVs to track the missiles.	<p>1. The legacy Defensive Engagement System (DES) cannot use external tracks to form a FCQ track.</p> <p>2. Within X seconds.</p> <p>3. Does the ADC have to do this manually?</p> <p>4. Would they start tracking automatically if the missiles were within their AoR?</p> <p>5. Would they have been tracking the aircraft?</p>
7	The 2 Alpha-controlled UAVs send FCQ tracks for the 5 missiles to both Alpha and Beta.	<p>1. The 2 UAVs can redirect their payload to do this within YY seconds. (use case)</p> <p>2. Takes XX seconds for the FCQ tracks to stabilize.</p> <p>3. What is the comms between UAVs and ships for maneuver and payload control?</p>
8	Alpha assigns 3 missile engagements to itself and 2 to Beta.	<p>1. Is this done manually by the ADC?</p> <p>2. Is there a threshold below which it is manual and above which it is automatic?</p> <p>3. Airspace deconfliction is done on all 5 missiles with helo in the air.</p> <p>4. The UAVs and SA are above the level that the intercepting missiles will fly.</p> <p>5. Current DES does not handle more than 2 engagements simultaneously for AD.</p> <p>6. The timeline for using a "Shoot-Look-Shoot" engagement has not yet been developed. (use case)</p> <p>7. If the timeline is not satisfied, a "Shoot-Shoot-Look" engagement should occur.</p> <p>8. The DES may not be able to support the deconfliction timeline for 5 incoming missiles. (additional analysis needed, modeling, simulation, and T&amp;E)</p> <p>9. The DES may not have the capability to acknowledge Beta's acceptance of its assignment of 2 missiles.</p>
9	Alpha receives FCQ tracks (continuously) for the missiles and initiates engagements for 3 missiles.	<p>1. The missile launches from different batteries are X seconds apart, and from the same batteries XX seconds apart.</p> <p>2. The engagement software chooses the missile batteries to launch; each battery chooses the missiles. (use case)</p>
10	Both Alpha and Beta launch 2 AD missiles.	<p>1. The timeline between detection, launch time, and projected intercept time must be known such that a "kill" will not impact the ships.</p>
10a	Alpha tracks all of the intercept missiles and sends track updates to the intercept missiles it launched.	<p>1. There is an initial period when they are not tracked.</p> <p>2. After this time, they are tracked and comms is</p>

		<p>established to them.</p> <ol style="list-style-type: none"> <li>How often are these sent and with what latency?</li> <li>What communications between Alpha and the missiles?</li> <li>Does it actually send intercept point updates, or is this computed by the missile?</li> <li>Is the DES capable of sending track updates to the interceptor missiles that Beta had launched within the intercept timeline? (need to evaluate the DES system and software architecture)</li> </ol>
11	SA detects submarine launch of missiles from the southwest	<ol style="list-style-type: none"> <li>These missiles are also within the AoR of the SA.</li> <li>If they were outside, then their entry into the AoR would be noted!</li> <li>The AoR would be such that the fleet can be protected against such an attack. This would be part of the AD daily planning.</li> </ol>
12	SA informs both Alpha and Beta.	<ol style="list-style-type: none"> <li>Within X seconds of detection</li> </ol>
13	Alpha commands Beta to use its 2 UAVs to track these submarine missiles.	<ol style="list-style-type: none"> <li>Same comments apply from Step 6</li> </ol>
14	Alpha launches a third AD missile against the western attack.	<ol style="list-style-type: none"> <li>Continues to execute its previous plan.</li> </ol>
15	Alpha assigns itself 3 of the 7 missiles and Beta the other 4.	<ol style="list-style-type: none"> <li>Deconfliction is done since there is a helo in the air.</li> <li>The DES may not support the deconfliction timeline for 7 incoming missiles if there were more friendly or own aircraft in the vicinity.</li> </ol>
16	Alpha launches 2 missiles against the southern attack.	<ol style="list-style-type: none"> <li>Same comments as Step 10.</li> </ol>
17	Analysis of UAV track data shows that one western missile was not killed.	<ol style="list-style-type: none"> <li>It's difficult to separate a "leaker" from the chaff after a nearby hit. The SA aircraft confirms the leaker.</li> <li>Need a use case for leaker detection from debris.</li> </ol>
18	Alpha picks up the missile on its own radar.	<ol style="list-style-type: none"> <li>The own-radar track information is correlated with the UAV track information. No new track assignments are made.</li> <li>The timeline in Step 10 should also take into account that there is still time to intercept a leaker.</li> <li>If there is (at Step 10) insufficient time, then a "Shoot-Shoot-Look" engagement should be done.</li> <li>Need a survivability use case for this.</li> </ol>
19	Alpha launches another missile at the leaker.	<ol style="list-style-type: none"> <li>Is the target HVA informed of the leaker? Is this automatic or manual?</li> <li>Was it warned on initial determination of incoming missiles? Is this warning guaranteed?</li> <li>Does the target HVA maneuver or change its comms profile, deploy decoys, or what?</li> </ol>
20	Ships radar confirms that the leaker was killed.	<ol style="list-style-type: none"> <li>Using own radar.</li> <li>Is some manual recognition needed?</li> </ol>
<b>Extensions</b>		
Ext 1	Only 2 UAVs are available, and they are assigned to Alpha and have the AoR to the west.	<ol style="list-style-type: none"> <li>The UAVs will probably have to be maneuvered to get the submarine missiles within their AoR. How is this done? Will there be sufficient time?</li> <li>Is the AoR for the UAVs broad enough to plan this in advance?</li> <li>Is there a display showing the areas where FCQ tracks can be generated (before any attacks take</li> </ol>

		<p>place)? Does this have to come from the UAV?</p> <p>4. The FCQ tracks can only be generated where there is an overlapping AoR.</p>
Ext 2	One missile cannot be deconflicted.	<p>1. Have to command the helo to move immediately and recompute interception.</p> <p>2. This would have to be from both ships, since Alpha does all deconfliction initially before assignment. But Beta would do confirmation, so what if it could not deconflict? Haven't thought through to this level.</p>

### Mission Thread Template: Overarching Quality Attributes

Quality Attribute	Engineering Considerations, Issues, and Challenges
Performance	<ol style="list-style-type: none"> <li>The airspace deconfliction latency is heavily dependent on the number of aircraft within the strike paths.</li> <li>The timeline function from missile detection at specific distance from target until point of impact, including detection by both UAVs, engagement assignments, missile launching sequence and fly out times, have not been analyzed in detail!</li> </ol>
Availability / Reliability	<ol style="list-style-type: none"> <li>What if both UAVs cannot maneuver to their respective AoRs in time? <ol style="list-style-type: none"> <li>Probably have to wait until within ship's radar to fire.</li> <li>Is this a manual decision? (tradeoff with automation)</li> </ol> </li> <li>What if the ship/missile communications fails? <ol style="list-style-type: none"> <li>Probably have to fire another intercept missile!</li> <li>Can the other ship try to control the missile?</li> </ol> </li> <li>What if Alpha/Beta Comms fails? <ol style="list-style-type: none"> <li>Revert to a predefined separate engagement.</li> </ol> </li> <li>What if Beta does not acknowledge engagement assignments? Revert to what was defined in ROE, assume that it will follow received orders, or what? <ol style="list-style-type: none"> <li>Degraded Mode Use Case needs developed.</li> </ol> </li> <li>Degraded modes of operation have not been detailed yet.</li> <li>Loss of comms to SA. <ol style="list-style-type: none"> <li>After initial detection and UAV coverage, it does not matter. Some coverage.</li> <li>Before initial detection, the UAVs will provide some coverage, but will probably have some unmonitored areas.</li> <li>What happens when missile goes beyond LoS radar coverage?</li> </ol> </li> <li>What if one of the UAVs is deemed nonfunctional during operations?</li> </ol>
Accuracy	<ol style="list-style-type: none"> <li>If the tracks are relayed (see I.2), what if they are not sufficiently accurate? Will they be?</li> <li>Given multiple relay hops, how will accuracy be impacted? (Performance/accuracy tradeoff implications). How can shared resources be managed to bound latencies in this environment?</li> </ol>
Interoperability	<ol style="list-style-type: none"> <li>Can a UAV that is assigned and controlled by one ship can be reassigned and controlled by another ship dynamically? (Degraded mode future support?)</li> <li>Can FCQ information be transferred in real time from Alpha to Beta in order to target one of the missiles?</li> </ol>
Extensibility	<ol style="list-style-type: none"> <li>If a land-based system is used instead of an SA, can this mission thread still be satisfied?</li> <li>Can a helo do the job of a UAV for degraded operations?</li> <li>These should be extensions to the MT?</li> </ol>
Supportability	<ol style="list-style-type: none"> <li>Can the UAV's flight duration support this mission thread?</li> <li>Once a pair of UAVs return to their associated ship, how long does it take to service them before they could be sent out on patrol again?</li> </ol>
IA / Security	<ol style="list-style-type: none"> <li>What if the enemy aircraft are actively jamming after they fire? How is this detected and handled?</li> </ol>

Backward Capability	1. <i>The legacy DES has a couple of capability gaps that drive the need for a new system to satisfy this mission thread. What functionalities/capabilities of the legacy system will need to be supported for the new DES?</i>
Openness	<i>These require a different MT (sustainment).</i>
Reusability	<i>These require a different MT (sustainment).</i>
Testability	<ol style="list-style-type: none"> <li>1. <i>What self-diagnostics are being done on the UAVs, and how often is this information provided back to the ship?</i></li> <li>2. <i>Can the training simulators support this mission thread?</i></li> </ol>

---

## Appendix B Glossary

<b>Actor</b>	In the Unified Modeling Language (UML), “specifies a role played by a user or any other system that interacts with the subject” [OMG 2007].
<b>architectural challenge</b>	Potential problem with achieving a capability based on the proposed architecture and the quality attributes’ impact to the architecture. An architectural challenge could become an architectural risk if a program does not perform additional investigation to assess and correctly address the challenge.
<b>architecture</b>	The structure of components, their relationships, and the principles and guidelines governing their design evolution over time [DoD CIO 2007, IEEE 1990].
<b>capability</b>	“The ability to achieve a desired effect under specified standards and conditions through combinations of means and ways across the ... DOTMLPF to perform a set of tasks to execute a specified course of action” [CJCSI 2009, p. 23].
<b>context diagram</b>	A high-level, informal view of three things: the system you’re gathering requirements for, the things that need to interact with the system, and a brief note about the interaction between each thing and the system [Terski 2008].
<b>DoDAF</b>	The Department of Defense Architecture Framework defines the artifacts on which current DoD SoS architecture development practices center.
<b>DOTMLPF</b>	Doctrine, organization, training, materiel, leadership and education, personnel, and facilities. The Joint Capabilities Integration & Development System (JCIDS) process provides solution space that considers solutions involving any combination of DOTMLPF [Acquikipedia 2012].
<b>engineering consideration</b>	Concerns such as unclear functional requirements, missing system-level use cases, or hardware mismatches.
<b>mission thread</b>	<p>A sequence of end-to-end activities and events that take place to accomplish the execution an SoS capability. The context of a mission thread is defined by a vignette. A mission thread is given as a series of steps. There are three main types of mission thread: operational, development, and sustainment.</p> <p>Chairman of the Joint Chiefs of Staff 6212.01F defines a Joint Mission Thread (JMT) as an operational and technical description of the end-to-end set of activities and systems that accomplish the execution of a joint mission [CJCSI 2012].</p>
<b>system of sys-</b>	“A set or arrangement of systems that results when independent and useful

<b>tems</b>	<p>systems are integrated into a larger system that delivers unique capabilities” [DoD 2004]. The OSD guide defines four types of SoS [Deputy Under Secretary of Defense 2008]:</p> <p><i>Directed.</i> In a directed SoS, the integrated SoS is built and managed to fulfill specific purposes. It is centrally managed during long-term operation to continue to fulfill those purposes as well as any new ones the system owners might wish to address. The constituent systems maintain an ability to operate independently, but their normal operational mode is subordinated to the central managed purpose.</p> <p><i>Acknowledged.</i> An acknowledged SoS has recognized objectives, a designated manager, and resources for the SoS; however, the constituent systems retain their independent ownership, objectives, funding, and development and sustainment approaches. Changes in the systems are based on collaboration between the SoS and the system.</p> <p><i>Collaborative.</i> In a collaborative SoS, the component systems interact more or less voluntarily to fulfill agreed-upon central purposes. The internet is a collaborative system. The Internet Engineering Task Force works out standards but has no power to enforce them. The central players collectively decide how to provide or deny service, thereby offering some means of enforcing and maintaining standards.</p> <p><i>Virtual.</i> A virtual SoS lacks a central management authority and a centrally agreed-upon purpose for the SoS. Large-scale behavior emerges—and may be desirable—but this type of SoS must rely on relatively invisible mechanisms to maintain it.</p>
<b>use case</b>	<p>A means for specifying required uses of a system. Typically, they are used to capture the requirements of a system, that is, what a system is supposed to do [OMG 2007].</p>
<b>vignette</b>	<p>A description of the geography, own force structure and mission, strategies and tactics, and the enemy forces and their attack strategies and tactics, including timing. There may be associated Measures of Performance (MOP) and Measures of Effectiveness (MOE). A vignette provides context for one or more mission threads [Gagliardi 2010].</p>



---

## References

*URLs are valid as of the publication date of this document.*

### **[Acquipedia 2012]**

Acquipedia. *DOTMLPF Analysis*. Defense Acquisition University, 2012.  
<https://dap.dau.mil/acquipedia/Pages/ArticleDetails.aspx?aid=d11b6afa-a16e-43cc-b3bb-ff8c9eb3e6f2>

### **[Barbacci 2003]**

Barbacci, M., Ellison, R., Lattanze, A., Stafford, J., Weinstock, C., & Wood, W. *Quality Attribute Workshops, Third Edition (CMU/SEI-2003-TR-016)*. Software Engineering Institute, Carnegie Mellon University, 2003. <http://www.sei.cmu.edu/library/abstracts/reports/03tr016.cfm>

### **[CJCSI 2009]**

Chairman of the Joint Chiefs of Staff Instruction. *Joint Capabilities Integration and Development System (CJCSI 3170.01G)*. Defense Acquisition University, 2009.

### **[CJCSI 2012]**

Chairman of the Joint Chiefs of Staff Instruction. *Net Ready Key Performance Parameter (CJCSI 6212.01F)*. Defense Technical Information Center, March 2012.  
[http://www.dtic.mil/cjcs\\_directives/cdata/unlimit/6212\\_01.pdf](http://www.dtic.mil/cjcs_directives/cdata/unlimit/6212_01.pdf)

### **[Clements 2001]**

Clements, P., Kazman, R., & Klein, M. *Evaluating Software Architectures: Methods and Case Studies*. Addison Wesley, 2001.

### **[Deputy Under Secretary of Defense 2008]**

Deputy Under Secretary of Defense (Acquisition and Technology). *Systems Engineering Guide for Systems of Systems, Version 1.0*. ODUSD(A&T)SSE, 2008.  
<http://www.acq.osd.mil/se/docs/SE-Guide-for-SoS.pdf>

### **[DoD 2004]**

Department of Defense. "System of Systems Engineering," Ch. 4. *Defense Acquisition Guidebook*. Pentagon, 2004.

### **[DoD CIO 2007]**

Department of Defense Chief Information Officer. *DoD Architecture Framework, Version 1.5, Volume 1: Definitions and Guidelines*. DoD, 2007.  
[http://dodcio.defense.gov/Portals/0/Documents/DoDaF/DoDAF\\_Volume\\_I.pdf](http://dodcio.defense.gov/Portals/0/Documents/DoDaF/DoDAF_Volume_I.pdf)

**[Gagliardi 2010]**

Gagliardi, M., Wood, W., Morrow T., & Klein, J. *System of Systems (SoS) Quality Attribute Specifications and Architecture Evaluation* [Webinar], Slide 10. Software Engineering Institute, Carnegie Mellon University, 2010.

<http://www.sei.cmu.edu/library/abstracts/presentations/20100120webinar.cfm>

**[IEEE 1990]**

IEEE Standards Association. *IEEE Standard Glossary of Software Engineering Terminology* (IEEE Std 610.12-1990). IEEE Computer Society Press, 1990.

<http://standards.ieee.org/findstds/standard/610.12-1990.html>

**[OMG 2007]**

Object Management Group. “Use Cases,” 585–586. *OMG Unified Modeling Language (OMG UML), Superstructure, V2.1.2*. OMG, 2007.

**[SEI 2010a]**

Software Engineering Institute. *Software Product Lines*. <http://www.sei.cmu.edu/productlines/> (2010).

**[SEI 2010b]**

Software Engineering Institute. *ATAM Evaluator Certificate*.

<http://www.sei.cmu.edu/training/certificates/architecture/atame.cfm> (2010).

**[SEI 2012]**

Software Engineering Institute. *ATAM Leader Training*. <http://www.sei.cmu.edu/training/p32.cfm> (2012).

**[Terski 2008]**

Terski, Matt. “The Context Diagram: Getting Started with Use Cases.” *The Use Case Blog*.

<http://blog.casecomplete.com/post/Getting-Started-with-Use-Cases-The-Context-Diagram.aspx> (2008).

<b>REPORT DOCUMENTATION PAGE</b>			<i>Form Approved</i> <i>OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE October 2013	3. REPORT TYPE AND DATES COVERED Final		
4. TITLE AND SUBTITLE Introduction to the Mission Thread Workshop		5. FUNDING NUMBERS FA8721-05-C-0003		
6. AUTHOR(S) Mike Gagliardi, Bill Wood, and Tim Morrow				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213			8. PERFORMING ORGANIZATION REPORT NUMBER CMU/SEI-2013-TR-003	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFLCMC/PZE/Hanscom Enterprise Acquisition Division 20 Schilling Circle Building 1305 Hanscom AFB, MA 01731-2116			10. SPONSORING/MONITORING AGENCY REPORT NUMBER ESC-TR-2013-003	
11. SUPPLEMENTARY NOTES				
12A DISTRIBUTION/AVAILABILITY STATEMENT Unclassified/Unlimited, DTIC, NTIS			12B DISTRIBUTION CODE	
13. ABSTRACT (MAXIMUM 200 WORDS)  In Department of Defense programs, a system of systems (SoS) is integrated to accomplish a number of missions that involve cooperation among individual systems. Understanding the activities conducted within each system and how they interoperate to accomplish the missions of the SoS is of vital importance. A mission thread is a sequence of end-to-end activities and events, given as a series of steps, that accomplish the execution of one or more capabilities that the SoS supports. However, listing the steps and describing them do not reveal all the important concerns associated with cooperation among the systems to accomplish the mission; understanding the architectural and engineering considerations associated with each mission thread is also essential. The Mission Thread Workshop (MTW) is a facilitated, stakeholder-centric workshop whose purpose is to elicit and refine end-to-end quality attribute, capability, and engineering considerations for SoS mission threads. This report introduces the MTW, describes the three phases of an MTW, and explains the steps of each phase in detail. It also describes the benefits that a program can expect from conducting an MTW, which include identifying quality attributes and architectural challenges but also reach beyond these goals to expose gaps in capability, functionality, documentation, and engineering.				
14. SUBJECT TERMS quality attribute, System-of-Systems Architectures, mission threads, vignettes, Mission Thread Workshop			15. NUMBER OF PAGES 43	
16. PRICE CODE				
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. Z39-18  
298-102