Systems of Systems Engineering Life Cycle

Dr. Judith S. Dahmann
The MITRE Corporation
7525 Colshire Drive
McLean, VA 22301
USA

jdahmann@mitre.org

ABSTRACT

This paper discusses Systems Engineering (SE) for Systems of Systems (SoS) from the perspective of the systems engineering life cycle as it applies to the particular circumstances of systems of systems. Based on the draft SoS Annex of the ISO 15288 systems engineering standard, the paper discusses how the current SE life cycle process are applied to SoS. The paper also presents a life cycle, implementer's view of systems engineering of SoS using an iterative approach to SoS evolution based on the changes made in the constituent systems supporting the system of systems.

1.0 INTRODUCTION

While there are many definitions of Systems of Systems (SoS) and systems engineering for SoS for the purposes of this lecture we will use the definitions from the US Defense Acquisition Guidebook (DAG):

> System of Systems is a "set or arrangement of systems that results when independent and useful systems are integrated into a larger system that delivers unique capabilities".
>
> Systems of Systems Engineering is "the process of planning, analyzing, organizing, and integrating the capabilities of a mix of existing and new systems into a system-of-systems capability that is greater than the sum of the capabilities of the constituent parts".

While SoS are themselves systems, their key characteristics (Maier, 1998)[1] affect the way systems engineering is applied for SoS[2]:

- Operational independence of constituent systems;
- Managerial independence of constituent systems;
- Geographical distribution;
- Evolutionary development processes; and
- Emergent behaviour.

These characteristics and their implications for systems engineering means that systems engineers tailor SE life processes and the overall implementation of systems engineering when applying SE to SoS as systems.

[1] Maier, M. W. (1998). "Architecting principles for systems-of-systems." Systems Engineering 1(4): 284.

[2] See NATO Lecture "Systems of Systems Characterization and Types" for full discussion of this topic.

2.0 SYSTEMS ENGINEERING LIFE CYCLE PROCESSES AS THEY APPLY TO SYSTEMS OF SYSTEMS

2.1 Overview of 15288 Life Cycle Processes

ISO 15288 is a widely accepted international standard for systems engineering[3] developed and promulgated under the International Organization for Standardization, the world's largest voluntary standards organization.

> ISO/IEC 15288:2008 establishes a common framework for describing the life cycle of systems created by humans. It defines a set of processes and associated terminology. These processes can be applied at any level in the hierarchy of a system's structure. Selected sets of these processes can be applied throughout the life cycle for managing and performing the stages of a system's life cycle. This is accomplished through the involvement of all interested parties, with the ultimate goal of achieving customer satisfaction.

The standard describes SE lifecycle processes in four major categories as shown in Figure 1. These include:

- Agreement Processes;
- Organizational Project-Enabling Processes;
- Technical Management Processes; and
- Technical Processes.

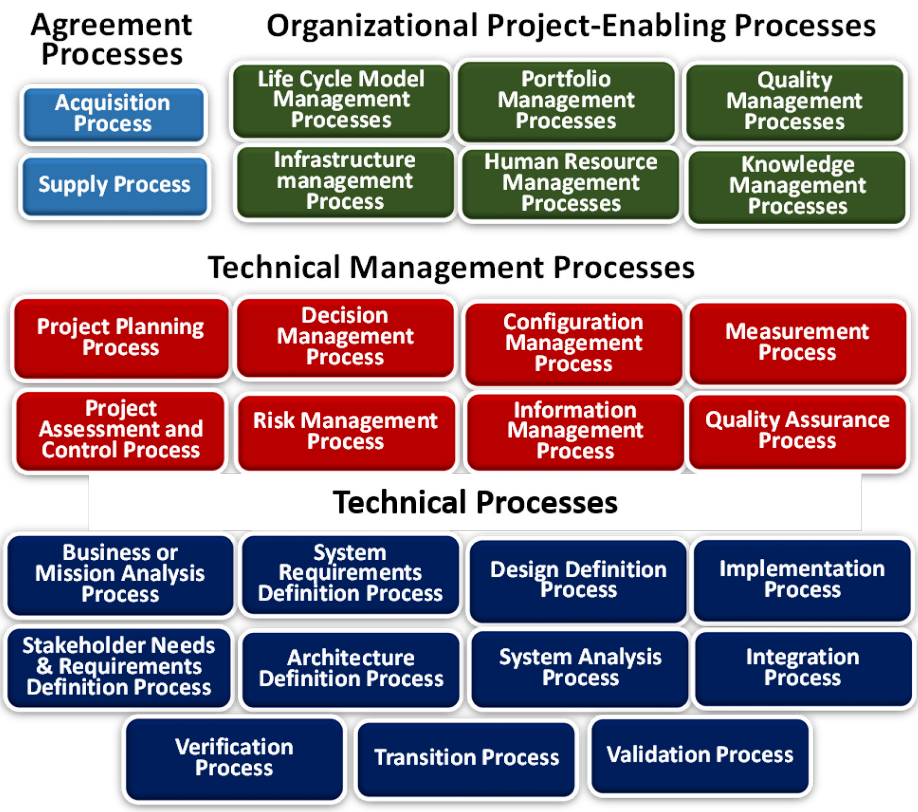

Figure 1: ISO 15288 Systems Engineering Life Cycle Processes.

[3] International Organization for Standardization online catalogue, http://www.iso.org/iso/home/about.htm, Accessed 11/27/2011.

The draft update includes an annex which specifically addresses how these processes apply to systems of systems. The following sections discuss each of these processes and how the draft standard adapts them to the particular characteristics of SoS.

2.2 Agreement Processes

Agreement processes establish the way that elements of the SoS will work together both in development and in operations. This includes both those responsible for the SoS and the constituent systems.

While this is typically fairly straightforward in systems where there is a single manager and systems engineer, in systems of systems this is more complex. Constituent systems of a SoS are often independent of the SoS. In most cases they are acquired and managed by different organizations and as a result they often hold original objectives that may not align with those of the SoS.

A recognized set of SoS types based on authority relationships is shown in Table 1. The SoS type has a direct effect on how agreement processes will apply.

Table 1: Systems of Systems Types[4].

Type	Definition
Directed	Directed SoS are those in which the SoS is engineered 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 component systems maintain an ability to operate independently, but their normal operational mode is subordinated to the centrally managed purpose.
Acknowledged	Acknowledged SoS have recognized objectives, a designated manager, and resources for the SoS; however, the constituent systems retain their independent ownership, objectives, funding, development, and sustainment approaches. Changes in the systems are based on cooperative agreements between the SoS and the system.
Collaborative	In collaborative SoS, the component systems interact more or less voluntarily to fulfill agreed-upon central purposes.
Virtual	Virtual SoS lacks a central management authority and a centrally agreed-upon purpose for the system of systems. Large-scale behavior emerges—and may be desirable—but this type of SoS relies upon relatively invisible, self-organizing mechanisms to maintain it.

Except in directed SoS, where the constituent systems are subordinated to the SoS, the SoS organization cannot task a constituent system organization without their cooperation. In acknowledged or collaborative SoS, where the SoS manager does not have authority over the constituent systems, SoS tasks are balanced against the tasks of the constituent system based on negotiation among the SoS elements. Finally, in virtual SoS, where there is no agreement processes may be informal, or considered only for analysis purposes.

2.3 Organizational Project Enabling Processes

In a system, *the organizational project enabling processes* establish the project development and management processes for all elements of the project from start to finish including establishing and curtailing projects, managing resources and monitoring quality.

[4] US Defense Acquisition Guidebook, 2014; https://acc.dau.mil/dag4, Accessed 11/23/2014.

In a SoS, *organizational project enabling processes* are implemented at two levels. Organizations responsible for the constituent systems implement these processes for their system independent of the SoS. The SoS organization implements these processes for the SoS particularly for those considerations that apply to the overall SoS. In collaborative systems of systems these SoS level process are implemented by agreement of the constituent systems.

A particular challenge in SoS engineering arises from the lack of alignment among the constituent system *organizational project enabling processes* and those of the SoS. Constituent systems processes are designed to meet their own outcomes and may not align with those of the SoS, posing a challenge for a coordinated implementation across the SoS.

2.4 Technical Management Processes

In a system, *technical management processes* are concerned with technically managing the resources and assets of the project and with applying them to fulfill the agreements made concerning the implementation of the project.

Again, in a SoS, these *technical management processes* are implemented at two levels. At the SoS level, *technical management processes* area applied to the particular considerations of SoS engineering - planning, analyzing, organizing, and integrating the capabilities of a mix of existing and new systems into a system-of-systems capability. In almost all cases, constituent systems organizations retain responsibility for engineering their systems and for their own *technical management processes.*

This means that SoS organizations must plan an integrated life cycle that recognizes the independent changes in the constituent systems, in addition to the SoS-initiated changes in a life cycle that treats the SoS as a system. This results in a SoS evolutionary implementation approach with incremental capabilities added among the constituent systems. This will be discussed explicitly in Section 3 of this paper.

2.5 Technical Processes

In a system *technical processes* are concerned with the technical actions throughout the life cycle, applied in order to create and use a system. *Technical processes* in a SoS are also implemented by both the SoS and constituent systems.

In particular each *technical process* is implemented for the SoS as well as for the constituent systems:

- **Business or Mission Analysis** for a SoS looks across the full SoS business and mission environment.
- **Stakeholder Needs and Requirements Definition** focus on the top level SoS, but also consider the disparate needs of the stakeholders for the systems.
- **System Requirements Definition** for the SoS is done at the level needed to satisfy stakeholder needs and mission objectives and translated into requirements for the constituent systems.
- **Architecture for the SoS** is a framework for organizing and integrating the capabilities of a mix of existing and new systems into a SoS capability, leaving the architectures of the constituent systems to their organizations.
- **Design Definition** provides sufficient detailed data and information to enable the SoS implementation. This involves collaboration with the constituent systems who will conduct their own design trades to identify the approach to address SoS requirements as they apply to their system.
- **Integration, Verification, Transition, Validation** implemented by the constituent systems for the changes they make to support requirements generated by the SoS**.** They also apply to the SoS when

the upgraded constituent systems are integrated into the SoS and performance is verified and validated. SoS-level evaluations may only be performed in the operational environment, in which case precautionary measures should be considered to avoid adverse SoS-behavior.

- **Operations, Maintenance and Disposal Processes** are typically implemented by the constituent systems, given their management and operational independence.

As with other processes, the independent and asynchronous nature of constituent systems in a SoS pose challenges to effective implementation of these *technical processes.*

3.0 IMPLEMENTER VIEW OF SYSTEMS ENGINEERING LIFE CYCLE FOR SYSTEMS OF SYSTEMS

3.1 Systems Engineering "V" Model and Systems of Systems Double V Model

Figure 2 below displays a classic view of systems engineering as a "V" process, adapted from the US Defense Acquisition Guidebook[5]. In this view a system moves from a statement of operational need through requirements and design to implementation, validation and delivery. Various forms of this model are regularly used to describe systems engineering for individual systems.

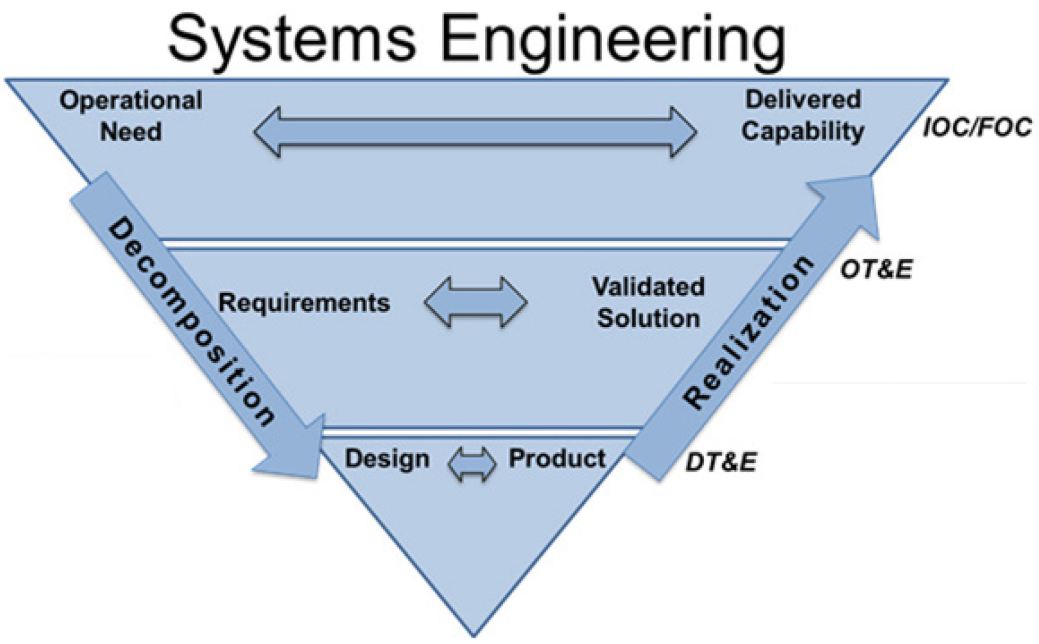

Figure 2: Systems Engineering "V" Model.

When applying systems engineering to SoS as a system, given the multiple levels of engineering at both the SoS and systems levels, one way to depict SE for SoS is as a "double V" model as shown in Figure 3.

[5] Adapted from US Defense Acquisition Guidebook Chapter 4 Systems Engineering, Figure 4.1.F2. Systems Engineering Processes, https://acc.dau.mil/CommunityBrowser.aspx?id=638297&lang=en-US, Accessed 11/26/2014.

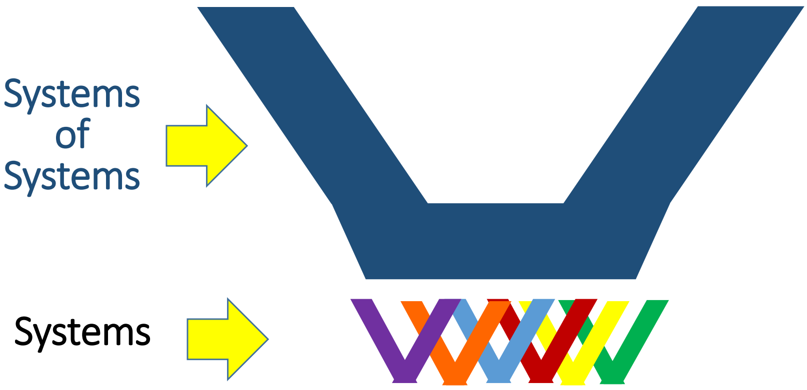

Figure 3: "Double V" Representation of SE for a System of Systems.

This representation emphasizes the fact that in a SoS, systems engineering continues for the individual systems, concurrently with application of SE for the SoS as a system. This can be considered a logical view of the SoS SE process.

3.2 Implementers View of SoS SE

These "V" and "double V" representations imply a single pass development of a system of a SoS. However, as noted above in Section 2.3 in the discussion of *technical management processes* as they apply to SoS, the nature of SoS as composed of multiple independent systems, with their own organization, technical and technical management processes all effect the implementation of SE for a SoS. In particular "this means that SoS organizations must plan an integrated life cycle that recognizes the independent changes in the constituent systems, in addition to the SoS-initiated changes in a life cycle that treats the SoS as a system. This results in a SoS evolutionary implementation approach with incremental capabilities added among the constituent systems." In this section of the paper we will discuss this as an implementer's view of SoS SE.[6]

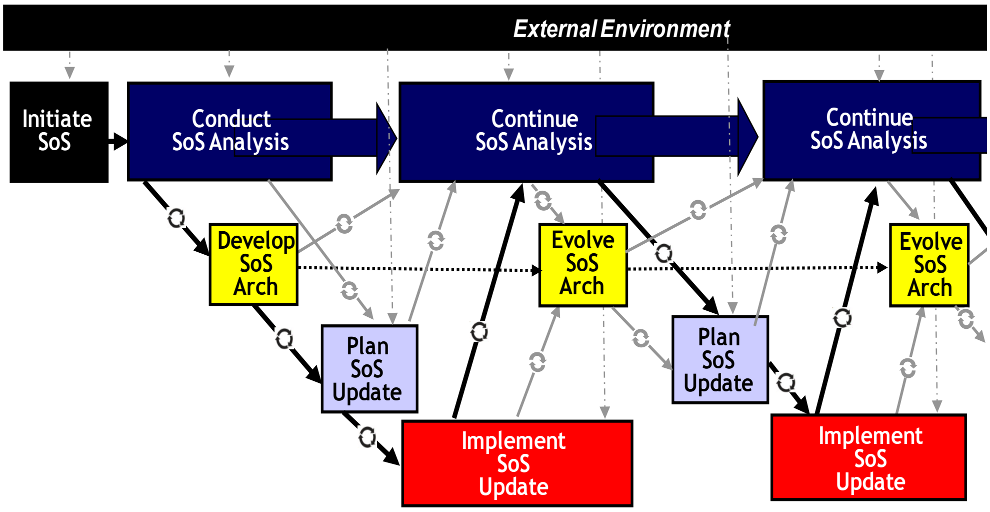

Figure 4: SE Life Cycle for SoS[7].

[6] The material in this section of the paper draws on IEEE 2011 paper entitled "Implementers' View of SoS SE" Dahmann J., G. Rebovich, J. Lane, R. Lowry and K. Baldwin. "An Implementers' View of Systems Engineering for Systems of Systems." *Proceedings of the IEEE Systems Conference*, April 4 - 7, in Montreal, QC, Canada, 2011.

[7] Ibid.

This depiction of the SoS SE life cycle[8] represents SoS SE as steps which are implemented in an iterative fashion, with each step providing feedback into the ongoing, evolutionary process. The major steps in the life cycle are:

- **Initiate SoS**: Provides foundational information to initiate the SoS.
- **Conduct SoS Analysis**: Provides analysis of the 'as is' SoS and basis for its evolution.
- **Develop SoS Architecture**: Develops/evolves the persistent technical framework for SoS evolution and a migration plan identifying risks and mitigations.
- **Plan SoS Update**: Evaluates SoS priorities, backlog of SoS changes, and options to define plans for the next SoS upgrade cycle.
- **Implement SoS Update**: Oversees system implementations and plans/conducts SoS level testing, resulting in a new SoS product baseline.
- **Continue SoS Analysis**: Ongoing SoS analysis revisits the state of and plans for the SoS as the basis for SoS evolution.

This overall process and steps will be used in the remainder of this paper to discuss SoS SE life cycle.

It should be noted that this is just one way to represent the SoS SE life Cycle which reflects a growing understanding of SoS life cycle implementation. Figure 5, shows another representation developed by the European Commission project **D**esigning for **A**daptability and Evolutio**N** in **S**ystems of Systems **E**ngineering (**DANSE**).[9]

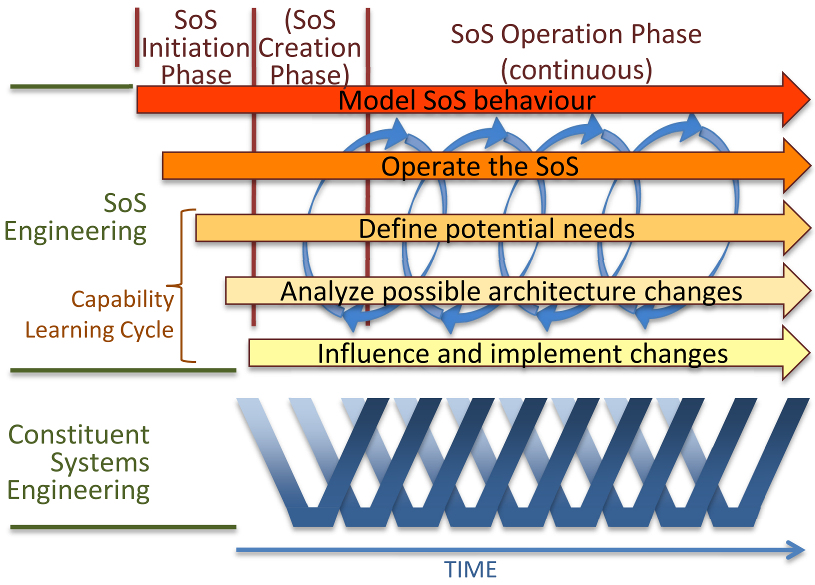

Figure 5: SoS Life Cycle as Depicted by Designing for Adaptability and EvolutioN in Systems of Systems Engineering (DANSE)[10].

[8] This model has been referred to as a 'wave model', Concept of Wave Planning was developed by Dr David Dombkins. See "Complex Project Management" Booksurge Publishing, South Carolina: 2007.

[9] Designing for Adaptability and evolutioN in System of systems Engineering (DANSE), www.danse-ip.eu/ Accessed 11/26/2014.

[10] Honour, Eric. "*Designing for Adaptability and evolutioN in System of systems Engineering*", Presented at National Defense Industry Association Systems Engineering Conference, October in Arlington, VA, 2013. www.dtic.mil/ndia/2013system/TH16282_Honour.pdf, Accessed 11/26/2014.

While this DANSE depiction is different in some ways, it shares essential characteristics of the SoS SE life cycle process. As described in the original paper on the SoS implementers view model[11] the key features of this life cycle model reflect the nature of SoS and the impact on SE:

- **Multiple Overlapping Iterations of Evolution** reflect the fact that most SoS leverage developments of their constituent systems, and consequently, SoS are characterized by incremental development.
- **Ongoing Analysis** provides an analytic basis for each iteration of SoS evolution. Unlike traditional systems engineering in which upfront analysis drives development, engineering of SoS requires continuous analysis to address the dynamic nature of the SoS and its context.
- **Continuous Input from External Environment** is key for SoS SE, since any manager or engineer of a SoS has control over only a small part of the environment that affects the SoS.
- **Architecture Evolution** is also important. While the architecture of a SoS ideally provides a persistent framework for the SoS evolution over time, the planned SoS architecture is typically implemented incrementally and may itself evolve.
- **Forward Movement with Feedback** drives the evolution of a SoS which typically adopts a "battle rhythm" driven by elements in the SoS context (e.g. the development plans of a key constituent system or the unit fielding schedule) which are not under the control of the SoS. These external driving events effectively "pace" the execution of the SoS evolution. While there may be feedback within an evolution, many SoS adopt a "bus stop" approach, where they deliver those changes that can be implemented during an iteration and defer the rest to subsequent evolutions (or the next time the bus stops.)

Associated with each step in the model are particular SoS SE artifacts or knowledge elements with reflect key information about the SoS to support its engineering[12]. These artifacts are shown in Table 2, with a comparison to similar artifacts associated with systems engineering for systems.

Table 2: SoS SE Artifacts as Compared to System Artifacts[13].

Artifact	SoS	System
Capability Objectives	Focused on capabilities at the SoS-level. Solution(s) typically require multiple constituent systems, not all of which may be known in advance. Scope typically initially defined in the charter for the SoS.	Addresses a gap in a user capability as defined by formal process (Joint Capabilities Development System (JCIDS) or Component equivalent process); may provide functionality that supports SoS capability objectives.
CONOPS	Multiple system focus. Often developed after constituent systems have been fielded; Evolves over time, sometimes substantially.	Single system focus. Defined when systems acquisition begins.
Systems Information	Focus is on system-level information that impacts SoS-level capability objectives. Extends beyond technical issues to include operational, fiscal, organizational, and planning issues.	Focus is on interfaces and inputs/outputs with external systems and how they support or inhibit single system performance. Focus is usually on technical issues.
Requirements	Requirements 'space' versus set of specific requirements. Defined at a level of detail that enables trades among potential and actual constituent systems and interfacing external systems.	Defined by needs of the operational users of the system and by the threat. Usually articulated as detailed operational requirements or specified technical requirements.

[11] Dahmann J., G. Rebovich, J. Lane, R. Lowry and K. Baldwin. "An Implementers' View of Systems Engineering for Systems of Systems." *Proceedings of the IEEE Systems Conference*, April 4 - 7, in Montreal, QC, Canada, 2011.

[12] Dahmann J., G. Rebovich, J. Lane and R. Lowry. "System Engineering Artifacts for Systems of Systems." *Proceedings of the IEEE Systems Conference*, April 5-8, in San Diego, CA, 2010.

[13] Ibid.

Artifact	SoS	System
Performance measures and methods	Focus is on performance of SoS solution. As independent as possible of the specific systems to allow for assessment of alternative implementation approaches.	Focus is on performance of the specific system and connections with external interfaces.
Performance Data	Often collected in operational environment. Used to support continuous improvement of the SoS.	Predominantly collected in traditional acquisition lifecycle T&E, including simulation/modelling. Used to support fielding decisions.
SE Planning Elements	Focus is on determining rhythm, organizational structure, technical reviews, and decision processes across SoS evolution. Ability and willingness of constituent systems to support SoS plans is an important consideration.	Focus is on an individual system typically part of the acquisition process; takes the form of an SE Plan.
Risks and Mitigations	Focus is on desired capabilities and undesirable emergent behaviors of the SoS. Includes single system risks or dependencies essential to SoS capabilities and plans.	Focus is on system issues and potential problems. Includes external dependencies that pose special risks.
Master Plan	Focus is on SoS-level view across multiple increments and touch points for constituent systems. Reflects the SoS evolution strategy. Focus is often on continuous improvement versus achievement of a defined end state.	Focus is typically on individual system and approach to achieve defined end-state. Reflects the system acquisition strategy.
Agreements	Focus is on managing relationships among multiple organizations. Agreements support SoS evolution including specific commitments to execute SoS increment development.	Focus is on defining specific system dependencies (e.g. commitments to provide components to a system through Government Furnished Equipment (GFE) or Commercial Of-the Shelf (COTS) components).
Architecture	A shared framework primarily aimed at informing analysis and decisions for developing or evolving SoS capabilities. A context for understanding the relationships among constituent systems and developing implementation options for meeting capability requirements. Includes key constituent systems information, connectors and protocols used to communicate and/or synchronize processing across the constituents, key data elements/structures that cross interfaces, and key data conversions to facilitate data sharing and communications between constituents.	A framework for analyzing and making decisions on system development and interfaces with external systems. For the single system, includes information about system's top level components, connectors between the components, protocols used to communicate between the components and synchronize processing across the components and key data elements/structures that cross interfaces between the components and any interfacing external systems.
Technical Baselines	Focus is on SoS-level description plus identification of constituent system baselines that are part of the SoS baseline.	System detailed artifacts/components that comprise the system baseline.
Technical plan(s)	Focus is on planning the implementation of changes to constituent systems to execute a SoS increment.	Focuses on implementation of changes to the system, including those required for the system to interface with external elements.
Integrated Master Schedule	Set of SoS SE activities and milestones plus key single system activities and milestones that are driving SoS critical path. Focus is on key synchronization points among SoS constituents and pointers to development schedules of constituent systems for the current SoS increment.	Detailed list of development activities, milestones, and associated schedule for the system.

In the subsequent sections of this paper each of the major steps in the SoS life cycle are discussed along with the key artifacts which are generated or updated at that step, as shown in Figure 6.

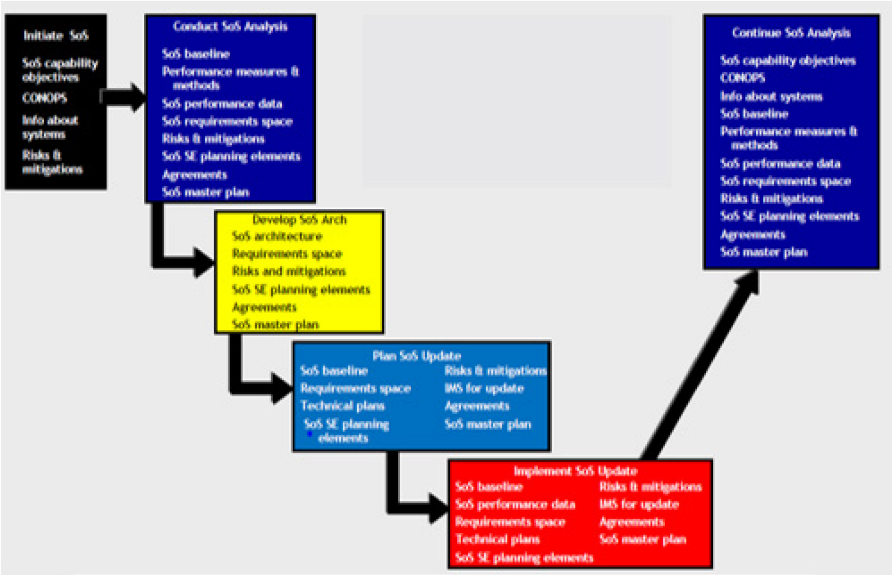

Figure 6: Artifacts Generated or Updated at Each Step in the SoS SE Lifecycle[14].

3.3 Initiate SoS

The first step in the SoS SE life cycle is *Initiate SoS*. The work done at this initial step provides the foundational information to start the SoS SE process. Key activities during *Initiate SoS* include developing a basic understanding of the SoS objectives, identifying the key users, user roles and expectations, and defining the core systems supporting capabilities.

Information important to the execution of this element includes:

- A statement of top-level objectives for the SoS (SoS capability objectives).
- A description of how systems in the SoS will be employed in an operational setting (SoS CONOPS).
- Programmatic and technical information about systems that affect SoS capability objectives (systems information).
- Initial risks facing the SoS.

Typically SoS are initiated after some type of trigger event which makes the need for specific attention to a systems of system, which may already exist in practice, but which has never been explicitly recognized as a SoS nor been subject to explicit systems engineering at the SoS level. Often the systems in the SoS exist and are deployed and, in many cases support the general capability called for in the SoS. In these situations, some event or change in circumstances elevates the importance of the SoS capabilities and initiates a formal effort to address this capability and hence the initiation of the SoS activity.

3.4 Conduct SoS Analysis

The next step in the life cycle is activity to *Conduct SoS Analysis*. This step provides analysis of the 'as is' SoS and basis for its evolution of the SoS to achieve user objectives.

[14] Lane, J., et al. "Key System of Systems Engineering Artifacts to Guide Engineering Activities", Presentation at NDIA Systems Engineering Conference. San Diego CA: October 2010.

Figure 7 shows the major activities typically implement during this step. These includes developing a CONOPs for the SoS, a functional baseline of the end-to-end functionality called for in the CONOPS, an assessment of the current systems available to support this functionality and, finally, an assessment of gaps in needed SoS capability.

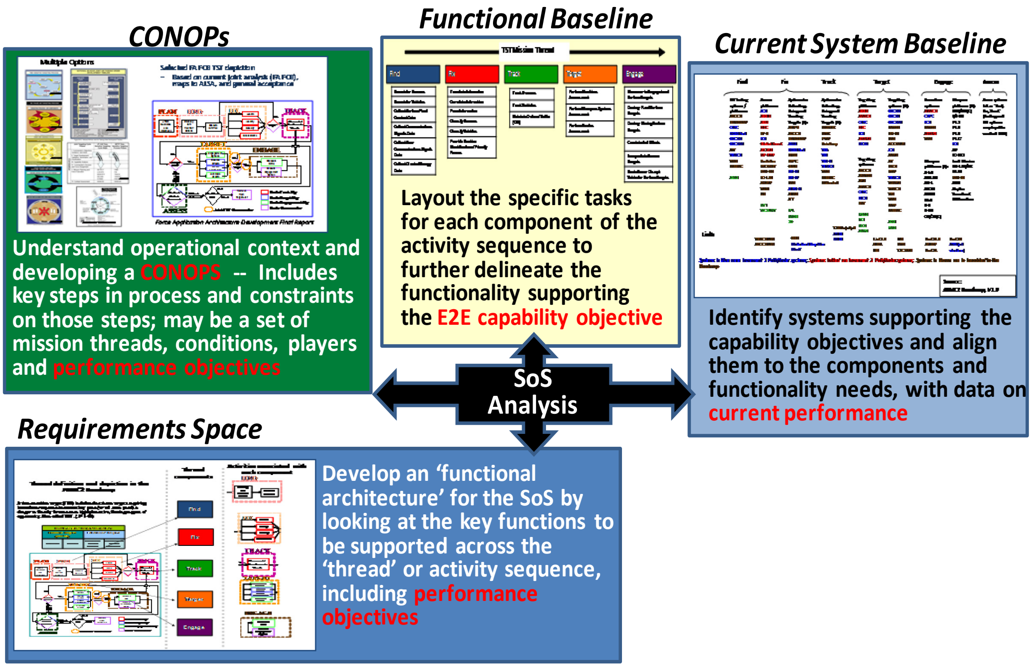

Figure 7: Major Activities Typically Supporting SoS Analysis[15].

The results of these analyses establish the initial view of the SoS based on information developed during SoS initiation and are reflected in a set of SoS SE artifacts:

- **SoS technical baselines** including requirements baseline, allocated baseline and product baseline.
- **SoS performance measures and methods,** a basis for overall SoS performance and continuous SoS improvement.
- **SoS performance data** to assess progress towards SoS capability objectives.
- **SoS requirements space,** a first order SoS user needs and functions to provide the capability in various environments.
- **SoS risks and mitigations** especially those that emanate from outside the SoS, including changes to constituent systems of the SoS.

In addition this information provides the basis for developing the initial plans for the SoS engineering, as reflected in other SoS SE artifacts:

- **SoS planning elements** provide structure and process for SoS SE, including pacing of SoS upgrades, organizational structure and decision processes, and technical reviews.
- **SoS master plan,** the SoS analog to a systems acquisition strategy.
- **Agreements** that delineate broad roles and responsibilities of SoS participants and their specific commitments in a development increment.

[15] From NDIA SoS and T&E Final Report, March 2012.

3.5 Develop SoS Architecture

In the next step in the life cycle, *Develop SoS Architecture*, the SoS SE develops and evolves the persistent technical framework for SoS evolution and a migration plan that identifies risks and mitigations.

The typical activities in this step are shown in Figure 8. These include an explicit delineation of the end-to-end SoS capabilities as defined in the CONOPS, alignment of current systems to these activities, including an understanding of gaps in ability to meeting SoS objectives, and the identification and analysis of alternative architectures which could improve SoS performance and other attributes including SoS resilience in light of the inevitable changes in systems and the SoS environment which could disrupt SoS operations.

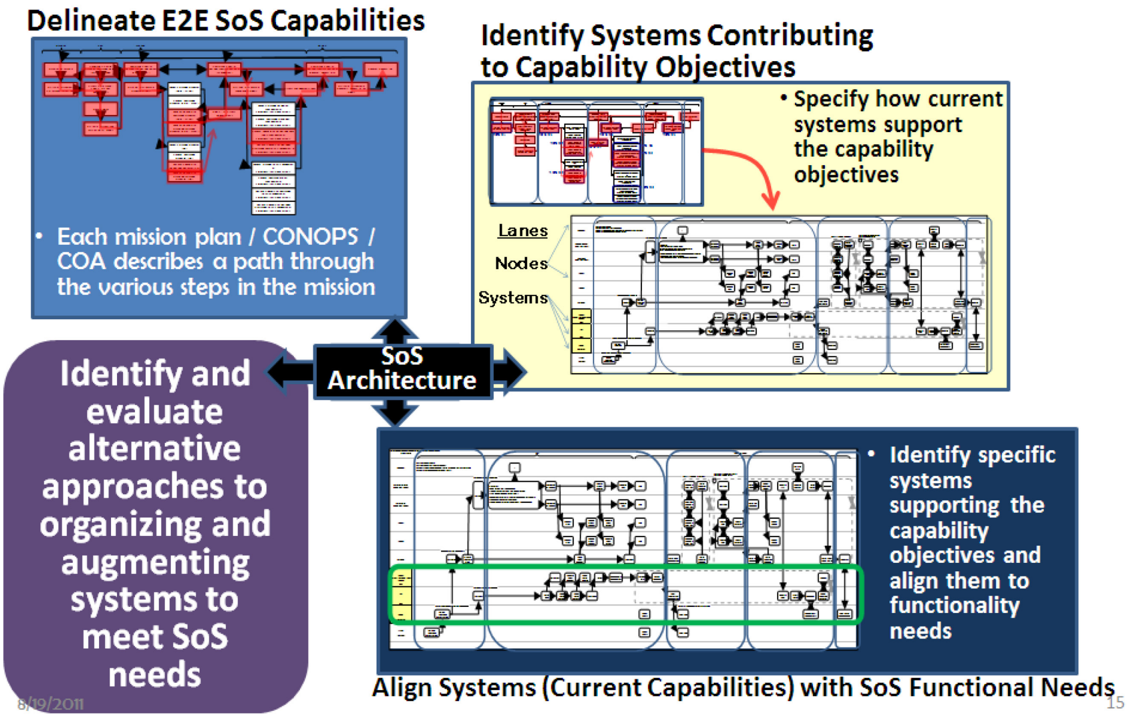

Figure 8: Major Activities Typically Supporting SoS Architecture Development[16].

A SoS architecture is the key artifact created and used in this element. The SoS architecture is typically created in the first or early execution of this element. Subsequently, the SoS architecture is evolved. Through the life cycle the architecture provides a shared representation of the SoS technical framework and is used to inform and document decisions and guide SoS evolution. It includes systems, key SoS functions, relationships and dependencies, as well as end-to-end functionality, data flow and communications protocols. It is used to address possible changes in functionality, performance or interfaces.

During this step, as the SoS SE learns more about the SoS and its associated context, there may be updates to artifacts created at initiation, including the SoS risks, SE planning elements, agreements and the master plan.

[16] Dahmann, J, R. Heilman et al, "*SoS Systems Engineering (SE) and Test & Evaluation (T&E): Final Report of the NDIA SE Division*", SoS SE and T&E Committees of the Systems Engineering Division of the National Defense Industry Association, March 2012, http://www.ndia.org/Divisions/Divisions/SystemsEngineering/Documents/Studies/SoS%20SE%20and%20TE%20%20NDIA%20Final%20Report%2028%20March%20with%20abstractnotes.pdf; Accessed 11/26/2014.

3.6 Plan SoS Update

In the next step in the life cycle, *Plan SoS Update*, the SoS SE evaluates the SoS priorities, options and backlog of SoS changes to define the plan for the next SoS upgrade cycle.

The major activities typically implemented in *Plan SoS Update* are shown in Figure 9. Once a set of priority needs are identified, an analysis of the current SoS architecture and supporting systems is conducted to identify options for making changes in the SoS to address the need. This is done in close cooperation with the constituent systems since they will be implementing any changes and they know the technical and programmatic considerations for the various options. Based on this collaborative analysis, a plan for the SoS update is formulated and coordinated.

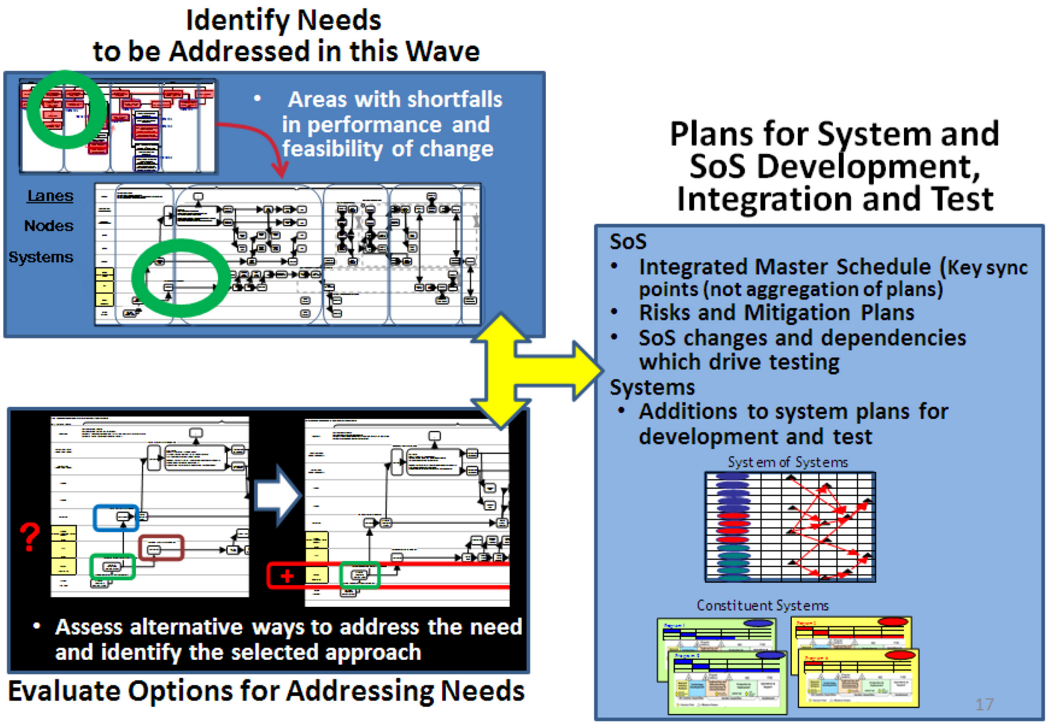

Figure 9: Major Activities Typically Supporting Planning the SoS Update[17].

Since this includes identifying the user needs to be addressed in this iteration, evaluating the options for addressing these needs, and developing the plans for the updates, the following artifacts are addressed at this step in the lifecycle:

- An **allocated baseline** is created for the update.
- **Risks and mitigations** are identified.
- **Agreements** are developed.
- Implementation and integration and **test plans** are created.
- An **Integrated Master Schedule** (IMS) is developed for update.
- The SoS **master plan** is updated.
- Update **SoS technical baselines** and **SoS requirements space**.

[17] Ibid.

3.7 Implement SoS Update

Finally in the *Implement SoS Update* step, the plans are executed and changes in the systems are implemented and tested by the systems as part of their systems development and engineering processes. During this step, the SoS systems engineer, monitors implementations at the constituent system level and plans and conducts SoS level testing, resulting in a new SoS product baseline, as shown in Figure 10.

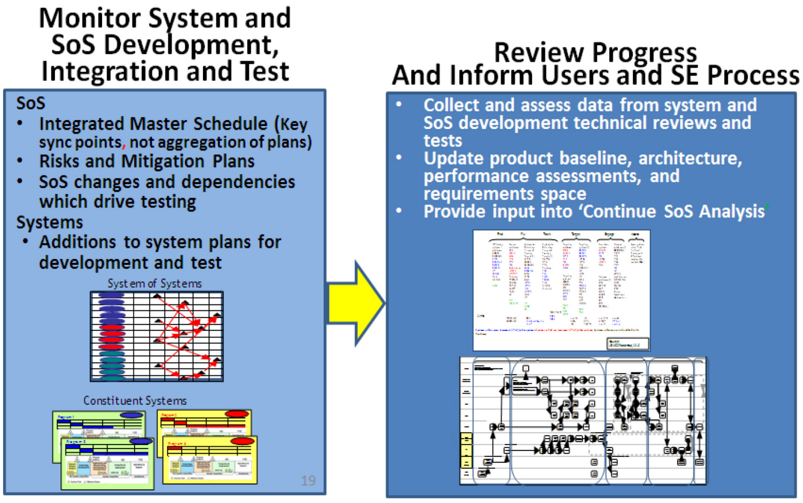

Figure 10: Major Activities Typically Supporting Implementation of SoS Update[18].

The systems implement and test changes at their level while the SoS SE team monitors progress and updates the Integrated Master Schedule (IMS) for the increment. The SoS SE team leads SoS integration and test, developing data on SoS performance and any unanticipated factors encountered. Artifacts containing information key to this step include:

- Technical plans and SE planning elements;
- IMS;
- SoS requirements space;
- SoS performance data;
- SoS technical baselines (more specifically, the product baseline); and
- Implementation risks.

3.8 Continued Evolution of the SoS

Looking back at Figure 4, note that this is an ongoing process and that there are regular feedback loops among the life cycle steps as emphasized in Figure 11.

[18] Ibid.

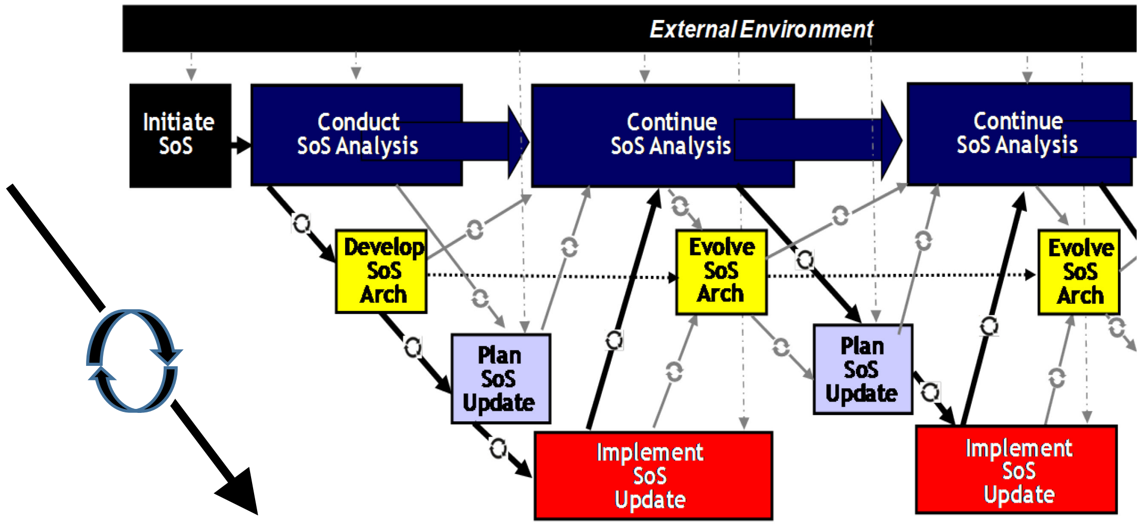

Figure 11: SoS Lifecycle Model Includes Regular Feedback Among Steps in the Model.

This is intended to convey that SoS is typically an ongoing adaptive evolutionary process which addresses the fact that most SoS are subject to multiple independent changes which affect the SoS performance and evolution.

4.0 SUMMARY

This paper has examined systems engineering for systems of systems from the perspective of the systems engineering life cycle as it applies to the particular circumstances of systems of systems. Based on the draft SoS Annex of the ISO 15288 systems engineering standard, the paper has presented a discussion of how the current SE life cycle processes are applied to SoS. The paper has also presented a life cycle, implementer's view of systems engineering of SoS based on an iterative approach to SoS evolution based on the changes made in the constituent systems supporting the system of systems.