



Calhoun: The NPS Institutional Archive
DSpace Repository

Faculty and Researchers

Faculty and Researchers' Publications

2016

The system of systems engineering and integration "Vee" model

Vaneman, W.K.

Vaneman, W.K., 2016, April. The system of systems engineering and integration "Vee" model. In Systems Conference (SysCon), 2016 Annual IEEE (pp. 1-7). IEEE. Link <http://hdl.handle.net/10945/58172>

Downloaded from NPS Archive: Calhoun



Calhoun is a project of the Dudley Knox Library at NPS, furthering the precepts and goals of open government and government transparency. All information contained herein has been approved for release by the NPS Public Affairs Officer.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

<http://www.nps.edu/library>

The System of Systems Engineering and Integration “Vee” Model

Warren K. Vaneman, Ph.D.
Systems Engineering Department
Naval Postgraduate School
Monterey, CA

Abstract— In the twenty-first century, mission success will require unprecedented interoperability among disparate constituent systems resulting in a System of Systems (SoS). Unlike traditional systems engineering where systems are created based on a set of user needs, a SoS is composed of multiple constituent systems, at various stages within their lifecycles (i.e. new start systems, systems in-development, legacy systems), to satisfy needed mission capabilities. System of Systems Engineering has been exploring appropriate methods and processes for almost two decades. This paper takes the discussion further by introducing a System of Systems Engineering and Integration (SoSE&I) methodology. SoSE&I is the planning, analyzing, and integrating of constituent systems into an SoS capability greater than the sum of those individual systems. The SoSE&I “Vee” process model is introduced and discusses how it is used to engineer the SoS throughout its lifecycle to increase systems integration and interoperability, and directly impact the operational success.

Keywords— *System of Systems; System of Systems Engineering and Integration; Systems Engineering “Vee” Model*

I. INTRODUCTION

“Our current system is like a machine to which we just keep adding important and wanted items but without a cohesive strategy for an elegant, interwoven system. Considered on their own, the addition and growth of individual elements may be useful. But when ownership organizations do not see how their contribution fits into the whole and think their element is an end-state in itself, effective communication and executions are inhibited.” [1]

With the exponential growth of computing, networks, and communications, twenty-first century systems are more highly capable, and more complex than ever before. Since its inception, systems engineering focused on the individual systems and platforms resulting in local optimization. With the complexity and pace of change in technology, it is evident that the systems engineering community must move away from the platform-centric perspective, and adopt a more comprehensive Systems of Systems (SoS) approach to ensure integrated and interoperable environment to achieve mission success.

One of the major challenges of establishing a SoS is, often the constituent systems have been developed years before it was envisioned that it would ever be required to function together in an interoperable environment. Therefore, most

constituent systems were designed to meet mission needs independent of the SoS. The challenge is integrating these disparate constituent systems, each at various points within its own lifecycle (i.e. new starts, systems in-development, and legacy systems), into a SoS that fulfills a capability that is not possible by a single system alone.

Traditional systems engineering includes defining and enforcing technical standards through architecting, analysis, and requirements development and management; making unbiased independent technical decisions; providing stewardship of technical capabilities; and ensuring reliable and safe operations of systems within the domain. The development and integration of a SoS requires a shift to a comprehensive SoS Engineering and Integration (SoSE&I) approach that begins with proven systems engineering methods to articulate the SoS architecture, requirements and interface standards that will govern development and procurement of future systems. Testing, evaluation and certification against the SoS architecture will ensure that user mission needs have been met prior to implementation into the operational environment. The SoSE&I process has been defined to increase the chances a SoS will achieve the desired capabilities with improved cost efficiency, increased quality and flexibility, and reduced time to field the new capabilities. Table 1 portrays the differences between traditional systems engineering and SoSE&I.

The remainder of this paper presents a SoSE&I approach to support mission requirements through: defining and controlling a managed SoS baseline that directly relates to the needed capabilities; establishes a test, evaluation, and certification methodology to evaluate delivered capabilities in the context of mission performance; and formalizes a method of governance that puts engineering discipline and rigor into SoS and system design and investment decisions. This approach is well-suited for directed and acknowledged SoS, but can be applicable with collaborative SoS given the proper governance structure.

II. SYSTEM OF SYSTEMS BACKGROUND

A System of Systems (SoS) is set or arrangement of systems that results when independent, and task-oriented systems are integrated into a larger systems construct, that delivers unique capabilities and functions in support of missions that cannot be achieved by individual systems alone. A SoS exists when constituent systems: (1) Can operate independently from the SoS; (2) Have lifecycles that are individual managed; and, (3)

Table 1. Traditional Systems Engineering vs. System of Systems Engineering.

	Traditional Systems Engineering	Systems of Systems Engineering
Purpose	Development of a single system to meet stakeholder requirements and defined performance	Evolving new SoS capability by leveraging synergies of legacy systems
System Architectures	System architecture established early in lifecycle and remains relatively stable	Dynamic reconfiguration of architecture as needs change; use of service oriented architecture approach as enabler
System Interoperability	Defines and implements specific interface requirements to integrate components in system	Component systems can operate independently of SoS in a useful manner. Protocols and standards essential to enable interposable systems
System “ilities”	Reliability, Maintainability, Availability are typical “ilities”	Added “ilities” such as Flexibility, Adaptability, Composability
Acquisition & Management	Centralized acquisition and management of the system	Component systems separately acquired and continue to be managed as independent systems
Anticipation of Needs	Concept phase activity to determine system needs	Intense concept phase analysis followed by continuous anticipation, aided by ongoing experimentation

Come together to achieve a capability that is unrealized by a single system alone [2].

System of Systems are often confused with complex systems. A complex system consists of systems whose elements are connected through diverse and complex interconnections. For example, an aircraft is a complex system with sub-systems so intricate that they are often thought to be systems by their developers (e.g. aircraft engines).

While many complex systems often behave like SoS, there are two primary differences. First the lifecycle of a complex system follows the same “cradle to grave” lifecycle that has been defined for a system – from the definition of need to the systems retirement. The lifecycle of a SoS is from the definition of the needed capability to a point in time when that capability is no longer required (e.g. the Ballistic Missile Defense Capability). Thus, during the lifecycle of a SoS, constituent systems periodically join and depart the SoS.

The second difference is in the definition of system boundaries. While system boundaries can be difficult to characterize, they ultimately can be defined, thus engineered and managed. In a SoS, boundaries are often difficult to define due to the fluidness of the composition of systems that comprise that SoS.

System of Systems is also confused with a Family of Systems (FoS). A FoS is a set of systems that provide similar capabilities through different approaches to achieve similar or complementary effects [3]. The primary difference in a SoS and FoS is in the operations and management of the constituent systems. Systems within a FoS are engineered, managed, and maintained using the same processes, but they do not necessarily operate together as constituents systems in a SoS would. For example, a fleet of aircraft work towards a common goal while operating as a single system (e.g. the transportation of passengers), but are maintained by the same processes and share a common logistical supply chain.

The term “Enterprise Systems” are often erroneously used to describe a SoS. Enterprise Systems are complex, socio-technical systems that comprise interdependent resources of people, information, and technology that must interact with each other and their environment in support of a common mission [4]. An organization may be considered an enterprise system, that may include a variety of systems, and SoS, to accomplish its goals.

A system simultaneously may be a member of a FoS, SoS, and Enterprise System. For example, fighter aircraft from a certain military service (i.e. Enterprise Systems), will be managed against a common system’s baseline (i.e. FoS), but may not necessarily operate together. When some of the aircraft are assigned to an air wing, they operate together forming a capability that cannot be achieved by a single system alone, thereby forming a SoS. Fig. 1 shows the relationship between an Enterprise System, a SoS, a FoS and a

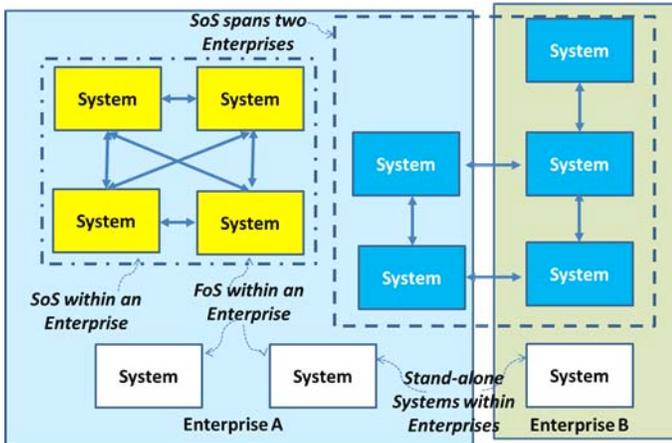


Fig. 1. Relationship between Enterprise Systems, SoS, FoS, and Systems [5].

system. Note, a SoS may be comprised of constituent systems from multiple enterprises [5].

There are four recognized SoS types [2, 3]:

- Virtual – SoS that lacks a central management authority, and a centrally agreed upon purpose. This type of SoS must rely on invisible mechanisms to maintain it. (e.g. Internet.)
- Collaborative – SoS where component systems interact voluntarily to fulfill agreed upon central purposes. The central players collectively decide how to provide or deny service, thereby providing a means for maintaining and enforcing standards. (e.g. consortium of university laboratories.)
- Acknowledged – SoS component systems have recognized goals and objectives, a designated manager, and resources. However, the constituent systems retain independent ownership, objectives, funding, development, and sustainment. (e.g. Navy carrier strike group that is made up of various ships and aircraft, linked together through voice and data communication networks.)
- Directed – SoS is designed, built, and managed to fulfill a central purpose, usually over a long period of time. Component systems can operate independently, but normally work as an SoS. (e.g. Satellite and ground station tasking, collection, communication, and processing activities.)

System of Systems Engineering and Integration (SoSE&I) is the planning, analyzing, and integrating of constituent systems into a SoS capability greater than the sum of those systems. SoSE&I differs from other definitions of Systems of Systems Engineering (SoSE) in that it spans the entire lifecycle of the SoS, placing an equal emphasis on each phase. Traditional SoSE concentrates on the planning activities early in the SoS lifecycle.

The key elements of SoSE&I are:

- Definition and control of a managed SoS baseline that directly tracks to delivered capabilities;
- An established SoS validation, verification, and certification process to evaluate delivered capabilities in context of mission performance;
- A formal method of governance and change control that puts discipline and rigor into investment decisions at the SoS level.

III. THE SOSE&I METHODOLOGY

The SoSE&I methodology is built on a Systems Engineering “Vee” process model. An abridged SoSE&I “Vee” is shown in Fig. 2, and consists of four primary tenets. The upper-left side of the “Vee”, referred to as SoS Architecture & Requirements Development, includes the early engineering and planning activities to include CONOPS development, architecture definition, architectures analysis, and SoS requirements derivation. The bottom of the “Vee”, referred to as “Systems Design & Development,” includes the traditional systems engineering activities for individual systems. Multiple “Vees” are depicted here to represent that many systems are being developed and managed in parallel, with each system at a unique point in its lifecycle. The upper-right side of the “Vee”, referred to as “Mission Assurance”, includes SoS interoperability and certification, SoS deployment, and operation and maintenance. The fourth area is the “Vee” is governance – policy processes that are overarching to all areas of the “Vee.”

Recently, the Systems Engineering “Vee” has been criticized as a sequential process model that lacks interaction between the lifecycle phases. While the “Vee” model does not explicitly show the interactions between the phases, those interactions are assumed to exist based on the structure of the model. The activities on the left-hand side of the “Vee” represent a top-down engineering approach, while the activities on the right-hand side represent a bottom-up approach. The “Vee” is symmetrically structured, with equal and corresponding activities on both the left and right sides of the model. Thus, the SoS requirements developed on the left-side of the “Vee,” are demonstrated and certified on the right-hand side.

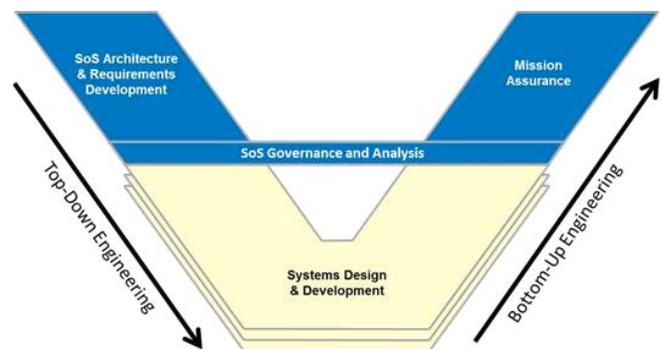


Fig. 2. The SoSE&I “Vee.”

A. SoS Architecture and Requirements Development

The SoSE&I “Vee” begins at the upper-left side with SoS Architecture & Requirements Development. In this phase the user needs are defined, and then transformed into technical requirements that can be executed by the system program office. The activities performed are depicted in Fig. 2 and further described in this section.

The Capability Collection/Customer Interface activity represents the primary interface between the Chief Architect, or Chief Engineer, and the stakeholders. The objective is to provide a clear and direct information exchange with a focus on the user mission needs and expectations in the context of their mission, and the environment that the mission is performed in. One outcome of this stakeholder interaction is a Design Reference Mission (DRM) - defining the mission problem, capability needs, and the threat and operating environment that the SoS will operate in [6]. It does not prescribe a particular SoS or system solution. Without this context, the best designed system could fail to meet expectations when implemented in the operational environment.

During the Capability Assessment and Analysis Phase, capability needs are extracted, organized, and documented in support of the capabilities-based acquisition process. The capabilities are analyzed to determine if a solution already exist, or if the desired state can be achieved by changing doctrine or policies, a re-organization of the current system structure or composition, training modification to enhance skills of the personnel, or modifying exiting systems.

The next activity, the SoS Architecture Development & Analysis Phase transforms the user’s capability needs and perspectives into static and dynamic architectural views. The architecture is defined using a mission-based, top-down approach which yields mission threads that consider the architecture from an end-to-end mission perspective. Static architectural views are developed and explored for functionality, interfaces, and standards. This activity requires collaboration amongst stakeholders at all levels to capture and organize the information in an efficient manner for supporting a variety of engineering and analysis activities. Using Model-Based Systems Engineering (MBSE) methods allows the SoS architect to manage relationships between operations,

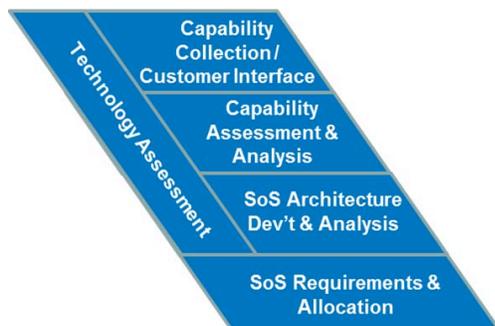


Fig. 3. SoS Architecture and Requirements Development.

functions and the associated systems in a variety of operational contexts to efficiently evaluate the SoS. Dynamic architectural views are constructed and used to analyze and forecast SoS performance, identify gaps, bottlenecks, and other constraints within the architecture, and explore solution alternatives to mitigate the issues found. These activities are performed in the context of the defined approved mission threads and relate back to the goals and objectives to meet the stakeholder’s mission needs.

System of systems architecting is an iterative process required throughout the entire SoS lifecycle. The SoS architectures provide context over many epochs. Three typical SoS architectural epochs are: the “as fielded” representing the current architecture; the “to be” representing funded changes to the SoS that are still in development or have not been fielded with the SoS; and the “vision” that represents the likely evolution of the SoS given the forecast changes in the environmental conditions coupled with the predicted technological changes with the constituent systems.

The Requirements and Allocation Phase derives SoS requirements and interface specifications from the SoS architecture, thus maintaining traceability from the user’s capability needs, to the architecture and analysis, to the requirements. Those SoS requirements are then allocated to the appropriate system program office, for inclusion in acquisition plans and program baselines. The allocated SoS requirements are managed under formal configuration control, and tracked for progress throughout the lifecycle of each constituent system.

Technology Assessment activities monitor emerging technologies, technology standards, and forecast emerging trends that can be employed in a SoS. Since a SoS is a dynamic technical environment - with new technologies and systems entering the SoS, and other systems from the SoS or being retired – decision-makers must understand how the SoS changes, and how it can support a given mission, over time. This activity also develops strategies on how to technically preserve and advance the SoS throughout it’s lifecycle.

The activities within SoS Architecture and Requirements benefit the SoS by: providing a comprehensive plan to align systems that are meant to work together for mission success; provides a foundation from which stakeholders can prioritize capability needs and budget and schedule issues; and establishes an overarching requirements baseline to improve integration and interoperability across the SoS.

While there are many benefits to the SoS Architecture and Requirements approach, there are also challenges. Frequently the SoS organization is steeped in traditional systems engineering processes. It is often difficult to transition to SoSE&I processes while continuing to deliver systems oriented analysis results.

B. SoSE&I Role in Systems Design and Development.

The bottom of the “Vee”, depicted in Fig. 4, represents the systems engineering activities that are performed by the developers of the constituent systems. Several individual system “Vees” are depicted to illustrate that many systems are

developed and managed concurrently, with each system at different maturity levels within its own lifecycle.

In this phase the focus is on the development, sustainment, and management of individual systems. However, two important SoSE&I activities must occur to ensure a successful SoS. First, the SoSE&I team must have sufficient insight into the constituent system’s development, sustainment, and management to ensure the systems are compatible with the SoS. This is an important point because as decisions are made for individual systems, it is easy for those decisions to be contrary to the stated mission of the SoS. When individual system decisions impact the interoperability of the system to be able to work with the SoS, the decision must be elevated to the SoS-level for resolution through the governance process.

Second, understanding constituent system functionality and programmatic issues is critical since constituent systems in a SoS rely on each other to achieve mission success. Issues such as system schedule delays, or technology issues leading to capability shortfalls, are critical since other systems that depend on upstream information may not be able to fulfill their missions within a SoS. System retirements are also an area of concern because a premature decommissioning may yield gaps that inhibit the SoS. A strong governance process must be in place to communicate and manage changes during the development cycle while maximizing SoS and mission effectiveness.

The SoS will benefit through the establishment of a framework for better coordination among individual systems and programs. For example, each program office identifies, tracks, and mitigates risks for their individual constituent system. By considering risks from a holistic SoS perspective, integration and interoperability risks can be identified, thereby providing systems engineers and program managers with insights into risks that occur in another constituent system, but negatively impact their system.

The goal of the SoSE&I in system development cycles is to best influence and impact SoS opportunities, and design flexibility into the constituents systems to best adapt to new interfaces thus extending functionality of the SoS.

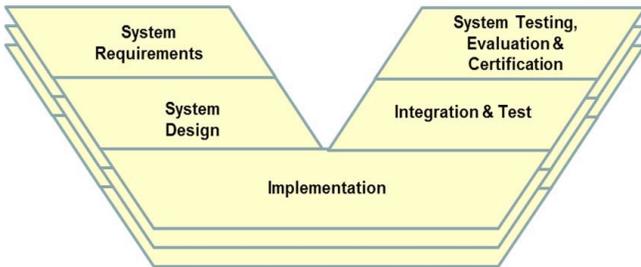


Fig. 4. SoSE&I Role in System Design and Development.

C. *Mission Assurance.*

The upper-right side of the “Vee” representing the SoS Mission Assurance activities is shown in Fig. 5. System of Systems Mission Assurance includes SoS interoperability and certification, deployment, operations and maintenance. The activities performed in this phase are complementary to the activities performed in the upper-left side of the “Vee.” The strength of mission assurance is the constant collaboration and coordination of major activities for the successful development and integration of the SoS.

The Mission Assurance phase presents some of the most challenges for the SoS because this is the phase where constituent systems come together to form the SoS. While this integration has likely been planned, simulated, and tested in a MBSE environment, this is the first time when true system performance is experienced. Integration of the SoS is challenging even when it is controlled.

SoS Interoperability and Certification focuses on the end-to-end functionality and performance across the SoS. One of the biggest challenges is how and when to integrate constituent systems into the SoS. If constituent systems are permitted to join the SoS asynchronously, predicting the behaviors and performance of the SoS becomes problematic because new constituent systems change the operational dynamics of the SoS. Since the cost of scheduling and conducting SoS-wide test and certification events can be disruptive to operations and cost prohibitive, it is impossible to test and certify the SoS on an on-going basis. Therefore a risk of negative consequences as these new constituent systems randomly join the SoS exists. One method that shows considerable promise for SoS integration is the Wave Model [7]. This model establishes entrance points when new capabilities can join a SoS, thereby better controlling the baseline, and allowing for better test and certification management. This disciplined approach discourages asynchronous system insertions, which could lead to a critical capability being delayed until the next entrance point.

By the time a constituent system is ready to join a SoS, traditional system-level validation and verification activities should be completed. Validating and verifying system-level



Fig. 5. Mission Assurance.

requirements is the responsibility of the system program office. System of Systems evaluation and certification is different from traditional system validation and verification because of the sheer size and complexity of most SoS. The term “evaluation and certification” is used for a SoS instead of “validation and verification” so not to confuse the rigorous testing of system requirements, with the capability focused evaluation of a SoS.

System of systems evaluation and certification is frequently performed by analysis because of the difficulty of bringing all relevant constituent systems together for a large demonstration event. As such, exercises and experiments are often used to provide insights into the operational performance, and can be used to identify interoperability problems prior to the SoS being deployed into an operational environment.

Acknowledged SoS are sometimes formed to satisfy unique missions of limited duration. It is during the deployment and execution of these missions when many of the constituent systems will be operating together for the first time. For example, many of the U.S. Military Units that assembled into a SoS for humanitarian relief in response to the 2004 Indian Ocean earthquake and tsunami did not previously operate together, and assembled for this one-time event. When such a SoS is assembled, innovative means to evaluate and predict performance must be employed. These means could include evaluation of SoS performance during operational exercises, and using MBSE to simulate and predict performance. In cases like this, it is important to define a strong governance model to be able to quickly mitigate issues after they are identified. In the 2004 Indian Ocean earthquake and tsunami example, the U.S. military’s strong command and control structure provided the governance model.

Another area that should be closely monitored is the emergence of new hazards as a result of constituent systems forming a SoS. Generally, hazards are a function of components at the lowest level of the system. The emergence of new hazards generally requires new functionality involving the control of new applications and interactions within a system [8].

System of Systems hazards are a function of the constituent systems. Often the sum of SoS hazards is greater than the sum of the constituent system hazards. The occurrence of new hazards within a SoS, whether known or inadvertent, are usually the result of new functionality or changes by integrating constituent systems into the SoS [8]. Current SoSE&I processes do not provide proper guidance on the detection and control of safety or hazard concerns. This is an area ripe for further research.

The goal of a SoS is to provide the functionality to address the real-world capability needs. Given the lifecycle of the typical SoS, constituent systems will periodically join and depart from the SoS for normal operational rotation, or for maintenance. The operations and maintenance phase develops a comprehensive plan to better align constituent systems operations, and ensure a constant level of SoS functionality is provided. Planning of this type can be challenging because constituent systems are often owned and controlled by several

different organizations with their own goals and priorities for their constituent systems. The comprehensive operations and maintenance plan identifies alternate constituent systems that can satisfy the functional requirements necessary for the SoS to continue providing full operational capabilities.

D. Governance

A cornerstone of an effective SoS is governance. Governance in this context means the set of rules, policies, and decision-making criteria that will guide the SoS to achieving its goals and objectives [9]. A solid governance plan is essential to: accelerated time to fielding; increased constituent system and SoS quality; increased SoS flexibility to rapidly adapt to changing conditions and threats; and, the identification of technical problems that will reduce mission effectiveness or increase SoS cost [10].

Governance asks three fundamental questions. First, what will be managed at the SoS level? The SoS architecture and requirements provide some insights to the answers to this question. Being able to distinguish which SoS elements should be governed at the system level, and which should be governed at the SoS level will help prevent an over-prescriptive governance structure.

The second question is, who has the accountability and decision rights across a broad set of stakeholders? This question determines the degree of participation, responsiveness, consensus, inclusiveness, and accountability needed in the governance strategy. It also guides how enforcement is managed within the SoS [9].

The third question is, how to implement the SoS governance structure effectively? The organizational structure, standards, policies, and the management environment must be understood to develop and implement effective governance. Governance must be consistent with the organization [9].

Prior to developing a governance structure, the type of SoS must be understood in-depth, as “one size does not fit all.” The level of involvement of constituent systems, transparency of decisions, and enforcement of standards varies considerably between virtual and directed SoS. However, good governance is needed to: identify clear boundaries of authority across a broad set of stakeholders; greater acceptance of newly implemented SoS Governance Authority; and, coordinated and transparent decision-making

IV. CONCLUSIONS

System of systems increase the complexity, scope, and cost of both the systems engineering and planning processes, and introduces the need to coordinate inter-program activities and manage agreements among multiple program managers and stakeholders who may not have a vested interest in the SoS. The problems that need to be addressed are large and complex, and not amenable to solutions of better systems engineering alone [10].

System of Systems Engineering and Integration provides the process to plan, engineer, and technically manage a SoS throughout the entire lifecycle. Fig. 6 shows the unabridged SoSE&I “Vee.” It is well-suited for use with directed and

acknowledged SoS, and possibly for collaborative SoS given the proper circumstances and governance structure. It can be applied to constituent systems at various phases in their lifecycles. In “new start” systems, SoSE&I will ensure that interoperability and integration is built into new programs and enforced through a governance structure throughout the development process. For systems that are in-development, but haven’t been fielded, SoSE&I can influence design trade decisions, and reduce risk, through early architecting and engineering. And in constituent systems that are already in service, SoSE&I can evaluate the impacts of planned upgrades, and how this systems will interact with new systems to achieve mission effectiveness.

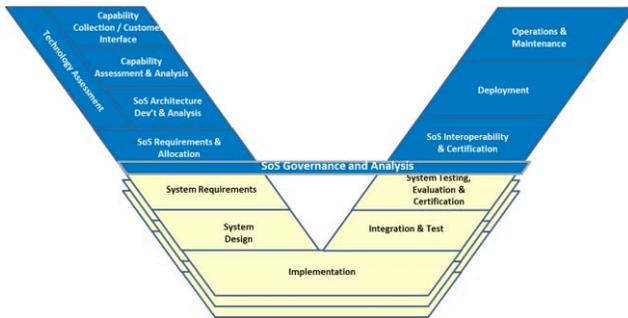


Fig. 6. Unabridged SoSE&I “Vee.”

REFERENCES

- [1] B. Gortney and H. Harris, “Minding the gaps,” U.S. Naval Institute Proceedings, pp. 36-40, May 2014.
- [2] M. Maier, “Architecting principles for system-of-systems,” Systems Engineering, vol 1, no 4, pp. 267-284, 1998.
- [3] Deputy Under Secretary of Defense (Acquisition and Technology), Systems engineering guide for systems of systems. Washington, D.C., 2008.
- [4] R.E. Giachetti, “Design of enterprise systems,” New York, NY: CRC Press, 2010.
- [5] M. Jamshidi, “System of Systems Engineering: Principles and applications,” New York, NY: CRC Press, 2009.
- [6] F.R. Skolnick and P.G. Wilkins, “Laying the foundation for successful systems engineering,” Johns Hopkins APL Technical Digest, Vol. 21, No. 2 (pp. 208-216), 2000.
- [7] J. Dahmann, G. Rebovich, J. Lane, R. Lowry, and K. Baldwin, “An implementers’ view of systems engineering for system of systems,” Proceedings of the IEEE Systems Conference, April 2007.
- [8] North Atlantic Treaty Organization, “AOP-52 Guidance on software design and assessment of munitions – Related Computing Systems (1st ed.)”, North Atlantic Treaty Organization Standardization Agency, 2008.
- [9] W.K. Vaneman and R. D. Jaskot, “A criteria-based framework for establishing system of systems governance,” Proceedings of the 7th Annual IEEE Systems Conference, April 2013.
- [10] W. K. Vaneman and R. Budka, “Defining a system of systems engineering and integration approach to address the Navy’s Information Technology Technical Authority,” Proceedings of the INCOSE International Symposium, June 2013.