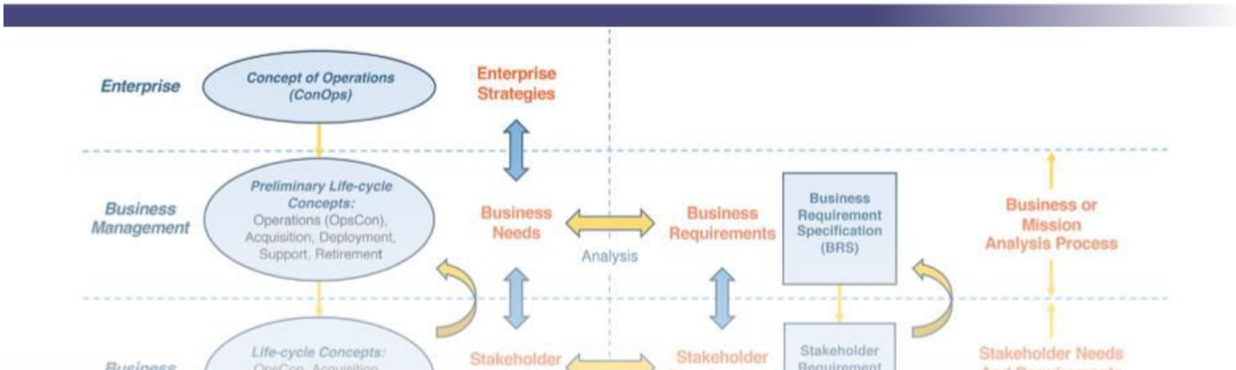
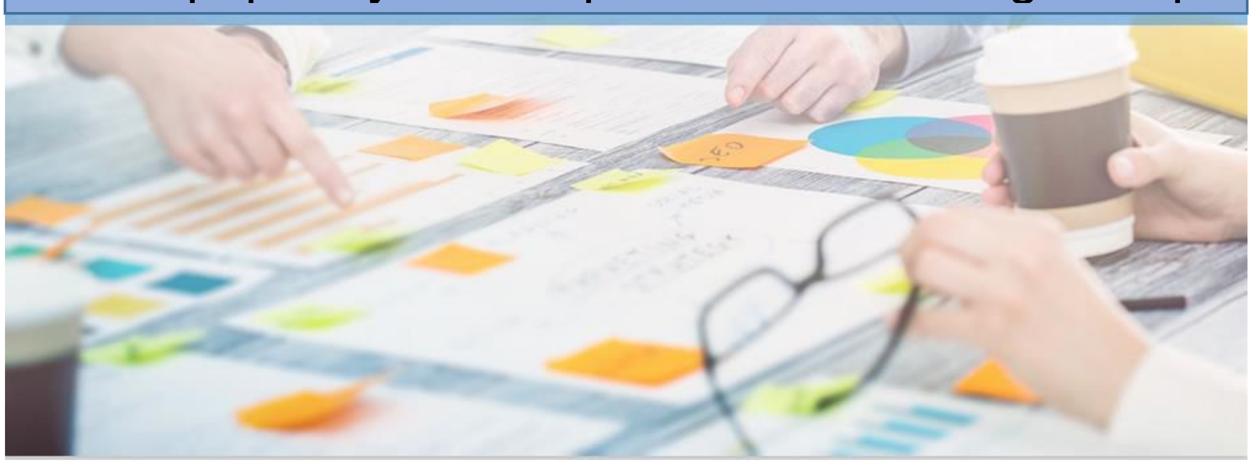


Integrated Data as a Foundation of Systems Engineering

December 2018

Whitepaper by the Requirements Working Group



Copyright Information

Additional Copies/General Information

Copies of *Integrated Data as a Foundation for Systems Engineering*, as well as any other INCOSE document can be obtained from the INCOSE Central Office. General information on INCOSE, the Requirements Working Group, any other INCOSE working group, or membership may also be obtained from the INCOSE Central Office at:

International Council on Systems Engineering
7670 Opportunity Road, Suite 220
San Diego, California 92111-2222 | USA
E-mail: info@incose.org

Telephone: +1 858-541-1725
Toll Free Phone (US): 800-366-1164
Fax: +1 858-541-1728
Web Site: http://www.incose.org

INTEGRATED DATA AS A FOUNDATION OF SYSTEMS ENGINEERING

Document No.: INCOSE-TP-2018-001-01.0
Version/Revision: 1.0
Date: December 31, 2018

Prepared by:
Requirements Working Group
International Council on Systems Engineering (INCOSE)
7670 Opportunity Road, Suite 220,
San Diego, California 92111-2222 USA

REVISION HISTORY

Revision	Revision Date	Change Description & Rationale
0	01/19/2017	Original copy for IW 2017
0.1	01/31/2017	Updated version based on comments received at IW 2017
0.2	02/07/2017	Updated version based on first review IW 2017 RWG review
0.3	03/15/2017	Updated version based on second review IW 2017 RWG review
0.4	04/17/2017	Updated version presented to Tech Ops for review
0.5	04/23/2018	Updated version based on comments from Tech Ops Review
1.0	12/31/2018	First release – updated based on second Tech Ops Review cycle

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

PREFACE

This whitepaper has been prepared and produced by a volunteer group of contributors within the Requirements Working Group (RWG) of the International Council on Systems Engineering (INCOSE). This whitepaper is an outgrowth from discussions concerning the RWG participation in and collaborating with the Model-Based Systems Engineering (MBSE) Initiative that occurred within the RWG sessions during INCOSE IW 2016 and IW2017 in Torrance, CA and subsequent communications between the authors, members of the RWG, and members of other INCOSE Working Groups.

The RWG is producing this whitepaper from the perspective that requirements, along with all work products (models, designs, documents, diagrams, drawings, etc.) generated during the performance of system life cycle process activities are represented by underlying sets of data and information. Data and information in these sets need be able to be accessible and shared between organizations and project management (PM) and systems engineering (SE) tools used within an organization as well as external organizations. This sharing will help to ensure consistency, correctness, and completeness of work products developed across all system life cycle stages. To enable this sharing of data and information, the project/program needs to manage these sets of data in a way that enables sharing of the data across all system life cycle stages.

Key attributes of the envisioned for a project's shareable sets of data include:

- There is a defined ontology to which all the sets of data within the project are consistent.
- There is a master project schema to which all the sets of data within the project comply.
- Data representing work products are appropriately linked across system life cycle stages.
- Relationships are defined that reflect dependencies and interactions between entities and associated work products.
- The PM and SE tools used to generate the sets of PM & SE data comply fully with the industry interoperability standards as discussed in section 4.7.2.

Using this perspective, **integrated (or federated), shareable sets of data can be viewed as a foundation of Systems Engineering**. From this data-centric perspective of SE, there are many key benefits that will aid organizations in successfully meeting the challenges associated with today's ever increasingly complex systems, meeting the intent of the MBSE Initiative, and helping organizations move toward INCOSE's Vision 2025. A data-centric perspective of SE also aligns well with the development in other domains of expertise such as Building Information Modelling (BIM) being incorporated in the construction industry (NBS 2018).

While practicing SE from this data-centric perspective affects all INCOSE Working Groups (WGs), the following WGs are key stakeholders: Life Cycle Management, Measurement, MBSE Initiative, MBSE Patterns, Ontology, Process Improvement, Requirements, SE Effectiveness, SE Transformation, and Tool Integration and Model Life Cycle Management. The activities of these working groups are not only enablers to the practice of SE from a data-centric perspective but many of these groups are directing their efforts on how to perform SE with a focus on the essential data/information/knowledge needed to perform SE as is advocated within this whitepaper.

AUTHORS

The principal authors of this Guide are:

Lou Wheatcraft, Requirements Experts, USA
Michael Ryan, University of New South Wales, Canberra, Australia

MAJOR CONTRIBUTORS

Those who made a significant contribution to the generation of this Guide are:

Kathy Baksa, Pratt & Whitney, USA
Jeremy Dick, Integrate Systems Engineering Ltd, United Kingdom
Rick Zinni, Harris Corporation, USA
Carl Svensson, Swedish Defense Materiel Administration, FMV, Sweden
David Long, Vitech, USA

Reviewers

In addition to the authors and major contributors, this whitepaper was provided for review and comment to the following attendees of the RWG sessions at IW2017, members of other WGs, and Technical Operations:

Celeste A. Drewien, Sandia National Labs, USA
Gauthier Fanmuy, Dassault Systems, France
Glenn A. Thibodeau, Autoliv, USA
Juan Llorens, Universidad Carlos III de Madrid, Spain
Leo van Ruihven, Croonwolter&dros, Netherlands
Luis M. Alonso, The REUSE Company, Spain
Marv Mealman, Whirlpool Corp, USA
Melissa Wallace, USA
Mia Zagar, Shell, USA
Michael E. Pafford, USA
Michael J. Riccio, The Aerospace Corporation, USA
Robert McAfoos, Barrios Technology, USA
Robert A. Noel, The Boeing Company, USA
Rudolph Oosthuizen, CSIR, South Africa
Steve Turner, WSP, United Kingdom
Tasha Penner, USA
Thomas V. Traczyk, Raytheon Corporation, USA
Tracy A. Scott-Parry, UTC Aerospace Systems, USA
Troy A. Peterson, SSI, USA
Warren Vaneman, Naval Postgraduate School, USA

1 Introduction

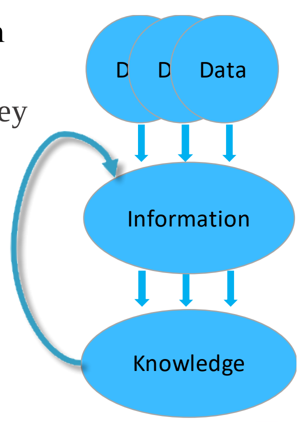
1.1 Purpose and Scope

While models are one foundational element of Systems Engineering (SE), there is more to SE than the models themselves. Models are represented by data and information as are other SE work products. In this context, models and other work products are either projections of the same data and information or represented by data and information generated from other system life cycle process activities. To effectively manage ever increasing complex systems of the future there are benefits to managing this underlying data and information in such a way it can be shared across the system life cycle process activities, shared between the various SE tools used to create and manage this data and information, and shared between organizations involved in the development and operations of the system of interest. This sharing will help ensure correctness, consistency, and completeness of the data and information typical of our ever increasingly complex systems.

The practice of SE is often viewed from many perspectives. Similar to the old story of the blind men and the elephant, SE cannot be effectively practiced when viewed from just one perspective (requirements, models, patterns, standards, industry specific application, etc.). To successfully practice SE, wise systems engineers recognize and use each perspective as appropriate to the activity they are performing. The perspective of this whitepaper addresses the intent of the MBSE Initiative by presenting a broader, data-centric view of SE. In this context, the development of a system of interest involves the creation and use of many work products, including models, generated during the execution of the system life cycle process activities.

The goal, as advocated in this whitepaper, is to have the capability to integrate PM and SE data and information into integrated (federated) shareable sets of data. (*Integrated vs Federated databases are discussed in more detail in Section 3.4.3*) These sets of data not only represent an architectural model of the system under development, but also a model of the system life cycle process activities and resulting work products that can be used to more effectively manage the system engineering efforts across all system life cycle stages. The degree to which the data and information is integrated is driven by the needs of the organization and programs from a business and technical perspective based on the size and complexity of their programs, their diversity and complexity of supply chains, and types of engineering information that comprises a technical baseline for their system of interest.

In discussing "data", it is important to understand the relationship between data, information, knowledge, and wisdom. (Bellinger 2004-2),



- ***Data***: individual facts/bits/datum without context, by themselves they have little value
- ***Information***: data with context allowing users to expand on the data and gain information, insight, and knowledge
- ***Knowledge***: aggregation of information, helps apply the information allowing users to define context, patterns, correlations, causations, inform standards, etc.
- ***Wisdom***: knowledge plus experience

The PM and SE tools used to generate and manage the various work products and underlying data and information contribute to the understanding of the context of this data. This context results in information. This information represents an information model of the system being developed as well as provides valuable rationale and insights developed while executing the system life cycle process activities involved in engineering the system. In practicing SE, the systems engineer's emphasis needs to be on the data and information shared across system life cycle stages rather than on the individual system life cycle activity processes and associated work products. Combining the systems engineer's experience and knowledge with the information contained in the integrated, shareable sets of data enables the systems engineer to use their wisdom to successfully deliver winning products – products that deliver what is needed, within cost and schedule, with the desired quality. Accepting this premise, it is useful to view SE from a data-centric perspective.

The authors believe this data-centric perspective of SE provides an alternate lens through which to acquire a more complete understanding of the system life cycle process activities and resulting work products and underlying data and information needed to manage the development of increasingly complex systems of the future. This perspective aids in acquiring a practical understanding of the system life cycle process activities from the perspective of not only models, but all work products that are generated from activities conducted during each of the system life cycle process activities and the underlying data and information used to represent these work products.

The purpose of this whitepaper is to help organizations implement Systems Engineering by addressing the following areas:

- Help organizations understand that the data-centric approach to practicing SE advocated in this whitepaper is a major part of MBSE.
- Outline the importance for an organization to keep track and manage data and information across all the system life cycle process activities.
- Provide guidance regarding the management principles and decisions important to the implementation and management of SE from a data-centric perspective.
- Propose a method that can be used to measure and benchmark an organization's capability to practice SE from a data-centric perspective.
- Furnish guidance and a roadmap to help an organization move towards practicing SE from a data-centric perspective.

Expanding on the concept of SE from a data-centric perspective, the goals of this whitepaper are to:

- Present a broader data-centric perspective of SE that meets the intent the MBSE initiative.
- Provide organizations an understanding that integrated (federated), shareable sets of data are a key foundation of SE.
- Provide an integrated context of the various perspectives of MBSE which can be built upon and expanded on by the various INCOSE Working Groups, SE tool vendors, and practitioners of SE.

The overall goal is to make this whitepaper a useful product to help organizations implement the level of SE capability that best fits their needs.

1.2 Audience

The intended audience of this whitepaper includes project and product managers and systems engineers who are stakeholders in activities defined by the SE discipline and are thinking about, or are in the process of, implementing SE within their organization. This whitepaper will help those who are wondering how to successfully implement the intent of the MBSE Initiative within their organization and those that are interested in maturing their current SE capabilities toward a more data-centric implementation of SE - irrespective of the size and complexity of the system under development and the size and culture of the organization developing the system.

From a requirements perspective, this whitepaper is also targeted to those who have been, or are currently, focused on defining, documenting, and managing requirements as a distinct and separate, siloed activity from other system life cycle process activities. From a tool vendor perspective, this whitepaper is targeted to those whose tools do not provide the capability to integrate and share requirements and the other work products and their underlying data and information across all system life cycle process activities. While these approaches may have worked in the past and may work for some present system development efforts, it is doubtful these approaches will allow organizations to meet the future challenges of increasingly complex systems.

1.3 Organization

This whitepaper is organized as follows:

- ***Section 2.0*** addresses the need for SE and the benefits of adopting SE from a data-centric perspective. A list of challenges that need to be addressed due to the increasingly complex systems is presented followed by a list of benefits organizations can realize by practicing SE from a data-centric perspective. A Questionnaire is provided in the appendices to help organizations identify issues and risks they may be having which can be mitigated by practicing SE from a data-centric perspective.
- ***Section 3.0*** introduces and defines the concept of practicing SE from a data-centric perspective. The section begins by discussing the SE work products and underlying data and information that are generated as part of each of the system life cycle process activities. Next the questions: "What is a model?" and "What is model-based SE (MBSE)?" are addressed from a data-centric perspective. An introduction to the concepts of ontology, schema, and integrated vs federated data is provided. Lastly, the concept of integrated/federated data as a foundation of SE is discussed followed by a revised definition of SE from a data-centric perspective.
- ***Section 4.0*** goes into more detail on what it means to practice SE from a data-centric perspective providing guidance that can be used to understand and successfully create and manage integrated/federated, shareable sets of data within an organization. This section starts with a discussion concerning the need for corporate management buy in and support needed to transition the organization from their present state to practicing SE from a data-centric perspective. Key concepts from big data are introduced including data governance, information technology, information management, and configuration management. Also included is a discussion concerning another key advantage to practicing SE from a data-centric perspective - the use of measures to help better manage the system development activities across all system life cycle stages. This section concludes with a description concerning the current state of most organizations concerning practicing SE and the path needed to move from the current state to a future state where the projects within an

organization practice SE from a data-centric perspective using integrated/federated, shareable sets of data.

- ***Section 5.0*** focuses on topics to help organizations develop a SE capability that meets the needs of their organization. To aid in this journey, SE capability levels (SCLs) are presented to help organizations assess what their current SE capability is from an integrated, shareable sets of data perspective and provide a roadmap to get to their desired level of SE capability based on their organization's specific needs. Next, the selection of an SE toolset that is needed to implement the chosen SCL is discussed. The final topic in this section provides guidance to help sell to management the idea of moving toward a data-centric practice of SE within the organization.

2 THE NEED FOR SYSTEMS ENGINEERING

This section addresses the need for SE and the benefits of adopting SE from a data-centric perspective. A list of challenges that need to be addressed due to the increasingly complex systems is presented followed by a list of benefits organizations can realize by practicing SE from a data-centric perspective. A questionnaire is provided in the appendices to help organizations identify issues they may be having which can be mitigated by practicing SE from a data-centric perspective.

2.1 Meeting the Challenges of Increasingly Complex Systems

As stated in INCOSE Vision 2025 (INCOSE 2014), a constant throughout the evolution of systems engineering "is an ever-increasing complexity of systems which can be observed in terms of the number of system functions, components, and interfaces and their non-linear interactions and emergent properties. Each of these indicators of complexity has increased dramatically over the last fifty years and will continue to increase due to the capabilities that stakeholders are demanding and the advancement in technologies that enable these capabilities."

Often this complexity involves large-scale systems whose system life cycle process activities are distributed across many organizations and locations. An example of system complexity is the Boeing 787 (Malone et al 2016), for which over 30 companies based in countries around the world built large portions of the airplane. To help manage this complex system, Boeing developed a model that had >2,000 functions, >5,000 data flows, >1,000,000 data parameters, and >50,000,000 objects, with an average of three relationships per object, as well as ~1,000 geographically dispersed users involved in the modeling effort.

The authors have compiled the following list of major challenges practitioners need to address to successfully develop current systems as well as systems with increasing complexity in the future.

Practitioners need the capability to:

- Manage large number of work products and the underlying data and information that represents them electronically, rather than in printed, standalone, documents, diagrams, or drawings,
- Replace organizational "silos" with a more holistic organizational approach establishing a collaborative environment with a multidiscipline team that uses shareable sets of data to holistically integrate with coherence and consistency work products and their underlying data and information across disciplines, organizations, and system life cycle processes,
- Capture, integrate, manage, access, and share increasingly large sets of system engineering and program/project management (PM) data and information and associated interrelationships,
- Identify and manage dependencies across not only the system architecture but dependencies across disciplines and system life cycle work products and entities,
- Identify, define, and manage interactions (interfaces) between parts of the complex system architecture and between the system and the macro-system of which it is a part, no matter the complexity of the system under development,

- Track progress, identify at-risk activities, and take actions before these risks become problems that could impact cost, schedule, or the ability to deliver a product that meets stakeholder needs in the operational environment,
- Transition from a "gut" decision-making culture to a data-driven decision culture that is more effective and appropriate for managing complex systems. Data-driven decisions are not only driven by data and information but are managed within an appropriate toolset, along with supporting information as to why the decision was made,
- Integrate SE activities with the PM activities and resulting work products and underlying data and information to better manage cost, schedule, and risk.

Technology is evolving at a rapid rate, especially information technology, not only resulting in more complex systems, but also enabling the documentation, management, and integration of large sets of data that represent the many work products and underlying data and information generated as part of the system life cycle process activities.

In order to keep up with rapidly evolving technology, organizations need to continuously evolve their SE capabilities to address the challenges described above for increasingly complex systems.

2.2 Benefits of Implementing Systems Engineering From a Data-Centric Perspective

A data-centric perspective of SE complements the system life cycle process activities by enabling the opportunity for system development with increased quality, lower cost, and lower risk. Implementing a data-centric perspective enables organizations to realize the following benefits:

- Meet the challenges associated with increasing complexity for current and future systems discussed in the previous section.
- Provide greater consistency of all products because any single piece of design data and information can be expressed authoritatively within integrated/federated, shareable sets of data that can later be referred to by others for decisions or formation of other work products.
- Provide better visibility into the principle characteristics of the whole system because multiple views from a project's integrated/federated, shareable sets of data can be created that succinctly address specific stakeholder needs, concerns, and interests.
- Provide greater congruence and configuration management between documentation and reality. Differing views of the underlying SE data and information can be automatically generated into SE work products, reducing the effort to keep the work products and their underlying data and information up to date and consistent, resulting in work products that match the best available, current data and information.
- Establish "single source of truth". Single source of truth (SSoT) represents the official state or baseline version of data and information—regardless of what someone says or thinks, no matter what they "remember" or what perspective they have concerning what is being done or built or a decision that was made, if it isn't in the project's configuration managed, shareable sets of data, it isn't the truth. (*Requires all the underlying data and information to be maintained and kept current and consistent.*)
- Facilitate the navigation, traceability, and interrogation of data and information across all system life cycle stage activities. Managers and engineers can have access to the correct,

complete, and consistent data and information more quickly, and on an as-needed basis, without going through manual distribution or search processes.

- Enable the reuse of SE and PM work products and underlying data and information. Considerable time and expense can be saved when an organization can reuse SE and PM data and information and not have to start from scratch for each new project (brown field systems). This reuse ability is key to effective product line management.
- Facilitate the management of the stakeholder needs, requirement definition, design, build/code, and system verification and validation activities in an integrated, consistent manner. Data and information associated with verification and validation activities across all system life cycle process activities can have higher quality, and provide greater insight concerning the status of verification and validation activities. This makes it easier to show compliance and that stakeholder needs are being met.
- Reduce the costs associated with erroneous design and resulting rework. Analysis of the SE work products and underlying data and information can reveal a flaw or inconsistency as soon as it is created, enabling correction before downstream work is done, work that would be invalid, and costly and time consuming to correct if the upstream mistake were not corrected immediately. This also helps to avoid huge expenses associated with recalls, returns, warranty work, and negative comments on social media.
- Facilitate the identification of interactions (interfaces), helping to ensure the system of interest can be successfully integrated into the macro system it is a part and reducing integration issues and costly rework and schedule slips associated with these issues.
- Provide identification, management, interoperability, and integration of work products and underlying data and information across business or organizational elements needed to support program budget and schedule goals. For example, with the ability to metatag data, information, and work products, they can directly be linked to the WBS, budget, schedule, and risk management activities.
- Ensure data and information needed by programs and projects (e.g., for milestones, gate reviews, mission operations, risk mitigation, and anomaly investigations, decisions, and outcomes) are identified and managed to provide traceability of the data and information used in decision-making.
- Use measures to better manage SE activities across all system life cycle process activities. Measures allow managers and systems engineers to monitor trends, assess progress, identify issues to help ensure the system being developed will meet stakeholder needs and expectations. (See section 4.6 for a more detailed discussion on measures.)

Note 1: This list of benefits is derived from a similar list documented in National Aeronautical and Space Administration (NASA) "Expanded Guidance for NASA Systems Engineering", Volume 2, Chapter 8.2. (NASA 2016.

Note 2: "Organizations" refer to all organizations involved in a system's ecosystem both internal as well as external to the developing organization or the owner and operator of the system. This includes organizations that are part of a systems of systems development effort. A key premise of practicing SE from a data-centric perspective is 1) data from all systems in available, 2) all organizations involved form and comply with a common ontology, and 3) All organizations practicing SE from a data-centric perspective form sharable sets of data that fully adhere to interoperability standards such that the data and information can be shared between organizations and the tools they use.

Organizations need to develop a level of organizational SE capability that will enable them to realize the benefits listed above. Since one size doesn't fit all, an organization needs to assess the SE capabilities that best fit its domain, product line (degree of complexity), and culture. The level of SE capability an organization establishes needs to be tailored to the size and complexity of systems developed by the organization, whether small, medium, or large projects. (*A more detailed discussion on levels of SE capability is included in section 5.1*.)

3 SYSTEMS ENGINEERING – A DATA-CENTRIC PERSPECTIVE

This section introduces and defines the concept of practicing SE from a data-centric perspective. The section begins by discussing the SE work products and underlying data and information that are generated as part of each of the system life cycle process activities. Next, the questions: "What is a model?" and "What is model-based SE (MBSE)?" are addressed from a data-centric perspective. Next, the concept of integrated data as a foundation of SE is discussed with an introduction to the concepts of ontology, schema, and integrated vs federated data. Lastly, a revised definition of SE from a data-centric perspective is proposed.

3.1 Systems Engineering Work Products

ISO/IEC TR 24748-1, *System and Software Engineering—Life Cycle Management* (ISO/IEC TR 24748-1 2016) defines the following system life cycle stages: concept, development, production, utilization, support, and retirement. The INCOSE SE Handbook (INCOSE 2015), expands these six stages into thirty life cycle processes grouped into four broad areas: technical process, technical management processes, agreement processes, and organizational project-enabling processes as shown in Figure 1.

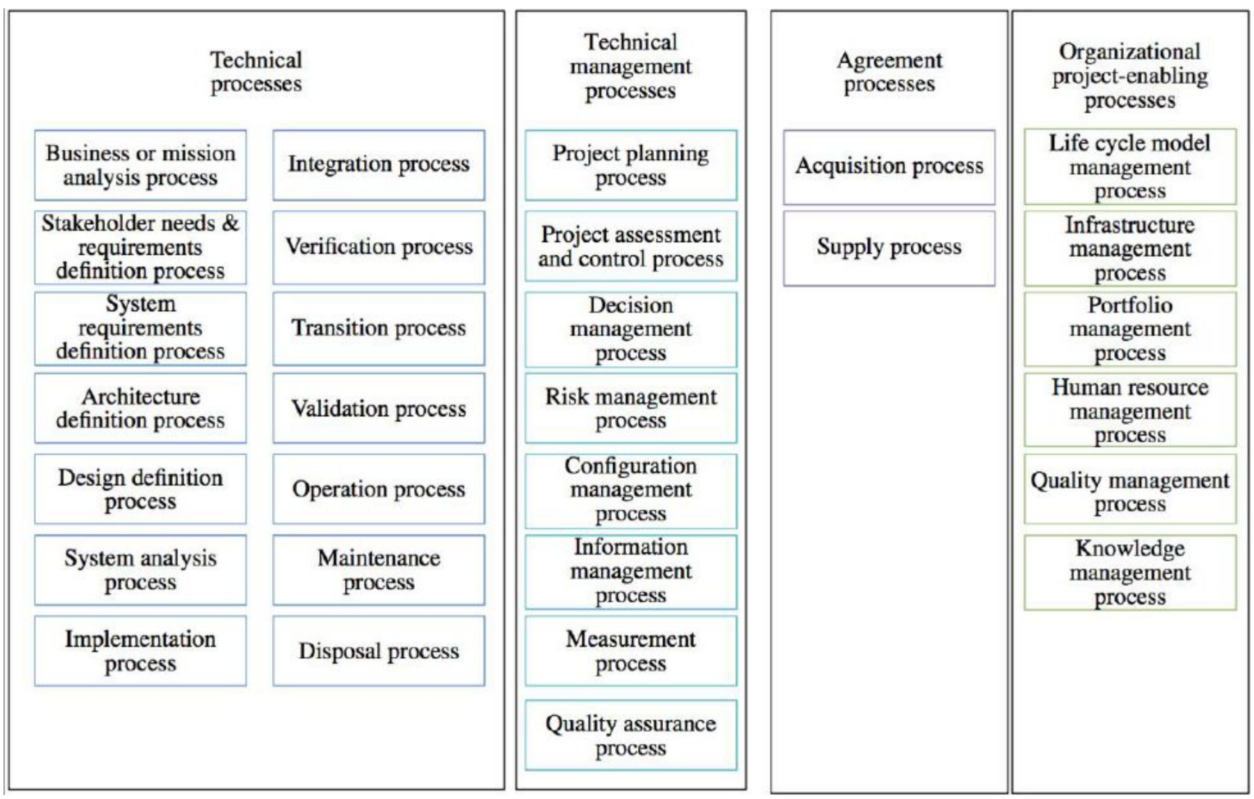


Figure 1, System life cycle processes per ISO/IEC/IEEE 15288.
(This figure is excerpted from ISO/IEC/IEEE 15288:2015, Figure 4 on page 17, with permission from the ANSI on behalf of the ISO. © ISO 2015. all rights reserved.)

As shown in Figure 2 below, (INCOSE 2015) each of these processes have inputs, activities, controls, enablers, and outputs. The inputs, controls, and enablers for any given process are outputs of the activities of other processes, some internal to a project/organization and some external. For purposes of this whitepaper, the outputs or artifacts of any process are work products and their underlying data and information.

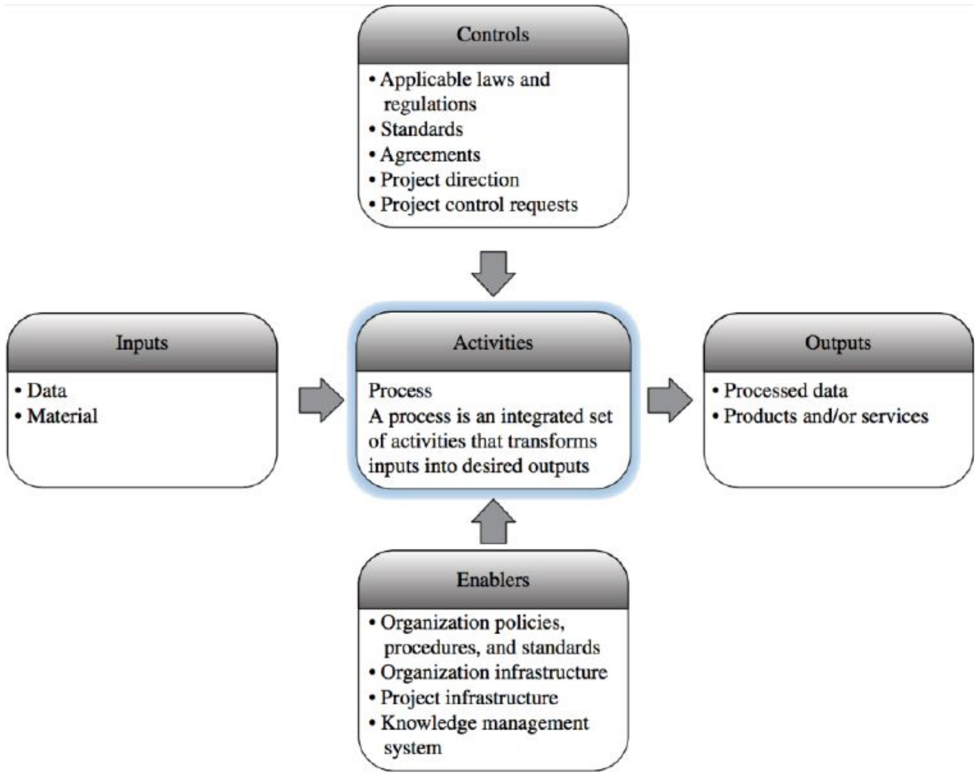


Figure 2: Sample of IPO diagram for SE processes. INCOSE SE Handbook.
(Original figure created by Shortell and Walden. Usage per the INCOSE notices page. All other rights reserved.)

These work products may be represented in a "hard copy" printed form (documents, drawings, diagrams, etc.) or in an electronic form (documents, drawings, diagrams, databases, models, spreadsheets, etc.). In some cases, the electronic form may be a file without any underlying data or may be represented by underlying data and information stored in a database.

Moving toward practicing SE from a data-centric perspective, with the goal of integrating all the work products and their underlying data and information into shareable sets of data, requires the electronic form of work products to be such that their underlying data and information is represented by a data set that can be shared and ideally integrated with other similarly formatted sets of data that adhere to industry interoperability standards (see section 4.7.2). This allows the project to develop integrated, shareable sets of data from which the various work products across all system life cycle stage activities can be visualized. From this perspective, all work products represented by and underlying set of shareable data would be visualizations of the project's integrated underlying data and information model.

Table 1 provides example work products generated during each of the six system life cycle stage activities defined in ISO/IEC TS 24748-1:2016. *A more detailed list of work products can be found in chapters 4, 5, 6, & 7 of the INCOSE SE Handbook (HB) (INCOSE 2015). (A similar list of work products can be generated from a program/project management (PM) perspective. The authors believe that both sets of work products (PM & SE) are dependent and need to be integrated for maximum management effectiveness.)*

LIFE CYCLE	PURPOSE (Activities)	EXAMPLES OF ACTIVITY OUTPUT WORK PRODUCTS	
CONCEPT	• Design inputs • Define problem space • Characterize solution space • Identify stakeholders' needs, goals, objectives, • Identify drivers and constraints • Explore concepts, ideas, & technologies • Develop initial concepts and models • Assess concept feasibility • Propose and baseline feasible concept/viable solution • Document stakeholder needs • Transform stakeholder needs into system requirements	• Need, Goals, Objectives • Measures: MOEs, MOPs, KPPs, TPMs. Leading indicators • Concepts of Operation, ConOps, OpsCons, Use Cases, User Stories, Operational scenarios • Voice of the Customer and other stakeholder expectations, • Functional architecture, • Product breakdown structure, Work breakdown structure • Mind maps, Power Point slide(s) • Sketches, diagrams, drawings • Proof of concept prototypes • Concept trade studies	• Scope document, budget, schedule • Descriptive models • Stakeholder needs • Interface diagrams and definitions, ICDs • Requirement attributes (rationale, trace, allocation, risk, priority, verification method, etc.) • Verification matrix • Allocation & trace matrices • Requirement verification and validation • Validated and baselined system design-input requirements
DEVELOPMENT	• Design outputs Refine system requirements, develop subsystem, assembly, component requirements • Verify and validate requirements • Refine models • Create solution description – architecture and system design • Document the design • Build engineering mockups • Integrate, verify, and validate design & system	• Analytical models, environment models, reliability prediction analysis, fault trees, simulations used in development, etc. • Logical decomposition, logic diagrams • Subsystem, assembly, component requirements (document or database) • Design trade studies • Physical architecture, product breakdown structure	• Design documents, drawings, algorithms - at system, subsystem, and component levels • Build-to/code-to specifications • Design verification and validation • Engineering mockups, prototypes • Source code, compiled applications • Test plans, procedures • System integration, verification, and validation plans, procedures
PRODUCTION	• Produce system • Inspect and test • Post production system validation in operational environment	• Manufacturing/coding plans • System being produced • As-built drawings, diagrams, algorithms, models	• Completed test, system verification & validation activities, data, and results; non-conformance reports
UTILIZATION	• Operate system to satisfy users' needs	• Updated models, • User, maintenance manuals/procedures	• Simulators for operator training • Surveillance test plans and data; customer feedback
SUPPORT	• Provide sustained system capability	• Sustaining Engineering • Upgrades and in-service modifications of the system.	• Updated drawings, diagrams, algorithms, models • Failure/finding reports
RETIREMENT	• Store, archive, or dispose	• End-of-life plan for retirement, disposal, recycle	

Table 1: Examples of system life cycle activities and work products

(Derived from INCOSE SE HB Chapters 3, 4, & 5).

The focus in the concept stage is design inputs that define the system of interest and investigate the extent of the effort, time, and cost to provide that system of interest. Most of the work in the concept stage is conducted to:

- FGacilitate a common understanding of the problem being solved;
- Identify and elicit stakeholder needs and expectations
- Define a common vision, goals, and objectives for the system of interest
- Identify drivers and constraints

- Evaluate the project, technical, and operational risks associated with candidate concepts by performing feasibility analyses (cost, schedule, technical, political, environmental, ethical, etc.) and trade studies;
- Define and baseline a feasible concept for the system of interest needed to address the problem being solved;
- Based on this concept, document a set of stakeholder needs;
- Transform the stakeholder needs into system requirements.

If the work products shown in Table 1 have been created using SE tools that support increasing granularity, work products and their underlying data and information developed during the concept stage can continue to be used in subsequent system life cycle stage activities with added data, information, and refinement.

As a system progresses from one system life cycle stage to the next, the number of work products and their underlying data and information increase rapidly. The data and information from the concept stage serve as input to the later life cycle process activities and help to identify areas where deeper analysis and refinement is necessary. Feedback from these analyses is used to update the various work products and their underlying data, information, and the system concept, as needed. Traceability between data and information from the previous stage work products and the subsequent stage work products is established and maintained, as is rationale for all changes. Linking the data and information within and between system life cycle stage activities, is fundamental to establishing integrated or federated, shareable sets of data and helping to ensure consistency, correctness, and completeness of the resulting data and information.

Within many organizations, these processes are frequently executed by different disciplines and organizations across the various system life cycle stages, often resulting in "silos" within the developing organization and especially between external organizations. A primary outcome of implementing SE from a data-centric perspective within an organization is to breakdown the silos and integrate these work products and their underlying data, into integrated, shareable sets of data and information.

3.2 What is a Model?

All the SE work products and underlying data and information discussed in the previous section are represented within the integrated or federated, shareable sets of data. These work products include various types of models. The use of models as part of the system life cycle process activities is an important part of the MBSE Initiative. In the context of SE, the INCOSE SE HB states that "a model that represents a system and its environment is of particular importance to the systems engineer who must analyze, specify, design, and verify systems, as well as share information with other stakeholders. Different types of models are used to represent systems for different modeling purposes."

Because of this, it is instructive to understand what a model is.

Definitions of "model" include:

- A physical, mathematical, or otherwise logical representation of a system, entity, phenomenon, or process (DoD 1998).
- A representation of one or more concepts that may be realized in the physical world (Friedenthal, Moore, and Steiner 2009).

- A simplified representation of a system at some particular point in time or space intended to promote understanding of the real system (Bellinger 2004).
- An abstraction of a system, aimed at understanding, communicating, explaining, or designing aspects of interest of that system (Dori 2002).
- A selective representation of some system whose form and content are chosen based on a specific set of concerns; the model is related to the system by an explicit or implicit mapping (Object Management Group 2010).
- An approximation, representation, or idealization of selected aspects of the structure, behavior, operation, or other characteristics of a real-world process, concept, or system. (IEEE 610.12-1990).

Analytical models include a series of diagrams or other work products represented by underlying data and information stored in a database. Example diagrams include: package diagram, requirement diagram, activity diagram, sequence diagram, state machine diagram, use case diagram, block definition diagram, internal block diagram, parametric diagram (Novel 2016). Each of the diagrams or other work products is a visualization of the data and information in the database from whatever perspective is needed by the user to communicate a specific message or address a specific need.

In the INCOSE SE HB, the word "model" shows up over 680 times! The term is used to refer the various kinds of models, visualizations of the data and information contained in an analytical model, as well as documents, diagrams, drawings, or any other representation of a system. Examples include: system life cycle model, modeling and simulation, SysML or other language based models, SE "Vee" model, spiral model, event model, modeling artifacts, model taxonomy, mental models, competency models, engineering model, development model, system model, product model, graphical models, mathematical models, physical models, operational analysis models, logical models, functional models, architectural models, behavioral models, meta-models, cost models, process models, rule models, ontological models, belief models, project models, capability models, data models, structural models, analytical models, business models, representation models, temporal models, mass models, probabilistic models, parametric models, layout models, network models, concept models, information models, maturity models, SE process model, reference model, domain models, and T-shaped model.

Note: the above list of the various types of models exceeds the set of models that any one project will need or use. Each model type is generated for a specific purpose or need the project or SE practitioner wishes to address. Projects need to decide, and document in their Systems Engineering Management Plan (SEMP), which types of modeling work products are needed to meet their needs. Those models will then be leveraged for a particular SE effort.

In David Long's blog (DLONG 2016), *One Model to Coordinate Them All,* he discusses the concept of an overall model that coordinates all other models. He states:

"*Requirements models, activity models, interface models, parametric models, reliability models, thermal models, power models, finite element models, … the list goes on and on. In this drive towards model-based systems engineering (MBSE) – and ultimately model-based engineering to connect the product life cycle – how can we make sense of this vast portfolio of models? How can we effectively manage the models and use them to gain leverage over the problem at hand so that we engineer the system rather than becoming distracted by our models? The models for these analytic dimensions are not new. These are the models that engineering disciplines have developed over the years.*

Which we choose differs based upon the system of interest, and the set of analytic models chosen bring rigor, effectiveness, and efficiency to the systems engineering."

"*Within the INCOSE community, we often focus on a second type of model – what many call the descriptive systems model, what I often term the architectural systems model. This covers the space from concept of operations through requirements, behavior, physical architecture, and verification & validation."*

"There is one and only one architectural model – broad in scope, fundamentally interconnected in nature – and that architectural model connects and coordinates the diverse analytic models. Done well, the architectural model addresses both the problem and solution, reflecting and integrating the key dimensions of both in a manner that clearly reflects the interconnected nature of the system. Done well, the architectural model aligns and maps key terminology across disciplines and concerns, connecting the various perspectives and analytical considerations. In addressing needs, logical solution, physical solution, and V&V, the descriptive model is highly connected."

Because of the usefulness and value of models, especially an integrated, shareable data and information model of the system, the MBSE Initiative was formed.

3.3 What is Model-Based Systems Engineering (MBSE)?

The INCOSE SE HB states: "MBSE is often contrasted with a traditional document-based approach to SE. In a document-based SE approach, there is often considerable information generated about the system that is contained in documents and other artifacts such as specifications, interface control documents, system description documents, trade studies, analysis reports, and verification plans, procedures, and reports. The information contained within these documents is often difficult to maintain and synchronize, and difficult to assess in terms of its quality (correctness, completeness, and consistency)."

INCOSE SE HB, Section 9.2 (INCOSE 2015) defines MBSE as: "the formalized application of modelling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout the development and later life cycle phases."

"In an MBSE approach, much of this information is captured [electronically] in a system model or set of models. The system model is a primary artifact of the SE process. MBSE formalizes the application of SE through the use of models. The degree to which this information is captured in models and maintained throughout the life cycle depends on the scope of the MBSE effort. Leveraging an MBSE approach to SE is intended to result in significant improvements in system requirements, architecture, and design quality; lower the risk and cost of system development by surfacing issues early in the system definition; enhance productivity through reuse of system artifacts; and improve communications among the system development team."

The artifacts are the work products and their underlying data and information that are generated during the SE process. It is important to understand that not all work products are models and not all models have to be analytical. In addition, it is helpful to understand that various views or visualizations of the data and information are not the same as the model which these views represent.

In Zane Scott's blog (ZSCOTT 2016), *Models and Views*, he addresses this confusion by making a distinction between models and views. *"Model" and "view" are terms that are used somewhat loosely in the world of systems engineering. Often, they are used* *interchangeably. Frequently, the imprecision of their usage causes confusion. Views, which might be pictures, diagrams, or textual descriptions of various aspects of the reality, can easily be considered models [*based on the definitions above*]. But doing so can lead to a loss of some of the power available in a system where models and views are understood differently. There is a strong value-add to be had from understanding documents and views in relation to models rather than seeing them as interchangeable concepts. A better approach involves a model that offers different views in order to serve different purposes. The documents/views flow from the model as structured answers to particular queries of the model."*

Given there can be multiple types of models, and visualizations of the data and information in those models generated while progressing through the system life cycle processes, a 10,000-foot level view of SE from a data-centric perspective is needed to help understand the context in which work products and their underlying data and information are generated and used. Thus, when a model needs to be shared, it is the sets of data representing the model that needs to be shared rather than the artifacts which are visualizations of the data and information in the sets of data. (Malone et al 2016)

MBSE is not really about any particular type of model or *visualization* of data and information – whether that be a diagram, report, or document – but is about the underlying *data and information model* that enables consistency across such models and visualizations. Hence the "Model" in Model-Based Systems Engineering" refers to the information model, not a specific type of model, diagram, or other visualization of the data in the model. (Malone et al 2016)

3.4 Integrated/Federated Shareable Sets of Data

The data and information model is represented by a project's integrated/federated), shareable sets of data that is a foundation of SE. These shareable sets of data represent the data an information model of the system of interest - work products and artifacts are visualizations of the data and information; they are not the model. (*See section 3.4.3 for a discussion concerning integrated vs federated sets of data*.)

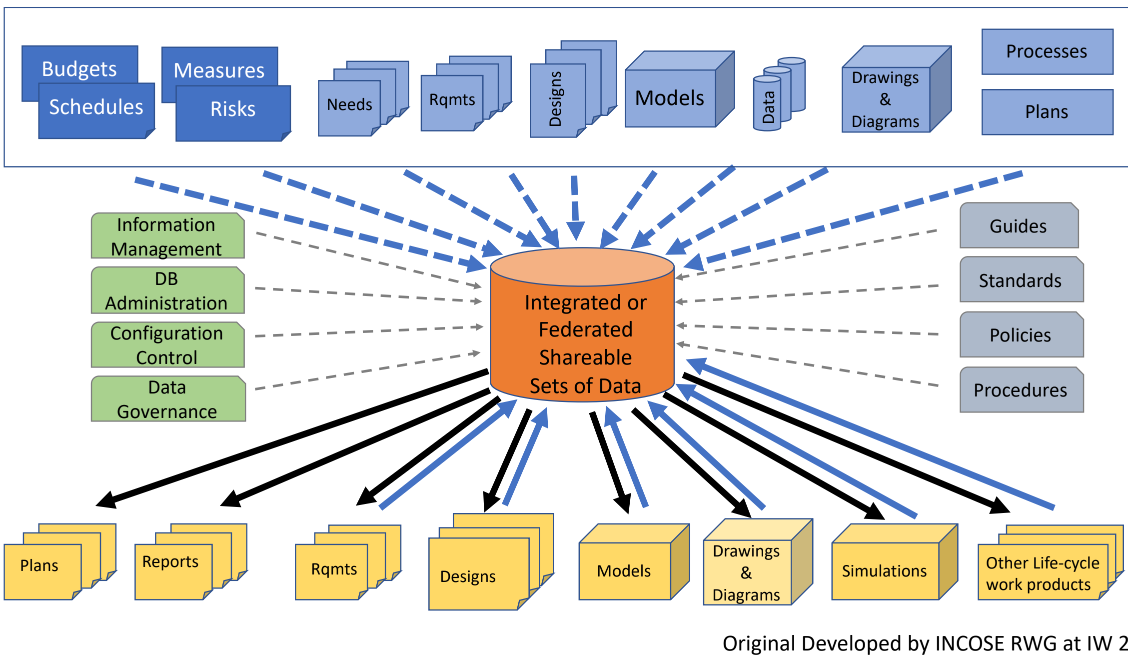


Figure 3: Integrated Data as a foundation for SE

As shown at the top of Figure 3, work products such as budgets, schedules, requirements, designs, diagrams, drawings, SysML or other language-based models, processes, and plans, etc. are created as part of the system life cycle process activities. The underlying data and information representing these work products is stored and managed either electronically in databases or as electronic files and documents. This data and information can be managed within shareable sets of data that represent those work products.

Guides, standards, policies, and procedures can also be included in the shareable sets of data. The shareable sets of data are managed via the enterprise and project data governance (Section 4.2), information management (Section 4.4), configuration management (Section 4.5), and DB administration requirements and processes. In order for the data and information to be considered the "single source of truth" as discussed earlier, the shareable sets of data must be maintained and managed under strict configuration control.

As shown at the bottom of Figure 3, once shareable sets of data have been formed, they become inputs to subsequent system life cycle process activities, resulting work products, and their underlying data and information. The database management tools allow the project's SE toolset to access data and information from the shareable sets of data which become a foundation of all the project's system life cycle process activities. This view of SE from a data-centric perspective is essential to manage the system development efforts across all system life cycle process activities and to address the challenges of increasingly complex systems.

3.4.1 The Importance of Defining a Project Common Ontology

Fundamental to forming shareable sets of data, the project needs to define and document a project ontology. The ontology includes the formal naming and definition of a set of terms, entities, data types, and properties as well as defining the relationships between these terms, entities, data types that are fundamental to the project and organization the project is part of. Having a documented project ontology helps ensure consistent use of this data and information across all system life cycle process activities and work products as well as across various groups within and external to the project. This common project ontology is key to interoperability and shareable sets of data and information.

A key consideration of a project is that organizations outside the project, either internal or external to the enterprise may use their own ontology which may not be compatible with the project's ontology. This can make sharing data, information, and work products between organizations problematic and often requires some sort of "middleware" to translate external ontology's to be consistent with the project's ontology.

To address this issue, there is a legitimate argument that focusing on just a project ontology does not go far enough. Those with this view, advocate that the SE community needs to define a universal SE ontology of agreeable terms that can be used across the SE discipline, across MBSE languages, and MBSE tools. By doing so, the SE community will be closer to realizing the vision advocated in this whitepaper for shareable sets of data and information both within a project and between projects. At INCOSE IS 2018, the Ontology WG met. At this session they discussed this need for a universal SE ontology and started defining a roadmap to making this happen.

3.4.2 The Importance of Defining a Project Master Schema

Consistent with the project's ontology, the project needs to define a master schema for the project's shareable sets of data. The master schema is a description, in a formal language, of the database structure that defines the objects in the databases, shows how real-world entities are modelled in the database, and integrity constraints that ensure compatibility between parts of the schema.

Ideally, all databases and file management repositories in the project's shareable sets of data need to have a schema consistent with the project's master schema to ensure compatibility of the data and information, allowing the data and information to be shared among the various SE tools in the project's toolset, and allowing data and information to be shared across various groups within and external to the organization.

Note: The concept of schema is a key part of database management. Having a master schema which other databases are consistent with is important from an interoperability, consistency, and data sharing perspective. Currently, various tool vendors define a proprietary schema for the data and information developed and managed within their tool's database. This can make sharing data and information between tools problematic and often requires some sort of "middleware" to extract data from one database, transform the data and information to be consistent with the project's master schema, and then load the data and information into the project's sharable data base. See section 4.7.2, concerning current efforts to help define standards that facilitate the sharing of data between various SE tools.

Note: While the concepts of ontology and schema are critical, the details of how they are structured and implemented are beyond the scope of this whitepaper, as are the inclusion of examples for different sizes and complexity of projects. These are topics that can be addressed by the appropriate working groups who focus on these areas of interest.

3.4.3 Integrated vs. Federated Sharable Sets of Data

A key point of this white paper is for organizations to move toward a data-centric driven approach that supports consistency, correctness, and completeness of the data and information that represents not only the system of interest, but all the SE work products developed as part of the SE lifecycle processes. Even if different SE tools are used within different parts of the organization or external organizations that have a part of the system under development, when a change is made to the data and information in one database that is either tightly linked to data in other databases or actually represents the same data and information, that change should be easily propagated across the other databases. This ability to share these types of data and changes to the data across databases is fundamental to the data-centric approach advocated in this whitepaper. Without this capability, creating a single source of truth is very difficult.

There are several approaches being explored to facilitate the integration and sharing of data. Two of these approaches are: **Data integration** (Wikipedia, 2018) and **federated database system** (Wikipedia, 2018).

Data integration involves combining data and information residing in different sources and providing users with a unified view of the data and information. The need for data integration as discussed in this whitepaper appears with increasing frequency as the volume and the need to share data and information contained in various databases increases. Data integration has become the focus of extensive theoretical work and numerous open issues and challenges remain unsolved. The reader is encouraged to refer to the reference to learn more about these issues.

A **federated database system** is a type of meta-database management system, which transparently maps multiple autonomous database systems into a single **federated database**. The constituent databases are interconnected via a computer network and may be geographically decentralized. Since the constituent database systems remain autonomous, a federated database system is a contrastable alternative to the (sometimes daunting) task of merging several disparate databases in to a single integrated database. A federated database, or **virtual database**, is a composite of all constituent databases in a federated database system. There is no actual data integration in the constituent disparate databases as a result of data federation. Again, the reader is encouraged to refer to the reference to learn more about the concept federated databases.

A major concern in taking a federated database approach is configuration management, defining baselines, and establishing a single "single source of truth" for the system under development. This is a key issue that projects need to address. One approach could be to define a specific time interval in which all the databases feeding into the federated database system are "synced up"

The specific approach is left up to other stakeholders and other INCOSE Working Groups that have expertise in this area. A key reason for this whitepaper is to get people thinking and hopefully if the practitioners demand the ability to better manage consistency, correctness, and completeness of data and information across the entire development lifecycle of ever increasingly complex systems, standards organizations and tool vendors will provide the means to do this.

Whether integrated or federated, the concepts of ontology, schemas, and standards are key enablers to the sharing of data and information.

3.5 Systems Engineering from a Data-Centric Perspective Defined

To help emphasize the concept that integrated data is a foundation of SE, the authors propose the following modified definition of SE from a data-centric perspective:

> "***SE, from a data-centric perspective, involves the formalized application of shareable sets of data to represent the SE work products and underlying data and information generated to support concept maturation, requirements development, design, analysis, and verification and validation activities throughout the system life cycle, from conceptual design to retirement***."

Taking a broader, data-centric view, the shareable sets of data represents not only a model of the system under development (architectural model) but also represents a model of all the system life cycle process activities, resulting work products, and their underlying data and information.

With a data-centric perspective of SE, the capability to capture, manage, access data, and manage the interrelationships between SE work products can be accomplished through a variety of methodologies, which range from the establishment of a single relational database to a virtually integrated, but distributed, database by means of a federation (or data map/index) of disparate data sources (as shown in Figure 3). As stated in NASA's Expanded Guidance for SE (NASA 2016):

> As stated in NASA's Expanded Guidance for SE (NASA 2016): "*In all cases, the interrelationships (both within and between data sources) among the various data items are captured. Establishment of a "master map" or ontology (i.e., a common vocabulary for the types and attributes of the data items and their associated interrelationships) up front, for all these data items and their associated interrelationships, facilitates the establishment of this capability.*"

4 PRACTICING SYSTEMS ENGINEERING FROM A DATA-CENTRIC PERSPECTIVE

This section goes into more detail concerning what it means to practice SE from a data-centric perspective providing guidance that can be used to understand and successfully create and manage the integrated, shareable sets of data within an organization. This section starts with a discussion concerning the need for enterprise and business management buy in and support needed to transition the organization from their present state to practicing SE from a data-centric perspective. Key concepts from big data are introduced including: data governance, information technology, and information management (Berson 2011), (Ladley 2012), (Soares 2014), and (Starling 2015). These are probably new concepts few systems engineers have thought about - let alone addressed in their organization - yet are essential concepts for organizations to understand to be successful in their journey towards implementing SE from a data-centric perspective. A brief discussion on configuration management, vs. information management is covered next. Also included is a discussion concerning another key advantage to practicing SE from a data-centric perspective - the use of measures to help better manage the system development activities across all system life cycle stages. This section concludes with a description concerning the current state of many organizations practicing SE and the path needed to move from their current state to a future state where the projects within the organization practice SE from a data-centric perspective using integrated/federated, shareable sets of data.

4.1 Success Starts At The Top

For projects to successfully implement SE from a data-centric perspective the journey must start at the top. Stakeholder needs and requirements exist at several levels (Ryan, 2013) within an organization as shown in Figure 4.

Note: An enterprise champion is essential for enterprise-wide changes. However, as reflected later in the whitepaper in Section 5.3, organizational change is a journey which may start with a pilot at a project level. This section describes the desired to-be state, but that should not inhibit an organization or a project from beginning the journey.

At the top, there is an **enterprise level** in which enterprise leadership sets the enterprise strategies; a **business management level** in which business management derives business needs, constraints, and requirements; a **business operations level** (*where the projects exist*) in which stakeholders define their needs and requirements; a systems' level in which the system of interest is defined in logical and physical views; and subsequently, there are lower levels for the subsystem and other system elements.

At the **enterprise level,** strategies are defined that will guide its future. Leadership communicates their intentions regarding the operation of the organization—in terms of existing systems, processes, and systems to be developed. Leadership defines the enterprise in terms of 'brand' and establishes a mission statement and corresponding goals and objectives which clearly state the reason for the enterprise and its strategy for moving forward.

The senior leadership develops a vision and advocates for the need for and acknowledges the benefits and Return on Investment (ROI) associated with implementing SE from a data-centric perspective.

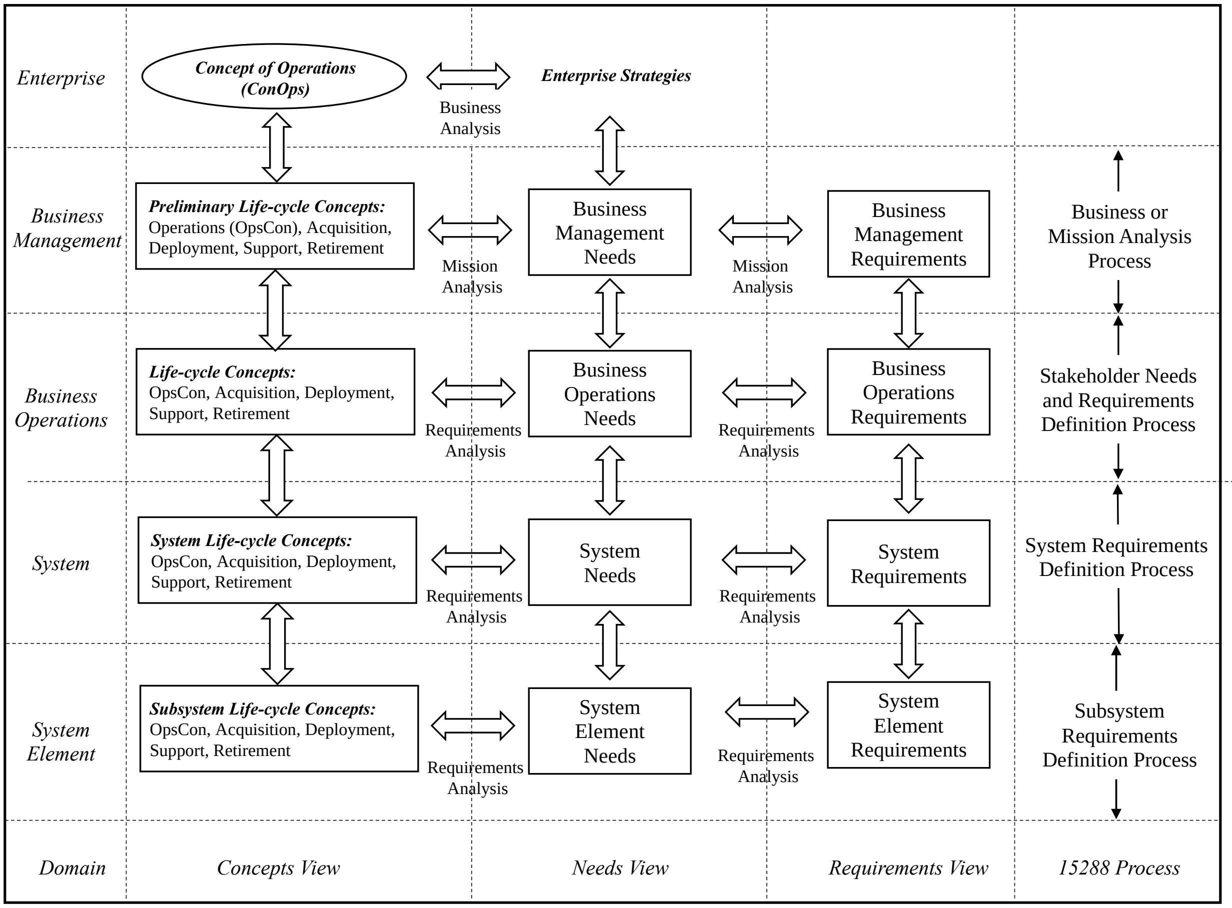


Figure 4: Transformation of concepts into needs into requirements (based on Ryan, 2013)

At the **business management level,** the concepts, needs, resulting requirements are documented that will result in an infrastructure that enables the enterprise to adopt SE from a data-centric perspective. This includes choosing the level of SE capability appropriate to the projects (Section 5.1), defining data governance policies (Section 4.2), developing the information technology (IT) architecture requirements (Section 4.3), and data and information management policies and plans (Section 4.4) tailored to the needs of the projects, product lines, and culture of the enterprise. Included at this level is the definition of the organization's configuration management (CM) policy (Section 4.5).

To successfully practice SE from a data-centric perspective, the levels of the enterprise above the project level need to address process, tools, and people:

- Processes need to be defined at the business management and business operations levels that support the chosen level of SE from a data-centric perspective capability. For those projects and organizations who are just beginning their journey, the project may wish to adopt data-centric approaches which focuses on satisfying the requirements and deliverables of business management level processes first. (*Note: Section 5.1 discusses SE Capability Levels (SCLs) that allow organizational elements to tailor their SE capabilities needed to successfully manage the development of the systems in their specific domain and types of systems they develop.*)

- SE tools and information technology (IT) infrastructure appropriate to the level of SE capability chosen needs to be provided by the IT organization at the business operations level; and
- People within the projects need the training, knowledge, and experience appropriate to the level of SE capability being implemented by the organization consistent with type and complexity of systems being developed and the SE toolset adopted by the enterprise.

Also, at the business management level, key measures of success are defined (Section 4.6) enabling management to track progress, identify and manage risk, identify issues and take action before the issues become problems. These measures include data to help quantify the ROI. For each project, business management defines "success" in terms of these measures which they use to track each project's progress.

Note: A key issue in many organizations is the separation between SE and PM. In a data-centric world, these concepts can be integrated under the concept of engineering management. In the context of this section, it is business management that defines both the SE processes and PM processes that are implemented at the business operations level. From an SE perspective the product breakdown structure (PBS) can be mapped to the PM work breakdown structure (WBS) enabling the SE activities to be included in the program/project schedule and budgeting activities and resulting work products.

At the **business operations leve**l, *where the projects operate*, the infrastructure is put in place to allow projects to develop and manage systems using an SE approach from a data-centric perspective at the level of SE capability defined by business management. This involves defining operating procedures, work instructions, processes, etc.; acquiring the IT infrastructure, defining a master schema for the project databases and file management systems, and acquiring an PM and SE toolsets with the capabilities and features needed to manage the project from a data-centric perspective. In addition, the infrastructure is put into place to train project and engineering teams in the processes and PM and SE toolsets as well as in the concepts associated with practicing SE from a data-centric perspective.

For organizations with multiple business units, each with different product lines, each business unit provides the infrastructure tailored to their unique needs. *Note that the various business units may decide on different implementations of SE from a data-centric perspective as discussed in Section 5.1. Green field projects vs brown field projects will need process tailored to their specific systems of interest.*

It is important to understand that often a system development effort involves multiple organizations who must work together. A prime example is the Boeing example of system complexity where the Boeing 787 (Malone et al 2016), over 30 companies based in countries around the world were involved in building large portions of the airplane. Systems of systems is another example where multiple organizations work together to achieve a common goal. This includes all organizations involved in a system's ecosystem both internal as well as external to the developing organization or the owner and operator of the system. A key premise of practicing SE from a data-centric perspective is:

- Data from all organizations and systems is available,
- All organizations involved form and comply with a common ontology, and
- All organizations practicing SE from a data-centric perspective form integrated sets of data that adhere to interoperability standards such that the data and information can be shared between organizations.

Assuming these activities discussed above are completed at the enterprise, business management, and business operations levels for all organizations involved in the system's development, the projects within the business operations level of each organization will have a much greater chance of success in implementation of SE from a data-centric perspective. For a project to be successful, the following actions must be completed:

- The senior management has agreed to implement SE from a data-centric perspective, and there is an enterprise level "champion".
- Data governance and information management policies have been defined.
- The level of data-centric SE capability consistent with the needs of the project has been agreed to.
- An IT infrastructure has been put into place that meets the needs of the project.
- PM and SE toolsets consistent with the needs of the project have been procured and licenses put in place.
- The project has a defined ontology and master schema for the project's shareable sets of data.
- Plans, processes, procedures, and work instructions have been defined by the program/project (plans include: Project Management Plan (PMP), Systems Engineering Management Plan (SEMP, and Information Management Plan (IMP)).
- Project team members are trained in practicing SE from a data-centric perspective, the PM and SE tools, defined schema, plans, processes, procedures, and work instructions.

At the **systems and systems element leve**ls the work products are developed by the project team for each life cycle stage based on the policies, plans, processes, procedures, and work instructions defined for the project at the business operations level using the PM and SE tools consistent with the needs of the project and the IT infrastructure that has been put in place. To help ensure consistency and shareability, the data and information needs to be consistent with the project's ontology and master schema.

4.2 Data Governance

Data Governance (DG) is the formulation of policy at the business management level to optimize, secure, manage, and leverage data and information as an enterprise asset.

The following basic principles of DG need to be established at the enterprise level. These basic principles guide all enterprise activities:

- *Data and information are assets* – Data and information are assets that have value to the enterprise and must be managed accordingly. Data and information are the life-blood of the enterprise.
- *Data and information must be Secur*e – Information security includes the principles of confidentiality, integrity, and availability (CIA). Data and information must be protected from unauthorized use and disclosure, must be able to be trusted (accurate, consistent, of high quality, managed), and accessible to authorized personnel.
- *Data and information risk must be mitigated* – There is risk associated with data and information which must be recognized and mitigated. This risk also can represent a liability if data and information is compromised or misused.
- *Data and information must be accessible and shareable* – Users must have access to the data and information necessary to perform their duties; therefore, data and information must be sharable across the enterprise functions and organizations that have a need for the data and information.

- *Data and information have an owner and steward* – Each data element and information have a data owner accountable for proper management, access, and usage of the data and information and a steward accountable for data and information quality.
- *A Common Vocabulary (ontology and schema) must be defined* - All data and information must be clearly defined consistently throughout the enterprise with the definitions understandable and available to all stakeholders.

DG for the enterprise is established and controlled at the business management level to implement the basic principles defined by the enterprise. The focus is on the "what". The "How", implementation, is defined at the business operations/project level. DG includes vision, principles, processes, and requirements to oversee and control the management of data and information and the use of data and data-related resources and information within the enterprise to:

- Ensure that data and information is managed in alignment with the basic principles and needs of the enterprise;
- Manage data and information within the largest relevant context of the enterprise strategy, goals, and objectives;
- Define the data and information to be governed and policies for: security, access, sharing, quality, and backup/archival storage, and retention;
- Ensure compliance with regulations, standards, policies, and requirements that govern access, privacy, quality, and security of the data and information;
- Support and enable knowledge-based decisions, analysis, and analytics;
- Ensure data and information usage achieves maximum value to the enterprise and its customers while managing the cost and quality of information handling; and
- Enforce the consistent, integrated, and disciplined use of data and information within the enterprise and partners.

DG requires cross-organizational cooperation to deliver timely, trustworthy data for better decisions and knowledge. DG is achieved through a partnership between the Business Management and Business Operations as shown in Figure 5.

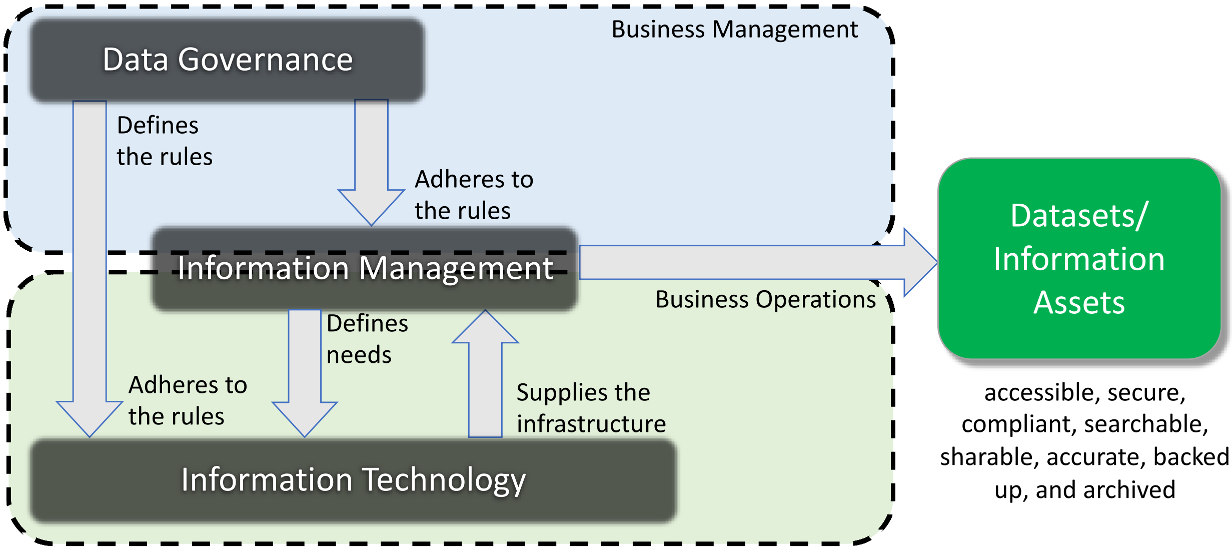


Figure 5: Cross-organizational cooperation

Data Governance defines the rules "what". Data Management and Information Technology adhere to the rules "how". Data Management defines the needs and requirements for the information technology infrastructure. Information Technology supplies and maintains that infrastructure per those requirements. Organizational elements (business units and projects) conduct business operations in adherence to the rules and within the supplied infrastructure. The organizational elements are responsible for their Data Management and the resulting data and information assets.

4.3 Information Technology

Information Technology (IT) is responsible for the IT infrastructure needed in support of business management level data governance and management needs and requirements. The IT organization exists at the business operations level. The role of IT is to:

- Develop, establish, and manage enterprise data architecture and platforms in alignment with data governance and data management policies, principles, processes, and requirements defined at the business management level.
- Supply, maintain, and provide support for hardware and software (PM and SE toolsets) needed to meet the needs of the organizational element data and information management activities.
- Design and implement data access, security, search, sharing, quality, backup, and archival storage control services in alignment with enterprise data governance and management and needs of the individual organizational elements.

4.4 Information Management

The purpose of Information Management (IM) is the management of data and information assets within an enterprise and organization element(s). IM occurs at the business operations level by the business units and programs/projects within the business units. IM addresses the "how" to implement data governance policies and data and information management requirements defined at the business management level. There are multiple levels of data management:

- Business Management: defines, controls, monitors implementation, and ensures compliance with enterprise data governance policies, requirements, and processes and provides the direction, philosophy, and mindset required to manage enterprise data assets.
- Organizational Element(s)/projects: develop, implement, and manage data and information and configuration management plans that implement enterprise data governance requirements and processes. The organizational elements are responsible for the day-to-day "activities" that must be performed to achieve the management of data and information assets within the organizational element. Practicing SE from a data-centric perspective is enabled by these data and information management activities.

In addition to the technical life cycle processes shown in Figure 1, the following crosscutting technical management processes are also shown: project planning, project assessment and control, decision management, risk management, configuration management, information management, measurement, and quality assurance (QA). It is equally important that organizational elements practicing Systems Engineering implement these processes as well.

The Project Planning Process (INCOSE SE HB section 5.1) includes the development of a Project Management Plan (PMP) that establishes the direction and infrastructure necessary to enable the assessment and control of the project progress and identifies the details of the work and the needed set of personnel, skills, and facilities with a schedule and budget for resources from within and outside the organization needed to produce the system of interest.

A major activity in project planning is preparing the Systems Engineering Management Plan (SEMP). The SEMP needs to:

- Establish that the project will conduct SE from a data-centric perspective and define the level of SE capabilities that will be used by the project. (Section 5.1)
- Include definitions of the system life cycle process activities and identification of all work products generated as part of the activities associated with these processes as well as the major deliverables of the project at each life cycle stage.
- Address the form of the work products (paper vs electronic), the SE tools to be used to generate and maintain and configuration manage (Section 4.5) the work products and underlying data and information, and the IT infrastructure needed.
- Define the key measures (Section 4.6) and work product attributes that will be used to manage the system development effort.
- Define a project ontology.

Both the PMP and SEMP need to identify the measures and reports that will be used to manage and track progress of the system life cycle process activities. These reports help define the data and information needed to be managed within the integrated, shareable sets of data. Knowing what data and information will be included in the reports helps inform the formation of the project master schema which individual sets of data and databases will conform.

The Information Management Process (IM) (INCOSE SE HB section 5.6) supplements the PMP and SEMP addressing the functions associated with project information management. The IM Process ensures the project's data and information is properly stored, maintained, secured, and accessible to those who need it, thereby establishing/maintaining integrity of relevant system life cycle work products and underlying data and information. The IM Process provides the basis for the management of and access to project data and information throughout the system life cycle stage activities.

Specific details concerning the IM Process are tailored to a specific project and included in the project's Information Management Plan (IMP). The IMP identifies the system-relevant data and information to be collected, retained, secured, and disseminated. The preparation of the project IMP at the beginning of the project is essential to reap the benefits of practicing SE from a data-centric perspective. The IMP needs to:

- Identify the resources and personnel skills specific to information management, including naming an IM lead for the project;
- Define the tasks to be performed;
- Define the rights, obligations, and commitments of parties for generation, management, and access to project data and information;
- Identify data and information management tools and processes, as well as methodologies, standards, and procedures that will be used by the project;
- Establish the scope of project data and information that is to be maintained and which data and information will under configuration control;
- Define a project master schema for the shareable sets of data and databases that will be used to store the data associated with the various work products and underlying data across all

system life cycle process activities. The project master schema includes formats and media for capture, retention, transmission, and retrieval of data and information. Different sets of data generated by different SE tools will most likely have their own schema, sometimes proprietary. Being able to share data between these sets of data is easier if the project defines a master schema first, and then the different SE tools used define their schemas to be as consistent as possible with the project's master schema;

- Establish and maintaining a system data dictionary;
- Define project relevant data and information, access privileges, and sharing criteria;
- Identify valid sources of data and information and designating authorities (owners) and responsibilities regarding the origination, generation, capture, archival, sharing, and disposal of information in accordance with the records and Configuration Management Process (Section 4.5) and governing standards and requirements; and
- Identify the standards by which the data and information will be created, managed, and stored. These standards enable the integration and sharing of the data and information contained in the integrated, shareable sets of data. (See section 4.7.2 for more information on data sharing standards.)

With effective data and information management, data and information are readily accessible to authorized project and organizational element personnel. Challenges related to maintaining databases, security of data, sharing data across multiple platforms and organizations, and transitioning when technology is updated are all need to be addressed by the PMP, SEMP, and IMP.

Effective data and information management is essential to successfully implementing SE from a data-centric perspective, enabling the projects to create and manage shareable sets of data that will be a foundation of all the project's SE activities.

4.5 Configuration Management

The Configuration Management Process (CM) (INCOSE SE HB section 5.5) supplements the PMP and SEMP addressing the functions associated with project configuration management. The CM Process ensures product functional, performance, and physical characteristics are properly identified, documented, validated, and verified to establish product integrity; that changes to these product characteristics are properly identified, reviewed, approved, documented, and implemented.

A key part of system development is the establishment, control, and maintenance of baselines. Baselines are business, budget, functional, performance, and physical reference points for the system under development. These baselines, or reference points, are established by review and acceptance of requirements, design, and product specifications, and other artifacts generated as part of the system life cycle activities as shown in Figure 1. The creation of a baseline may coincide with a project milestone or decision gate. As the system matures and moves through the life cycle stages, the software or hardware baseline is maintained under configuration control.

Not all artifacts need to be put under configuration control. Which artifacts or work products that are put under configuration control defined in the project plan and SEMP and often represented via a "document tree". From a vendor contract perspective, contract deliverables are usually put under configuration control while intermediate work products are not.

It is the set of configuration managed artifacts that represent the system hardware and software baseline. This baseline represents the "single source of truth" concerning what was agreed to and is being developed per those agreements.

Historically, the configuration managed artifacts are represented by documents in either a hardcopy or softcopy form. A key issue with these documents is that they are often generated at different times and are valid only at the time they were baselined and put under configuration control. To make this clear, most organizations adhere to ISO 9001 standards and include a statement on the front page of these documents stating this fact. Because of different timing of the baselined documents, they are often out of sync and inconsistent, thus the "single source of truth" isn't always clear.

A key benefit of a data-centric approach is that many of the artifacts are different visualizations of the project's integrated data set. When this is the case, changes to the data set are propagated across all the various visualizations of that data set. In the future, projects that have matured their configuration management process to be consistent with a data-centric approach, may baseline the data set from which the visualizations are generated rather than the individual artifacts. Using this perspective, all the artifacts will be consistent with the baselined data set. SE tools support this approach, enabling the creation of baselines by removing write permission from the dataset, freezing the data at the time of baseline. A copy of the data set can be made and changes to the baseline data set are allowed only if they go through a formal CM change process.

4.6 Measures – Using Data To Better Manage Se Projects

Another key benefit of adopting SE from a data-centric perspective is being able to use measures to better manage, across all system life cycle process activities, the SE activities associated with increasingly complex systems. Measures allow managers and systems engineers to monitor and assess progress, identify issues, and ensure the system being developed will meet stakeholder needs and expectations.

As stated in the INCOSE "Systems Engineering Measurement Primer", v2.0, (INCOSE 2010) using measures can "efficiently deliver information to systems engineering managers who use it for decision-making." Measurements help the project manager and systems engineer to:

- Monitor the progress and performance of SE and PM activities
- Communicate effectively throughout the organization
- Identify and correct problems early
- Make key tradeoffs
- Track specific project objectives
- Defend and justify decisions

Due to the importance of a measure to project success, several key measures are commonly used that reflect overall customer/user satisfaction (e.g., performance, safety, reliability, availability, maintainability, and workload requirements): measures of Suitability (MOSs), measures of performance (MOPs), and measures of effectiveness (MOEs). Once the project/program has identified and defined these measures, the measures are managed and monitored closely throughout the system life cycle process activities and used by technical and programmatic leadership so that they make informed decisions and take appropriate and timely actions.

A subset of these measures and associated requirements that are high priority and considered critical to successful development and operations, are also referred to as Key Performance Parameters (KPPs), as a failure to meet a KPP requirement may put the project at risk of cost and/or schedule overruns, or at risk of performance shortfalls. A subset of KPPs that are also "at risk" are often included in a project's list of Technical Performance Measures (TPMs) and closely monitored. For systems based on new technologies that have not been proven in the actual operational environment, additional development and operational risk is added to the project. Measures are an important tool to help project managers and systems engineers manage these risks.

The set of measures will be monitored closely during implementation by comparing trends for the current actual achievement of the parameters with the values that were anticipated for the current time and projected for future dates. Each major review needs to include a status of these measures.

A major source of measures are the attributes that can be defined as part of a requirement expression. These attributes are discussed and defined in INCOSE-TP-2010-006-02, INCOSE "*Guide to Writing Requirements*", Chapter 5 (INCOSE-TP-2010-006-02 2015).

Using requirement attributes helps to better define measures to manage projects. Given that requirements are the common threads that tie all systems engineering product development life cycle process activities together, having insight into these processes is necessary to manage a project effectively. Using attributes, management is able to generate reports or create dash boards from metrics managed within as SE toolset like:

- How many, or what percentage, of requirements have been approved?
- How many, or what percentage, of requirements have been implemented in the design?
- How many open change requests are there?
- What is the status of the high priority, high risk requirements (KPPs, TPMs)?
- For system verification, how may requirements have a verification approach defined?
- How many system verification activities have been successfully competed? Have failed?
- What percentage of system validation activities have been completed?

The purpose of using measures is to provide management with metrics that can be monitored and tracked closely throughout the system life cycle stages to assess schedule and budget status and to help ensure a successful program and mission. The importance of identifying, managing, and using measures is emphasized by the number of documents that INCOSE has published on the subject:

- INCOSE SE HB, Section 5.7, Measurement Process (INCOSE-TP-2003-002-04, 2015)
- Metrics Guidebook for Integrated Systems and Product Development (INCOSE-TP-1995-002-01)
- Systems Engineering Measurement Primer (INCOSE-TP-2010-005-02)
- Technical Measurement Guide (INCOSE-TP-2003-020-01)
- Systems Engineering Leading Indicators Guide (INCOSE-TP-2005-001-03)
- Project Managers Guide to SE Measurement for Project Success (INCOSE-TP-2015-001-01)

4.7 Developing Shareable Sets Of Data

SE, from a data-centric perspective, involves the formalized development of shareable sets of data to represent the SE work products and underlying data and information generated throughout the system life cycle stage activities**.**

The shareable sets of data are at the core of practicing SE from a data-centric perspective. The shareable sets of data include the data and information from several databases and files created by the various SE tools used to develop, document, and manage the various work products (e.g., use cases, diagrams, requirements, models, designs, etc.) and their underlying data and information. In adopting SE from a data-centric perspective, the end state is to integrate these databases and files into shareable sets of data and information as was shown in Figure 3.

As stated previously, it is critical that the project defines and documents an ontology and a master schema for the project's integrated, shareable sets of data at the beginning of a project, Doing this allows the work products and their underlying data and information to be shared between SE tools and other organizations. Key considerations in defining the schema include defining the entities, consistent with the ontology, that will be stored in the databases, defining the attributes that will be included as part of a requirement expression, measures that will be used to track the project status, and defining the reports and associated data and information needed by management to monitor the health and status of the project. This includes both PM and SE management reports, data, and information.

Key attributes of the envisioned integrated or federated, shareable sets of data include:

- There is a defined ontology to which all the sets of data within the project are consistent.
- There is a master project schema to which all the sets of data within the project comply.
- Data representing work products are appropriately linked across system life cycle stages.
- Relationships are defined that reflect dependencies and interactions between entities and associated work products.
- The PM and SE tools used to generate the sets of PM and SE data comply fully with the industry interoperability standards as discussed in section 4.7.2.

Achieving an SE capability level where the project has integrated or federated, shareable sets of data will not happen overnight. It will take a journey lasting several years.

Even though the focus in the following sections and figures is on an integrated data approach, in many cases, a federated database system (Section 3.3) is probably a more practical approach to enabling the ability to share data and information. Which approach is the most appropriate is probably different depending on whether a system or system of systems is being discussed, whether the majority of the development is being developed within a single organization or is shared across multiple, geographically separated organizations, as well as whether the system is being developed from a green or brown field perspective.

4.7.1 Moving from The Current State to a State that Includes Integrated or Federated, Shareable Sets of Data

As shown in Figure 6, different organizations or parts of an organization are often involved in the various SE processes for each system life cycle stage, using a variety of PM and SE tools to produce and manage the various work products and their underlying data and information. There is no defined master ontology for the enterprise nor project. There is no master schema for the sets of data representing the work products. The SE tools store the data and information representing the various work products either as electronic files or documents (shown as a solid line) or in their own proprietary database using a proprietary schema. Unless these tools support a standard for sharing data with other tools, the data in these individual databases are not compatible making it difficult to share data between tools and organizations.

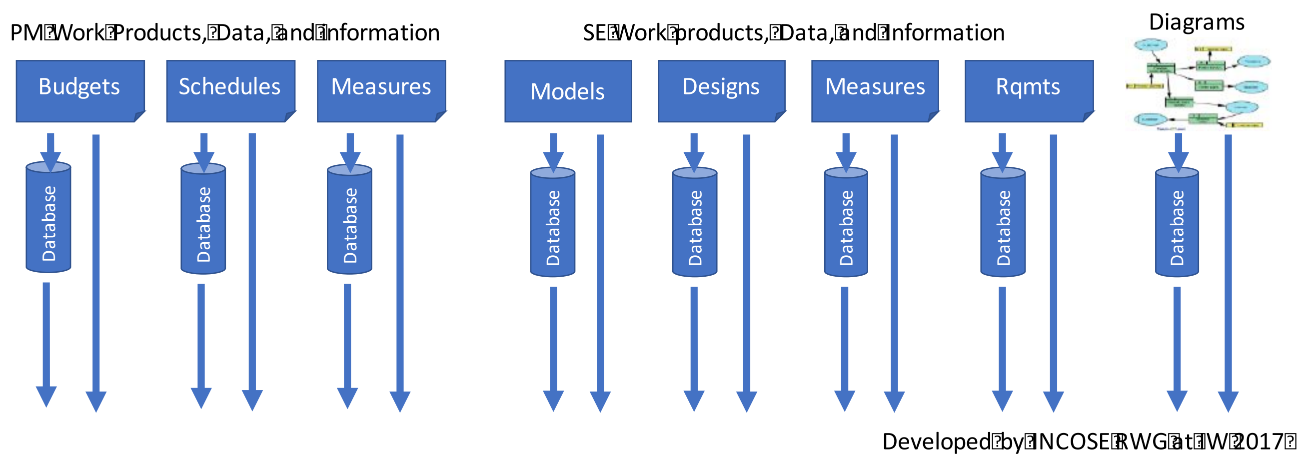


Figure 6: Current State: Siloed organizations and sets of data

Without a view of SE from a data-centric perspective, data and information generated by one group or organizational element using a specific tool are either not made available to those involved in the other system life cycle process activities or the tools used. The result is data and information that is difficult to share as well as hindering the ability to maintain consistency, completeness, and correctness of the data and information across all system life cycle stages.

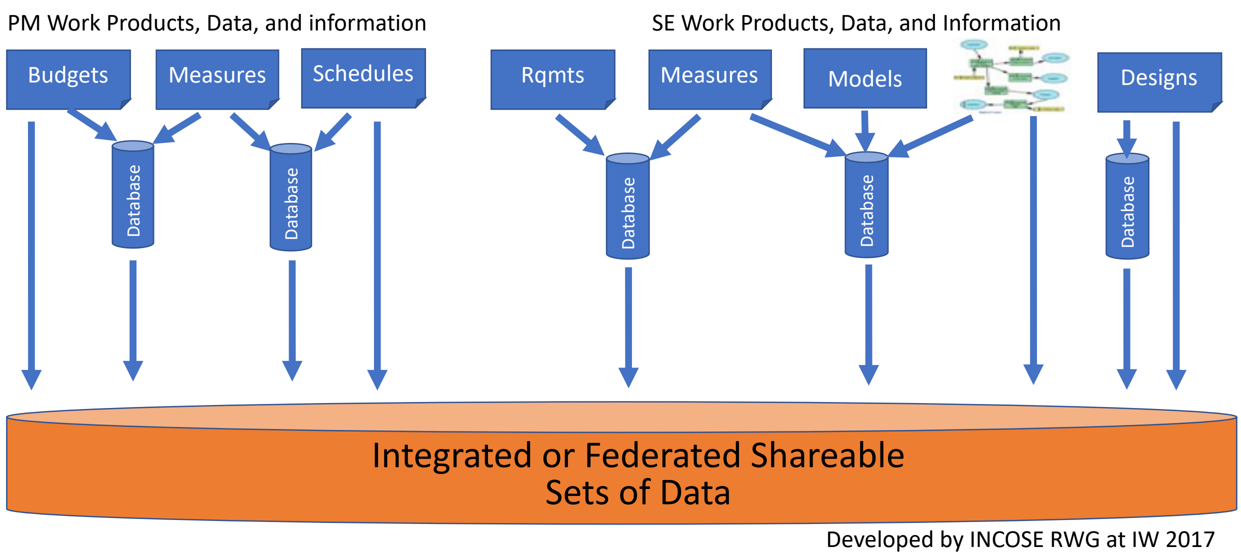


Figure 7: Interim State: Data from existing databases imported into the integrated, shareable sets of data

As shown in Figure 7 and Figure 8, interim states, the project has a master ontology and a master schema defined for their integrated, shareable sets of data. The tools in the organization's PM and SE toolsets used to generate and manage work products and their underlying data and information have either:

- special software procured or developed to **extract** data and information from the individual SE tool databases, **transform** that data and information to conform to the project's master schema, and **load** the transformed data and information into the integrated, shareable sets of data (*this process is referred to as the ETL process*); or
- databases having a schema that is consistent with the project's master schema allowing the data and information in these databases to be integrated or federated directly into the project's shareable sets of data without having to go through an ETL process.

The first case, shown in Figure 7, will be the most common for most organizations as they start their journey towards establishing integrated, shareable sets of data enabling them to practice SE from a data-centric perspective. PM has their own legacy tools for budget and scheduling - each with their own methods of tracking various project management performance measures. Some organizational elements have legacy tools to develop diagrams that are stored as electronic files (as compared to storing work products and their underlying data and information in a database). One organizational element may have a robust legacy requirements management tool (RMT) that has been in use for many years but has a proprietary database schema. Another part of the organization has just started using an analytical modeling tool that can be used to support the generation and management of various system life cycle work products and their underlying data and information, but doesn't have all the robustness of the RMT, so requirements continue to be managed in the RMT and imported into the modeling tool via an ETL process. Depending on which standards are supported by the tools, this process could be either manual or automated. Another part of the organization has a legacy design tool that has been in use for many years that is not compatible with the modeling tools nor the RMT. The tracking of the system verification and validation activities may be done in the RMT tool, but not integrated with the various modeling work products and their underlying data and information.

This case is less desirable in that the data and information from these legacy databases will have to go through the ETL process to get the data and information into the integrated, shareable sets of data and any changes made to the SE tool databases must go through the often expensive and time consuming ETL process before the changes can be reflected in the integrated, shareable sets of data. This makes it harder to keep the data in the integrated or federated, shareable sets of data current, correct, and consistent across all system life cycle process activities. Also, anyone doing analysis, modifying/updating work products and their underlying data and information, or generating reports based on the data and information from the shareable sets of data, will have to make sure that the data from these external databases is current and consistent.

The second case, shown in Figure 8, is preferred in that the integrated or federated, shareable sets of data contain the data and information developed within the individual PM and SE tool databases. Because their data is consistent with the project ontology, their schemas are consistent with the master schema, and these tools fully support standards for interoperability, the data in the databases are compatible and can be shared. This is also preferred because there is only one single source of truth for the project - the data in the integrated or federated, shareable sets of data is under strict configuration control and therefore represents the baseline status of the project at any given time. Any of the "visualizations" of the data will represent the current state of the project.

The second case will most likely mean the organization will need to procure new PM and SE toolsets. This can be an expensive and time-consuming step for most organizations. If setting on a path to procure new PM and SE toolsets, it is advisable to choose PM and SE tools that support the generation and management of multiple system life cycle work products and their underlying data and information and especially that fully support interoperability standards for compatible tools, schemas, and databases. *The perfect case would be to procure a single SE tool that "does it all", i.e., the one tool would result in having an integrated project dataset by default. That would help to ensure all data and information is shareable, current, and consistent across all system life cycle stages. There are several vendors in the process of developing tools that will support the development and shareability of PM and SE data and information across all life cycle stages. The form of these tools ranges from "glue ware" that various tools to share data or single product lifecycle management (PLM) tools.*

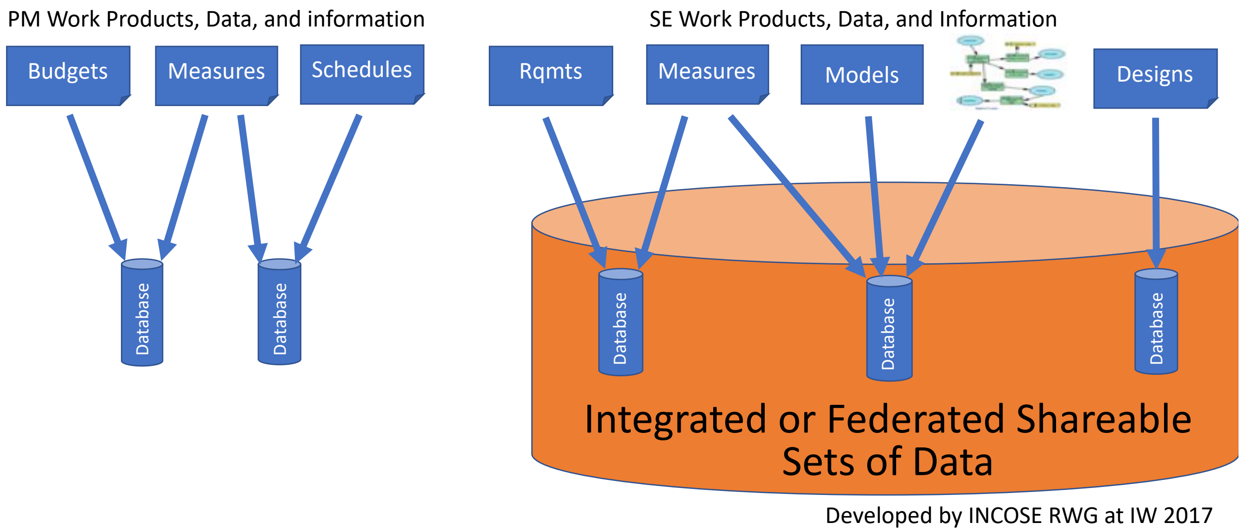


Figure 8: Interim State: Most systems engineering tool databases are included in the integrated, shareable sets of data

Note: in the second case as shown in Figure 8, even though most of the SE tools have compatible databases included in the integrated, shareable sets of data, the organization may still choose to continue to use some legacy systems, such as budgeting and scheduling applications the project is required to use, whose schema is not compatible projects master schema. In this case, this data and information will need to go through an ETL process in order for the data to be useable by other SE tools.

As shown in Figure 9, the end state, the project has a master ontology defined and a master schema defined for their integrated, shareable sets of data. All the PM and SE tools used to generate work products create and maintain their data and information in a database that has a schema consistent with the master project sets of data schema and conform to interoperability standards. This allows these PM and SE tool databases to be compatible and be included within the project's integrated or federated, shareable sets of data.

Work products such as budgets, schedules, requirements, designs, diagrams, drawings, analytical models, etc. and their underlying data and information are created as part of the system life cycle process activities. The data and information representing these work products are either stored and managed electronically in databases or as electronic files which can be linked to other work products. These databases and files are made part of the project's integrated or federated, shareable sets of data that represent the project's work products and underlying data and information. Guides, standards, policies, and procedures are included in the integrated, shareable sets of data. The project's integrated, shareable sets of data are managed via the enterprise and organization's data governance, records management, information management, and DB administration requirements and processes. In order for the data and information to be considered the "single source of truth" as discussed earlier, the integrated or federated, shareable sets of data are under strict configuration control.

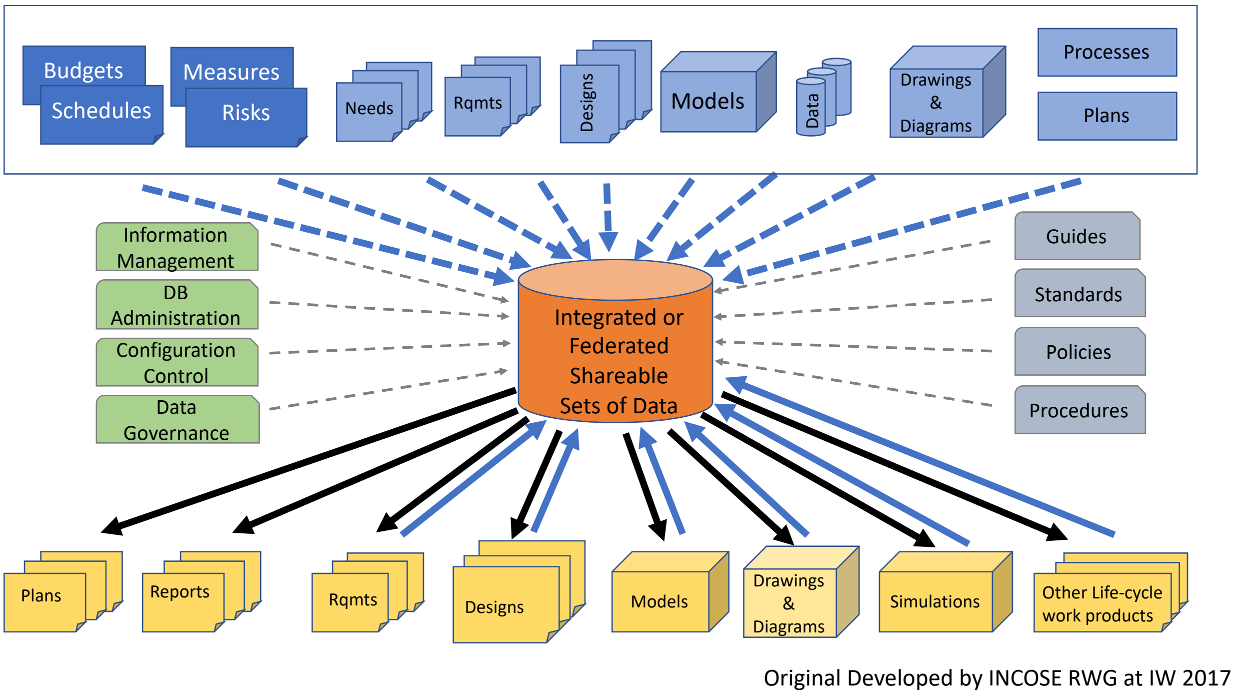


Figure 9: End State: The project has integrated, shareable sets of data - INCOSE's Vision 2025 realized.

Once the integrated or federated, shareable sets of data have been populated, it represents the single source of truth concerning the state of the project and becomes the source for subsequent system life cycle process activities and resulting work products and underlying data and information. Interoperability standards enable SE tools to share data. The database management tools allow the project's SE toolset to access data from the integrated or federated, shareable sets of data.

The integrated or federated, shareable sets of data become a foundation of all system life cycle process activities for the project. This data-centric SE perspective is essential to manage the system development efforts across all system life cycle stages and address the challenges of increasingly complex systems.

4.7.2 Standards

To meet the intent of the MBSE initiative and move towards INCOSE's Vision 2025, standards must be matured and adopted by the various PM and SE tool and database management system vendors.

This is a major issue that organizations and many of the PM and SE tool vendors are addressing. As discussed in the INCOSE SE HB, (INCOSE 2015) section 5.6, information management process, there are several activities in work to develop tool interchange specifications so that 'models' and other work products and their underlying data and information can be shared among tools.

The INCOSE SE HB states: "The STandard for the Exchange of Product (STEP)—ISO 10303 standard provides a neutral computer-inter-operable representation of product data throughout the life cycle.

- ISO 10303-239 (AP 239), Product Life Cycle Support (PLCS), is an international standard that specifies an information model that defines what information can be exchanged and represented to support a product through life (PLCS, 2013).
- INCOSE is a cosponsor of ISO 10303-233, *Application Protocol: Systems Engineering* (2012). AP 233 is used to exchange data between a SysML TM and other SE applications and then to applications in the larger life cycle of systems potentially using related ISO STEP data exchange capabilities."

ISO 15926 is an interoperability standard concerning system life cycle processes which is a further development of ISO 10303 and defines a common ontology for achieving interoperability between SE tools (also supported by Fiatech USA). ISO 15926 part 11 contains a method for achieving interoperability by means of information models representing information needs of ISO 15288 system life cycle processes (based on the W3C RDF standard.)

Another Initiative by tool vendors to develop common schemas for system life cycle data is OSLC = Open Services for Life cycle Collaboration (https://open-services.net/)."

Developing databases compliant with a project master schema would make it much simpler for SE tools to share sets of data and integrate these databases into the project's integrated or federated, shareable sets of data. This would make the SE tools interoperable by default. Conceptually, if the data is stored per a common standard and master schema, the data can be shared between the various SE tools and these same tools can then be used to visualize the data in whatever form is needed by any stakeholder in the organization as shown in Figure 9.

This concept is communicated clearly in the Boeing paper mentioned earlier (Malone et al 2016) "A perennial problem restricting data sharing is that modeling tools tend to be created independently, resulting in the tools having different and, often, incompatible data models. To enable data sharing, these separate data models need to be mapped, and a data transfer utility produced to perform intermediate data transformation as the data are passed between the tools. Creating and managing data utilities can easily become more expensive than managing the MBSE environments themselves. Compounding this problem is that data sharing among several tools becomes an ((N)(N-1))/2 scenario as individual data sharing utilities are built between tools."

Note that another part of the problem is standards proliferation by which multiple ontologies and exchange approaches are advocated. To make progress, the profession should step back and establish the common data model required to engineer any system (hopefully building upon work from the past such as AP-233). While that data model would require extensions based upon the specific domain and system of interest, it would establish the necessary core to reasonably underpin data interoperability and the practice of Systems Engineering.

Whether the SE tools being used by various organizational elements fully comply with a standard, an approach needs to be defined to integrate the different sets of data into the project's integrated, shareable sets of data. Fortunately, the INCOSE Tool Integration and Model Life cycle Management Working Group (TIMLM WG) is working toward this goal. Their mission is to capture best practices and guidelines for using computer-based tools, exchanging data between tools, and allowing users to operate on this data.

Once the ability to establish integrated or federated, shareable sets of data becomes a reality, ideally, PM and SE tools would be able to use this data to develop, display, and manage the various system life cycle work products and their underlying data and information. Done properly, all the benefits of SE from a data-centric perspective stated at the beginning of this whitepaper can be realized.

The infrastructure identified in Figure 9 and the processes to perform the ETL functions, need to be enabled by the enterprise, business management, and business operations levels. With this infrastructure in place, the program/project can then define their unique needs in their PMP, SEMP, and IMP.

5 DEVELOPING A SYSTEMS ENGINEERING CAPABILITY THAT MEETS THE NEEDS OF YOUR ORGANIZATION

This section focuses on topics to help organizations develop an SE capability that meets the needs of their organization. To aid in this journey, SE Capability Levels (SCLs) are presented to help organizations assess what their current SE capability is from data-centric perspective and provide a roadmap to get to their desired level of SE capability based on their organization's specific needs. Next, the selection of an SE toolset that is needed to implement the chosen SCL is discussed. The final topic in this section provides advice to help sell the idea of moving toward a data-centric practice of SE to management and then implementing the chosen SCL.

5.1 Levels of Systems Engineering Capability

It is important for the enterprise to first decide how, and to what extent, they are going to provide the capability for projects to implement SE from a data-centric perspective. This decision must be based on the needs of the enterprise while being scaled to the level of rigor that allows the system life cycle process activities to be performed by the projects with an acceptable level of risk. The INCOSE Systems Engineering Handbook (INCOSE 2015), Chapter 8, Tailoring Process and Application of Systems Engineering, provides excellent guidance in tailoring the system life cycle process activities to meet the needs of the project.

Once this is decided, the individual projects need to determine what level of SE capabilities they need to successfully manage the development of their system of interest from a data-centric perspective. Once they have decided on the level of SE capabilities needed, they can then take the necessary actions needed to provide those capabilities. These actions include:

- Developing enterprise, business management, and business operations level policies, processes, and procedures needed to implement SE from a data-centric perspective;
- Providing requirements to the IT department concerning the IT infrastructure needed, so these capabilities can be realized;
- Selecting and procuring PM and SE toolsets that support the level of SE capability decided on; and
- Training their mangers and systems engineers in the use of the PM and SE toolsets and processes from a data-centric perspective.

It is important to realize that this journey towards practicing SE from a data-centric perspective can be made in a series of small steps. The enterprise doesn't have to jump to completely integrated or federated, shareable sets of data at the beginning of their journey.

Some organizations may want to start with an electronic (vs. hard copy documents) requirement management capability with the ability to support allocation and traceability. Later, they can add the capability to manage the requirements and other work products across all system life cycle process activities linking requirements to the stakeholder needs from which they were transformed, to design outputs, and to verification and validation work products. The project can identify measures to track system development activities and identify and manage risks. They can then add the capability to use non-language-based diagrams as single entities without the underlying data, e.g., functional flow diagrams or context diagrams, and link the requirements to those diagrams. From there, the capability for analytical modeling can be added where the various diagrams, requirements, and other work products are visualizations of underlying sets of data (only if there is some benefit to be gained from doing so.) Taking this path can be a slow journey and it will be some time before the benefits of practicing SE from a data-centric perspective discussed earlier can be realized.

An alternate approach some organizations may want to implement (and may need to be based on the complexity of their systems) is to start with an analytical modeling capability from the beginning. This will allow them to incrementally integrate requirements development and management with other work products and their underlying data and information into shareable sets of data as well as link together work products across all system life cycle process activities. This path to achieving shareable sets of data will be shorter, speeding up the journey resulting in the organization being able to realize the benefits of practicing SE from a data-centric perspective sooner than the previous evolutionary approach.

No matter which approach taken, each step in the journey adds capabilities that will lead towards establishing shareable sets of data as a foundation of all system life cycle process activities, enabling the project to meet the challenges of increasingly complex systems and realizing the benefits of practicing SE from a data-centric perspective.

From an IT infrastructure requirements perspective, it is best for the projects to communicate the end state envisioned, so their IT department can provide the IT infrastructure and PM and SE toolsets that are scalable to be able to handle the needs of the organization for the envisioned end state.

To help in this journey to implement SE from a data-centric perspective, it is useful to define different levels of SE "capability". What specific SE capabilities an organization needs depends on their product line, its complexity, green-field vs brown-field products, issues they are having and want to address, workforce knowledge and experience, the PM and SE toolsets being used, and the organization's culture, processes, standard operating procedures, and work instructions.

Below are proposed SE Capability Levels (SCLs). Each level assumes the previous level has been experienced and surpassed. As the organization progresses through the levels, their SE capability level increases from a data-centric perspective. As the SCL increases, the organization is getting closer to realizing the intent of the MBSE Initiative and will be moving closer to realizing INCOSE's Vision 2025. The journey ends when the organization has reached the SCL that meets the needs of their organization.

To help organizations in their journey to adopt MBSE, the Model-Based Enterprise Capabilities Matrix (MBECM) INCOSE Challenge Team was formed at the INCOSE International Workshop (IW) 2018. The purpose of the MBECM is to provide a reference for enterprise and program/project organizations to assess their current and desired implementation of modeling. The Team has also developed a User's Guide to help enterprises to use the matrix for developing a strategic vision, roadmap, apply a yardstick, and perform tactical planning in their journey to implementing MBSE/Model-Based Engineering (MBE). The matrix addresses a range of MBSE/MBE factors and attributes, grouped by areas, that directly or indirectly support/enable the implementation of MBSE/MBE across the enterprise. For each of these, there are five stages of capability defined from no or little capability to full capability to practice MBSE/MBE. As such, the MBECM is focused on MBSE/MBE and its scope of is broader than the SCLs defined in this whitepaper, For more information concerning the MBECM Challenge Team products and activities you can follow this link: http://wiki.omg.org/MBSE/

Conceptually, the SCLs defined in the following paragraphs are similar to other capability maturity models defined by other institutions, however, the SCL concept described below is not the same. Other capability maturity models tend to focus on an organization's processes, their definition, their execution, and their enforcement. However, the SCLs defined herein focus on the capability of an organization to practice SE from the data-centric perspective discussed in this whitepaper, with the end state where all projects in the enterprise establish integrated or federated, shareable sets of data. Tables 2 - 5 below, list key "factors or attributes" to measure an organization's SE capability level in terms of practicing SE from the data-centric perspective.

SCL 0: Unfortunately, this level represents many legacy system development processes and associated shortcomings seen in today's world of more complex systems. These organizations are not equipped to deal with the ever increasingly complex systems and find it difficult to realize the benefits of practicing SE from a data-centric perspective. The current state of the key factors/attributes of the organization's SCL are shown in Table 2.

Factor/Attribute	Current State
Organization	The system life cycle process activities are divided across organizational units operating in silos.
Enterprise Level Data & Governance Policy, Processes, & Procedures	The enterprise has no documented data and information governance policy, processes, and procedures.
Project level Data and Information Management	Projects have not included data management concepts in their PMP nor SEMP and have no IMP.
Master Ontology	There is no defined master ontology for the enterprise nor project.
Master Schema	The project does not store data in a database so there is no defined master schema for the project.
Systems Engineering (SE) Tool Set	The primary SE toolset used by the project is limited to common office applications: word-processing, spreadsheets, presentations, and basic drawing and diagraming tools.
Project Management (PM) Tool Set	The primary PM toolset used by the project is common office applications: word-processing, spreadsheets, and presentations.
Work product Format	The primary focus of the project is on hardcopy, printed documents, design description documents, ICDs, CAD drawings, etc.
Shareability of data and information	While the files representing work products are stored electronically, they exist as independent files (vs. in a database containing underlying data) making it difficult to share information contained within the files other than with other office applications (copy/paste).
Linking (Traceability) between work products developed in different development life cycles	Few, if any, work products are linked together across system life cycle process activities making it is difficult to identify and manage dependencies between work products. Any traces between requirements is done manually, if at all.
Consistency of data across work products	There are often inconsistencies between and within work products.
Completeness of work products	It is difficult to assess completeness of data and information within work products.
Use of work product attributes and associated measures	Few, if any, SE work product attributes and measures are defined and used to help manage the project. PM measures focus on schedule and budget.
Configuration Management	The project baselines and configuration manages individual printed documents or electronic versions of the printed documents (e.g., pdf files). The single source of truth of project's data and information is represented by these baselined and configuration managed documents.

Table 2: SCL 0

SCL 1: The organization has started their journey to increase their SE capabilities but are mostly practicing SE from a document centric perspective rather than a data-centric perspective. A diagram of the various work products sets of data for organizations at SCL 1 closely resembles that shown in Figure 6. The current state of the key factors/attributes of the organization's SCL are shown in Table 3.

Factor/Attribute	Current State
Organization	The system life cycle process activities are divided across organizational units operating in silos.
Enterprise Level Data & Governance Policy, Processes, & Procedures	The enterprise has no documented data and information governance policy, processes, and procedures.
Project level Data and Information Management	Projects have not included data management concepts in their PMP nor SEMP and have no IMP.
Master Ontology	There is no defined master ontology for the enterprise nor project.
Master Schema	There is no project level master schema, each SE tool has its own proprietary schema making it difficult to share and integrate data sets from different SE & PM tools.
Systems Engineering (SE) Tool Set	Some parts of the enterprise may be using diagraming or modeling tools, requirement management tools (RMTs), CAD tools, etc. other than standard office tools, but in isolation from other parts of the system life cycle process activities and organization elements responsible for those activities.
Project Management (PM) Tool Set	Some projects may use legacy PM tools to develop and manage PM work products and their underlying data and information (e.g., budgets, schedules) but in isolation from the information in the SE tool set.
Work product Format	PM and SE tools store the data and information representing the various work products as electronic files. However, the primary focus of most projects is still on hardcopy, printed documents, design description documents, ICDs, CAD drawings, etc.
Shareability of data and information	While parts of the project may be using various PM & SE tools, the files representing work products are stored electronically as independent files (vs. in a database containing underlying data) making it difficult to share information contained within the files other than with other office applications (copy/paste). The data and information in the SE & PM tools are stored in tool specific databases. Often, these databases not compliant with interoperability and shareability standards. Because of this, files representing the data and information are not compatible - making it difficult to share data between tools and organizations.
Linking (Traceability) between work products developed in different development life cycles	Some work products may be linked within a system life cycle silo, but not necessarily across other system life cycle stages. For example, allocation, traceability, parent/child, and interface requirement relationships are managed within the RMT, but the requirements are not linked to user stories, stakeholder needs, diagrams, models, design, systems verification, nor system validation work products.
Consistency of data across work products	There are often inconsistencies between and within work products. Often requirements are developed with little or no underlying data and information model making it hard to maintain consistency - especially between dependent or related requirements. Because requirements and design models are done by separate groups, the requirements and design models are inconsistent.
Completeness of work products	It is difficult to assess completeness of data and information within work products. Often requirements are developed with little or no underlying data and information model making it hard to assess completeness. Models developed by the project focus on functionality, performance, and interfaces, but often do not reflect quality, design and construction

	standards, compliance with regulations, nor physical attributes of the system. Because requirements and design models are done by separate groups, the requirements and design models are incomplete.
Use of work product attributes and associated measures	A minimum of work product attributes and associated measures are defined within the individual tools, however the measures (and reports based on the measures) are often not consistent across organizational units and system life cycle process activities are often out of date and inconsistent.
Configuration Management	The project baselines and configuration manages individual printed or electronic file versions of the work products (e.g., pdf files) developed within the tools. The single source of truth of project's data and information is represented by these baselined and configuration managed files that represent the various work products. Because these files may be baselined at different times, the files are often inconsistent making it difficult to establish a single source of truth.

Table 3: SCL 1

SCL 2: For less-complex systems, many of the benefits of SE from a data-centric perspective listed earlier can be realized. A diagram of the various system life cycle process activity sets of data closely resembles that shown in Figure 7. The current state of the key factors/attributes of the organization's SCL are shown in Table 4.

Factor/Attribute	Current State
Organization	Organizational silos are disappearing, but some still exist. The focus is increasingly on multidiscipline, collaborative, teams (e.g. Integrated Product Teams).
Enterprise Level Data & Governance Policy, Processes, & Procedures	The enterprise may or may not have documented and implemented a data and information governance policy, processes, and procedures.
Project level Data and Information Management	Some of the projects within the enterprise have included data management concepts in their PMP and SEMP and have an IMP consistent with the enterprise level information governance policy, processes, and procedures.
Master Ontology	There is no defined master ontology for the enterprise, however some projects have a defined master ontology.
Master Schema	Some of the projects have defined a project level schema. Individual tool schemas are partially consistent with the project master schema making it easier to share and integrate data sets from different SE & PM tools
Systems Engineering (SE) Tool Set	Most of the work products are being developed using SE tools vs office applications. However, a variety of legacy SE tools (e.g., language-based modeling tools, requirement management tools) are still used to develop and manage some SE work products and their underlying data independent from other tools.
Project Management (PM) Tool Set	Most of the project work products are being developed using PM tools vs office applications. However, a variety of legacy PM tools are still used to develop and manage some PM work products and their underlying data independent from the SE tool set. Some of the PM tools may be partially compliant with interoperability and data sharing standards.
Work product Format	There may still be some use of common office applications, however work products and their underlying data and information are managed electronically with any paper- or electronic-based documentation considered as "reports" that represent the electronic data and information at the time of printing. Most work products are communicated as "pdf" files.
Shareability of data and information	Projects have started to establish shareable sets of data based on their master schema and uses these shareable sets of data to manage work products and their underlying data across many system life cycle process activities. To share

	the data. SE & PM tools are at least partially in compliance with data sharing and interoperability standards. Extract, Transform, Load (ETL) applications are developed or procured and used to input the data from the external databases into the project's integrated or federated, shareable sets of data. Changes to the external SE & PM tool databases must go through the ETL process before the changes are included in the shareable sets of data. This makes it difficult to keep the data and information in the integrated or federated, shareable sets of data current and consistent across all system life cycle process activities and with the external databases. Anyone doing analysis, modifying work products and their underlying data and information, or generating reports based on the data in the integrated or federated, shareable sets of data, must make sure the data from the external databases is current and has went through the ETL process before being brought into the integrated or federated shareable sets of data before using that data.
Linking (Traceability) between work products developed in different development life cycles	Many, but not all, of the PM and SE work products and underlying data and information are linked not only within system life cycle silos, but also across system life cycle stages. For example, requirements are linked to the stakeholder needs and higher-level requirements allocated to the system, requirements are linked to models, design artifacts are linked to requirements, system verification and system validation is linked to design artifacts and requirements. There is traceability between stakeholder needs, requirements, analysis, models, design artifacts, verification, validation.
Consistency of data across work products	The project has shareable sets of data and traceability between work products across all life cycle states, making it easier to establish and maintain consistency between work products.
Completeness of work products	The project has shareable sets of data and traceability between work products across all life cycle states, making it easier to assess completeness of the data and information within work products. Requirement sets include both functional and performance requirements developed as part of the modeling effort, but also include non-functional requirements including quality, design and construction standards, regulations, and physical attributes of the system that are difficult to include in logical models of the system.
Use of work product attributes and associated measures	The PMP and SEMP define work product attributes to be used to manage the overall SE and PM effort across all system life cycle stages. The PMP and SEMP define measures like MOSs, MOEs, MOPs, KPPs, TPMs, LIs to be included in the integrated, shareable sets of data. Project data and information are linked with the SE process data and information. The data representing measures and work product attributes are accessible and used to generate reports, dashboards, etc. which are used to better manage the project and system engineering processes.
Configuration Management	The project manages the various system life cycle process data and information from the integrated or federated, shareable sets of data. Various individual work products are "reports" generated from these sets of data. However, these reports are what are still baselined and configuration managed by the organization (as contrasted with baselining the sets of data(s) that represent those work products). The single source of truth of project's data and information is represented by these baselined and configuration managed files that represent the various work products. Because these files may be baselined at different times, the files are often inconsistent making it difficult to establish a single source of truth.

Table 4: SCL 2

SCL 3: Most the benefits of SE from a data-centric perspective listed earlier can be realized for more complex systems. A diagram of the various system life cycle process activity sets of data closely resembles that shown in Figure 8. The current state of the key factors/attributes of the organization's SCL are shown in Table 5.

Factor/Attribute	Current State
Organization	Silos within the project do not exist, or at least are minimized. The focus is on multidiscipline, collaborative, teams (e.g. Integrated product teams).
Enterprise Level Data & Governance Policy, Processes, & Procedures	The enterprise has documented and implemented a data and information governance policy, processes, and procedures.
Project level Data and Information Management	Most if not all of the of the projects within the enterprise have included data management concepts in their PMP and SEMP and have an IMP consistent with the enterprise level information governance policy, processes, and procedures.
Master Ontology	A master ontology for the enterprise and project has been developed. Project's develop a project level ontology consistent with the enterprise level ontology.
Master Schema	A master schema for the projects have been defined. Most SE & PM tool schemas are consistent with the project master schema.
Systems Engineering (SE) Tool Set	The project has transformed their SE process such that most of the SE work products are being developed using SE tools that are fully compliant with interoperability and data sharing standards. SE tool data and information is shareable with PM tool data and information.
Project Management (PM) Tool Set	The project has transformed their PM process such that most of the PM work products are being developed using PM tools that fully conform to interoperability and data sharing standards. PM data and information is shareable with SE tool data and information.
Work product Format	The project manages the various system life cycle process activities and work products and their underlying data and information from the project's integrated or federated, shareable sets of data. Most of the PM and SE tools store the data and information either as electronic files or in a database whose schema is consistent with the project master schema allowing the PM & SE tool databases to be included directly as part of the project's integrated, shareable sets of data.
Shareability of data and information	Because the Project's SE Tools adhere fully to interoperability and data sharing standards and consistent schemas, the data and information in these individual databases is compatible - enabling the SE & PM tools to share data and information both internally and with organizations external to the enterprise. Note: Even though most of the SE tools have compatible databases included in the integrated or federated, shareable sets of data, the enterprise may require the project to continue to use some legacy systems, whose schema is not compatible with the integrated, shareable sets of data. In this case, this data must go through an ETL process before the data can be included in the integrated, shareable sets of data and be accessible by other tools.
Linking (Traceability) between work products developed in different development life cycles	Most of the PM and SE work products and underlying data and information are linked also across system life cycle stages. For example, requirements are linked to the stakeholder needs and higher-level requirements allocated to the system, requirements are linked to models, design is linked to requirements, system verification and system validation is linked to design and requirements. There is traceability between stakeholder needs, requirements, analysis, models, design, verification, validation.
Consistency of data across work products	Because the project has shareable sets of data and traceability between work products across all life cycle states, it is much easier to establish and maintain consistency between work products.
Completeness of work products	The project has shareable sets of data and traceability between work products across all life cycle states, making it easier to assess completeness of the data and information within work products. Requirement sets include both functional and performance requirements developed as part of the modeling effort, but also include non-functional requirements including quality, design and construction standards, regulations,

	and physical attributes of the system that are difficult to include in a logical model of the system.
Use of work product attributes and associated measures	The PMP and SEMP define work product attributes to be used to manage the overall SE effort across all system life cycle stages. The PMP and SEMP define measures like MOSs, MOEs, MOPs, KPPs, TPMs, LIs to be included in the integrated, shareable sets of data. Project data and information are linked with the SE process data and information. The data representing measures and work product attributes are accessible and used to generate reports, dashboards, etc. which are used to better manage the project and system engineering processes.
Configuration Management	There is only one single source of truth for the project - the data and information in the integrated or federated, shareable sets of data. The project's data and information in these sets of data is under strict configuration control and therefore represents the baseline state of the project at any given time. The work products and underlying data and information are developed, analyzed, and managed holistically as an integrated system made possible because of the existence of these shareable sets of data. Any "visualizations" of the data and information in the these sets of data represent the current state of the project. Even when these visualizations are extracted as reports, the single source of truth is still the data and information model from which they were generated. There may still be some use of common office applications, however the master, ground-truth, data and information are managed electronically with any paper-based documentation visualizations of the data and information considered as "reports" that only represent the electronic data and information at the time of printing.

Table 5: SCL 3

SCL 4: SCL 3 plus:

- All the underlying data representing the system life cycle work products is included in the project's integrated or federated, shareable sets of data. All PM and SE work products and underlying data and information are developed using SE tools that conform fully to interoperability and data sharing standards and store the data and information in a database whose schema is consistent with the project's master schema. This allows all SE tool databases to be included directly as part of the project's integrated or federated, shareable sets of data, enabling all SE tools to share data and data to be shared with other internal and external organizations. All the work products are linked not only within a system life cycle stage, but also across system life cycle stages.

- Common office applications are used to view reports that only represent the electronic data and information in the integrated, shareable sets of data at the time of printing. Rather than baselining these reports, the sets of data and information from which the reports are generated are baselined and configuration managed. The project manages the various system life cycle process activities from the integrated, shareable sets of data. This integrated or federated, shareable sets of data represent not only an integrated architectural model of the system under development but also represents a model of all the system life cycle process activities and resulting work products and their underlying data and information.

- All the benefits of SE from a data-centric perspective listed earlier can be realized. A diagram of the various work products sets of data closely resembles that shown in Figure 9.

SCL 5: The enterprise has an enterprise level ontology defined and documented. The enterprise has defined and documented an enterprise level data and information governance policy, processes, and procedures. The enterprise has developed an enterprise level IMP. Two or more projects within the enterprise are operating at SCL 4.

Not every enterprise needs to be at SCL 5. Not every project needs to be at SCL 4. Most projects should strive to be at least at SCL 2 but are encouraged to get to SCL 3 or higher, that is, IF there is an ROI to the enterprise/project for doing so. Take baby steps. The enterprise may set a goal of being at level 4 or 5, assessing their current level, identifying the SCL appropriate for the organization, and then developing a roadmap for getting there.

Note: A project may be currently using analytical models as part of their system life cycle process activities, but unless they are managing the models and all other SE work products and their underlying data and information in integrated or federated, shareable sets of data, they have not yet reached SCL 4. Depending on the degree of data and information integration, these projects may be at SCL 1, 2, or 3.

5.2 Choosing an Appropriate Systems Engineering Toolset

Currently, in order to meet the intent of the MBSE Initiative, many organizations want to increase their capability to practice SE from a data-centric perspective. Individual SE tools tend to focus on specific needs and types of work products. Organizations (at the enterprise level) need to perform trade-studies to see if the expense of purchasing a specific SE tool or toolset, maintaining the licenses, training people to use the tool(s), maintaining the tool(s), maintaining models and other work products and their underlying data and information developed by the tool(s), etc. are going to provide a positive ROI, improved time to market, reduce the number of product defects, reduce the amount of rework, or reduce warranty work and recalls.

Given today's systems are increasingly complex, an SE toolset, including requirement management tools, diagraming tools, modeling tools, budgeting tools, scheduling tools are needed to help manage the challenges associated with these increasingly complex systems.

An SE toolset provides a more effective way of carrying out portions, or in some cases, all the system life cycle process activities using a data-centric perspective.

> To more effectively develop systems, the SE toolset needs to be tailored to an organization's needs, as evidenced by statements such as the following in NASA's NPR 7123.1, NASA Systems Engineering Processes and Requirements (NASA 2013) "*...technical teams and individuals should use the appropriate and available sets of tools and methods to accomplish required common technical process activities. This would include the use of modeling and simulation as applicable to the product-line phase, location of the WBS model in the system structure, and the applicable phase exit criteria*."

Going beyond requirements, there are SE tools that can support the entire system lifestyle including budgeting, scheduling, defining, designing, building/coding, verifying, validating, and sustaining engineering activities. These tools are used to collect, link, visualize, analyze, manage, and communicate data and information across all system life cycle stages. These more robust SE tools allow the organization to produce various views of the system under development and create and maintain the various work products (documents, databases, reports, diagrams, drawings, models, etc.) and their underlying data and information needed to more effectively manage the system development efforts as shown in Figure 3 and Figure 9.

A key issue with tools that claim to support the entire life cycle of a product, lack the robustness of tools that focus on a specific capability. For example, while a requirement management tool is great at supporting the development and management of sets of requirements, these tools often lack the capability to include diagraming and modeling as part of requirement development and modeling. Similarly, tools that are very good at supporting diagraming and modeling often lack key features of RMTs to adequately support the development and management of sets of requirements.

What capabilities are needed from an SE toolset depends on the product line, its complexity, green-field vs brown-field products, issues the organization is having and wants to address, and the workforce knowledge and experience. Organizations need to understand what data and information best meets their needs and which set of SE tools they need to work with and manage this data. SE tools are like any other software application…one size does not fit all. The SE toolset that is best for an organization is the toolset that meets the organization's requirements management, systems engineering, and modeling needs. Consider the outcomes needed as a result of using SE tools and the ROI resulting from these outcomes.

Before embarking on an SE toolset evaluation and selection initiative, work with management, project teams, engineering staff, and other key stakeholders to determine what the organization needs to help better manage the development of the systems in their domain. What features and functionality are needed in an SE toolset so the projects can effectively and efficiently manage their requirements, design, and other system life cycle process activities throughout all system life cycle stages? Specifically, choose the SE toolset that supports the SCL the project has decided to strive for. Above all, select SE tools that support the concept of SE from a data-centric perspective using the project's integrated or federated, shareable sets of data.

As stated earlier, it is advisable to choose SE tools that support the generation and management of multiple system life cycle work products and their underlying data and information and especially SE tools that fully support interoperability standards for compatible tools, schemas, and databases. The perfect case would be to procure a single SE tool that "does it all", i.e., the one tool would result in the project having integrated or federated, shareable sets of data by default. That would help to ensure all data and information is current and consistent across all system life cycle stages. Again, *vendors are in the process of developing tools that will support the development and shareability of PM and SE data and information across all life cycle stages*.

For a detailed list of features an SE toolset should have, see Appendix C.

5.3 Integrating Systems Engineering from a Data-Centric Perspective into Your Organization.

A major challenge to implementing the chosen SCL for a project is convincing management and co-workers that it is time to implement or improve the organization's SE capabilities and knocking down the walls of resistance. Some common reasons for them not wanting to implement or improve SE capabilities include: (Note: see also David Long's blog: "9 Imaginary Roadblocks to MBSE".)

- "We have been doing product development using our current processes for years, why should we change?"
- "Implementing SE from a data-centric perspective may work for others, but not for us."
- "This all seems very complicated, we don't have the knowledge, experience, or tools."
- "Our current SE work products, like requirements, are managed in an RMT. FFBDs, etc. are models, so aren't we already doing model-based SE?"

- "It is too expensive to procure the needed SE toolset, maintain the tools, and train our people to use those tools."
- "We don't have the budget to incorporate SE from a data-centric perspective at this time."
- "The expense and associated process to get new SE toolset installed on organizational computers is too great."
- "We would have to make signification IT infrastructure upgrades to accommodate the additional volume of data and performance requirements of the new SE tools."
- "We deal with the development of classified systems; controlling access and maintaining security will be too difficult."

Sound familiar? Often the pushback can be attributed to a lack of understanding risks associated with the current state of the organization, the benefits of moving toward a more data-centric practice of SE, and what level of SE capability is appropriate for the organization.

So how can management convince be convinced that some level of SE capability is needed? Three words - RETURN ON INVESTMENT! Think about it … what has been the impact of the current, poorly executed product development efforts? Failures, recalls, returns, warranty costs, lawsuits, negative reviews on social media, decreasing market share - not good for profitability! The ROI argument usually works with management especially when they can be convinced that by investing in an SCL tailored to the organization's needs, the overall product development process, product quality, time to market, and, profitability can be improved.

The more effective the SE processes, the less rework and fewer cost and schedule overruns. By implementing the appropriate level of SE capability, the probability of achieving a competitive advantage can be improved by removing obstacles to being able to deliver products on time, on budget, and that meet or exceed customer and quality expectations.

First, use the tables in section 5.1 to help assess what an organization's current SCL is currently at. Next identify the issues the organization is having given their current SCL. Appendix D, Systems Engineering Issues Questionnaire, can be used to identify specific issues an organization may be having because of the organization's current SCL. Try to quantify, with examples, these issues (could be monetary, opportunity costs given limited resources, time to market, etc.). Also address the costs associated with having multiple databases and the cost, effort, and issues associated with integrating and sharing data across organizational silos and other external organization that may be involved in a system's development.

Next, determine which SCL needed by the organization to address these issues. Provide rationale for being at this level. How will being at this level address the issues identified because of operating at the present SCL. Again, management likes to see the numbers. Do a gap analysis to determine what changes will need to be made and a rough order of magnitude of the costs and time to get to the new SCL. What ROI should they see if they approve moving the organization to this new SCL? What would be gained by investing in the infrastructure needed to get to this SCL and what would be the savings (costs of labor, tools, less defects, less recalls, less rework, etc.) If a positive ROI can be shown, management will be much more receptive.

Even if a good case is made for implementing a chosen SCL or improving the existing SE capability and have management backing in doing so, the rank and file may still push back. When this happens, the source of the push back is more often due to cultural dread and anxiety associated with change; not process aversion.

5.3.1 Changing Culture is Often Met with Opposition.

To combat this opposition, determine which stakeholders are for and against the change and why. For each individual or group, identify his or her concerns and devise a strategy to get the change adversaries to become change advocates. Start with those stakeholders that have the most influence and convince them by addressing their concerns. The aid of other stakeholders may need to be obtained that can help influence those that oppose the change. Success often depends on having a champion at the enterprise level.

Many engineering organizations tend to be very conservative in the way they do business. Rather than making a big revolutionary change, develop a strategy that involves incremental, evolutionary change over time. For example, initially, the core SE team may be the only ones who actually work directly within the SE toolset for the short term. They can work with the other engineering and management team members to ensure the data is accurate and sharable. The broader team does not have to learn the ins and outs of the actual SE toolset immediately but can have access to the various reports and view the outputs at any time. As the team gains experience with the new SE processes, they can increase their direct use of the SE tools associated with their specific function.

Using the SCLs defined earlier will enable an organization to take small bites rather than moving the whole organization to the new processes at once. True, it may take a long time to get where the organization needs to be, but as long as the small changes result in improvements and make the engineers' jobs easier, they will go along with the change. Most people follow the path of least resistance.

Make the path easier to follow easier and more beneficial than the current one. If the change results in more work or makes communication harder, the battle will be lost. For example, the lead engineer or project manager may already be over their head and working 50-60 hour weeks. Having them learn and implement a new process and a new tool(s) may be too much of a load for them to bear! However, if they are provided with a dedicated engineer that has the training, knowledge, and experience in the chosen SCL and associated processes and SE toolset to help implement the changes, they will be much more receptive. They will also be much more receptive if this results in them having to work less hours and having fewer crises to deal with each day!

People at all levels must be convinced of the utility of the changes, how the changes result in a better product, and result in less rework for them. *[Frequently the reason they are working the long hours is because they are always fighting fires, going from one crisis to another, that resulted from the lack of the proper SE tools, processes, data, and information in the first place! The culture needs to be changed from one of firefighting to one of fire prevention.]* As time passes, they may even start to be advocates for the changes that have been made and welcome further change. Depending on the size of an organization and culture, this process could take several years.

Even if individual projects only last a year or two, the organization will be in this for the long haul and must persevere and not give up and lose faith. Be prepared to make minor course corrections when things don't work as expected. Always keep the end in mind. Incorporate the incremental changes into the existing process rather than replacing them wholesale. Work to gain acceptance slowly. Get the people used to seeing and using bits of the processes in small ways and grow it outwards. People will never appreciate being told their current way is poor or old fashioned - they must discover the benefits of incorporating proposed changes themselves.

5.3.2 Use a Pilot Project

It is advisable to study the current organization and workflow and look for a project that can be used to demonstrate the benefits and ROI of the proposed changes in the short term. Use this pilot project as a viable example to effect larger scale change within the organization.

A small project will allow the organization to gain an understanding of what works (provides value, ROI), what doesn't, what is liked, and what isn't liked.

This pilot project can develop an example PMP, SEMP, and IMP that can be used as a template for other projects. A project ontology and schema can be developed that can also be reused by other projects. Finally, the implementation of integrated or federated, shareable sets of data for the project can be worked out. Armed with the lessons learned from the pilot project, the organization can develop a roadmap for new projects to practice SE from a data-centric perspective.

Several key steps include:

1. Develop a practical process that implements the chosen SCL. A good process is one that people can follow as part of their job, not something they have to do in addition to their job. Also, the process needs to fit the product line, domain, and culture of the organization. The implementation needs to be tailored to the project. Don't try to follow a process developed by a tool vendor for some other organization or product line.
2. Invest in training in the proposed chosen SCL, the SE tools involved, and the associated processes.
3. Pick a pilot project to apply the process and assign the grass roots data-centric SE advocates to that project.
4. Define and use measures so metrics can help document the ROI of implementing the chosen level of SE capacities can be clearly communicated.
5. Show management how measures and metrics maintained within the PM and SE tools will help them better track the health and status of the project, enabling them to be better project managers and systems engineers.
6. Encourage team members to be actively involved in organizations like INCOSE and join working groups whose members can aid the implementation process.
7. Invest in an outside consultant who has a proven track record in implementing SE capabilities consistent with the chosen SCL and chosen SE toolset.

Once the project is completed successfully (an assumption) then the project can be used as an example to get other projects to follow. The core grass roots folks can be spread out among other projects and mentor other project managers and systems engineers and train them and their teams in the concept of practicing SE from a data-centric perspective and in the use of the chosen SE toolset.

In many of the cases where adoption has been successful, there has been both advocacy at the top as well as a strong grass roots support that has gradually gained acceptance across the organization, but typically only after one team has proven success.

An organization will be successful in practicing SE from a data-centric perspective when it is considered to be the standard for system development. However, the road to success is long - it takes very strong, unwavering leadership and experience to get this done right. It is human nature to try to push back and say that it isn't possible. **It is possible!**

By implementing the chosen SCL the organization will be able to better address the challenges discussed earlier, meet the intent of the MBSE Initiative, reap the benefits of data-centric SE, and move closer to INCOSE's Vision 2025!

APPENDIX A: REFERENCED DOCUMENTS

Bellinger, G., 2004, "Modeling & Simulation: An Introduction", http://www.systems-thinking.org/modsim/modsim.htm .

Bellinger, G., Castro, D., Mills, A., 2004, "Data, Information, Knowledge, and Wisdom", http://www.systems-thinking.org/dikw/dikw.htm .

Berson, A. and Dubov, L., 2011, "Master Data Management and Data Governance", Second Edition, McGraw Hill, ISBN 978-0-07-174458-4

Dick, J. and Chard, J., 2004, "The Systems Engineering Sandwich: Combining Requirements, Models and Design", *INCOSE International Symposium IS2004*, July 2004.

Dori, D., 2002, "Object-Process Methodology: A Holistic System Paradigm". New York, NY, USA: Springer.

DoD 5000.59-M, 1998, "DoD Modeling and Simulation – Glossary", Defense Modeling and Simulation Office.

Friedenthal, S., Moore, A. and Steiner, R., 2008. "A Practical Guide to SysML: Systems Modeling Language", Morgan Kaufmann Publishers, Inc.

Ghavami, P., 2016, "Big Data Governance: Modern Data Management Principles for Hadoop, NoSQL, & Big Data Analytics", ISBN: 978-1519559720

Hull, E., K. Jackson, J. Dick, *Requirements Engineering*, Springer, 2011.

IEEE STD 610.12-1990. "IEEE Standard Glossary of Software Engineering Terminology". superseded by ISO/IEC/IEEE 24765:2010, Systems and Software Engineering – Vocabulary. Washington, DC, 2010

INCOSE-TP-1995-002-01, 1995. "Metrics Guidebook for Integrated Systems and Product Development", Wilbur, A., Towers, G., Sherman, T., Yasukawa, D. and Shreve, S.

INCOSE-TP-2003-020-01, 2005, Technical Measurement, Version 1.0, 27 December 2005. Prepared by Roedler. G. and Jones, C.

INCOSE-TP-2003-002-04, 2015, Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities. Version 4, Edited by Walden, David D., et al.

INCOSE-TP-2004-004-02, Systems Engineering Vision 2020, Version 2.03, September 2007, http://www.incose.org/ProductsPubs/pdf/SEVision2020_20071003_v2_03.pdf

INCOSE-TP-2005-001-03, 2010, Systems Engineering Leading Indicators Guide, Version 2.0, January 29, 2010; available at http://seari.mit.edu/documents/SELI-Guide-Rev2.pdf . Edited by Roedler, G., Schimmoller, H., Jones, C. and Rhodes, D.

INCOSE-TP-2010-005-02, 2010, Systems Engineering Measurement Primer: A Basic Introduction to Measurement Concepts and Use for Systems Engineering, INCOSE

INCOSE-TP-2010-006-02, 2015, Guide to Writing Requirements, Chapter 5, Prepared by the Requirement Working Group, INCOSE

INCOSE-TP-2015-001-01, 2015, Project Manager's Guide to Systems Engineering Measurement for Project Success, prepared by the Measurement Working Group (MWG), INCOS

ISO/IEC/IEEE 15288, 2015, Systems and Software Engineering— System Life Cycle Processes. Geneva, Switzerland: International Organization for Standardization

ISO/IEC, 2001, ISO/IEC 29148 FDIS Systems and Software Engineering—Life Cycle Processes—Requirements Engineering,

ISO/IEC TS 24748-1:2016 Systems and Software Engineering -- Life Cycle Management -- Part 1: Guidelines for Life Cycle Management

Ladley, John, 2012, "Data Governance: How to Design, Deploy, and Sustain an Effective Data Governance Program", Morgan Kaufmann, ISBN 978-0-12-415-829-0

Long, D., 2016,"One Model to Coordinate them All", http://community.vitechcorp.com/home/post/One-Model-to-Coordinate-them-All.aspx , Blog, Vitech Corporation

Long, D., 2016, "9 Roadblocks to MBSE", http://community.vitechcorp.com/home/post/9-Imaginary-Roadblocks-to-MBSE.aspx , Blog, Vitech Corporation

Long, D. and Scott, Z., 2011. "A Primer for Model-Based Systems Engineering", 2nd Edition, Vitech Corporation

Malone, R., Herrord, J., Friedland, B., Fogarty, D., 2016. "Paper Insights from Large Scale Model Based Systems Engineering at Boeing". 26th Annual INCOSE International Symposium (IS 2016), Edinburgh, Scotland, UK, July 18 - 21, 2016

NASA, 2016, Expanded Guidance for NASA Systems Engineering: Volume 2: Crosscutting Topics, Special Topics, and Appendices, NASA Headquarters

NBS, 2018, "What is Building Information Modelling (BIM)?", https://www.thenbs.com/knowledge/what-is-building-information-modelling-bim

Novel Engineering, 2016, "An Introduction to Model-Based Systems Engineering", Presentation

NPR 7123.1B, 2011. "Systems Engineering Processes and Requirements", NASA Procedural Requirements, NASA

OMG, 2010. "MDA Foundation Model". Needham, MA, USA: Object Management Group. ORMSC/2010-09-06

Scott, Z., 2016, "Models and Views", http://community.vitechcorp.com/home/post/Models-and-Views.aspx , Blog, Vitech Corporation

Soares, S., 2014, "The Chief Data Officer Handbook for Data Governance", MC Press Online, LLC, ISBN 978-1-58347-417-4

Starling, H., 2015, "Data Governance Simplified: Creating and Maintaining Trusted Data for Business"

Ryan, M., 2013. "An Improved Taxonomy for Major Needs and Requirements Artefacts", 23rd INCOSE International Symposium, Philadelphia

Ryan, M., Wheatcraft, L, Dick, J, and Zinni, R., 2014. "An Improved Taxonomy for Definitions Associated with a Requirement Expression", Systems Engineering/Test and Evaluation Conference SETE2014, Adelaide

Wheatcraft, L., 2015, "Features an SE Toolset Should Have", blog, http://reqexperts.com/blog/2015/12/features-a-re-tool-set-should-have/

Wiegers, K.E., 2003, "*Software Requirements"*, Redmond, WA: Microsoft Press

Wiegers, K.E.,2006, "*More About Software Requirements"*, Redmond, WA: Microsoft Press

Wikipedia, 2018, Data Integration, article, https://en.wikipedia.org/wiki/Data_integration

Wikipedia, 2018, Federated Database System, https://en.wikipedia.org/wiki/Federated_database_system

APPENDIX B. ACRONYMS AND ABBREVIATIONS

API	Application Program Interface
BA	Business analyst
CAD	Computer-aided design
ConOps	Concept of Operations
DB	Database
DBMS	Database Management System
DG	Data Governance
DoD	Department of Defense
ETL	Extract, Transform, Load
FFBD	Functional Flow Block Diagrams
HB	Handbook
ICD	Interface Control Document
IEC	International Electro-technical Commission
IEEE	Institute of Electrical and Electronics Engineers
IM	Information Management
IMP	Information Management Plan
INCOSE	International Council on Systems Engineering
ISO	International Standards Organization
IT	Information Technology
IW	International Workshop
KPP	Key Performance Parameter
LI	Leading Indicator
MBSE	Model-Based Systems Engineering
MOE	Measure of Effectiveness
MOP	Measure of Performance
MOS	Measure of Suitability

NASA	National Aeronautical and Space Administration
OMG	Object Management Group
OpsCon	Operational Concept
OSLC	Open Services for Life cycle Collaboration
PLCS	Product Life Cycle Support
PM	Project Management
PMP	Project Management Plan
QA	Quality Assurance
RFA	Request For Action
RMT	Requirement Management Tool
ROI	Return on Investment
RWG	Requirements Working Group
SE	Systems Engineering
SEMP	Systems Engineering Management Plan
SBP	Strategic Business Plan
SCL	Systems Engineering Capability Level
SSoT	Single Source of Truth
STEP	STandard for the Exchange of Product
SysML	Systems Modeling Language
TBD	To Be Determined
TBR	To Be Resolved
TIMLM WG	Tool Integration and Model Life cycle Management Working Group
TPM	Technical Performance Measure
UML	Unified Modeling Language
V&V	Verification and Validation

APPENDIX C: FEATURES AN SYSTEMS ENGINEERING TOOLSET SHOULD HAVE

Below are the features to be considered for an SE toolset to be selected by an organization. Including these features will enable the organization to achieve the capabilities they need to effectively implement their PM and SE processes from a data-centric perspective.

The order of the list does not in any way imply priority. Priority of these features and functions and the "importance-weighing factor" for each is left up to the evaluating organization based on its unique product development and management needs.

The authors do not expect any one SE tool to include everything in this list as many vendors tailor their tool to a specific client base needs or a specific system life cycle stage and set of work products. However, it would be preferable to minimize the number of different applications for the organization's SE toolset tailored to their specific domain, product line, and processes consistent with the SCL they are moving toward.

The INCOSE Tools Integration and Model Lifecycle Management (TIMLM) WG is leading an effort to develop a SE Tools Database (SETDB) and offer it as a service to members. This database will include a listing of all SE tools available, with specific capabilities and features defined. The plan is to make the SETDB interactive, allowing users to specify the capabilities and features they need and to set up priorities/weighing factor of each to allow them to select and rank SE tools available to meet the needs of the organization. INCOSE members can learn more and participate in this effort by going to the SETDB Connect site.

Functionality: (What capabilities and feature are recommended?)

Requirement Quality: Does the SE toolset include the capability to support SE best practices concerning the quality of requirement statements? For example, does the SE toolset enforce requirement standards such as those defined in the *INCOSE Guide to Writing Requirements*? This includes the ability of the SE toolset to help requirement authors to write properly formed requirements (spelling, grammar, unambiguous terms, requirement statement structure, consistency, etc.) and to assess the quality of a set of requirements based on the organization's standards for writing requirements.

Grammatical Structure of Requirements: Rather than treating requirement statements as an indivisible entity, does the SE toolset allow textual requirements to be managed as a grammatical structure? Within this structure the requirement statements refer to various entities: architectural parts, functions, conditions, states, modes, events, transition, interfaces, units of measure. To relate those references to the corresponding entities in the overall system model involves getting inside the grammar of the requirement statement. This capability will allow requirements to be an integral part of the overall system model. This capability also is key to ensuring all the entity names are consistent with the system and project-wide ontology (discussed below.)

Allocation and Traceability: Does the SE toolset support the key concepts of allocation and traceability between not just requirements but all work products and their underlying data - no matter the level? For requirements, this includes allocating requirements from one level of the architecture to parts of the architecture at the next level. Once children requirements are derived, traceability involves linking child requirements to their parent and linking requirements to their source. If developing analytical models, this allows requirements to be linked to the applicable parts of the architecture in the model.

Interface management: Does the SE toolset support the documentation of interface definitions (e.g., ICDs) and the corresponding interface requirements that are linked to those definitions? Can the toolset be configured to link an interface requirement from one element of the system architecture to the corresponding interface requirement for another element with which the first element interacts with? Can the toolset be configured to notify owners of a proposed change to a complementary interface requirement(s) or their definition? (This topic deals with the not only internal interfaces, but also interfaces between the system under development and external systems it is required to interact with.)

Dependencies between work products and their underlying data and information: Does the SE toolset allow users to link dependent requirements and other work products across all system life cycle process activities and their underlying data and information to each other? (This is important when a change to one work product could necessitate a change in another work product and underlying data and information.) The dependent work product may be part of the system under development or another system. Does the SE toolset allow users to do consistency assessments between dependent requirements and work products? Can the toolset be configured to notify owners of dependent work products when a change is made to one of the dependent work products? Note: this feature is supported by the traceability feature.

Impact Assessment: Does the SE toolset allow the user to assess the impact of a change vertically among levels of the architecture as well as horizontally across all work products from all system life cycle process activities helping the user to understand the impact of a change to other work products and the project's delivery schedule, cost, quality? Does the SE toolset allow the user to do change impact assessment to other work products generated in other system life cycle process activities whose underlying data resides in a separate database? For example, what is the impact of a requirement change on design? A change in an analytical model on requirement linked to that model? On a dependent requirement? A requirement change to system verification planning? System validation planning?

Ontology: Does the SE toolset allow users to define a system ontology as well as include the capability to expand its inherent ontology to define a project-wide ontology? An ontology includes the formal naming and definition of a set of terms, entities, data types, and properties as well as defining the relationships between these terms, entities, data types that are fundamental to a domain. (See Section 3.4.1) Defining an ontology is critical to ensuring consistency and allowing sharing of data and information across system life cycle process activities as well as reusability within the enterprise. *Note: Most SE tools provide the capability to define an ontology for the data and information managed within the tool, but often the ontology is proprietary. From a shareability perspective, what is needed is for all tools in the SE toolset to have an ontology that is consistent with the project-wide ontology.*

Schema: Does the project's SE toolset include the capability to define a master project schema for its integrated or federated, shareable sets of data consistent with the -wide ontology? The schema is a description, in a formal language, of the database structure that defines the objects in the databases, shows how real-world entities are modelled in the database, and integrity constraints that ensure compatibility between parts of the schema. (See Section 3.4.2) Do the tools in the SE toolset conform to standards for development of a common schema (e.g., OSLC)? SE tools in the projects SE toolset need to ensure their schemas are consistent with the project's master schema defined for the project's integrated or federated, shareable sets of data to ensure compatibility of the data, allowing the data to be shared among the various SE tools in the project's SE toolset. (*See also interoperability and tool integration later in the list.)*

Embedded Objects: Does the SE toolset allow the user to embed objects with various electronic formats (tables, pictures, drawings, diagrams, RTF files, word processing documents, spreadsheet documents, test procedures, etc.) that can be linked to other work products?

Diagrams and Drawings: Does the SE toolset support the development and management of diagrams and drawings as electronic files that can be linked to other work products and their underlying data independently from an analytical modeling tool?

Modeling: Does the SE toolset support the development and use of architectural, analytical, and physical models? Does the SE toolset allow the development and documentation of use cases, functional flow block diagrams, states and modes diagrams, timing diagrams, requirement expressions, and other types of models needed by the project and store the underlying data and information representing these activities and diagrams in a database consistent with a project's master schema? Will the toolset allow the user to develop high fidelity logical and physical models that support simulations – if that capability is needed by the organization? Does the toolset allow the creation of an extensible data model that can be easily constrained by a rule set; an extensible API to allow incorporation of custom data creation and manipulation utilities; a rich, natural language query engine?

Reusability: What features does the SE tool have that will enable re-use of work products and their underlying data and information for similar projects or projects involved in updating an existing product? Can the work products and underlying data for one version of a product be duplicated and used as the basis for the next version?

Product-line management: In addition to reusability, what features does the SE toolset have that supports product-line management? Does the tool allow branching of work products, e.g., for a class of systems, the same root requirement can be branched to multiple versions of the root requirement?

Tool Attributes: (What features do the tools need to have to allow the tool to be tailored to support the organization's specific needs?)

Tailorablity: Can the SE toolset be tailored within the organization based on a project's need e.g., complexity, team knowledge, development methodology, size, processes, timeframe, customer requirements, ontology, schema). While the enterprise may select a toolset consistent with SCL 3, 4 or 5, can projects within the organization use a subset of the features that best meets their needs? It is not good practice to require the use of a large mallet when a small hammer is all that is needed.

Configuration/Customization: Does the SE toolset include the ability to configure and/or customize the toolset to the customer's domain, culture, and organizational processes? Does the SE toolset allow individual users to configure their interface based on their unique role and use of the SE toolset? (With minimum help from the vendor.) *Note: Configure refers to the ability to configure the tool to meet user needs without changing the code. Customize involves changes to the tool code to provide new or tailored features needed by the customer. Having a tool that can be configured to meet an organization's needs is cheaper than paying a vendor for the tool to be customized to meet those needs.*

Learnability/usability: Does the SE tool have a user interface that is intuitive, user friendly, and easy to use with a small learning curve? How much training is necessary and available? Is online documentation and help functionality included?_Does the SE toolset provide methods allowing the user to navigate between various work products and visualizations such as: requirements, documents, configuration management information, reports, diagrams, design artifacts, etc.?

Security: Does the SE toolset provide security of the data and information in terms of access (at multiple levels, within levels, and different user classes), protection (from loss), and integrity? Does the SE toolset support the security standards applicable to the organization's domain and product type(s) and data governance policy? – Information security includes the principles of confidentiality, integrity, and availability (CIA). Data and information must be protected from unauthorized use and disclosure, must be able to be trusted (accurate, consistent, of high quality, managed), and accessible to authorized personnel.

Accessibility (devices/location): Does the SE toolset allow users to access data securely via desktops, laptops, portable devices (tablets, smart phones) both inside and outside the organization's firewalls? Are the sets of data created by the toolset accessible by another organization's (vendor/supplier) SE toolset?

Online vs Offline Modes: Does the SE toolset require the user to be connected to the server continuously (online) to use the toolset or does the toolset allow offline work to be accomplished with synchronization after going back online?

Concurrent Access: How many concurrent users does the SE toolset allow to work within the same area? What happens when more than one user wants to edit the same work products and underlying data? For some complex systems there may be over a hundred users modifying various work products in the project's integrated or federated, shareable sets of data simultaneously.

Performance: What is the maximum wait time between user actions? How does the SE toolset minimize performance impacts as the number and complexity of work products increase as well as the number of concurrent users increase?

Collaboration: Does the SE toolset support collaboration among the users within the tool across all life cycle process activities? Does the SE toolset allow users to collaborate no matter where their workplace is located? Globally? Does the SE toolset allow external organizations (vendors/suppliers) to collaborate with the project team?

Tool integration – requirements: Does the SE toolset allow the integration of requirements developed in a Requirement Management Tool (RMT) to be linked to entities defined and managed in other diagraming and modeling tools? For example, can requirements developed and managed within an RMT be linked to entities within a SysML or other types of models?

Interoperability/tool integration – data sharing: Does the SE toolset allow the sharing of data with other SE tools (ReqIF compliant) as well as with other word processing and spreadsheet application supported formats? How easy is it for information to be transferred into and out of the SE toolset to support the organization's processes and people? Does the SE toolset provide a standardized interface for importing or exporting data from/to other applications (e.g. Model-based tools), rather than requiring specialized scripting, etc. to achieve a transfer/interaction? Can the SE tool perform extraction, transformation, and loading (ETL) of data created by other

SE toolsets external to the project's integrated or federated, shareable sets of data so the data and information in the external tool's database can be imported into the project's integrated or federated, shareable sets of data and used by the project? How well can the individual tools be integrated with each other, i.e., can one tool access and manipulate the data and information (single source of truth) created by a different tool? Do the tools in the toolset conform to a common data exchange standard (e.g. AP239, AP233 XML)? (See section 4.7.2). *When a tool vendor says their tool conforms with a standard, doesn't conform partially or fully?* Do the tools in the SE toolset allow data exchange between tools seamlessly, with minimal and straightforward data model mapping required on the part of the user? How well does the degree of integration of the tools in the toolset meet the needs of the organization?

Sharing of Data - External: Does the SE toolset allow the project to identify and securely share specific sets of data with external organizations, e.g., customer or vendors? Does the toolset include an industry standard import/export utility? (See section 4.7.2). *Is the tool partially or fully compliant with that standard?*

Storage Location: Does the SE toolset require the work products and their underlying data to be stored in the "cloud" provided by the SE tool vendor or stored in-house on the organization's server(s) or cloud-based storage? While in the could aids in collaboration, security and protection of sensitive, classified, or IP information is a concern when storing data in the cloud. Accessibility, access, and protection (from loss) are also key considerations for any storage location. Is the storage location consistent with enterprise data governance policy?

Scalability/Extendibility: Will the SE toolset be able to support the development and management of the volume of work products consistent with the size and complexity of the systems the organization develops? If the enterprise is procuring the SE toolset, will the SE toolset be able to support the number of projects within the enterprise given the size and complexity of the systems developed within the enterprise.

Archive/Backup/Long Term Availability: Does the SE toolset provide the capability to archive and backup all the data and information consistent with enterprise data governance policy? Do the formats used provide long term availability as storage and retrieval technologies evolve or a specific tool changes or the user changes their toolset? (Need to *avoid using a backup/archive format that uses a proprietary format that is no longer accessible if the tool vendor goes out of business.)*

Management and Reporting*: (What features are needed to help more effectively manage the project?)*

Attributes: Does the SE toolset allow the user to define and manage attributes for work products. For example, for requirements, does the SE tool allow the user to define attributes needed to help manage their requirements? (*A discussion on the use of attributes to manage a project and list of attributes is included in* INCOSE-TP-2010-006-02, *INCOSE Guide to Writing Requirements.)*

Measures: Does the SE toolset allow the enterprise and project to define specific measures that will allow managers and systems engineers to monitor and assess progress, identify issues, and ensure the system being developed will meet stakeholder needs and expectations? Several key measures are commonly used that reflect overall customer/user satisfaction (e.g., performance, safety, reliability, availability, maintainability, and workload requirements): measures of suitability (MOSs), measures of performance (MOPs), and measures of effectiveness (MOEs), and leading indicators (LIs). *(See section 4.6)*

Reports: Does the SE toolset include a robust, well documented report feature that allows users to create unique reports (using the attributes and measures defined previously) as well as customize standard reports provided with the tool? Does the SE toolset allow reports to be exported in multiple formats (MS Word, Pages, RTF, spreadsheet, presentation, graphical, etc.)? At the beginning of the quest for an SE toolset, one of the first things that needs to be done is develop the process the tool needs to support. Include in this process description the specific reports needed. That will drive the schema of the data, meta-data, measures, and attributes to be included in the project integrated or federated database.

Metrics/dashboards: Closely related to a report feature, does the SE toolset provide the capability to do "data mining" and analytics of the measures and information in the attributes to enable the display of and report historical and trend data? It is often not enough to just know what percent of the requirements for the system have been successfully verified. Often it would be useful to see if the trend to completion of system verification activities is on the right pace, is slowing down, or speeding up. If slowing down, the project may not be able to complete all the system verification and system validation activities in time for the customer acceptance review. Management needs to know this!

Notifications: Can the SE toolset send notifications via email or texting concerning changes to work products; design work products, system V&V work products? Can the SE toolset send notifications from one user to another user (or group of users) concerning actions, comments, and questions? Can the SE toolset send notifications to the appropriate users when a specific measure is predicted to or has exceeded a pre-specified threshold?

Project Management work products: Does the SE toolset allow various PM work products to be managed within the SE toolset or at least linked to artifacts within the SE toolset? This includes budget, schedule, and risk management work products. Can these work products and underlying data and information be linked to parts of the product breakdown structure and other SE work products and their underlying data?

Life cycle Support: Does the SE toolset support system development across all system life cycle process activities: scope definition, requirement definition and management, gate reviews, design, manufacturing, coding, system verification, system validation, and sustaining engineering? For example, system verification & validation: Does the SE toolset allow the linkage of requirements to their system verification and system validation requirements, procedures, results of the procedures, and close out documentation of the system verification and validation activities?

Workflow: Does the SE toolset provide the ability to define and support the organization's SE process workflow within the tool (e.g., for requirements does the SE toolset allow the project to track the requirement's status: draft, review, approve, baseline; design, test, code/manufacture, system verification, and system validation)? Can the SE toolset allow the creation, management, and execution of SE processes, procedures, and work instructions within the toolset?

Configuration Management: Does the SE toolset provide robust configuration management of all system life cycle work products and underlying data and information including change, version, and baseline control? Does the SE toolset allow the user to access the change history of any work product? If work products are developed and maintained within a database, does the SE toolset allow configuration control of the database (versus the various reports/visualizations representing the data and information in the database)?

General:

Price: Is the SE toolset affordable in relation to the size of the project and number of requirements and requirement sets, modeling activities and resulting artifacts, design and verification and validation work products, number of concurrent users? Concerning affordability, is there a single upfront application fee, individual license fee (if a license fee, one-time or yearly)? Are the licenses fixed or floating? Does the price include initial setup, installation, configuration or customization or is that extra? Is ongoing technical support included or extra? Is training included or extra? Would it be more cost effective to spend more on a single tool that has most of the above features or multiple tools to give the organization all the features needed for their chosen SCL?

Cost of infrastructure to support the use of the SE toolset: What are the IT requirements to host and deploy the toolset? What specialty skills beyond engineering are required to operate, extend, and maintain the SE toolset?

Vendor/product maturity: How long has the SE tool been on the market? How long has the vendor been in business?

User feedback and satisfaction: In today's social media driven world, access is provided to actual user comments concerning the tool, the tool vendor, ease of use, reliability, tech support, etc. Don't get overwhelmed by the hype and sales pitch from the vendor. See what the actual users have to say about the SE tools being considered for inclusion in the organization's SE toolset.

APPENDIX D: SYSTEMS ENGINEERING ISSUES QUESTIONNAIRE

Instructions: The statements below are worded such that they represent the current state of an organization. Check the column that most closely reflects the perspective of the organization's current state: True, Mostly True, Neutral, Mostly False, False. If most of the responses are either "True" or "Mostly True", that is a good indication the organization needs to adopt Systems Engineering (SE) or mature their current SE processes, moving toward SE with a data/ information-centric perspective.

	Issue/Challenge	T	MT	N	MF	F
1.	We develop very complex systems with a large number of work products and sets of data. Many of our work products are managed as printed, standalone documents. We are having problems managing this complex system with our current approach to documentation of work products.					
2.	Our current organization is divided into "silos" for each system life cycle stage. This makes it difficult share data across organizational elements to holistically integrate with coherence and consistency work products and their underlying data and information across disciplines, organizations, and system life cycle stages.					
3.	Our current system life cycle capabilities do not allow us to capture, integrate, manage, and access increasingly large sets of system engineering and program/project management data and information and their associated interrelationships.					
4.	Our current organization and SE process makes it very hard to identify and manage dependencies across not only the system architecture but dependencies across disciplines and system life cycle work products and entities.					
5.	Our current organization and SE process makes it difficult to identify, define, and manage interactions (interfaces) between parts of our complex system architecture and between the system and the macro-system of which it is a part. Because of this we often have costly integration problems resulting in costly and time-consuming rework.					
6.	Our current organization and SE process makes it difficult to track progress, identify at-risk activities, and take actions before these risks become problems that could impact cost, schedule, or the ability to deliver a product that meets stakeholder needs in the operational environment. The result is we spend a lot of time being firefighters to put out fires rather than being able to prevent the fires from starting in the first place.					
7.	Our current decision-making culture is based on a "gut" feel mainly because of a lack of easy and timely access to data and information needed to make informed decisions. Once a decision is made, frequently the decision is not documented nor is the supporting information as to why the decision was made documented.					
8.	Currently, PM activities and resulting work products and underlying data and information are segregated from the SE activities and resulting work products and underlying data and information, making it difficult to manage cost, schedule, and risk.					
9.	Consistency of work products and their underlying data is problematic. Work products and their underlying data and information are spread across multiple databases and servers. These sets of data are not compatible (schemas are not consistent) making it difficult to share data. The result is that there is no single source of design data and information that can be expressed authoritatively in order to be referred to by others for decisions, derivations, or formation of other work products.					
10.	We have poor visibility into the principle characteristics of our whole system preventing us from creating multiple views from integrated,					

	shareable sets of data that succinctly address specific stakeholder concerns and interests.					
11.	Our current processes result in poor congruence and configuration management between documentation and reality. Many of our SE work products must be generated manually to obtain differing views of the system under development. The labor and associated costs are expensive to generate, configuration manage, and keep these work products and their underlying data and information up to date. Frequently many work products and their underlying data and information are out-of-date and do not match the best available, current data and information.					
12.	Because our data and information are distributed across many databases and servers there is no “single source of truth”. Because of this we are at the mercy of what someone says or thinks, what they “remember”, or what perspective they have concerning what is being done or built or a decision that was made. Truth is in the eye of the beholder.					
13.	With our current SE toolset, it is very difficult to navigate, trace, or interrogate system engineering data and information across all system life cycle stage activities. Managers and engineers do not have ready access to correct and consistent information on an as-needed basis. Meaningful reports take a lot of labor to produce manually, having to search individual databases and integrate the data for the desired reports.					
14.	Currently we are not able to reuse SE and PM work products. The result is considerable time and expense because each brown field project must start from scratch resulting in wasted funds and increased time to manage our product line.					
15.	Because of stove piping and a lack of traceability, we are unable to adequately manage stakeholder needs, requirement definition, design, build/code, and system verification and validation activities in an integrated, consistent manner. Our current processes make it difficult to monitor the status of verification and validation activities in order to show compliance with stakeholder needs and drivers and constraints (e.g., regulations, customer requirements).					
16.	Because of our current SE processes, the costs associated with erroneous design and resulting rework is very high. Lack of an integrated or federated, shareable sets of data makes analysis of the SE work products and underlying data and information difficult to identify a flaw or inconsistency as soon as it is created, preventing us to take corrective action before downstream work is done, making that work invalid, and increasing costs and time to correct because the upstream mistake was not identified and corrected immediately. One result of this is huge expenses associated with recalls, returns, warranty work, and negative comments on social media.					
17.	Our current, siloed organization and lack of an integrated or federated, shareable sets of data directly impacts the identification, management, interoperability, and integration of work products and underlying data and information across business or organizational elements. This makes it difficult to support program budget and schedule goals.					
18.	Our current distributed organization of data and information makes it difficult to metatag data, information, and work products. Because of this, we cannot currently tie these things directly to our WBS, budget, schedule, and risk management systems.					
19.	Data and information needed by programs and projects (e.g., for milestones, reviews, mission operations, risk mitigation, and anomalies or investigations, decisions, and outcomes) are not identified and managed. Because of this, there is currently no way to provide traceability of the data used in decision-making.					
20.	It is difficult to establish, maintain, and track key measures needed to monitor trends, assess progress, and identify issues. Because of this, tend monitoring of leading indicators is not possible resulting major issues popping up with little or no warning, actual progress is hard to determine. This makes it difficult to ensure the system being developed will meet stakeholder needs and expectations.					

APPENDIX E: COMMENT FORM

Reviewed Document:	
Name of submitter (first name & last name):	
Date Submitted:	
Contact information (email address):	
Type of submission (individual/group):	
Group name and number of contributors (if applicable)	

Comment sequence number	Commenter name	Category (TH, TL, E, G)	Section number (e.g., 2.1.1, <u>no alpha</u>)	Specific reference (e.g., paragraph, line, Figure no., Table no.)	Issue, comment and rationale (rationale must make comment <u>clearly evident</u> and <u>supportable</u>)	Proposed Changed/New Text -- MANDATORY ENTRY -- (must be <u>substantial</u> to increase the odds of acceptance)	Importance Rating (R = Required, I = Important, T = Think About for future version)

Submit comments to Requirement Working Group chair. Current WG chair will be listed at:

http://www.incose.org/techcomm.html

If this fails, comments may be sent to info@incose.org (the INCOSE main office), which can relay to the appropriate WG, if so requested in the comment cover page.

INCOSE Publications Office

7670 Opportunity Road, Suite 220

San Diego, CA 92111-2222 USA