

**SYSTEMS ENGINEERING  
HANDBOOK**



# **SYSTEMS ENGINEERING HANDBOOK**

---

## **A GUIDE FOR SYSTEM LIFE CYCLE PROCESSES AND ACTIVITIES**

**FOURTH EDITION**

**INCOSE-TP-2003-002-04  
2015**

**Prepared by:**

**International Council on Systems Engineering (INCOSE)  
7670 Opportunity Rd, Suite 220  
San Diego, CA, USA 92111-2222**

**Compiled and Edited by:**

**DAVID D. WALDEN, ESEP  
GARRY J. ROEDLER, ESEP  
KEVIN J. FORSBERG, ESEP  
R. DOUGLAS HAMELIN  
THOMAS M. SHORTELL, CSEP**

**WILEY**

Copyright © 2015 by John Wiley & Sons, Inc. All rights reserved

Published by John Wiley & Sons, Inc., Hoboken, New Jersey  
Published simultaneously in Canada

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 750-4470, or on the web at [www.copyright.com](http://www.copyright.com). Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

**Limit of Liability/Disclaimer of Warranty:** While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services or for technical support, please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic formats. For more information about Wiley products, visit our web site at [www.wiley.com](http://www.wiley.com).

***Library of Congress Cataloging-in-Publication Data:***

Systems engineering handbook : a guide for system life cycle processes and activities / prepared by International Council on Systems Engineering (INCOSE) ; compiled and edited by, David D. Walden, ESEP, Garry J. Roedler, ESEP, Kevin J. Forsberg, ESEP, R. Douglas Hamelin, Thomas M. Shortell, CSEP. – 4th edition.

pages cm

Includes bibliographical references and index.

ISBN 978-1-118-99940-0 (cloth)

I. Systems engineering—Handbooks, manuals, etc. 2. Product life cycle—Handbooks, manuals, etc. I. Walden, David D., editor. II. Roedler, Garry J., editor. III. Forsberg, Kevin, editor. IV. Hamelin, R. Douglas, editor. V. Shortell, Thomas M., editor. VI. International Council on Systems Engineering.

TA168.S8724 2015

620.001'1—dc23

2014039630

ISBN: 9781118999400

Set in 10/12pt Times LT Std by SPi Publisher Services, pondicherry, India

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

1 2015

# CONTENTS

<b>INCOSE Notices</b>	<b>vii</b>	<b>3 Generic Life Cycle Stages</b>	<b>25</b>
<b>History of Changes</b>	<b>viii</b>	3.1 Introduction	25
<b>Preface</b>	<b>ix</b>	3.2 Life Cycle Characteristics	26
<b>List of Figures</b>	<b>x</b>	3.3 Life Cycle Stages	27
<b>List of Tables</b>	<b>xii</b>	3.4 Life Cycle Approaches	32
<b>1 Systems Engineering Handbook Scope</b>	<b>1</b>	3.5 What Is Best for Your Organization, Project, or Team?	36
1.1 Purpose	1	3.6 Introduction to Case Studies	39
1.2 Application	1	<b>4 Technical Processes</b>	<b>47</b>
1.3 Contents	1	4.1 Business or Mission Analysis Process	49
1.4 Format	3	4.2 Stakeholder Needs and Requirements Definition Process	52
1.5 Definitions of Frequently Used Terms	4	4.3 System Requirements Definition Process	57
<b>2 Systems Engineering Overview</b>	<b>5</b>	4.4 Architecture Definition Process	64
2.1 Introduction	5	4.5 Design Definition Process	70
2.2 Definitions and Concepts of a System	5	4.6 System Analysis Process	74
2.3 The Hierarchy <i>within</i> a System	7	4.7 Implementation Process	77
2.4 Definition of Systems of Systems	8	4.8 Integration Process	79
2.5 Enabling Systems	10	4.9 Verification Process	83
2.6 Definition of Systems Engineering	11	4.10 Transition Process	88
2.7 Origins and Evolution of Systems Engineering	12	4.11 Validation Process	89
2.8 Use and Value of Systems Engineering	13	4.12 Operation Process	95
2.9 Systems Science and Systems Thinking	17	4.13 Maintenance Process	97
2.10 Systems Engineering Leadership	21	4.14 Disposal Process	101
2.11 Systems Engineering Professional Development	22		

<b>5</b>	<b>Technical Management Processes</b>	<b>104</b>	9.4	Object-Oriented Systems Engineering Method	193
5.1	Project Planning Process	104	9.5	Prototyping	197
5.2	Project Assessment and Control Process	108	9.6	Interface Management	197
5.3	Decision Management Process	110	9.7	Integrated Product and Process Development	199
5.4	Risk Management Process	114	9.8	Lean Systems Engineering	203
5.5	Configuration Management Process	122	9.9	Agile Systems Engineering	207
5.6	Information Management Process	128			
5.7	Measurement Process	130			
5.8	Quality Assurance Process	135			
<b>6</b>	<b>Agreement Processes</b>	<b>139</b>	<b>10</b>	<b>Specialty Engineering Activities</b>	<b>211</b>
6.1	Acquisition Process	140	10.1	Affordability/Cost-Effectiveness/ Life Cycle Cost Analysis	211
6.2	Supply Process	142	10.2	Electromagnetic Compatibility	219
<b>7</b>	<b>Organizational Project-Enabling Processes</b>	<b>145</b>	10.3	Environmental Engineering/Impact Analysis	220
7.1	Life Cycle Model Management Process	145	10.4	Interoperability Analysis	221
7.2	Infrastructure Management Process	149	10.5	Logistics Engineering	222
7.3	Portfolio Management Process	151	10.6	Manufacturing and Producibility Analysis	225
7.4	Human Resource Management Process	154	10.7	Mass Properties Engineering	225
7.5	Quality Management Process	156	10.8	Reliability, Availability, and Maintainability	226
7.6	Knowledge Management Process	158	10.9	Resilience Engineering	229
<b>8</b>	<b>Tailoring process and Application of Systems Engineering</b>	<b>162</b>	10.10	System Safety Engineering	231
8.1	Tailoring Process	163	10.11	System Security Engineering	234
8.2	Tailoring for Specific Product Sector or Domain Application	165	10.12	Training Needs Analysis	237
8.3	Application of Systems Engineering for Product Line Management	170	10.13	Usability Analysis/Human Systems Integration	237
8.4	Application of Systems Engineering for Services	171	10.14	Value Engineering	241
8.5	Application of Systems Engineering for Enterprises	175			
8.6	Application of Systems Engineering for Very Small and Micro Enterprises	179			
<b>9</b>	<b>Cross-Cutting Systems Engineering Methods</b>	<b>180</b>		<b>Appendix A: References</b>	<b>246</b>
9.1	Modeling and Simulation	180		<b>Appendix B: Acronyms</b>	<b>257</b>
9.2	Model-Based Systems Engineering	189		<b>Appendix C: Terms and Definitions</b>	<b>261</b>
9.3	Functions-Based Systems Engineering Method	190		<b>Appendix D: N<sup>2</sup> Diagram of Systems Engineering Processes</b>	<b>267</b>
				<b>Appendix E: Input/Output Descriptions</b>	<b>269</b>
				<b>Appendix F: Acknowledgements</b>	<b>284</b>
				<b>Appendix G: Comment Form</b>	<b>286</b>
				<b>Index</b>	<b>287</b>

# INCOSE NOTICES

This International Council on Systems Engineering (INCOSE) Technical Product was prepared by the INCOSE Knowledge Management working group. It is approved by INCOSE Technical Operations Leadership for release as an INCOSE Technical Product.

Copyright ©2015 by INCOSE, subject to the following restrictions:

*Author Use:* Authors have full rights to use their contributions unfettered, with credit to the INCOSE technical source, except as noted in the following text. Abstraction is permitted with credit to the source.

*INCOSE Use:* Permission to reproduce and use this document or parts thereof by members of INCOSE and to prepare derivative works from this document for INCOSE use is granted, with attribution to INCOSE and the original author(s) where practical, provided this copyright notice is included with all reproductions and derivative works. Content from ISO/IEC/IEEE 15288 and ISO/IEC TR 24748-1 is used by permission, and is not to be reproduced other than as part of this total document.

*External Use:* This document may not be shared or distributed to any non-INCOSE third party. Requests for permission to reproduce this document in whole or in part, or to prepare derivative works of this document for external and/or commercial use, will be denied unless covered by other formal agreements with INCOSE. Copying, scanning, retyping, or any other form of reproduction or use of the content of whole pages or source documents are prohibited, except as approved by

the INCOSE Administrative Office, 7670 Opportunity Road, Suite 220, San Diego, CA 92111-2222, USA.

*Electronic Version Use:* All electronic versions (e.g., eBook, PDF) of this document are to be used for personal professional use only and are not to be placed on non-INCOSE sponsored servers for general use. Any additional use of these materials must have written approval from the INCOSE Administrative Office.

*INCOSE Corporate Advisory Board Use:* INCOSE has granted permission to member organizations of the INCOSE Corporate Advisory Board (CAB) to post an electronic (PDF) version of this document on their internal servers for use by their employees, subject to the external use restrictions noted earlier. Additional use of this document by CAB organizations for internal purposes is permitted per INCOSE policy CAB-100.

*Notice:* Hardcopy versions of this document may not be the most current. The current approved version is always the electronic version posted on the Product Area of the INCOSE website.

*General Citation Guidelines:* References to this handbook should be formatted as follows, with appropriate adjustments for formally recognized styles:

INCOSE (2015). *Systems Engineering Handbook: A Guide for System Life Cycle Process and Activities* (4th ed.). D. D. Walden, G. J. Roedler, K. J. Forsberg, R. D. Hamelin, and, T. M. Shortell (Eds.). San Diego, CA: International Council on Systems Engineering. Published by John Wiley & Sons, Inc.

# HISTORY OF CHANGES

Revision	Revision date	Change description and rationale
Original	Jun 1994	Draft <i>Systems Engineering Handbook</i> (SEH) created by INCOSE members from several defense/aerospace companies—including Lockheed, TRW, Northrop Grumman, Ford Aerospace, and the Center for Systems Management—for INCOSE review
1.0	Jan 1998	Initial SEH release approved to update and broaden coverage of SE process. Included broad participation of INCOSE members as authors. Based on Interim Standards EIA 632 and IEEE 1220
2.0	Jul 2000	Expanded coverage on several topics, such as functional analysis. This version was the basis for the development of the Certified Systems Engineering Professional (CSEP) exam
2.0A	Jun 2004	Reduced page count of SEH v2 by 25% and reduced the US DoD-centric material wherever possible. This version was the basis for the first publically offered CSEP exam
3.0	Jun 2006	Significant revision based on ISO/IEC 15288:2002. The intent was to create a country- and domain-neutral handbook. Significantly reduced the page count, with elaboration to be provided in appendices posted online in the INCOSE Product Asset Library (IPAL)
3.1	Aug 2007	Added detail that was not included in SEH v3, mainly in new appendices. This version was the basis for the updated CSEP exam
3.2	Jan 2010	Updated version based on ISO/IEC/IEEE 15288:2008. Significant restructuring of the handbook to consolidate related topics
3.2.1	Jan 2011	Clarified definition material, architectural frameworks, concept of operations references, risk references, and editorial corrections based on ISO/IEC review
3.2.2	Oct 2011	Correction of errata introduced by revision 3.2.1
4.0	Jan 2015	Significant revision based on ISO/IEC/IEEE 15288:2015, inputs from the relevant INCOSE working groups (WGs), and to be consistent with the Guide to the Systems Engineering Body of Knowledge (SEBoK)



# PREFACE

The objective of the International Council on Systems Engineering (INCOSE) *Systems Engineering Handbook* (SEH) is to describe key process activities performed by systems engineers. The intended audience is the systems engineering (SE) professional. When the term *systems engineer* is used in this handbook, it includes the new systems engineer, a product engineer or an engineer in another discipline who needs to perform SE, or an experienced systems engineer who needs a convenient reference.

The descriptions in this handbook show what each SE process activity entails, in the context of designing for required performance and life cycle considerations. On some projects, a given activity may be performed very informally; on other projects, it may be performed very formally, with interim products under formal configuration control. This document is not intended to advocate any level of formality as necessary or appropriate in all situations. The appropriate degree of formality in the execution of any SE process activity is determined by the following:

1. The need for communication of what is being done (across members of a project team, across organizations, or over time to support future activities)
2. The level of uncertainty
3. The degree of complexity
4. The consequences to human welfare

On smaller projects, where the span of required communications is small (few people and short project life cycle) and the cost of rework is low, SE activities can be

conducted very informally and thus at low cost. On larger projects, where the span of required communications is large (many teams that may span multiple geographic locations and organizations and long project life cycle) and the cost of failure or rework is high, increased formality can significantly help in achieving project opportunities and in mitigating project risk.

In a project environment, work necessary to accomplish project objectives is considered “in scope”; all other work is considered “out of scope.” On every project, “thinking” is always “in scope.” Thoughtful tailoring and intelligent application of the SE processes described in this handbook are essential to achieve the proper balance between the risk of missing project technical and business objectives on the one hand and process paralysis on the other hand. Chapter 8 provides tailoring guidelines to help achieve that balance.

## APPROVED FOR SEH V4:

Kevin Forsberg, ESEP, Chair, INCOSE Knowledge Management Working Group

Garry Roedler, ESEP, Co-Chair, INCOSE Knowledge Management Working Group

William Miller, INCOSE Technical Director (2013–2014)

Paul Schreinemakers, INCOSE Technical Director (2015–2016)

Quoc Do, INCOSE Associate Director for Technical Review

Kenneth Zemrowski, ESEP, INCOSE Assistant Director for Technical Information

# LIST OF FIGURES

- 1.1. System life cycle processes per ISO/IEC/IEEE 15288
- 1.2. Sample of IPO diagram for SE processes
- 2.1. Hierarchy within a system
- 2.2. Example of the systems and systems of systems within a transport system of systems
- 2.3. System of interest, its operational environment, and its enabling systems
- 2.4. Committed life cycle cost against time
- 2.5. Technology acceleration over the past 140 years
- 2.6. Project performance versus SE capability
- 2.7. Cost and schedule overruns correlated with SE effort
- 2.8. Systems science in context
- 2.9. SE optimization system
- 2.10. Professional development system
- 3.1. Generic business life cycle
- 3.2. Life cycle model with some of the possible progressions
- 3.3. Comparisons of life cycle models
- 3.4. Importance of the concept stage
- 3.5. Iteration and recursion
- 3.6. Vee model
- 3.7. Left side of the Vee model
- 3.8. Right side of the Vee model
- 3.9. IID and evolutionary development
- 3.10. The incremental commitment spiral model (ICSM)
- 3.11. Phased view of the generic incremental commitment spiral model process
- 4.1. Transformation of needs into requirements
- 4.2. IPO diagram for business or mission analysis process
- 4.3. Key SE interactions
- 4.4. IPO diagram for stakeholder needs and requirements definition process
- 4.5. IPO diagram for the system requirements definition process
- 4.6. IPO diagram for the architecture definition process
- 4.7. Interface representation
- 4.8. (a) Initial arrangement of aggregates; (b) final arrangement after reorganization
- 4.9. IPO diagram for the design definition process
- 4.10. IPO diagram for the system analysis process
- 4.11. IPO diagram for the implementation process
- 4.12. IPO diagram for the integration process
- 4.13. IPO diagram for the verification process
- 4.14. Definition and usage of a verification action
- 4.15. Verification level per level
- 4.16. IPO diagram for the transition process
- 4.17. IPO diagram for the validation process
- 4.18. Definition and usage of a validation action
- 4.19. Validation level per level
- 4.20. IPO diagram for the operation process
- 4.21. IPO diagram for the maintenance process
- 4.22. IPO diagram for the disposal process
- 5.1. IPO diagram for the project planning process
- 5.2. IPO diagram for the project assessment and control process
- 5.3. IPO diagram for the decision management process

- 5.4. IPO diagram for the risk management process
- 5.5. Level of risk depends on both likelihood and consequences
- 5.6. Typical relationship among the risk categories
- 5.7. Intelligent management of risks and opportunities
- 5.8. IPO diagram for the configuration management process
- 5.9. Requirements changes are inevitable
- 5.10. IPO diagram for the information management process
- 5.11. IPO diagram for the measurement process
- 5.12. Measurement as a feedback control system
- 5.13. Relationship of technical measures
- 5.14. TPM monitoring
- 5.15. IPO diagram for the quality assurance process
- 6.1. IPO diagram for the acquisition process
- 6.2. IPO diagram for the supply process
- 7.1. IPO diagram for the life cycle model management process
- 7.2. Standard SE process flow
- 7.3. IPO diagram for the infrastructure management process
- 7.4. IPO diagram for the portfolio management
- 7.5. IPO diagram for the human resource management process
- 7.6. IPO diagram for the quality management process
- 7.7. IPO diagram for the knowledge management process
- 8.1. Tailoring requires balance between risk and process
- 8.2. IPO diagram for the tailoring process
- 8.3. Product line viewpoints
- 8.4. Capitalization and reuse in a product line
- 8.5. Product line return on investment
- 8.6. Service system conceptual framework
- 8.7. Organizations manage resources to create enterprise value
- 8.8. Individual competence leads to organizational, system and operational capability
- 8.9. Enterprise SE process areas in the context of the entire enterprise
- 9.1. Sample model taxonomy
- 9.2. SysML diagram types
- 9.3. Functional analysis/allocation process
- 9.4. Alternative functional decomposition evaluation and definition
- 9.5. OOSEM builds on established SE foundations
- 9.6. OOSEM activities in context of the system development process
- 9.7. OOSEM activities and modeling artifacts
- 9.8. Sample FFBD and N<sup>2</sup> diagram
- 9.9. Examples of complementary integration activities of IPDTs
- 9.10. Lean development principles
- 10.1. Contextual nature of the affordability trade space
- 10.2. Systems operational effectiveness
- 10.3. Cost versus performance
- 10.4. Affordability cost analysis framework
- 10.5. Life cycle cost elements (not to scale)
- 10.6. Process for achieving EMC
- 10.7. Supportability analysis
- 10.8. Reliability program plan development
- 10.9. Resilience event model
- 10.10. Sample Function Analysis System Technique (FAST) diagram

# LIST OF TABLES

- 2.1. Important dates in the origins of SE as a discipline
- 2.2. Important dates in the origin of SE standards
- 2.3. Current significant SE standards and guides
- 2.4. SE return on investment
- 3.1. Generic life cycle stages, their purposes, and decision gate options
- 4.1. Examples of systems elements and physical interfaces
- 5.1. Partial list of decision situations (opportunities) throughout the life cycle
- 8.1. Standardization-related associations and automotive standards
- 8.2. Attributes of system entities
- 9.1. Types of IPDTs and their focus and responsibilities
- 9.2. Pitfalls of using IPDT

---

# 1

---

## SYSTEMS ENGINEERING HANDBOOK SCOPE

### 1.1 PURPOSE

This handbook defines the discipline and practice of systems engineering (SE) for students and practicing professionals alike and provides an authoritative reference to understand the SE discipline in terms of content and practice.

### 1.2 APPLICATION

This handbook is consistent with ISO/IEC/IEEE 15288:2015, *Systems and software engineering—System life cycle processes* (hereafter referred to as ISO/IEC/IEEE 15288), to ensure its usefulness across a wide range of application domains—man-made systems and products, as well as business and services.

ISO/IEC/IEEE 15288 is an international standard that provides generic top-level process descriptions and requirements, whereas this handbook further elaborates on the practices and activities necessary to execute the processes. Before applying this handbook in a given organization or project, it is recommended that the tailoring guidelines in Chapter 8 be used to remove conflicts with existing policies, procedures, and standards

already in use within an organization. Processes and activities in this handbook do not supersede any international, national, or local laws or regulations.

This handbook is also consistent with the *Guide to the Systems Engineering Body of Knowledge* (SEBoK, 2014) (hereafter referred to as the SEBoK) to the extent practicable. In many places, this handbook points readers to the SEBoK for more detailed coverage of the related topics, including a current and vetted set of references.

For organizations that do not follow the principles of ISO/IEC/IEEE 15288 or the SEBoK to specify their life cycle processes (including much of commercial industry), this handbook can serve as a reference to practices and methods that have proven beneficial to the SE community at large and that can add significant value in new domains, if appropriately selected and applied. Section 8.2 provides top-level guidance on the application of SE in selected product sectors and domains.

### 1.3 CONTENTS

This chapter defines the purpose and scope of this handbook. Chapter 2 provides an overview of the goals and value of using SE throughout the system life cycle.

Chapter 3 describes an informative life cycle model with six stages: concept, development, production, utilization, support, and retirement.

ISO/IEC/IEEE 15288 identifies four process groups to support SE. Each of these process groups is the subject of an individual chapter. A graphical overview of these processes is given in Figure 1.1:

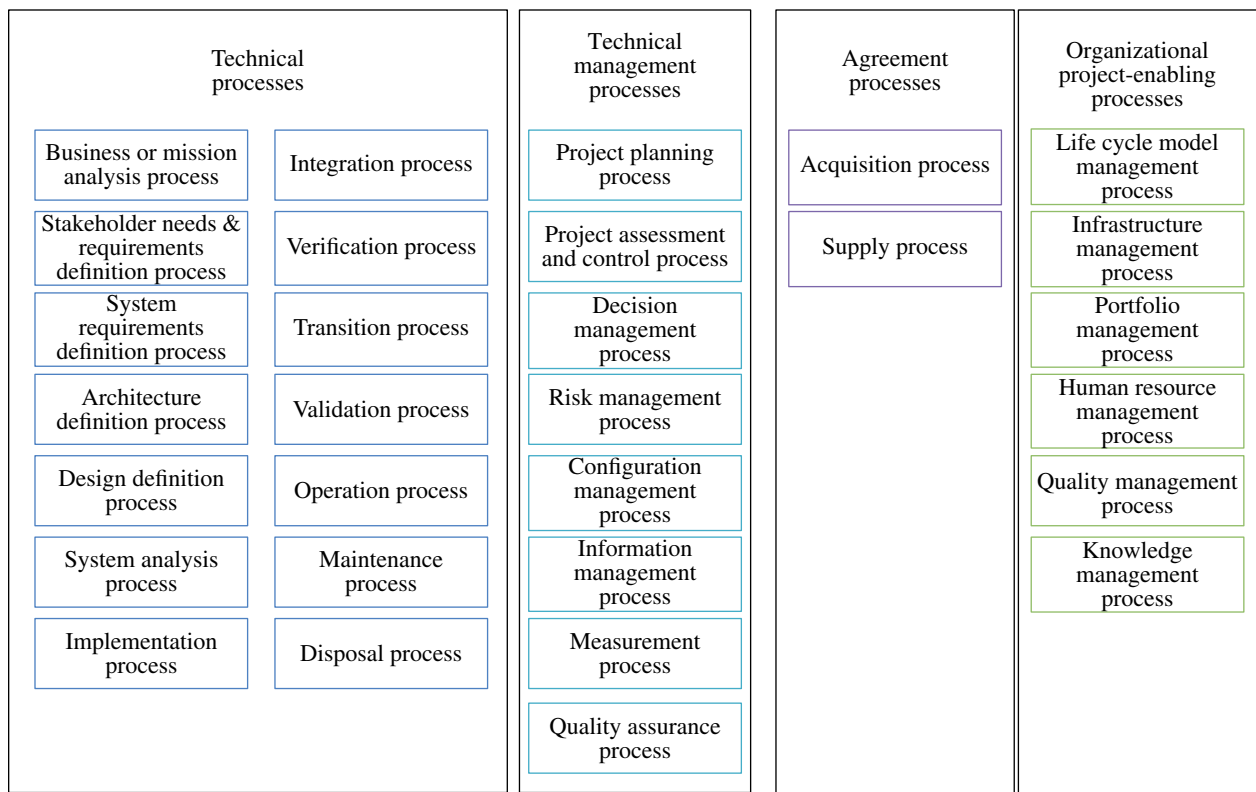
- *Technical processes* (Chapter 4) include business or mission analysis, stakeholder needs and requirements definition, system requirements definition, architecture definition, design definition, system analysis, implementation, integration, verification, transition, validation, operation, maintenance, and disposal.
- *Technical management processes* (Chapter 5) include project planning, project assessment and control, decision management, risk management,

configuration management, information management, measurement, and quality assurance.

- *Agreement processes* (Chapter 6) include acquisition and supply.
- *Organizational project-enabling processes* (Chapter 7) include life cycle model management, infrastructure management, portfolio management, human resource management, quality management, and knowledge management.

This handbook provides additional chapters beyond the process groups listed in Figure 1.1:

- *Tailoring processes and application of systems engineering* (Chapter 8) include information on how to adapt and scale the SE processes and how to apply those processes in various applications. Not every process will apply universally. Careful selection



**FIGURE 1.1** 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.

from the material is recommended. Reliance on process over progress will not deliver a system.

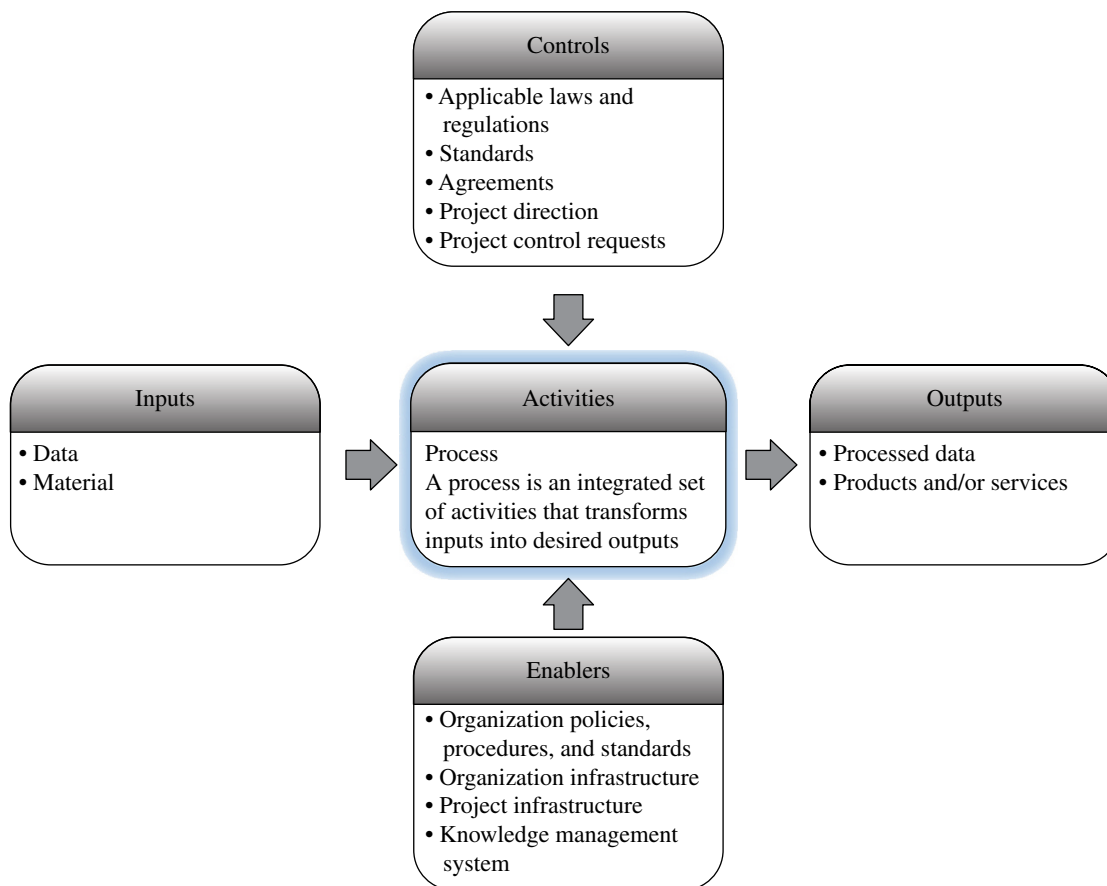
- *Crosscutting systems engineering methods* (Chapter 9) provide insights into methods that can apply across all processes, reflecting various aspects of the iterative and recursive nature of SE.
- *Specialty engineering activities* (Chapter 10) include practical information so systems engineers can understand and appreciate the importance of various specialty engineering topics.

Appendix A contains a list of references used in this handbook. Appendices B and C provide a list of acronyms and a glossary of SE terms and definitions, respectively. Appendix D provides an N<sup>2</sup> diagram of the SE processes showing where dependencies exist in the form

of shared inputs or outputs. Appendix E provides a master list of all inputs/outputs identified for each SE process. Appendix F acknowledges the various contributors to this handbook. Errors, omissions, and other suggestions for this handbook can be submitted to the INCOSE using the comment form contained in Appendix G.

### 1.4 FORMAT

A common format has been applied in Chapters 4 through 7 to describe the system life cycle processes found in ISO/IEC/IEEE 15288. Each process is illustrated by an input–process–output (IPO) diagram showing key inputs, process activities, and resulting outputs. A sample is shown in Figure 1.2. Note that the IPO



**FIGURE 1.2** Sample of IPO diagram for SE processes. INCOSE SEH original figure created by Shortell and Walden. Usage per the INCOSE Notices page. All other rights reserved.

diagrams throughout this handbook represent “a” way that the SE processes can be performed, but not necessarily “the” way that they must be performed. The issue is that SE processes produce “results” that are often captured in “documents” rather than producing “documents” simply because they are identified as outputs. To understand a given process, readers are encouraged to study the complete information provided in the combination of diagrams and text and not rely solely on the diagrams.

The following heading structure provides consistency in the discussion of these processes:

- Process overview
- Purpose
- Description
- Inputs/outputs
- Process activities
- Process elaboration

To ensure consistency with ISO/IEC/IEEE 15288, the purpose statements from the standard are included verbatim for each process described herein. Inputs and outputs are listed by name within the respective IPO diagrams with which they are associated. A complete list of all inputs and outputs with their respective descriptions appears in Appendix E.

The titles of the process activities listed in each section are also consistent with ISO/IEC/IEEE 15288. In some cases, additional items have been included to provide summary-level information regarding industry best practices and evolutions in the application of SE processes.

The controls and enablers shown in Figure 1.2 govern all processes described herein and, as such, are not repeated in the IPO diagrams or in the list of inputs associated with each process description. Typically, IPO diagrams do not include controls and enablers, but since they are not repeated in the IPO diagrams throughout the rest of the handbook, we have chosen to label them IPO diagrams. Descriptions of each control and enabler are provided in Appendix E.

## 1.5 DEFINITIONS OF FREQUENTLY USED TERMS

One of the systems engineer’s first and most important responsibilities on a project is to establish nomenclature and terminology that support clear, unambiguous communication and definition of the system and its elements, functions, operations, and associated processes. Further, to promote the advancement of the field of SE throughout the world, it is essential that common definitions and understandings be established regarding general methods and terminology that in turn support common processes. As more systems engineers accept and use common terminology, SE will experience improvements in communications, understanding, and, ultimately, productivity.

The glossary of terms used throughout this book (see Appendix C) is based on the definitions found in ISO/IEC/IEEE 15288; ISO/IEC/IEEE 24765, *Systems and Software Engineering—Vocabulary* (2010); and SE VOCAB (2013).



---

# 2

---

## SYSTEMS ENGINEERING OVERVIEW

### 2.1 INTRODUCTION

This chapter offers a brief overview of the systems engineering (SE) discipline, beginning with a few key definitions, an abbreviated survey of the origins of the discipline, and discussions on the value of applying SE. Other concepts, such as systems science, systems thinking, SE leadership, SE ethics, and professional development, are also introduced.

### 2.2 DEFINITIONS AND CONCEPTS OF A SYSTEM

While the concepts of a *system* can generally be traced back to early Western philosophy and later to science, the concept most familiar to systems engineers is often traced to Ludwig von Bertalanffy (1950, 1968) in which a system is regarded as a “whole” consisting of interacting “parts.” The ISO/IEC/IEEE definitions provided in this handbook draw from this concept.

#### 2.2.1 General System Concepts

The systems considered in ISO/IEC/IEEE 15288 and in this handbook

[5.2.1] ... are man-made, created and utilized to provide products or services in defined environments for the benefit of users and other stakeholders.

The definitions cited here and in Appendix C refer to systems in the real world. A system concept should be regarded as a shared “mental representation” of the actual system. The systems engineer must continually distinguish between systems in the real world and system representations. The INCOSE and ISO/IEC/IEEE definitions draw from this view of a system:

... an integrated set of elements, subsystems, or assemblies that accomplish a defined objective. These elements include products (hardware, software, firmware), processes, people, information, techniques, facilities, services, and other support elements. (INCOSE)

[4.1.46] ... combination of interacting elements organized to achieve one or more stated purposes. (ISO/IEC/IEEE 15288)

Thus, the usage of terminology throughout this handbook is clearly an elaboration of the fundamental idea that a system is a purposeful whole that consists of interacting parts.

An external view of a system must introduce elements that specifically do not belong to the system but do interact with the system. This collection of elements is called the *operating environment or context* and can include the users (or operators) of the system.

The internal and external views of a system give rise to the concept of a *system boundary*. In practice, the system boundary is a “line of demarcation” between the system itself and its greater context (to include the operating environment). It defines what belongs to the system and what does not. The system boundary is not to be confused with the subset of elements that interact with the environment.

The *functionality* of a system is typically expressed in terms of the interactions of the system with its operating environment, especially the users. When a system is considered as an integrated combination of interacting elements, the functionality of the system derives not just from the interactions of individual elements with the environmental elements but also from how these interactions are influenced by the organization (interrelations) of the system elements. This leads to the concept of *system architecture*, which ISO/IEC/IEEE 42010 (2011) defines as

the fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution.

This definition speaks to both the internal and external views of the system and shares the concepts from the definitions of a system.

### 2.2.2 Scientific Terminology Related to System Concepts

In general, *engineering* can be regarded as the practice of creating and sustaining services, systems, devices, machines, structures, processes, and products to improve the quality of life—getting things done effectively and efficiently. The repeatability of experiments demanded by science is critical for delivering practical engineering solutions that have commercial value. Engineering in general and SE in particular draw heavily from the terminology and concepts of science.

An *attribute* of a system (or system element) is an observable characteristic or property of the system (or system element). For example, among the various attributes

of an aircraft is its air speed. Attributes are represented symbolically by variables. Specifically, a *variable* is a symbol or name that identifies an attribute. Every variable has a domain, which could be but is not necessarily measurable. A *measurement* is the outcome of a process in which the system of interest (SOI) interacts with an observation system under specified conditions. The outcome of a measurement is the assignment of a *value* to a variable. A system is in a *state* when the values assigned to its attributes remain constant or steady for a meaningful period of time (Kaposi and Myers, 2001). In SE and software engineering, the *system elements* (e.g., software objects) have *processes* (e.g., operations) in addition to attributes. These have the binary logical values of being either *idle* or *executing*. A complete description of a system state therefore requires values to be assigned to both attributes and processes. *Dynamic behavior* of a system is the time evolution of the system state. *Emergent behavior* is a behavior of the system that cannot be understood exclusively in terms of the behavior of the individual system elements.

The key concept used for problem solving is the *black box/white box* system representation. The black box representation is based on an external view of the system (attributes). The white box representation is based on an internal view of the system (attributes and structure of the elements). There must also be an understanding of the relationship between the two. A system, then, is represented by the (external) attributes of the system, its internal attributes and structure, and the interrelationships between these that are governed by the laws of science.

### 2.2.3 General Systems Methodologies

Early pioneers of SE and software engineering, such as Yourdon (1989) and Wymore (1993), long sought to bring discipline and precision to the understanding and management of the dynamic behavior of a system by seeking relations between the external and internal representations of the system. Simply stated, they believed that if the flow of dynamic behavior (the system state evolution) could be mapped coherently into the flow of states of the constituent elements of the system, then emergent behaviors could be better understood and managed.

Klir (1991) complemented the concepts of a system in engineering and science with a general systems methodology. He regarded problem solving in general to rest upon a principle of alternatively using abstraction and

interpretation to solve a problem. He considered that his methodology could be used both for system inquiry (i.e., the representation of an aspect of reality) and for system definition (i.e., the representation of purposeful man-made objects).

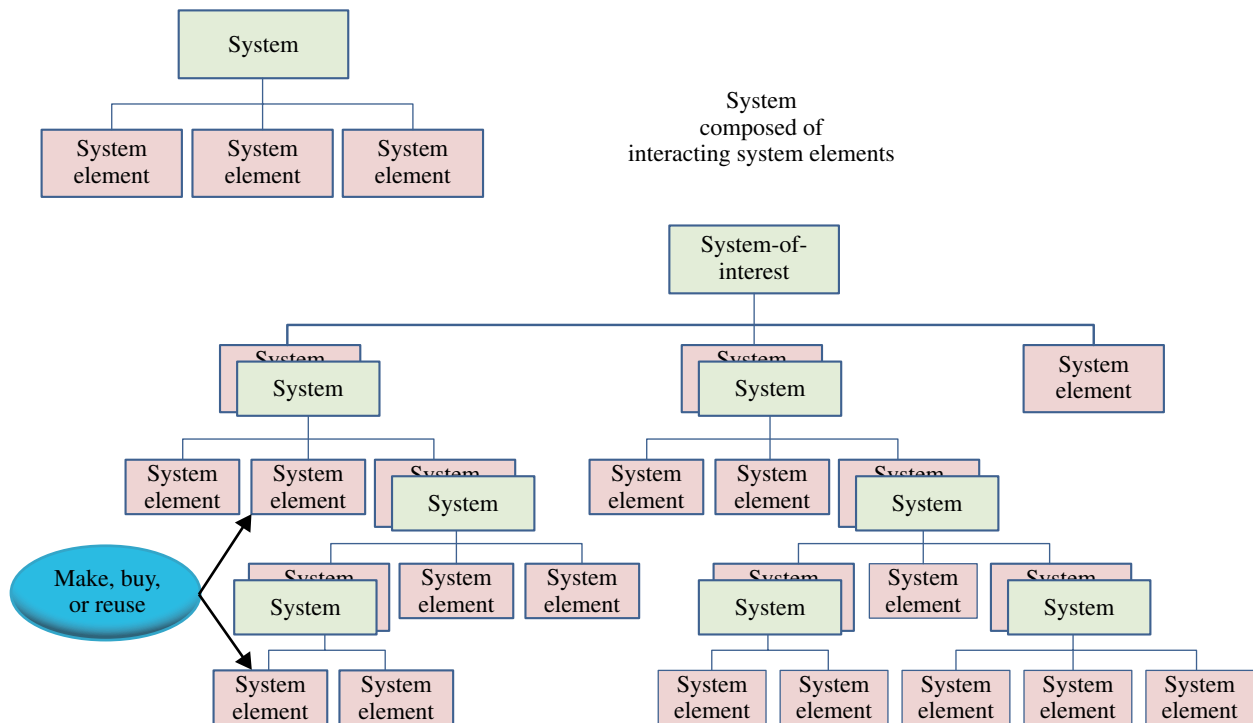
### 2.3 THE HIERARCHY WITHIN A SYSTEM

In the ISO/IEC/IEEE usage of terminology, the *system elements* can be *atomic* (i.e., not further decomposed), or they can be *systems on their own merit* (i.e., decomposed into further subordinate system elements). The *integration* of the system elements must establish the relationship between the effects that *organizing* the elements has on their *interactions* and how these effects enable the system to achieve its *purpose*.

One of the challenges of system definition is to understand what level of detail is necessary to define each system element and the interrelations between elements. Because the SOIs are in the real world, this means that

the response to this challenge will be domain specific. A system element that needs only a black box representation (external view) to capture its requirements and confidently specify its real-world solution definition can be regarded as atomic. Decisions to make, buy, or reuse the element can be made with confidence without further specification of the element. This leads to the concept of hierarchy within a system.

One approach to defining the elements of a system and their interrelations is to identify a complete set of distinct system elements with regard only to their relation to the whole (system) by suppressing details of their interactions and interrelations. This is referred to as a partitioning of the system. Each element can be either atomic or it can be a much higher level that could be viewed as a system itself. At any given level, the elements are grouped into distinct subsets of elements subordinated to a higher level system, as illustrated in Figure 2.1. Thus, hierarchy within a system is an organizational representation of system structure using a partitioning relation.



**FIGURE 2.1** Hierarchy within a system. This figure is adapted from ISO/IEC/IEEE 15288:2015, Figure 1 on page 11 and Figure 2 on page 12, with permission from the ANSI on behalf of the ISO. © ISO 2015. All rights reserved.

The concept of a system hierarchy described in ISO/IEC/IEEE 15288 is as follows:

[5.2.2] The system life cycle processes ... are described in relation to a system that is composed of a set of interacting system elements, each of which can be implemented to fulfill its respective specified requirements.

The art of defining a hierarchy within a system relies on the ability of the systems engineer to strike a balance between clearly and simply defining span of control and resolving the structure of the SOI into a complete set of system elements that can be implemented with confidence. Urwick (1956) suggests that a possible heuristic is for each level in the hierarchy to have no more than  $7 \pm 2$  elements subordinate to it. Others have also found this heuristic to be useful (Miller, 1956). A level of design with too few subordinate elements is unlikely to have a distinct design activity. In this case, both design and verification activities may contain redundancy. In practice, the nomenclature and depth of the hierarchy can and should be adjusted to fit the complexity of the system and the community of interest.

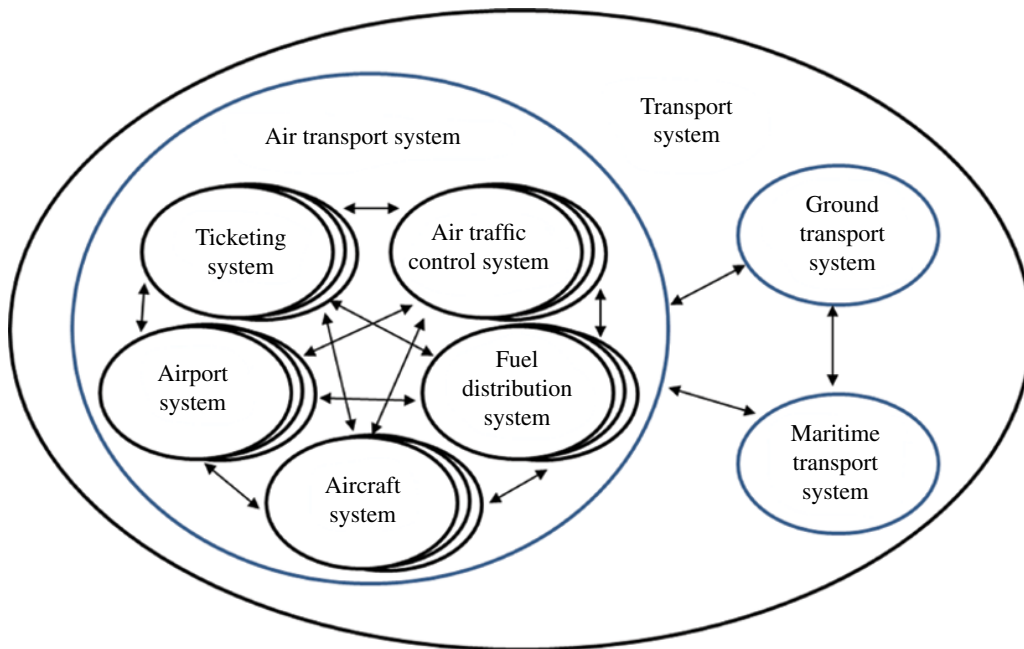
## 2.4 DEFINITION OF SYSTEMS OF SYSTEMS

A “system of systems” (SoS) is an SOI whose elements are managerially and/or operationally independent systems. These interoperating and/or integrated collections of constituent systems usually produce results unachievable by the individual systems alone. Because an SoS is itself a system, the systems engineer may choose whether to address it as either a system or as an SoS, depending on which perspective is better suited to a particular problem.

The following characteristics can be useful when deciding if a particular SOI can better be understood as an SoS (Maier, 1998):

- Operational independence of constituent systems
- Managerial independence of constituent systems
- Geographical distribution
- Emergent behavior
- Evolutionary development processes

Figure 2.2 illustrates the concept of an SoS. The air transport system is an SoS comprising multiple aircraft,



**FIGURE 2.2** Example of the systems and systems of systems within a transport system of systems. Reprinted with permission from Judith Dahmann. All other rights reserved.

airports, air traffic control systems, and ticketing systems, which along with other systems such as security and financial systems facilitate passenger transportation. There are equivalent ground and maritime transportation SoS that are all in turn part of the overall transport system (an SoS in the terms of this description).

The SoS usually exhibits complex behaviors, often created by the existence of the aforementioned Maier's characteristics. "Complexity" is essentially different from "complicated." In complicated systems, such as an automobile, the interactions between the many parts are governed by fixed relationships. This allows reasonably reliable prediction of technical, time, and cost issues. In complex systems, such as the air transport system, interactions between the parts exhibit self-organization, where local interactions give rise to novel, nonlocal, emergent patterns. Complicated systems can often become complex when the behaviors change, but even systems of very few parts can sometimes exhibit surprising complexity.

The best way to understand a complicated system is to break it down into parts recursively until the parts are so simple that we understand them and then to reassemble the parts to understand the whole. However, this approach does not help us to understand a complex system, because the emergent properties that we really care about disappear when we examine the parts in isolation. A fundamentally different approach is required to understand the whole in context through iterative exploration and adaptation. As a result, SE requires a balance of linear, procedural methods for sorting through complicatedness ("systematic activity") and holistic, nonlinear, iterative methods for harnessing complexity ("systemic" or systems thinking and analysis—always required when dealing with SoS). The tension between breaking things apart and keeping them in context must be dynamically managed throughout the SE process.

The following challenges all influence the engineering of an SoS (Dahmann, 2014):

1. *SoS authorities*—In an SoS, each constituent system has its own local "owner" with its stakeholders, users, business processes, and development approach. As a result, the type of organizational structure assumed for most traditional SE under a single authority responsible for the entire system is absent from most SoS. In an SoS, SE relies on crosscutting analysis and on composition and integration of constituent systems, which in turn depend on an agreed common purpose and motivation for these systems to work together toward collective objectives that may or may not coincide with those of the individual constituent systems.
2. *Leadership*—Recognizing that the lack of common authorities and funding poses challenges for SoS, a related issue is the challenge of leadership in the multiple organizational environment of an SoS. This question of leadership is experienced where a lack of structured control normally present in SE requires alternatives to provide coherence and direction, such as influence and incentives.
3. *Constituent systems' perspectives*—SoS are typically composed, at least in part, of in-service systems, which were often developed for other purposes and are now being leveraged to meet a new or different application with new objectives. This is the basis for a major issue facing SoS SE, that is, how to technically address issues that arise from the fact that the systems identified for the SoS may be limited in the degree to which they can support the SoS. These limitations may affect initial efforts at incorporating a system into an SoS, and systems' commitments to other users may mean that they may not be compatible with the SoS over time. Further, because the systems were developed and operate in different situations, there is a risk that there could be a mismatch in understanding the services or data provided by one system to the SoS if the particular system's context differs from that of the SoS.
4. *Capabilities and requirements*—Traditionally (and ideally), the SE process begins with a clear, complete set of user requirements and provides a disciplined approach to develop a system to meet these requirements. Typically, SoS are comprised of multiple independent systems with their own requirements, working toward broader capability objectives. In the best case, the SoS capability needs are met by the constituent systems as they meet their own local requirements. However, in many cases, the SoS needs may not be consistent with the requirements for the constituent systems. In these cases, SoS SE needs to identify alternative approaches to meeting those needs either through