SEBoK GUIDE TO THE SYSTEMS ENGINEERING BODY OF KNOWLEDGE

Guide to the Systems Engineering Body of Knowledge (SEBoK), version 2.2

Part 4

Please note that this is a PDF extraction of the content from www.sebokwiki.org

INCOSE

SYSTEMS ENGINEERING RESEARCH CENTER

IEEE computer society

Guide to the Systems Engineering Body of Knowledge, Part 4

version 2.1

PDF generated using the open source mwlib toolkit. See http://code.pediapress.com/ for more information.
PDF generated at: Thu, 31 Oct 2019 15:17:52 EDT

Contents

Articles

References

Front Matter

Letter from the Editor

Hi there! Welcome to the October 2019 instantiation of the Systems Engineering Body of Knowledge. Since version 1.0 appeared in September 2012, that means the SEBoK just celebrated its 7th birthday! This release, version 2.1 is also my third release as Editor in chief. This release brings what I hope are some exciting changes for the readers and authors.

The **first change** I hope you notice is that we have added bylines to those articles for which we can track their origins. As of this release, we are recognizing the contribution of lead authors and the additional contributing authors. It is our hope that these contributions will be beneficial to the authors in their professional lives - being able to prove their contributions to this important knowledge base.

The **next obvious change** should be the way glossary bubbles have been updated. They are more readable now, with a grey background and black text.

Other changes include **new articles** on:

- Digital Engineering
- Mission Engineering
- Set Based Design
- MBSE Adoption Trends 2009-2018

Additionally, we have **updated content** on Resilience, Human Systems Integration, and Capability Engineering. Part 1 also received a wire brushing. We have also begun incorporating video. You will find a short video on the Main page. We are also going to begin to look at existing INCOSE YouTube channel content to look for 1-3 minute clips we can strategically place throughout the SEBoK to add value.

There is a big announcement to be made relative to the SEBoK. The IEEE Computer Society has been one of the three stewards of the SEBoK from the beginning. They have had a seat on the Board of Governors, and have provided invaluable counsel. In January 2020, that stewardship will be transferred from the IEEE Computer Society to the IEEE Systems Council. The Editorial Board looks forward to the continued support and participation of IEEE. Thank you IEEE Computer Society, and in particular to Rich Hilliard and Andy Chen.

Regarding the reach of the SEBoK, there were over 29,000 visitors and 68,781 page views during the the month of July 2019. That brings our total page views to over 3.45M since 2012. Top content pages in July: 1) Stakeholder Needs and Requirements, 2) Types of Models, 3) Types of Systems, 4) Systems Requirements, and 5) Reliability, Availability, and Maintainability. Top countries accessing the SEBoK in July:

1. US
2. India
3. Australia
4. United Kingdom
5. Philippines

Looking forward to the next release, it is my hope that those of you that enjoy working with video will think about creating video content now that we have that capability. Please limit your submissions to no more than 3 minute clips, and the preferred format is mp4.

I am still looking for additional authors and folks interested in taking a leadership role as editors to help manage and grow our content for specific areas. It would be nice to add some more content in Part 6: Related Disciplines, and Part 7: SE Implementation Examples. If you would like to author an article for those sections, please reach our to Nicole Hutchinson (emtnicole@gmail.com) or myself (rcloutier@southalabama.edu).

That is it for now ... I hope to see you at the upcoming International Workshop being held in Torrence, CA in January 2020. If you have ideas for the SEBoK, or would like to get involved, be sure to find me there and we can have some coffee and chat. But, do not feel you have to wait until then to get involved - reach out now! Thanks for your ongoing support.

BKCASE Governance and Editorial Board

BKCASE Governing Board

The three SEBoK steward organizations – the International Council on Systems Engineering (INCOSE), the Institute of Electrical and Electronics Engineers Computer Society (IEEE-CS), and the Systems Engineering Research Center (SERC) provide the funding and resources needed to sustain and evolve the SEBoK and make it available as a free and open resource to all. The stewards appoint the BKCASE Governing Board to be their primary agents to oversee and guide the SEBoK and its companion BKCASE product, GRCSE.

The BKCASE Governing Board includes:

- **The International Council on Systems Engineering (INCOSE)**
  - Art Pyster (Governing Board Chair), Paul Frenz
- **Systems Engineering Research Center (SERC)**
  - Jon Wade, Cihan Dagli
- **IEEE Computer Society (IEEE CS)**
  - Andy Chen, Rich Hilliard

Past INCOSE governors Bill Miller, Kevin Forsberg, David Newbern, David Walden, Courtney Wright, Dave Olwell, Ken Nidiffer, Richard Fairley, Massood Towhidnejad, and John Keppler. The governors would also like to acknowledge John Keppler, IEEE Computer Society, who has been instrumental in helping the Governors to work within the IEEE CS structure.

The stewards appoint the BKCASE Editor in Chief to manage the SEBoK and GRCSE and oversee the Editorial Board.

Editorial Board

The SEBoK Editorial Board is chaired by an Editor in Chief, supported by a group of Associate Editors.

SEBoK Editor in Chief

Robert J. Cloutier

University of South Alabama

rcloutier@southalabama.edu [1]

Responsible for the appointment of SEBoK Editors and for the strategic direction and overall quality and coherence of the SEBoK.

SEBoK Managing Editor

Nicole Hutchison

Stevens Institute of Technology

nicole.hutchison@stevens.edu [2] or emtnicole@gmail.com [3]

Responsible for the the day-to-day operations of the SEBoK and supports the Editor in Chief.

Each Editor has his/her area(s) of responsibility, or shared responsibility, highlighted in the table below.

SEBoK Part 1: SEBoK Introduction

Lead Editor: Robert J. Cloutier

University of South Alabama

rcloutier@southalabama.edu [1]

Responsible for Part 1

SEBoK Part 2: Foundations of Systems Engineering

Lead Editor: Gary Smith

Airbus

gary.r.smith@airbus.com [4]

Assistant Editor: Dov Dori

Massachusetts Institute of Technology (USA) and Technion Israel Institute of Technology (Israel)

dori@mit.edu [5]

Responsible for the Representing Systems with Models knowledge area

Assistant Editor: Duane Hybertson

MITRE (USA)

dhyberts@mitre.org [6]

Jointly responsible for the Systems Fundamentals, Systems Science and Systems Thinking knowledge areas

Assistant Editor: Peter Tuddenham

College of Exploration (USA)

Peter@coexploration.net [7]

Assistant Editor: Cihan Dagli

Missouri University of Science & Technology (USA)

dagli@mst.edu [8]

Responsible for the Systems Approach Applied to Engineered Systems knowledge areas.

SEBoK Part 3: Systems Engineering and Management

Assistant Editor: Barry Boehm

University of Southern California (USA)

boehm@usc.edu [9]

Jointly responsible for the Systems Engineering Management and Life Cycle Models knowledge areas

Assistant Editor: Kevin Forsberg

OGR Systems

kforsberg@ogrsystems.com [10]

Jointly responsible for the Systems Engineering Management and Life Cycle Models knowledge areas

Assistant Editor: Gregory Parnell

University of Arkansas (USA)

gparnell@uark.edu [11]

Responsible for Systems Engineering Management knowledge area.

Assistant Editor: Garry Roedler

Lockheed Martin (USA)

garry.j.roedler@lmco.com [12]

Responsible for the Concept Definition and System Definition knowledge areas.

Assistant Editor: Phyllis Marbach

Incose LA (USA)

prmarbach@gmail.com [13]

Assistant Editor: Ken Zemrowski

ENGILITY

kenneth.zemrowski@incose.org [14]

Responsible for the Systems Engineering Standards knowledge area.

SEBoK Part 4: Applications of Systems Engineering

Lead Editor: Judith Dahmann

MITRE Corporation (USA)

jdahmann@mitre.org [15]

Jointly responsible for Product Systems Engineering and Systems of Systems (SoS) knowledge areas

Assistant Editor: Michael Henshaw

Loughborough University (UK)

M.J.d.Henshaw@lboro.ac.uk [16]

Jointly responsible for Product Systems Engineering and Systems of Systems (SoS) knowledge areas

Assistant Editor: James Martin

The Aerospace Corporation

james.martin@incose.org [17]

Responsible for the Enterprise Systems Engineering knowledge area.

SEBoK Part 5: Enabling Systems Engineering

Assistant Editor: Emma Sparks

Cranfield University

Jointly responsible for the Enabling IndividualsandEnabling Teams knowledge area

Assistant Editor: Rick Hefner

California Institute of Technology

Rick.Hefner@ngc.com [18]

Assistant Editor: Tim Ferris *Cranfield University*

Timothy.Ferris@cranfield.ac.uk [19]

Assistant Editor: Bernardo Delicado

MBDA / INCOSE

bernardo.delicado@mbda-systems.com [20]

SEBoK Part 6: Related Disciplines

Lead Editor: Alice Squires

Washington State University (USA)

alice.squires@wsu.edu [21]

SEBoK Part 7: Systems Engineering Implementation Examples

Lead Author: Clif Baldwin

FAA Technical Center

cliftonbaldwin@gmail.com [22]

Responsible for Part 7: Systems Engineering Implementation Examples, which includes Case Studies and examples.

Graduate Reference Curriculum for Systems Engineering (GRCSE)

David H. Olwell

St. Martin's University (USA)

dolwell@stmartin.edu [23]

Associate Editor for SEBoK.

Graduate Student Support

With SEBoK v. 2.1, the Governing Board has hired a graduate student to support the Editor in Chief and Managing Editor. Madeline Haas, a master's student at George Mason University, is the current graduate student supporting the SEBoK and we gratefully acknowledge her exemplary efforts.

Interested in Editing?

The Editor in Chief is looking for additional editors to support the evolution of the SEBoK. Editors are responsible for maintaining and updating one to two knowledge areas, including recruiting and working with authors, ensuring the incorporation of community feedback, and maintaining the quality of SEBoK content. We are specifically interested in support for the following knowledge areas:

- System Deployment and Use
- Product and Service Life Management
- Enabling Businesses and Enterprises
- Systems Engineering and Software Engineering
- Procurement and Acquisition
- Systems Engineering and Specialty Engineering

If you are interested in being considered for participation on the Editorial Board, please visit the BKCASE website http://www.bkcase.org/join-us/or contact the BKCASE Staff directly at bkcase.incose.ieeecs@gmail.com [24].

SEBoK v. 2.1, released 31 October 2019

References

[1] mailto:rcloutier@southalabama.edu
[2] mailto:nicole.hutchison@stevens.edu
[3] mailto:emtnicole@gmail.com
[4] mailto:gary.r.smith@airbus.com
[5] mailto:dori@mit.edu
[6] mailto:dhyberts@mitre.org
[7] mailto:Peter@coexploration.net
[8] mailto:dagli@mst.edu
[9] mailto:boehm@usc.edu
[10] mailto:kforsberg@ogrsystems.com
[11] mailto:gparnell@uark.edu
[12] mailto:garry.j.roedler@lmco.com
[13] mailto:prmarbach@gmail.com
[14] mailto:kenneth.zemrowski@incose.org
[15] mailto:jdahmann@mitre.org
[16] mailto:M.J.d.Henshaw@lboro.ac.uk
[17] mailto:james.martin@incose.org
[18] mailto:Rick.Hefner@ngc.com
[19] mailto:Timothy.Ferris@cranfield.ac.uk
[20] mailto:bernardo.delicado@mbda-systems.com
[21] mailto:alice.squires@wsu.edu
[22] mailto:cliftonbaldwin@gmail.com
[23] mailto:dolwell@stmartin.edu
[24] mailto:bkcase.incose.ieeecs@gmail.com

Acknowledgements and Release History

This article describes the contributors to the current version of the SEBoK. For information on contributors to past versions of the SEBoK, please follow the links under "SEBoK Release History" below. To learn more about the updates to the SEBoK for v. 2.1, please see the Letter from the Editor.

The BKCASE Project began in the fall of 2009. Its aim was to add to the professional practice of systems engineering by creating two closely related products:

- *Guide to the Systems Engineering Body of Knowledge (SEBoK)*
- *Graduate Reference Curriculum for Systems Engineering (GRCSE)*

BKCASE History, Motivation, and Value

The **Guide to the Systems Engineering Body of Knowledge (SEBoK)** is a living authoritative guide that discusses knowledge relevant to Systems Engineering. It defines how that knowledge should be structured to facilitate understanding, and what reference sources are the most important to the discipline. The curriculum guidance in the **Graduate Reference Curriculum for Systems Engineering (GRCSE)** (Pyster and Olwell et al. 2015) makes reference to sections of the SEBoK to define its core knowledge; it also suggests broader program outcomes and objectives which reflect aspects of the professional practice of systems engineering as discussed across the SEBoK.

Between 2009 and 2012 BKCASE was led by Stevens Institute of Technology and the Naval Postgraduate School in coordination with several professional societies and sponsored by the U.S. Department of Defense (DoD), which provided generous funding. More than 75 authors and many other reviewers and supporters from dozens of companies, universities, and professional societies across 10 countries contributed many thousands of hours writing the SEBoK articles; their organizations provided significant other contributions in-kind.

The SEBoK came into being through recognition that the systems engineering discipline could benefit greatly by having a living authoritative guide closely related to those groups developing guidance on advancing the practice, education, research, work force development, professional certification, standards, etc.

At the beginning of 2013, BKCASE transitioned to a new governance model with shared stewardship between the Systems Engineering Research Center (SERC) [1], the International Council on Systems Engineering (INCOSE) [2], and the Institute of Electrical and Electronics Engineers Computer Society (IEEE-CS) [3]. This governance structure was formalized in a memorandum of understanding between the three stewards that was finalized in spring of 2013. The stewards have reconfirmed their commitment to making the SEBoK available at no cost to all users, a key principle of BKCASE.

As of the end of July 2019, SEBoK articles have had over 3.4M pageviews from 1.7M unique visits. We hope the SEBoK will regularly be used by thousands of systems engineers and others around the world as they undertake technical activities such as eliciting requirements, creating systems architectures, or analysis system test results; and professional development activities such as developing career paths for systems engineers, deciding new curricula for systems engineering university programs, etc.

Governance

The SEBoK is shaped by the BKCASE Editorial Board and is overseen by the BKCASE Governing Board. A complete list of members for each of these bodies can be found on the BKCASE Governance and Editorial Board page.

Content and Feature Updates for 2.1

This version of the SEBoK was released 31 October 2019. This is a significant release of the SEBoK which includes new articles, new functionality and minor updates throughout. The SEBoK PDF was also updated (see Download SEBoK PDF).

For more information about this release please refer to Development of SEBoK v. 2.1.

SEBoK Release History

There have been 21 releases of the SEBoK to date, collected into 13 main releases.

Main Releases

- Version 2.1 - Current version. This is a significant release with new articles, new functionality, and minor updates throughout.
- Version 2.0 - This was a major release of the SEBoK which included incorporation of multi-media and a number of changes to the functions of the SEBoK.
- Version 1.9.1 - This was a micro release of the SEBoK which included updates to the editorial board, and a number of updates to the wiki software.
- Version 1.9 - A minor update which included updates to the System Resilience article in Part 6: Related Disciplines, as well as a major restructuring of Part 7: Systems Engineering Implementation Examples. A new example has been added around the use of model based systems engineering for the thirty-meter telescope.
- Version 1.8 - A minor update, including an update of the Systems of Systems (SoS) knowledge area in Part 4: Applications of Systems Engineering where a number of articles were updated on the basis of developments in the area as well as on comments from the SoS and SE community. Part 6: Related Disciplines included updates to the Manufacturability and Producibility and Reliability, Availability, and Maintainability articles.
- Version 1.7 - A minor update, including a new Healthcare SE Knowledge Area (KA), expansion of the MBSE area with two new articles, Technical Leadership and Reliability, Availability, and Maintainability and a new case study on the Northwest Hydro System.
- Version 1.6 - A minor update, including a reorganization of Part 1 SEBoK Introduction, a new article on the Transition towards Model Based Systems Engineering and a new article giving an overview of Healthcare Systems Engineering, a restructure of the Systems Engineering and Specialty Engineering KA.
- Version 1.5 - A minor update, including a restructure and extension of the Software Engineering Knowledge Area, two new case studies, and a number of corrections of typographical errors and updates of outdated references throughout the SEBoK.
- Version 1.4 - A minor update, including changes related to ISO/IEC/IEEE 15288:2015 standard, three new case studies and updates to a number of articles.
- Version 1.3 - A minor update, including three new case studies, a new use case, updates to several existing articles, and updates to references.
- Version 1.2 - A minor update, including two new articles and revision of several existing articles.
- Version 1.1 - A minor update that made modest content improvements.
- Version 1.0 - The first version intended for broad use.

Click on the links above to read more information about each release.

Wiki Team

In January 2011, the authors agreed to move from a document-based SEBoK to a wiki-based SEBoK, and beginning with v. 0.5, the SEBoK has been available at www.sebokwiki.org [4] Making the transition to a wiki provided three benefits:

1. easy worldwide access to the SEBoK;
2. more methods for search and navigation; and
3. a forum for community feedback alongside content that remains stable between versions.

The Managing Editor is responsible for maintenance of the wiki infrastructure as well as technical review of all materials prior to publication. Contact the managing editor at emtnicole@gmail.com [3]

The wiki is currently supported by Ike Hecht from WikiWorks.

SEBoK v. 2.1, released 31 October 2019

References

[1] http://www.sercuarc.org
[2] http://www.incose.org
[3] http://www.computer.org
[4] http://www.sebokwiki.org

Cite the SEBoK

When **citing the SEBoK in general**, users must cite in the following manner:

> SEBoK Editorial Board. 2019. *The Guide to the Systems Engineering Body of Knowledge (SEBoK)*, v. 2.1, R.J. Cloutier (Editor in Chief). Hoboken, NJ: The Trustees of the Stevens Institute of Technology. Accessed [DATE]. www.sebokwiki.org. BKCASE is managed and maintained by the Stevens Institute of Technology Systems Engineering Research Center, the International Council on Systems Engineering, and the Institute of Electrical and Electronics Engineers Computer Society.

To **cite a specific article** within the SEBoK, please use:

> SEBoK Authors. Author name(s). "Article Title." in SEBoK Editorial Board. 2019. *The Guide to the Systems Engineering Body of Knowledge (SEBoK)*, v. 2.1 R.J. Cloutier (Editor in Chief). Hoboken, NJ: The Trustees of the Stevens Institute of Technology. Accessed [DATE]. www.sebokwiki.org. BKCASE is managed and maintained by the Stevens Institute of Technology Systems Engineering Research Center, the International Council on Systems Engineering, and the Institute of Electrical and Electronics Engineers Computer Society.

Note that each page will include the by line (author names) for the article. If no byline is listed, please use "SEBoK Authors".

When **using material** from the SEBoK, attribute the work as follows:

> This material is used under a Creative Commons Attribution-NonCommercial ShareAlike 3.0 Unported License from The Trustees of the Stevens Institute of Technology. See Stevens Terms for Publication located in Copyright Information.

Cite this Page

This feature is located under "Tools" on the left menu. It provides full information to cite the specific article that you are currently viewing; this information is provided in various common citation styles including APA, MLA, and Chicago.

Bkcase Wiki:Copyright

Please read this page which contains information about how and on what terms you may use, copy, share, quote or cite the Systems Engineering Body of Knowledge (SEBoK):

Copyright and Licensing

Attribution

When **using text material from the SEBoK**, users who have accepted one of the Creative Commons Licenses described above terms noted below must attribute the work as follows:

This material is used under a Creative Commons Attribution-NonCommercial ShareAlike 3.0 Unported License from The Trustees of the Stevens Institute of Technology.

When **citing the SEBoK in general**, please refer to the format described on the Cite the SEBoK page.

When **using images, figures, or tables from the SEBoK**, please note the following intellectual property (IP) classifications:

- Materials listed as "SEBoK Original" may be used in accordance with the Creative Commons attribution (above).
- Materials listed as "Public Domain" may be used in accordance with information in the public domain.
- Materials listed as "Used with Permission" are copyrighted and *permission must be sought from the copyright owner* to reuse them.

Part 4: Applications of Systems Engineering

Knowledge Area: Product Systems Engineering

Product Systems Engineering

Lead Authors: *Bud Lawson, Ricardo Pineda*

Product systems engineering (PSE) is at the core of the new product development process (NPDP) that is needed to successfully develop and deploy products into different market segments. A market can be consumer based (e.g., private enterprises or general consumers) or it can be public (not-for-profit). Public markets address the strategic needs of a country or region, such as military, healthcare, educational, transportation, and energy needs. NPDP has two significantly overlapping and integrated activities:

1. **Systems engineering:** This includes concept generation, engineering design/development, and deployment
2. **Market development:** This includes market research, market analysis, product acceptance and market growth (diffusion), and rate of adoption

NPDP also includes manufacturability/producibility, logistics and distribution, product quality, product disposal, conformance to standards, stakeholder's value added, and meeting customer's expectations. The internal enterprise competence and capabilities such as customer support, sales & marketing, maintenance and repair, personnel training, etc., must also be taken into account.

Topics

Each part of the SEBoK is divided into knowledge areas (KAs), which are groupings of information with a related theme. The KAs in turn are divided into topics. This KA contains the following topics:

- Product Systems Engineering Background
- Product as a System Fundamentals
- Business Activities Related to Product Systems Engineering
- Product Systems Engineering Key Aspects
- Product Systems Engineering Special Activities

The Product Systems Engineering Background article discusses product types and the need for a product to be aligned with the business objectives of the enterprise. It also discusses the relationships between PSE and product development and technology development.

Various types of connections between product elements, and the concept of enabling systems are introduced in the Product as a System Fundamentals article. It also discusses product architecture, modeling, analysis, and integration with various specialty engineering areas.

Product launching and product offerings have close linkages to different business processes. The major linkages are to business development, marketing, product lines, quality management, project management, operations management, supply chain management, etc. These and other topics are described in the Business Activities Related to Product Systems Engineering article.

Products emerge when they are realized based upon a system definition. Realizing the system results in instances of the system that are either delivered as products to a specific acquirer (based upon an agreement) or are offered directly to buyers and users. Key Aspects of PSE are discussed in the Product Systems Engineering Key Aspects article which discusses aspects such as acquired vs. offered products, product lifecycle and adoption rates, integrated product teams (IPTs) and integrated product development teams (IPDTs), product architectures, requirements and standards, etc.

The last article, Product Systems Engineering Special Activities, covers some of the special activities carried out by PSE during the different stages from concept through product deployment.

Key Terms and Concepts

Product

A product is an artifact that is created by some person or by some process such as a manufacturing process, software source code compilation and integration, building construction, creative writing process, or data processing.

In general, a business product is defined as a *thing produced by labor or effort*, or the *result of an act or a process*. It stems from the verb *produce*, from the Latin *prōdūce(re) (to) lead or bring forth*. Since 1575, the word *product* has referred to anything produced, and since 1695, the word *product* has referred to *a thing or things produced*.

In economics and commerce, products belong to a broader category of *goods*. The economic meaning of the word *product* was first used by political economist Adam Smith. In marketing, a product is anything that can be offered to a market that might satisfy a want or a need. In retail industries, products are called merchandise. In manufacturing, products are purchased as raw materials and sold as finished goods. Commodities are usually raw materials, such as metals and agricultural products, but a commodity can also be anything widely available in the open market. In project management, products are the formal definitions of the project deliverables that make up or contribute to delivering the objectives of the project. In insurance, the policies are considered products offered for sale by the insurance company that created the contract.

Product System

A product system is the combination of end products and the enabling products for those end products. This concept of a product system is illustrated in Figure 1. In the ANSI/EIA 632-2003 standard, just the term *system* is used, but the scope of the standard is clearly restricted to product systems.

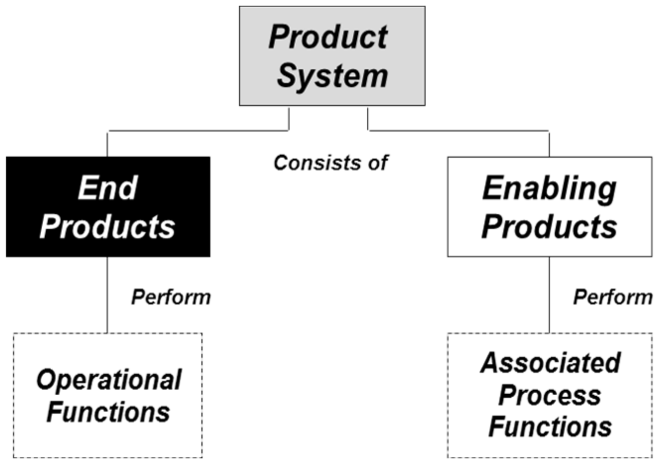


Figure 1. Product System Components (ANSI/EIA 632). Excerpts from "Processes for Engineering a System" (EIA-632), Copyright © (1999) TechAmerica. All Rights Reserved. Reprinted by Permission. All other rights are reserved by the copyright owner.

The end product can also be considered as a system with its own elements or subsystems, each of which has its own enabling products as illustrated in Figure 2. The product development process usually focuses only on the engineering of the end product. PSE is essential when the enabling products are by themselves complex or their relationship to the end product is complex. Otherwise, the use of a traditional product development process is sufficient.

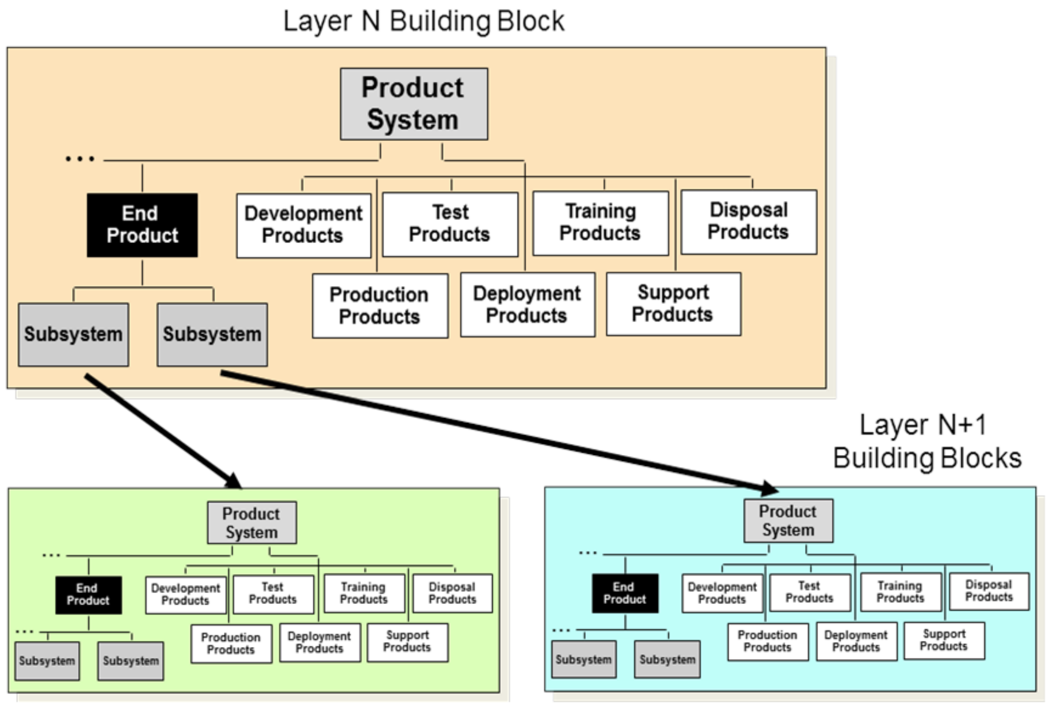


Figure 2. End Products and Enabling Products (ANSI/EIA 632). Excerpts from "Processes for Engineering a System" (EIA-632),

Product Realization System

There is a related system that enables the *realization* of the product system, which is the product realization system. It consists of *all the resources to be applied in causing the Intervention System [i.e., the product system, in this case] to be fully conceived, developed, produced, tested, and deployed* (Martin 2004). Lawson (2010) refers to this as a respondent system in the system coupling diagram. The intervention system is the system that is to be realized (or conceived and brought into being) in order to address some perceived problem in the context as shown in Figure 3.

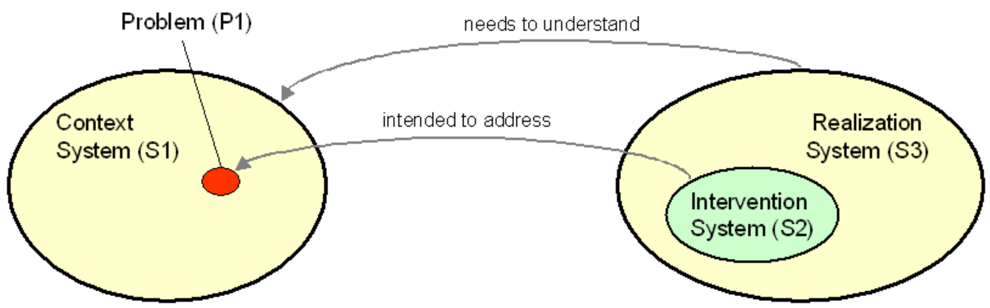


Figure 3. Realization System That Creates the Intervention to Solve a Problem (Martin 2004). Reprinted with Permission of Aerospace. All other rights are reserved by the copyright owner.

The realization system can be a service system (as described in knowledge area Service Systems Engineering) or an enterprise system (as described in the knowledge are Enterprise Systems Engineering). When the realization system is a service system, then the service could partially realize the system by just designing the product system without developing or creating it. This design service system can pass the design to a manufacturing service system that turns the design into a physical artifact. Usually an enterprise is established to orchestrate the disparate services into a cohesive whole that is efficiently and effectively performed to achieve the strategic goals of that enterprise.

The product realization system utilizes a product realization process as described in (Magrab et al 2010) or a product development process as described in (Wheelwright and Clark 1992).

Product Sustainment System

When the realization system delivers the product system into its intended environment, the product often needs a set of services to keep that product operational. This other system, when needed, is called the product sustainment system. It consists of various enabling products and operational services. The sustainment system in relation to the realization system and the deployed product system is illustrated in Figure 4. Notice that the realization may need to develop or modify the sustainment for the particular intervention (product) system under development.

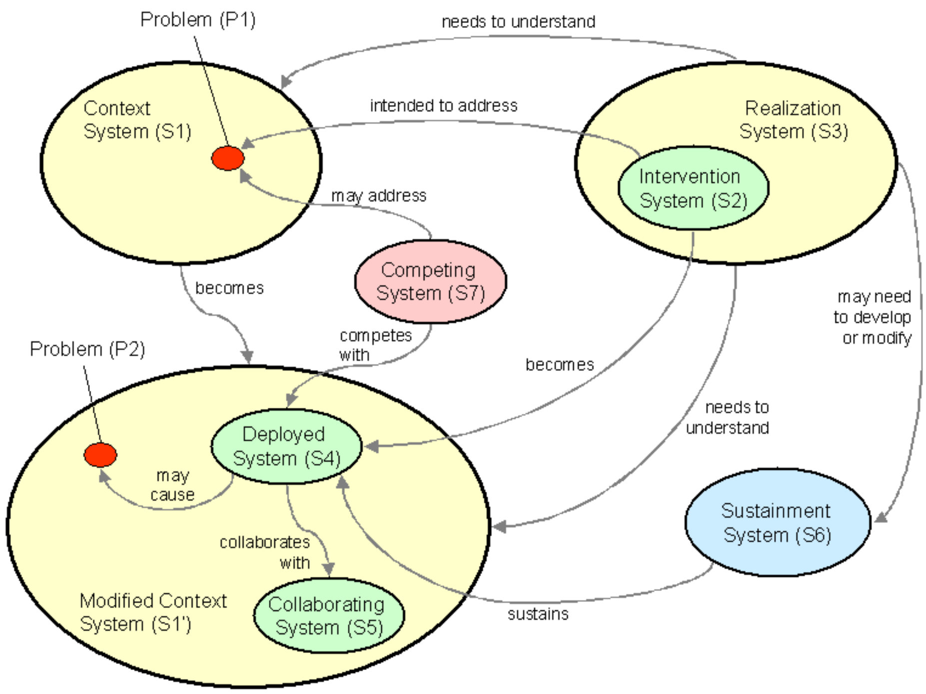


Figure 4. Product Sustainment System in Support of the Deployed Product System (Martin 2004). Reprinted with Permission of Aerospace.

Product Systems Engineering, Service Systems Engineering and Enterprise Systems Engineering

PSE is in line with Traditional Systems Engineering (TSE) as captured in most textbooks on the subject, such as Wasson (2006), Sage and Rouse (2009), and Blanchard and Fabrycky (2011). However, they do not cover the full breadth of PSE since they tend to focus on hardware and software products only. Other kinds of products to be engineered include personnel, facilities, data, materials, processes, techniques, procedures, and media (Martin 1997; Lawson 2010). Further discussions on the distinctions between the various kinds of products is provided in the Product Systems Engineering Background article. Product system domains could be data-intensive (e.g. transportation system), facilities-intensive (e.g. chemical processing plant), hardware-intensive (e.g. defense systems), or technique-intensive (e.g. search and rescue system). Most product systems are a composite of several different kinds of products that must be fully integrated to realize the complete value added potential for the different stakeholders.

When compared to Service Systems Engineering (SSE) and Enterprise Systems Engineering (ESE), PSE has some unique considerations:

- Often a product is part of a product line where both the product line and the products that make up that product line must be engineered simultaneously.
- Products are often composed of parts and sub-assemblies produced by several suppliers. This entails the need to work closely with the supply chain to ensure a successful product offering.
- Large complex products often require a lengthy and complicated series of steps for assembly, integration and test. During integration, many of the key assumptions made during the initial product design could be challenged.

- Products will usually require certification as to their safety or other factors like energy conservation and environmental compatibility. Electronic products often require certification to ensure electromagnetic compatibility and limited electronic emissions into the radio frequency spectrum. Transportation products require certification for safe operations.
- Products often have a complicated distribution network since they are not always developed where the end user may require it. There could be depots, warehouses, multi-modal transportation, wholesalers, dealers, and retail stores as part of this distribution network. This introduces challenges in delivery, maintenance and support of the product.
- Products must be engineered along with the realization system and the sustainment system. Sometimes it is necessary to make tradeoffs between the features and functions of the product system, the realization system and the sustainment system.

These considerations and others will be addressed in the articles under this knowledge area. One of the responsibilities of ESE is to manage these various considerations across multiple product lines across the enterprise while maximizing profits and customer satisfaction. SSE is often dependent on the products resulting from the PSE. A service will often be based on a product infrastructure that includes elements like data processing, hardware, software, data storage, data entry devices, display devices, service delivery facilities and techniques, service desk technicians, maintenance personnel, service offering catalogs and printed materials. Each of these products in the service infrastructure may need to be separately engineered using its own PSE lifecycle.

Creating Value

An enterprise that creates products must also create value in the eyes of the customer; a value that exceeds the cost of that product. This applies to both private and public enterprises operated for profit or not-for-profit. The creation and delivery of products may be the result of an acquisition agreement or an offering directly to buyers or users. To remain competitive, enterprises also need to understand the effects of the global economy, trends in industry, technology development needs, effects of new technology breakthroughs, market segments creation and their expectations, and most importantly, ever evolving customer expectations.

Ring (1998) defines a system value cycle with three levels that a systems approach must consider to deliver real world benefit:

1. stakeholder value
2. customer need
3. system solution

Value will be fully realized only when it is considered within the context of time, cost, and other resources appropriate to key stakeholders.

Aligning product characteristics with associated operational activities

The user of a product views the product as an asset that can be utilized in one's own systems of interest (Lawson 2010). Thus, in supplying the product, the expected form of operation becomes a driving factor in determining the characteristics of the product. In several contexts, in particular for military related products, the desired operational activities are termed concept of operations (ConOps) and in the case of commercial enterprises the intended use of the system is described through some form of *Market Service Description* of the product. The intended use of the product is market/customer driven and so the product characteristics must be aligned with the operational intent.

Architectures as basis for value assessment

Architectures can be used by enterprises to shift product development from individual products to an underlying product line architecture that incorporates the flexibility required by the enterprise to rapidly tailor new technologies and features to specific customer requirements (Phillips 2001). In determining the architecture of the product system, various alternative designs may arise. Each of the architecture alternatives is to be evaluated with respect to its value contribution to end users and other stakeholders.

Role of evaluation criteria in selection between product alternatives

In assessing the product system value, one must consider the measures that are to be used to determine the *goodness* of the product alternatives (alternative architectures and technologies) with respect to producibility, quality, efficiency, performance, cost, schedule and most importantly the coverage provided in meeting the customer's requirement or market opportunity.

Role of tradeoffs in maximizing value

The evaluation of alternatives must include the tradeoffs between conflicting properties. For example, in striving for superior quality and efficiency, tradeoffs must be made with respect to schedule and cost. See article on Measurement in Part 3. Tradeoffs are made during different stages of the development process: at the product or system level, at the subsystem and architecture definition level, and at the technology level (Blanchard and Fabrycky 2011).

There are a variety of methods for performing tradeoff analysis such as: utility theory, analytic hierarchical process, the Pugh selection method, multi-objective decision, multi-attribute utility analysis, and multi-variate analysis. For software, the Software Engineering Institute (SEI) provides 'The Architecture Tradeoff Analysis Method (ATAM)' (Kazman et al., 2000) for evaluating software architectures relative to quality attribute goals. ATAM evaluations expose architectural risks that potentially inhibit the achievement of an organization's business goals. The ATAM not only reveals how well an architecture satisfies particular quality goals, but also provides insight into how those quality goals interact with each other and how they trade off against each other.

Expanding role of software in creation of product value

Software has an increasing role in providing the desired functionality in many products. The embedding of software in many types of products (such as transportation vehicles, home appliance, and production equipment) accounts for an ever increasing portion of the product functionality. The current trend is the development of a network of systems that incorporate sensing and activating functions. The use of various software products in providing service is described in the Service Systems Engineering knowledge area.

References

Works Cited

ANSI/EIA. 2003. *Processes for Engineering a System.* Philadelphia, PA, USA: American National Standards Institute (ANSI)/Electronic Industries Association (EIA). ANSI/EIA 632-2003.

Blanchard, B.S., and W.J. Fabrycky. 2011. *Systems Engineering and Analysis*, 5th ed. Prentice Hall International Series in Industrial and Systems Engineering. Englewood Cliffs, NJ, USA: Prentice Hall.

ISO/IEC/IEEE. 2011. *Systems and software engineering - Architecture description.* Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC)/Institute of Electrical and Electronics Engineers (IEEE), ISO/IEC/IEEE 42010.

Kazman, R., M. Klein, and P. Clements. 2000. *ATAM: Method for Architecture Evaluation*. Pittsburgh, PA, USA: Software Engineering Institute (SEI)/Carnegie Mellon University (CMU). CMU/SEI-2000-TR-004, ESC-TR-2000-004.

Lawson, H. 2010. *A Journey Through the Systems Landscape*. London, UK: College Publications.

Magrab, E., S. Gupta, P. McCluskey, and P. Sandborn. 2010. *Integrated Product and Process Design and Development - The Product Realization Process*. Boca Raton, FL, USA: CRC Press.

Martin, J.N. 1997. *Systems Engineering Guidebook: A process for developing systems and products*, 1st ed. Boca Raton, FL, USA: CRC Press.

Martin, J. 2004. "The Seven Samurai of Systems Engineering: Dealing with the Complexity of 7 Interrelated Systems." Proceedings of the International Council on Systems Engineering (INCOSE) International Symposium, 2004, Toulouse, France.

Phillips, F. 2001. *Market-oriented Technology Management - Innovating for Profit in Entrepreneurial Times*. Berlin, Germany: Springer-Verlag.

Pugh, S. 1990. *Total Design: Integrated Methods for Successful Product Engineering*. Upper Saddle River, NJ, USA: Prentice Hall.

Ring, J. 1998. "A Value Seeking Approach to the Engineering of Systems." Proceedings of the IEEE Conference on Systems, Man, and Cybernetics, 1998, San Diego, CA, USA. p. 2704-2708.

Sage, A., and W. Rouse. (eds.) 1999. *Handbook of Systems Engineering and Management*. Hoboken, NJ, USA: John Wiley and Sons, Inc.

Wasson, C.S. 2006. *System Analysis, Design, and Development*. New York, NY, USA: John Wiley & Sons.

Wheelwright, S., and K. Clark. 1992. *Managing New Product and Process Development: Text and Cases*. Columbus, OH, USA: Free Press.

Primary References

ANSI/EIA. 2003. *Processes for Engineering a System*. Philadelphia, PA, USA: American National Standards Institute (ANSI)/Electronic Industries Association (EIA). ANSI/EIA 632-2003.

Blanchard, B.S., and W.J. Fabrycky. 2011. *Systems Engineering and Analysis*, 5th ed. Prentice Hall International Series in Industrial and Systems Engineering. Englewood Cliffs, NJ, USA: Prentice Hall.

Magrab, E., S. Gupta, P. McCluskey, and P. Sandborn. 2010. *Integrated Product and Process Design and Development - The Product Realization Process*. Boca Raton, FL, USA: CRC Press.

Martin, J. 2004. "The Seven Samurai of Systems Engineering: Dealing with the Complexity of 7 Interrelated Systems." Proceedings of the International Council on Systems Engineering (INCOSE) International Symposium, 2004, Toulouse, France.

Wasson, C.S. 2006. *System Analysis, Design, and Development*. New York, NY, USA: John Wiley & Sons.

Additional References

None.

< Previous Article | Parent Article | Next Article >

SEBoK v. 2.1, released 31 October 2019

Product Systems Engineering Background

Lead Author: *Ricardo Pineda*

Product Types

A system is by definition composed of elements that interact. The system itself usually is an element of a larger system, and you can often also view each element as a system on its own.

A system element consists of one or more products. Products need to be produced or acquired. Some can be acquired or procured as-is, without need for fabrication or modification. Others need to be engineered, and in some cases systems-engineered (Martin 1997). Basic product types are depicted in the figure below.

Types of products are not limited to hardware or software. Many other types of products perform functions necessary to meet stakeholder needs. Some are only relevant to certain industries or domains, such as structures for civil engineering, or ships for shipping or the naval domain. Systems engineers must remember not to allocate the required behavior for a system to hardware and software elements alone.

While we may associate the idea of a product with concrete objects like computer chip, phones, aircraft, or even command, control and communications centers, an organization or a process can also be a product. Sometimes a product is not complex enough to justify performing Product SE, and only needs product design engineering. Enterprise SE and Service SE should determine whether a product needs Product SE.

Product Taxonomy

For any system being developed, the systems engineers must decide what are the right elements to be included. This is not self-evident, because basic product types are not necessarily mutually exclusive. For example, some would consider that facilities contain hardware and people. Others would consider facilities to be separate from hardware and people. Some would include material as part of hardware, while others would not. Creating a taxonomy of product types can help the systems engineer think clearly and thoroughly about what components to include.

Business Objectives and Products

When it develops and launches a new product, an enterprise must align that product with its business goals, internal capabilities, and competition. It must align the end product with the systems expected to realize and sustain it.

The new product concept must be based on analysis that, besides product potential, also explores the ability of the enterprise to exploit that potential, including factors like organizational culture, focus, goals, and processes. Present and future markets and technology must be analyzed. So must several dimensions of competition: competitors' offerings and their plans, for entry into new markets and for product expansion including new functionality, features, or services. These, and the ability of the enterprise to react to them, must also be monitored for the enterprise to remain competitive in the long term.

Accelerating economic globalization since the 1970s has forced enterprises to respond to global needs, not just local or regional ones. Enterprises in the resulting hyper-competitive business environment must analyze their financial goals, their market positions, and the business segments in which they participate, in order to understand what products are required.

This is true for completely new products, and also for product enhancements, penetration of new markets, and growth within existing markets.

Relationship between Product Systems Engineering and Product Development

Product development is the process of bringing a new product to market. Product SE (PSE) considers the complete product system–that is the product in the context of all its enabling elements. PSE takes a full life cycle perspective, "from cradle to grave" or "dust to dust."

Technology-based product development may be thought of as coming from two sources. One, where innovation enhances existing technology, is aimed at relatively short-term market windows. The other involves long-term research to identify the technology developments required to realize the concept. These may be technologies whose availability is not foreseen in the near future, meaning that substantial investment and long lead times may be required before the proof of concept, initial operational capabilities (IOCs) or prototyping stages are reached, let alone the commitment to realize the actual product offering. Some authors claim that the systems engineering process and the new product development (NPD) process for this second source are one and the same.

It is from the second source that strategic initiatives (long-term applied research) realize new products in areas like military aircraft or bioengineering. When research resolves fundamental questions on matters of science or national/regional interest technology breakthroughs occur.

This article concentrates on the first source of technology-based product development, that is, the one driven by ever-evolving market needs to enhance existing technology.

Product Development Patterns

When existing or near-future technology innovations are exploited to generate new product ideas, product development may follow any one of following scenarios (Phillips 2001):

- Product development may use well-established technologies to help the enterprise improve the efficiency of current operations.
- Product development may use well-established technologies to help the enterprise into new kind of operations.
- Product development may use leading edge technologies to improve the efficiency of current operations.
- Product development may use leading edge technologies to help the enterprise into new kinds of operations.

The product itself may simply be a modification of an existing product or its presentation, it may possess new or different characteristics that offer additional benefits to the customer, and/or it may be entirely new and satisfy a newly-defined customer want or market niche (http:/ / www. businessdictionary. com/ definition/ product-development.html [1]).

Existing realization or sustainment systems may not be adequate to develop a given product. For example, it might be necessary to change development practices, use different testing methods or facilities, or upgrade manufacturing equipment and procedures. There might need to be improved customer support procedures and newly trained support personnel, upgraded maintenance facilities and tools, or modified spare parts delivery techniques.

Market Pressures

The product development process must be dynamic, adaptive, and competitive on cost, time-to-market, performance, and quality on a global scale. This is because in the global economy continuous technology innovation and constantly evolving markets and customer needs demand a rapid response.

Products themselves are often multidisciplinary in nature; product development must have close participation, not only from the different specialty engineering fields (mechanical, electrical, industrial, materials, and so on), but also from the finance field to analyze the total cost of development (including production), marketing and sales to understand market behavior and acceptance, manufacturers and distributors, and legal and public relations experts.

All this has mandated enterprises to assess how they create their products and services. The result has been an effort to streamline the development process. One example of this is seen by the deployment of integrated product teams (IPTs) sometimes known as integrated product development teams (IPDTs).

Product Systems Engineering

Product systems engineering strives for the efficient use of company resources in order to achieve business objectives and deliver a quality product. Product systems engineering activities range from concept to analysis to design and determine how conceptual and physical factors can be altered to manufacture the most cost-effective, environmentally friendly product that satisfies customer requirements. Engineering the product system requires an interdisciplinary approach that includes an analysis of the product and its related elements such as manufacturing, maintenance, support, logistics, phase-out, and disposal; these are all activities which belong to either the realization system or the sustainment system. The proper application of systems engineering and analysis ensures the timely and balanced use of human, financial, technological assets, and technology investments to minimize problems, harmonize overall results, and maximize customer satisfaction and company profits.

Products are as diverse as the customers that acquire them and there are no universally accepted methods, processes, and technologies (MPTs) for end-to-end analysis of products and their supporting subsystems. Every product needs to adapt existing MPTs based on prior experiences and best practices, such as Toyota (Hitchens 2007), MITRE (Trudeau 2010), and NASA (NASA SELDP 2011). Product systems engineering helps develop the end-to-end analysis of products and sub-systems by performing the following tasks:

- determining the overall scope of needs for the product system;
- defining product and system requirements;
- considering all interactions between the different elements of the product system;
- organizing and integrating engineering disciplines; and
- establishing a disciplined approach that includes review, evaluation and feedback, and ensures orderly and efficient progress.

Constantly evolving needs and requirements, along with constant technology innovations, may render a committed product development obsolete even before deployment. This has led to debate among systems engineering professionals on the need for the systems engineering process to become more rapidly adaptable. Platform-based solutions to resolve some of these challenges (infrastructure as a service, platform as a service, and software as a service) are being studied and proposed (MITRE 2010; Boehm 2010).

Integrated Product Development Process

The integrated product development process (IPDP) starts with understanding market needs and developing a strategy that creates products that satisfy or exceed customer expectations, respond to evolving customer demands, adapt to changing business environments, and incorporate systems thinking to generate novel ideas and co-create value with extensive stakeholders' participation. IPDP is a continuously evolving process that strives to realize products whose cost, performance, features, and time-to-market help increase company profitability and market share. Magrab, et al. (2010) discussed the IPDP in terms of four different stages; Figure 1 provides a snapshot of an IPDP and the main tasks carried out at each stage.

Stage I: Product Identification

During the product identification stage, the enterprise aims to identify an enterprise-wide strategy that flows down to individual product strategies resulting in a good business investment for the company. During this stage addressable markets for the product are identified in addition to geographical coverage of the product. The developments through this stage result in demonstration of strong customer need, determination of potential markets and geographic scope, the fitness of enterprise core capabilities to the product strategy, business profitability (return on investment, profit & loss), etc.

During this stage an integrated product team (IPT) first develops the IPDP for the project, usually by tailoring a corporate IPDP standard. The IPT assesses required technology innovation, feasibility of existing technologies, estimated time and cost of technology development, and the risks associated with markets, finances and technologies risk, etc. This stage also takes into account inputs from the continuous improvement (CI) process to develop new features, enhancements in existing products to address new market needs or customer demands.

Stage II: Concept Development

The main goal of the Concept Development stage is to generate feasible concepts designs for the potential product and develop MPTs that will satisfy the product's performance goals of economic viability and customer satisfaction. These concept designs must ensure that the company's core competencies can satisfy the requirements to produce the products while keeping into account the market viability, manufacturability, and technical feasibility through an extensive analysis of alternative process.

During this stage SE supports the IPT in identifying different operational scenarios and modes of operation, functional requirements of the products, technology risks and performance risks, and the main components of the products and required interfaces among them, etc. This stage involves a highly interactive and iterative exchange of concepts among several IPTs and depending on complexity of the products, a Systems Engineering Integration Team may be required to ensure analysis of all the possible solutions. During this stage inputs from the CI process helps analyze new technologies/processes including upgrades to existing technologies, and create products that results in enhanced customer experiences.

Stage III: Design and Manufacturing

During the design and manufacturing stage the actual product is realized and manufactured. This stage starts with creating engineering drawings for the product, product configuration items specs, "design for X" (DFX), manufacturing design plans, production plans and schedules, test production run to ensure that product meets customer requirements and quality criteria, and a plan for full production, logistics and distribution.

During this stage the product design & manufacturing engineering team works closely with operations managers to create MPTs to manage the technical effort for the product from an end-to-end perspective. Some of the SE activities during this stage include product integration, verification and validation plans; modeling, simulation, test & evaluation of the product system under critical scenarios; launch readiness plans including end-user test plans, operational readiness, etc. During this stage MPTs are developed and documented for proper handling of defective parts, processes, or functionalities. The CI process inputs include product and process performance enhancements and sustained life-cycle operations support.

Stage IV: Product Launch

During the product launch stage the product is delivered to its potential markets. During production and deployment, MPTs are developed to ensure that the product meets its quality goals, satisfies customer requirements, and realizes the business plan goals. This requires provisions for customer care, logistics, maintenance, training etc., and a CI process to monitor product and product system technical performance and product quality. The CI process is realized through extensive data collection using customer satisfaction surveys and remotely or manually observing, recording, and analyzing process performance metrics, technical performance measure, quality metrics, etc.

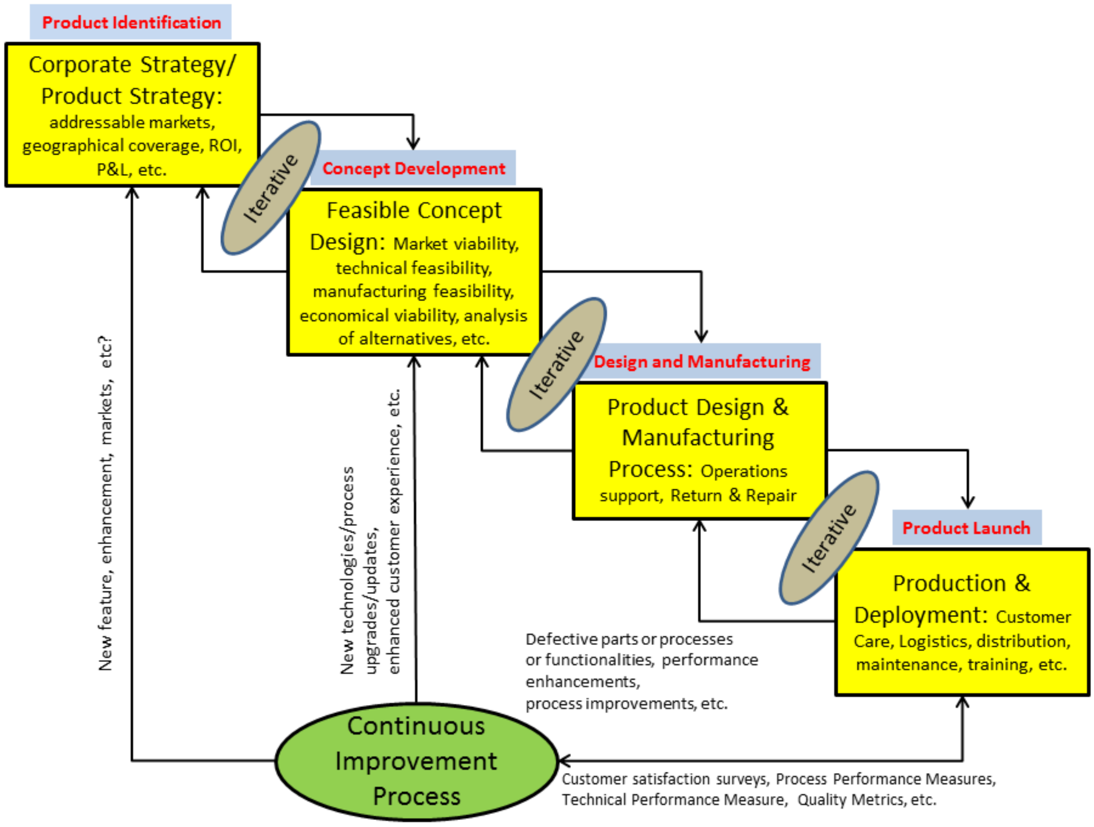


Figure 1. Integrated Product Development Process. (SEBoK Original)

Relationship between Product Systems Engineering and Technology Development

As technological advancement accelerates, product life cycles become shorter, especially for high technology products. As a result, enterprises risk having outdated or obsolete products that have lost pace with markets trends, technology trends, or customer expectations.

Product systems engineering should bring awareness of technology changes and trends to the analysis of new product ideas or innovations. This affects the time and cost inputs into the technical feasibility analysis of the product. The result should include a road map of required technology developments, which is then used to create the overall road map for the new product offering.

In these cases, new product ideas impose requirements on new technology developments.

On the other hand, when technology developments or breakthroughs drive product innovation or the generation of new markets, the technology developments may also generate requirements on product features and functionalities. Factors which dictate decisions about introducing products include the technology readiness levels (TRL), the integration readiness levels (IRL), the manufacturing readiness levels (MRL), the system readiness levels (SRL), and the operational readiness of the enterprise to launch the product system. See the "Readiness Levels" section in the Product Systems Engineering Special Activities article.

Understanding the entities (i.e., components or elements) that compose the product is not a trivial task for systems engineers. It is not unusual for a new product to require developments in several technologies, including new materials, electronic components, software, maintenance and repair procedures, processes, or organizational structures. All of these developments must be factored into the IPDP for the successful deployment and proper use of the product.

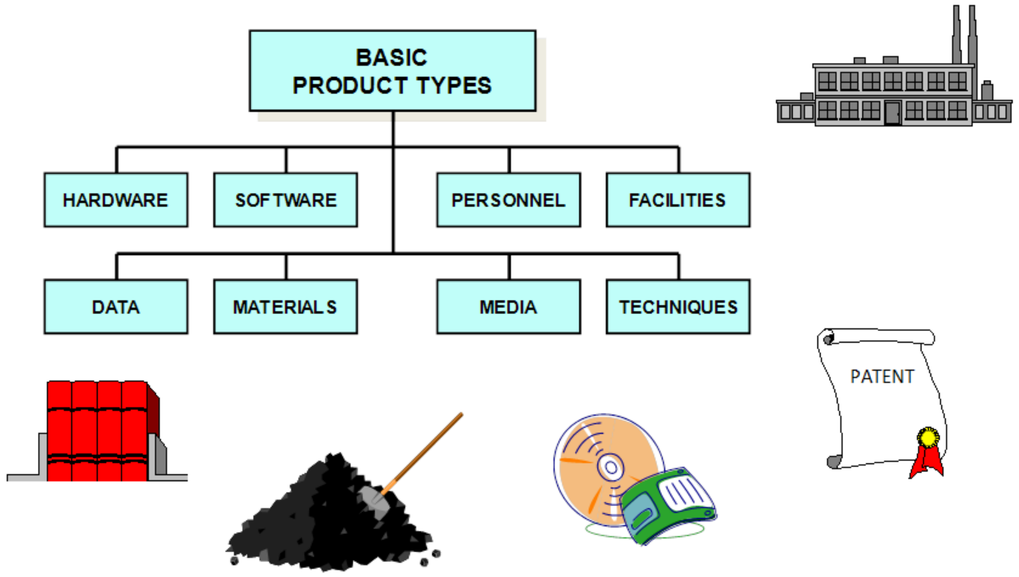


Figure 2. Basic Product Types that Constitute a Product System (Martin 1997). This material is reproduced with permission of John Wiley & Sons, Inc. All other rights are reserved by the copyright owner.

Product Type Examples

Examples of each product type are shown below (Martin 1997).

Table 1. Product Types (Martin 1997). This material is reproduced with permission of John Wiley & Sons, Inc.

Type	Examples
Hardware	Computer processor unit, radar transmitter, satellite payload, telephone, diesel engine, data storage device, network router, airplane landing gear
Software	Computer operating system, firmware, satellite control algorithm, robot control code, telephone switching software, database application
Personnel	Astronaut, computer operator, clerk, business executive, Laika (the cosmonaut dog), bus driver, cashier, maintenance technician
Facilities	Space rocket launch pad, warehouse building, shipping docks, airport runway, railroad tracks, conference room, traffic tunnel, bridge, local area network cables
Data	Personnel records, satellite telemetry data, command and control instructions, customer satisfaction scores
Materials	Graphite composite, paper, gold, concrete, stone, fiberglass, radar absorption material, cladded metals, integrated circuit substrate, magnetic memory core
Media	Data storage media (tape, disc, memory card), signal transport media (twisted pair wire, fiber optic cable, RF spectrum), communications media (television, radio, magazines), social media (blogs, Twitter, Facebook)
Techniques	Soldering, trouble trick response process, change notice handling, telephone answering protocol, project scheduling, data sorting algorithm

Materials could be thought of as basic raw materials, like steel, or as complex materials, like cladded metals, graphite composites, or building aggregate material. Personnel are not normally thought of as a “product,” but that can change depending on the type of system in question. The National Aeronautics and Space Administration (NASA) space program “system” certainly produces astronauts. When personnel are considered system(s), it is not usually possible to simply find and hire personnel with the requisite knowledge, skills, and experience. These personnel “products” can often be developed using a product SE approach (Martin 1996). For example, you could specify requirements (i.e., required knowledge, skills, and experience) for each person that is part of the system. Interfaces can be specified for each person, and an assessment can be made as to the maturity of each person (i.e., each potential product). These are a few examples of how product SE can be applied to personnel products.

In enterprise systems engineering, we may need education and training systems to make up a part of our personnel system in order to produce people with the right competencies and capabilities.

References

Works Cited

Academy of Program/Project and Engineering Leadership (APPEL). 2009. *NASA's Systems Engineering Competencies*. Washington, DC: US National Aeronautics and Space Association. Available at: http://www.nasa.gov/offices/oce/appel/pm-development/pm_se_competency_framework.html [2].

Blanchard, B.S., and W.J. Fabrycky. 2011. *Systems Engineering and Analysis*, 5th ed. Prentice Hall International Series in Industrial and Systems Engineering. Englewood Cliffs, NJ, USA: Prentice Hall.

Boehm, B. 2010. *Systems 2020 Strategic Initiative*. Hoboken, NJ, USA: Systems Engineering Research Center (SERC), SERC-2010-TR-009.

Grady, J. 2010. *Systems Synthesis - Product and Process Design*. Boca Raton, FL, USA: CRC Press.

INCOSE. 2012. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, version 3.2.2. San Diego, CA, USA: International Council on Systems Engineering (INCOSE), INCOSE-TP-2003-002-03.2.2.

Magrab, E., S. Gupta, P. McCluskey, and P. Sandborn. 2010. *Integrated Product and Process Design and Development - The Product Realization Process*. Boca Raton, FL, USA: CRC Press.

Martin, J.N. 1997. *Systems Engineering Guidebook: A process for developing systems and products*, 1st ed. Boca Raton, FL, USA: CRC Press.

MITRE. 2010. *Platform as a Service: A 2010 Marketplace Analysis*, Cloud Computing Series. Bedford, MA, USA: Systems Engineering at MITRE.

Morse, L., and D. Babcock. 2007. *Managing Engineering and Technology*, 4th ed. Upper Saddle River, NJ, USA: Prentice Hall.

Phillips, F. 2001. *Market Oriented Technology Management: Innovating for Profit in Entrepreneurial Times*. New York, NY, USA: Springer.

Trudeau, P.N. 2010. *Designing and Enhancing a Systems Engineering Training and Development Program*. Bedford, MA, USA: The MITRE Corporation.

Wasson, C.S. 2006. *System Analysis, Design, and Development*. New York, NY, USA: John Wiley & Sons.

Primary References

Grady, J. 2010. *Systems Synthesis - Product and Process Design*. Boca Raton, FL, USA: CRC Press.

Magrab, E., S. Gupta, P. McCluskey, and P. Sandborn. 2010. *Integrated Product and Process Design and Development - The Product Realization Process*. Boca Raton, FL, USA: CRC Press.

Martin, J.N. 1997. *Systems Engineering Guidebook: A process for developing systems and product*, 1st ed. Boca Raton, FL, USA: CRC Press.

Additional References

Academy of Program/Project and Engineering Leadership (APPEL). 2009. *NASA's Systems Engineering Competencies*. Washington, DC: US National Aeronautics and Space Association. Available at: http://www.nasa.gov/offices/oce/appel/pm-development/pm_se_competency_framework.html [2].

Blanchard, B.S., and W.J. Fabrycky. 2011. *Systems Engineering and Analysis*, 5th ed. Prentice Hall International Series in Industrial and Systems Engineering. Englewood Cliffs, NJ, USA: Prentice Hall.

Boehm, B. 2010. *Systems 2020 Strategic Initiative*. Hoboken, NJ, USA: Systems Engineering Research Center (SERC), SERC-2010-TR-009.

INCOSE. 2012. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, version 3.2.2. San Diego, CA, USA: International Council on Systems Engineering (INCOSE), INCOSE-TP-2003-002-03.2.2.

MITRE. 2010. *Platform as a Service: A 2010 Marketplace Analysis*, Cloud Computing Series. Bedford, MA, USA: Systems Engineering at MITRE.

Morse, L., and D. Babcock. 2007. *Managing Engineering and Technology*, 4th ed. Upper Saddle River, NJ, USA: Prentice Hall.

Phillips, F. 2001. *Market Oriented Technology Management: Innovating for Profit in Entrepreneurial Times*. New York, NY, USA: Springer.

Trudeau, P.N. 2010. *Designing and Enhancing a Systems Engineering Training and Development Program*. Bedford, MA, USA: The MITRE Corporation.

Wasson, C.S. 2006. *System Analysis, Design, and Development*. New York, NY, USA: John Wiley & Sons.

< Previous Article | Parent Article | Next Article >

SEBoK v. 2.1, released 31 October 2019

References

[1] http://www.businessdictionary.com/definition/product-development.html
[2] http://www.nasa.gov/offices/oce/appel/pm-development/pm_se_competency_framework.html

Product as a System Fundamentals

Lead Author: *Ricardo Pineda*

This article introduces fundamental concepts of product systems.

Product Elements and Connections

Product systems consist of product elements and two kinds of connections: connections among elements, and connections between elements and things in the system environment. That portion of the environment that can be influenced by the system or that can influence the system is called the “context.”

Connections between elements contain interactions and relationships (Hybertson 2009). A connection is more than a mere interface.

Interactions occur across *interfaces* between the elements inside or outside the system, and can be defined as exchanges of data, materials, forces, or energy. Connections with an interactive nature can be represented in various engineering artifacts: schematic block diagrams, data flow diagrams, free body diagrams, interface control diagrams, port specifications, energy transfer diagrams, and so on. Product systems engineering (PSE) usually defines interactions in an interface control document, interface design document, interface requirements document, or the equivalent.

Connections also encompass relationships between elements. These relationships can be spatial, motion-related, temporal, or social.

Spatial relationships:

- one element is underneath another
- two elements are *x units* apart
- one element is inside another

Motion-related relationships:

- the relative velocity of two elements is *v units*
- the relative acceleration between two elements is *a units*

Temporal relationships:

- one element exists before another
- two elements must exist at the same time
- two elements must be separated in time by *t units*

Social relationships:

- a human element feels a particular way about a system
- a human element owns another (non-human) element
- a human element understands the operation of a system in a particular way

Relationships that are not about time can still change over time. For example, an element that is inside another element during one mode of operation can be outside of it during a different mode of operation. Therefore, one should not assume that non-temporal relationships are necessarily static in time.

Relationships can be represented in engineering artifacts, including the timing diagram, timeline diagram, mission reference profile, capability road map, and project schedule chart.

Social relationships include the implicit or explicit social obligations or expectations between the roles that human elements play in a system. These roles may be assigned different authorities, responsibilities, and accountabilities. See the discussion on organization behavior in the article Team Dynamics. Organizational behavior theories and human factors may need to be considered when engineering such a product system.

There can also be social relationships between the humans and the non-human elements of the system. This may involve how the human “feels” about things in the system or perhaps even the system as a whole. Humans inside or outside the system-of-interest may have different degrees of “understanding” with respect to how the system operates, its limitations and capabilities, and the best way to operate it safely and effectively. The “ownership” relationship can be important in determining things like who can operate or change some configuration or mode of the system.

There are many such social relationships in a product system that are often ignored or misunderstood when performing PSE. Social relationships can affect the overall performance or behavior of a product system to the point of determining its success or failure.

Core Product and its Enabling Products & Operational Services

A variety of systems (themselves being products or services) enable the development, delivery, operation and eventual disposal of a product, as shown in Figure 1. The concept of enabling systems is defined in the ISO/IEC 15288 standard (2015).

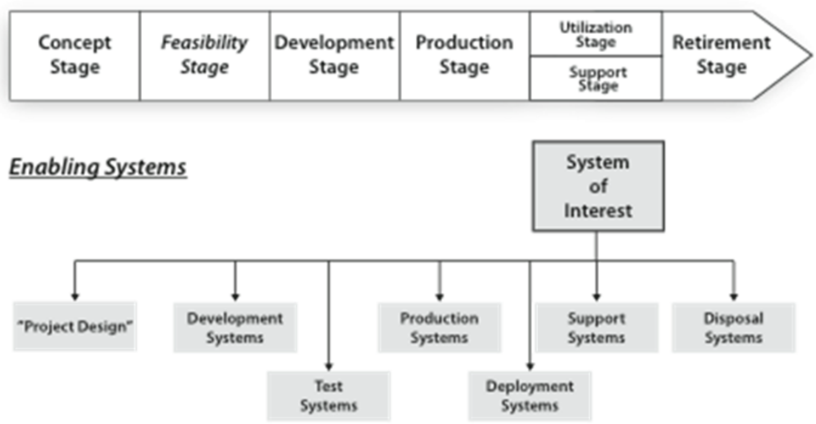


Figure 1. Example of Enabling Systems (Lawson 2010). Reprinted with permission of Harold "Bud" Lawson. All other rights are reserved by the copyright owner.

In the figure, the system-of-interest (SoI) goes into operation as a delivered product or offered service in the utilization stage while maintenance and logistics are provided (by a product sustainment system) simultaneously in the support stage. These two stages are commonly executed in parallel, and they offer opportunities to observe any need for changes in the properties of the product or service or how it is operated and supported. Making changes iterates the life cycle and results in new or improved products or features.

The delivered product and its enabling systems collectively form a wider system-of-interest (WSOI). The project design enabling system is an enterprise based system asset that establishes the strategy and means of organizing the projects to be executed along the life cycle. In many larger organizations, this type of enabling system is institutionalized and can be based upon recommendations of the Project Management Institute (PMI).

Product systems should be viewed as enabling service systems. That is, once deployed, a product system provides a service that contributes to an enterprise's operations. To the acquirer, the SoI provides operational services to users. This is true at several levels:

- Hardware and software products are used to enable the provisioning of service systems,
- Enterprises incorporate products as system assets and use them to enable operations, and
- Provided products are integrated into the system of systems.

Product Architecture, Modeling, and Analysis

IEEE standard 1471-2000 defines architecture as "the fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution" (IEEE 2000). Similarly, ISO/IEC 42010-2011 defines architecture as "fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution" (ISO/IEC 2011).

A product's purpose (stakeholder's need) is realized by a product system (the SoI). Because product systems are composed of different entities (components, assemblies, subsystems, information, facilities, processes, organizations, people, etc.) that together produce the results unachievable by any of the entities alone, architecting the product is based on a whole systems approach. To architect with a whole systems approach means to define, document, design, develop, manufacture, distribute, maintain, improve, and to certify proper implementation of the product's objectives in terms of the functional (the "what"), the behavioral (the use, or intended operations), the logical (interaction and relationships between entities) and the physical constructs (Wasson 2006; Maier 2009; Blanchard and Fabrycky 2011).

The system architect starts at the highest level of abstraction, concentrating on needs, functions, systems characteristics and constraints (concerns) before identifying components, assemblies, or subsystems. This is the systems view, and it is used to represent the stakeholder's market service description or the concept of operations (understanding of the opportunity/problem space).

Next to be documented, as needs become better understood, are architectural descriptions at different levels of abstraction, representing various stakeholders interests. These are the architecture models. They define the possible solution spaces for the product purpose in the form of detailed system, operational, behavioral, and physical requirements of the product system.

Different modeling techniques are then used to analyze different types of requirements. For operational scenarios and different modes of operation, there are hierarchical decomposition and allocation, architectural block diagrams (ABD), functional block diagrams (FBD), functional flow block diagrams (FFBD), and use case diagrams. For interactions and relationships among hardware and/or software components there are sequence diagrams, activity diagrams, state diagrams, and data flow diagrams. See (Maier 2009) Chapter 8 for an introduction to models and modeling.

Analysis of the solution space makes it possible to produce detailed technical specs, engineering drawings, blueprints, software architectures, information flows, and so on, that describe the entities in the product system. An entity's requirements bound its attributes and characteristics, levels of performance, operational capabilities, and design constraints. During design and integration, entity characteristics can be traced back to requirements (requirements traceability being a key aspect of SE). Verification and validation plans created during the requirements phase are the basis of testing certification that the product does what it was intended to do.

Overall, what occurs is the transformation of a set of requirements into products and processes that satisfy the stakeholder's need. The architecture is represented by a set of models that communicate an integrated view of the product's intent and purpose, and the interactions and interfaces required among all the different participating entities. The product's purpose is articulated in terms of business objectives (market, cost, functionality, performance, and time to deliver). The set of models includes sufficient variety to convey information appropriately to the stakeholders, designers/developers, specialty engineering, operations, manufacturers, management, and marketing and sales personnel.

Different architecture frameworks have been developed to guide product teams in defining the product architecture for commercial and for public enterprises. In general, an architecture framework describes a "view," meaning a "collection of models that represent the whole system from the perspective of a set of related stakeholder concerns." Prime examples of architecture frameworks are the Zachman framework (Zachman 1992), The Open Group Architecture Framework (TOGAF) (TOGAF 2011), the Enhanced-Telecom Operations Map (e-TOM), just to mention a few in the commercial sector. In the case of public enterprises a few architecture frameworks include the Department of Defense Architecture Framework (DODAF 2.0) (DoD 2010), the Federal Enterprise Architecture Framework (FEAF) (FEA 2001), the British Ministry of Defense Architecture Framework (MODAF) (MOD 2004), etc.

Differences between acquired products and offered products play an important role in defining product system requirements. Acquired products are life cycle-managed directly by the acquirer; for instance, acquired defense systems are defined, developed, tested, owned, operated, maintained and upgraded by the defense agency. See the article Product Systems Engineering Key Aspects within this KA.

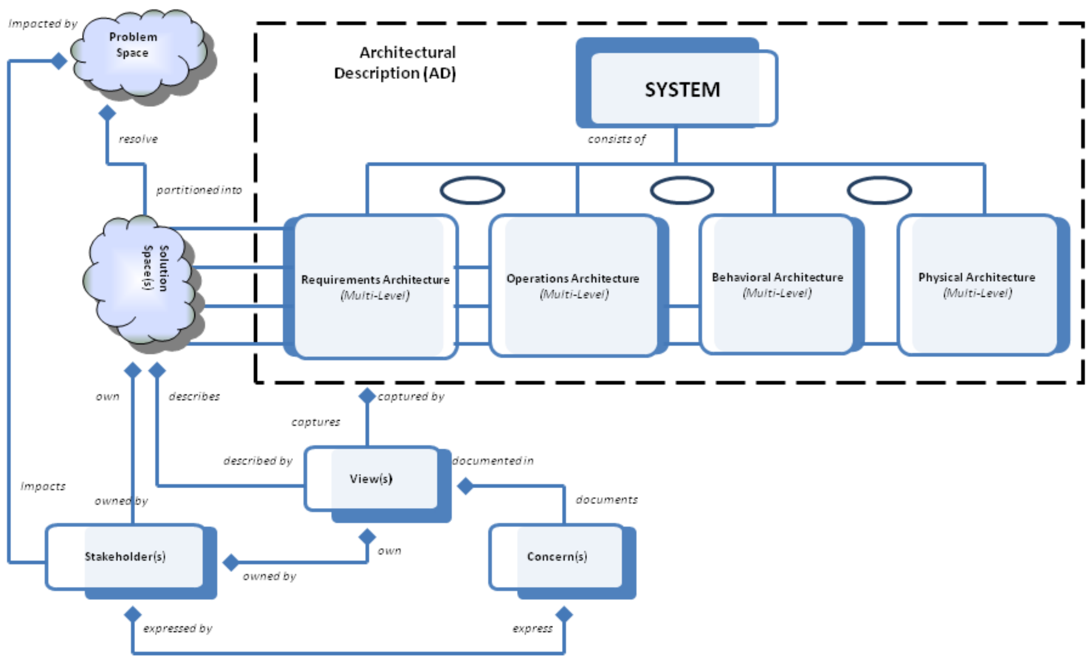


Figure 2. System Architectural Description Elements (Adapted from Wasson 2006). Reprinted with permission of John Wiley & Sons Inc. All other rights are reserved by the copyright owner.

Specialty Engineering Integration

The INCOSE *SE Handbook* defines specialty engineering as:

"Analysis of specific features of a system that requires special skills to identify requirements and assess their impact on the system life cycle."

Areas of expertise that fall under this umbrella definition include logistics support, electromagnetic compatibility analysis, environmental impact, human factors, safety and health analysis, and training. The unique characteristics, requirements, and design challenges of a system-of-interest all help determine the areas of specialty that apply.

A number of specialty engineering areas are typically important to systems engineers working on the development, deployment, and sustainment of product systems. For example, logistics support is essential for fielded product systems that require maintenance and repair. The delivery of services, materials, parts, and software necessary for supporting the system must all be considered very early in the development activity. These factors should usually be considered before the system requirements and concept definition are complete. To integrate these specialty disciplines sufficiently early on, the systems engineer needs to know what specialties relate to the system under development, how they relate to the systems engineering process, and how to integrate them into the life cycle process.

For product systems with significant hardware content and that operate in challenging environments, the following specialty engineering areas must usually be considered:

- manufacturability,
- reliability and maintainability,
- certification (essential where human safety is an issue),
- logistics support,
- electromagnetic compatibility (if they radiate),
- environmental impact,
- human factors,
- safety and health, and
- training.

The relationship of these specialty areas to the systems engineering process must be understood and considered. The key aspects of the relationship are:

- when the specialty needs to be considered,
- what essential data or information it provides,
- the consequences of not including the specialty in the systems engineering process, and
- how the systems engineers should interact with the specialty engineers.

Grady (2006) provides an overview, with references, for many of the specialty engineering disciplines, including reliability engineering; parts, materials, and process engineering (PMP); maintainability engineering, availability, producibility engineering, design to cost/life cycle cost (DTC/LCC), human factors engineering, corrosion prevention and control (CPC), system safety engineering, electromagnetic compatibility (EMC) engineering, system security engineering, mass properties engineering, and environmental impact engineering.

Eisner (2008) lists specialty engineering as one of the "thirty elements" of systems engineering. "Specialty engineering refers to a set of engineering topics that have to be explored on some, but not all, systems engineering efforts. In other words, some systems involve these special disciplines and some do not. Examples of specialty engineering areas include electromagnetic compatibility and interference, safety, physical security, computer security, communications security, demand forecasting, object-oriented design, and value engineering." Some of what we consider specialty engineering in the present article, Eisner includes among his "thirty elements" of systems engineering, but not as part of the specialty engineering element.

There is no standard list of specialty engineering disciplines. What is considered specialty engineering varies according to the community to which the systems engineering belongs, and sometimes to the preferences of the customer.

References

Works Cited

ANSI/IEEE. 2000. *Recommended practice for architectural description for software-intensive systems*. New York, NY, USA: American National Standards Institute (ANSI)/Institute of Electrical and Electronics Engineers (IEEE), ANSI/IEEE 1471-2000.

Blanchard, B.S., and W.J. Fabrycky. 2011. *Systems Engineering and Analysis*, 5th ed. Prentice Hall International Series in Industrial and Systems Engineering. Englewood Cliffs, NJ, USA: Prentice Hall.

Eisner, H. 2008. "Chapter 7. Essentials of Project and Systems Engineering Management," in *The Thirty Elements of Systems Engineering*, 3rd ed. New York, NY, USA: John Wiley & Sons.

Grady, J. 2006. "Chapter 3.7. System Requirements Analysis," in *Specialty Engineering Requirements Analysis*. New York, NY, USA: Academic Press.

Grady, J. 2006. *System Requirements Analysis*. New York, NY, USA: Academic Press.

Grady, J. 2010. *Systems Synthesis - Product and Process Design*. Boca Raton, FL, USA: CRC Press.

Hybertson, D. 2009. *Model-oriented Systems Engineering Science: A Unifying Framework for Traditional and Complex Systems*. Boston, MA, USA: Auerbach Publications.

INCOSE. 2012. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, version 3.2.2. San Diego, CA, USA: International Council on Systems Engineering (INCOSE), INCOSE-TP-2003-002-03.2.2.

ISO/IEC/IEEE. 2015.*Systems and software engineering - system life cycle processes*.Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC), Institute of Electrical and Electronics Engineers.ISO/IEC 15288:2015.

ISO/IEC/IEEE. 2011. *Systems and software engineering - Architecture description*. Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC)/Institute of Electrical and Electronics Engineers (IEEE), ISO/IEC/IEEE 42010.

Lawson, H. 2010. *A Journey Through the Systems Landscape*. London, UK: College Publications.

Maier, M., and E. Rechtin. 2009. *The Art of Systems Architecting*, 3rd ed. Boca Raton, FL, USA: CRC Press.

MOD. 2004. *Ministry of Defence Architecture Framework (MODAF)*, version 2. London, UK: UK Ministry of Defence.

The Open Group. 2011. *TOGAF*, version 9.1. Hogeweg, The Netherlands: Van Haren Publishing. Accessed August 29, 2012. Available at: https:/ / www2. opengroup. org/ ogsys/ jsp/ publications/ PublicationDetails. jsp?catalogno=g116.

Wasson, C.S. 2006. *System Analysis, Design, and Development*. Hoboken, NJ, USA: John Wiley & Sons.

Zachman, J.A. 1992. "Extending and Formalizing the Framework for Information Systems Architecture." *IBM Systems Journal*. 31 (3): 590-616.

Primary References

Eisner, H. 2008. "Chapter 7. Essentials of Project and Systems Engineering Management," in *The Thirty Elements of Systems Engineering*, 3rd ed. New York, NY, USA: John Wiley & Sons.

Lawson, H. 2010. *A Journey Through the Systems Landscape*. London, UK: College Publications.

Wasson, C.S. 2006. *System Analysis, Design, and Development*. Hoboken, NJ, USA: John Wiley & Sons.

Additional References

ANSI/IEEE. 2000. *Recommended practice for architectural description for software-intensive systems.* New York, NY, USA: American National Standards Institute (ANSI)/Institute of Electrical and Electronics Engineers (IEEE), ANSI/IEEE 1471-2000.

Blanchard, B.S., and W.J. Fabrycky. 2011. *Systems Engineering and Analysis*, 5th ed. Prentice Hall International Series in Industrial and Systems Engineering. Englewood Cliffs, NJ, USA: Prentice Hall.

Grady, J. 2006. "Chapter 3.7. System Requirements Analysis," in *Specialty Engineering Requirements Analysis*. New York, NY, USA: Academic Press.

Grady, J. 2006. *System Requirements Analysis*. New York, NY: Academic Press.

Grady, J. 2010. *Systems Synthesis- Product and Process Design*. Boca Raton, FL, USA: CRC Press.

Hybertson, D. 2009. *Model-oriented Systems Engineering Science: A Unifying Framework for Traditional and Complex Systems*. Boston, MA, USA: Auerbach Publications.

INCOSE. 2012. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, version 3.2.2. San Diego, CA, USA: International Council on Systems Engineering (INCOSE), INCOSE-TP-2003-002-03.2.2.

ISO/IEC. 2008. *Systems and Software Engineering -- System Life Cycle Processes*. Geneva, Switzerland: International Organisation for Standardisation / International Electrotechnical Commissions. ISO/IEC/IEEE 15288:2008.

ISO/IEC/IEEE. 2011. *Systems and software engineering - Architecture description*. Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC)/Institute of Electrical and Electronics Engineers (IEEE), ISO/IEC/IEEE 42010.

Maier, M., and E. Rechtin. 2009. *The Art of Systems Architecting*, 3rd ed. Boca Raton, FL, USA: CRC Press.

Zachman, J. 2008. "John Zachman's Concise Definition of The Zachman Framework™." Zachman International Enterprise Architecture. Accessed August 29, 2012. Available at: http:/ / www. zachman. com/ about-the-zachman-framework.

< Previous Article | Parent Article | Next Article >

SEBoK v. 2.1, released 31 October 2019

Business Activities Related to Product Systems Engineering

Lead Author: *Ricardo Pineda*

This topic discusses the interfaces between product systems engineering and other 'back office' and management activities that take place in an enterprise.

Marketing, Product Life Cycle Management, & Quality Management

Product systems engineering (PSE) includes critical and robust interfaces with related business activities, such as marketing, product life cycle management (PLM), and quality. Traditionally, PLM has been a critical stage in the integrated product development process (IPDP) and continues to be an important tool for life cycle management after product deployment. PLM provides an important component of the PSE end-to-end view. The other component is the “breadth” component that captures everything relevant to the system at each life cycle stage. Recently, the focus has started to shift from the idea of managing just the life of the product, to an expanded view that includes the management of product-lines (families) or product platforms themselves. This provides an increase in sustainability, flexibility, reduced development times, and important reductions in costs as new or enhanced products are not launched from scratch every time.

PSE also includes interfaces with the marketing function; in particular, PSE works closely with the business and market development organizations to elicit product needs and intended operations in target markets to define product roll-out and possible phases of product introduction. Analysis of the market is critical during the entire product life cycle from conception through retirement with the understanding that each life cycle phase requires very different marketing approaches. During concept development, marketing has to help determine the potential market, the addressable market segments, define products, and product/innovations requirements for those markets. During the product introduction stage, marketing has to create demand and prompt early customers to try the product. During the growth and maturity phases, marketing has to drive public awareness, develop the brand, and differentiate the product and its features and feature releases to compete with new market entrants. During saturation, marketing must help manage diminishing volumes and revenues as focus shifts from top line (increased market share) to bottom line (increased production and distribution efficiencies) considerations. See the article on Procurement and Acquisition.

The links between PSE and quality are just as critical. The relationships between PSE and quality also reflect the broad view which includes the product and opportunity, but also the company’s internal goals, processes, and capabilities. Quality schemes which focus on a tangible product have been extensively used historically. More recent approaches that acknowledge and match PSE's holistic view have come into use. Issued during 1988, ISO 9000 is a family of standards which focuses on processes and the organization instead of the product itself. In addition, it calls out specific requirements for the design of products and services. ISO 9001 has served as a “platform” for many other schemes which are tailored to specific domains. A collaborative effort of the International Aerospace Quality Group, AS9100 contains all of the base requirements of ISO 9100 and expands further requirements which are critical and specific to the aviation, space, and defense industries. Similarly QS-16949 is a technical standard based on ISO 9001 but expanded to meet specific requirements in the worldwide automotive industry. PSE should play an important role in the design and implementation of any quality management system. See the article on Quality Management.

Capability Maturity Model Integrated (CMMI) for Development is a process improvement approach whose goal is to help organizations improve their performance. CMMI can be used to guide process improvement across a project, a division, or an entire organization. Although initially used in software engineering and organizational development, CMMI use is spreading to other domains since it provides organizations with the essential elements for effective process improvement. According to the Carnegie Mellon Software Engineering Institute, CMMI describe "an evolutionary improvement path from ad hoc, immature processes to disciplined, mature processes with improved quality and effectiveness." (SEI 2010).

Project Management & Business Development

The end-to-end view mandated by PSE requires strong relationships with project management and business development activities. The ‘concurrent’ thinking encouraged by PSE necessarily requires multiple projects to move forward in parallel, but with a high level of coordination. In this sense, PSE and project management (see Systems Engineering and Project Management) are two heavily intertwined disciplines which have been shown to generate synergy and added value for the stakeholders.

The systems engineering management plan (SEMP) is the key document covering the activities, milestones, organization, and resource requirements necessary to ensure the system-of-interest accomplishes its intended goals and mission. A key objective of the SEMP, which is usually developed during the conceptual design phase, is to provide the structure, policies, and procedures to foster the optimum management of projects and activities needed for system development and implementation (Blanchard and Fabrycky 2011).

An effective and agile PSE function can make important contributions to business development for an enterprise or company. The primary goal of business development activities is to identify new types of business/product/services which are believed to address existing or potential needs and gaps (new markets), to attract new customers to existing offerings, and to break into existing markets. PSE’s end-to-end view of the life cycle can support market development by intelligence gathering, feedback on market acceptance or rejection, strategic analysis, and proposition development and campaign development. Finally, PSE should encourage the consideration of several factors within the new product development which may enhance market development. For example, in well-established companies, business development can often include setting up strategic alliances with other, third-party companies. In these instances, the companies may leverage each others expertise and/or intellectual property to improve the probability for identifying, researching, and bringing to market new businesses and new products. See (Sørensen 2012).

Supply Chain Management & Distribution Channel Management

PSE provides the following information to the supply chain management function in an enterprise:

- product specifications (including intended uses of the product),
- product acceptance criteria (for accepting delivery of the product from the supplier),
- product testing and qualification plans and procedures, including which ones are responsibility of the supplier and which ones are responsibility of the acquirer,
- interface specifications associated with each product,
- supplier certification criteria (including a list of pre-certified suppliers), and
- feedback on quality of products delivered by suppliers.

Supply chain management will, as necessary, manage the identification and certification of qualified suppliers with the concurrence of, and coordination with, systems engineering and product engineers.

PSE provides the following information to the distribution channel management function in the enterprise:

- product specifications (including intended uses of the product),
- product user manuals (including installation and maintenance documentation),
- product packaging (for safe delivery of product and for display in retail channels),
- product qualification data (to prove that product meets its design requirements),
- product certification data (to prove product is certified for safe and secure operation),
- user support instructions, and
- operator certification criteria.

Distribution channel management will, as necessary, manage the identification and certification of qualified distributors with the concurrence of, and coordination with, systems engineering and product engineers.

Capability Management & Operations Management

Capability is defined in various ways, but each definition is consistent with the notion of "the ability to do something useful." Products and services are acquired by end users to enable and improve their operational capability to let them do something useful, whether in a military context (e.g., weapon systems improve the capability to conduct effective military operations), or a social context (e.g., a car may improve the ability to satisfy the transport needs of a family). Users acquire products (e.g., military equipment, cars, “productized” service offerings from airlines and taxi companies, etc.) to contribute to satisfying their capability needs.

Capability management involves identifying and quantifying capabilities (existing, new, or modified) that will be needed to meet enterprise goals, as well as selecting a coherent set of product and services across all components of capability that will be integrated to provide the needed capabilities. So normally, requirements for "product systems" are derived from capability management. Capability management is likely to include trade-off processes to make the best use of existing products or low-risk evolutions of them, and conversely identifying when a capability need can only be satisfactorily met by a new-generation product, or even a new type of product altogether. In some cases, new offered products or disruptive technologies (e.g., jet engine, nuclear weapons, and internet) create opportunities for new or improved capabilities, in which case capability management focuses on ensuring that all needed components of capability are put in place to exploit the opportunity provided by the new product or technology. See Capability Engineering.

Operations management uses an integrated set of product systems to deliver value to the enterprise and its stakeholders. Operations management involves bringing new product systems into operation, normally while maintaining business continuity, so transition plans and relevant metrics are critical; next, operations management addresses some of the following: working up to full operational efficiency across all components of capability, coping with incidents, contingency plans to deal with major disruptions, adjusting the system to cope with new ways of working and to deliver new services to meet new enterprise requirements and accommodate new product systems entering service, and eventually planning transitions out of service or major in-service upgrades. PSE supports operations management by defining all dependencies for successful operation on other systems and services, and by providing ongoing engineering support for spares and repairs, obsolescence management, and system upgrades. Systems engineering in the in-service phase has been analyzed (Elliott et al. 2008) and is best viewed as the same basic systems engineering process conducted at a much higher tempo (Kemp and Linton, 2008) and requiring detailed understanding of constraints imposed by the current environment and usage. Configuration management and configuration status accounting during operation is very important for high value and high integrity systems to ensure that any changes are designed to fit the "as-is" system, which may be significantly different from the "as-originally intended" specification and design.

Product Engineering, Assembly, Integration, & Test

Product engineering typically results in an engineering model that is used as the “blueprint” for assembling, integrating, and testing (AIT) a product system. These AIT activities may be performed on prototype versions, as well as final production versions to be delivered to end users. There is significant experience in domain specific industries in performing AIT for complex products. Unfortunately, very little is written in the general literature. Wasson (2006) and de Jong (2008) cover some of these aspects. See also System Integration and System Verification.

For software products, the collection of code modules are integrated via some form of integration program (typically called “make”). The integrated modules are then subjected to tests to exercise the various potential paths through the software. Since software can be easily changed, it is common to use some form of regression testing based upon test suites in order to verify software correctness. Another common means of testing is by fault injection as described by Voas and McGraw (1998).

Manufacturing, Test, & Certification

Systems engineers usually work with manufacturing indirectly through the electrical and mechanical design teams. There are times in the development cycle when a direct interface and working relationship between systems engineering and manufacturing is appropriate and can improve the probability of program and system success. Early in the program the system concept must be examined to determine if it is manufacturable. The requirements and the concept design should be reviewed with the manufacturing engineers to obtain an assessment of the risks associated with the production of the system. If substantial risks are identified, then actions that improve the manufacturing capabilities of the organization, modify the design, and perhaps change the requirements may be needed to reduce the identified risks to acceptable levels. Manufacturing prototypes or proof of manufacture (POM) units may be necessary to reduce the risk and to demonstrate readiness to proceed with the design and the system development. Similarly, the systems engineers must establish that the system will be testable early in the product development phase. The requirements should be mapped to verification methods of inspection, analysis, demonstration, and test before they are released to the design team. All requirements mapped to test must be examined to determine the test methods and the risk associated with accomplishing the necessary tests as part of the product qualification, acceptance, and release process. Where risks are identified, the systems engineers must work with the test engineers to develop the necessary test capabilities.

Product Delivery & Product Support

Most products live much longer in the usage phase than in the development phase. The costs associated with product support are usually greater than the cost of developing the product. These two facts make it very important for the product systems engineer to consider the product delivery and support as part of the earliest activities during development. The design of the product dictates the maintenance and support that will be required. The systems requirements are the first means of influencing the design to achieve the desired product support. If maintenance, reliability, and support requirements have not been defined by the customer, then the systems engineer must define these to achieve the support methods and costs that the customer, users, and the organization responsible for support will find financially acceptable.

References

Works Cited

de Jong, I. 2008. *Integration and Test Strategies for Complex Manufacturing Machines: Integration and Testing Combined in a Single Planning and Optimization Framework*. Saarbrücken, Germany: Verlag.

Elliott, B. et al. INCOSE UK Chapter Working Group on Applying Systems Engineering to In-Service Systems, Final Report. Somerset, UK: INCOSE UK Chapter Working Group. 2008. Accessed November 11, 2014 at INCOSE UK http://www.incoseonline.org.uk/Documents/Groups/InServiceSystems/is_tr_001_final_report_final_1_0.pdf.

Kemp, D., and R. Linton. 2008. "Service Engineering." Proceedings of the 18th Annual International Symposium of the International Council on Systems Engineering, June 15-19, 2008, Utrecht, The Netherlands.

CMMI Product Team. *CMMI for Development Version 1.3* (CMU/SEI-2010-TR-033). 2010. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University. Accessed on 10 Nov 2014 at Software Engineering Institute Library http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=9661

Sørensen, H.E. 2012. *Business Development: a market-oriented perspective*. Hoboken, NJ, USA: John Wiley & Sons.

Voas, J.M., and G. McGraw. 1998. *Software Fault Injection*. Hoboken, NJ, USA: John Wiley & Sons.

Wasson, C.S. 2006. *System Analysis, Design, and Development*. Hoboken, NJ, USA: John Wiley & Sons.

Primary References

de Jong, I. 2008. *Integration and Test Strategies for Complex Manufacturing Machines: Integration and Testing Combined in a Single Planning and Optimization Framework*. Saarbrücken, Germany: Verlag.

Voas, J.M., and G. McGraw. 1998. *Software Fault Injection*. Hoboken, NJ, USA: John Wiley & Sons.

Wasson, C.S. 2006. *System Analysis, Design, and Development*. Hoboken, NJ, USA: John Wiley & Sons.

Additional References

Phillips, F.Y. 2001. *Market-Oriented Technology Management: Innovating for Profit in Entrepreneurial Times*, 1st ed. New York, NY, USA: Springer.

< Previous Article | Parent Article | Next Article >

SEBoK v. 2.1, released 31 October 2019

Product Systems Engineering Key Aspects

Lead Author: *Ricardo Pineda*

Acquired Products versus Offered Products

The emphasis for traditional systems engineering (TSE) is in the provisioning of products and related services that meet stakeholder needs and requirements. For acquired products, an acquirer specifies the needs and requirements, selects a supplier for development and provisioning, and then receives the needed products and services. The acquirer, after acceptance, usually owns, operates, and maintains the product and the support systems supplied by the developer. Offered products are provided by suppliers based on opportunities to develop and offer products and services to potential users of the product based on business objectives usually measured in terms of value addition to the stakeholder.

In the provisioning of product systems and related services, the enterprise owning and provisioning the product and services typically makes agreements with other suppliers to also provide elements, methods, and tools that are used during their entire life cycle. The supplying enterprises, in turn, may make further agreements with suppliers in regards to building a supply chain. The complexities of dealing with supply chains must be accounted for with respect to cost, risk, and schedule and thus can have an impact upon product or service maturity. (See articles under the Systems Engineering Organizational Strategy knowledge area (KA) in Part 5.)

Acquired Products

Specific needs for a product or service typically result in some form of an agreement between the acquirer and a supplier as specified in the agreement processes of ISO/IEC 15288 (2015). The acquirer specifies the need and requirements for the properties of the expected product or service and may or may not place specific requirements upon how the supplier plans to organize their life cycle treatment of the product or system.

The degree of formality involved with the agreement varies and is strongly influenced by whether the customer is a government entity or a commercial entity. Government contracts usually incorporate strict specifications and other unique requirements that are rarely found in commercial contracts. Government acquisition agents often specify design characteristics in addition to functional and performance specifications. Design specifications place constraints on product systems engineering (PSE) by explicitly defining the details of a product's physical characteristics. The government acquirer may also specify how the product is to be designed and developed or how it is to be produced. Government specifications tend to be longer, more detailed, and more complex than functional specifications and much longer than specifications used in a commercial environment.

When contracting with the government or similar enterprises, the PSE must identify disagreements related to the meaning of a particular provision in a contract, and work with contracts to get a written resolution of all ambiguities and issues in the specifications. Failure to do this can lead to legal disputes and government claims of product substitution which can prevent acceptance of the product system and result in financial penalties.

Developing product systems for government customers requires PSE to do a thorough review and perform internal coordination within the enterprise to prevent it from submitting proposals that are non-compliant because the requirements are not fully understood.

Offered Products

Given an opportunity or perceived opportunity, an enterprise may decide to develop and offer products or services to a broader potential marketplace. The properties of the product or service are often determined through surveying and/or forecasting the potential market penetration. The supplier determines the structure and operation of an appropriate life cycle model for achieving the desired results (Pugh 1990).

Supply Chains and Distribution Channels

The supply of products and services to the owner of a product or service that is acquired or offered at various points during the life cycle is vital to success. It is this wider system-of-interest (WSOI) that is the outsourcing holism that must be treated properly in order to provide successful products or services. A portrayal of supply chain structure is provided in Figure 1 below.

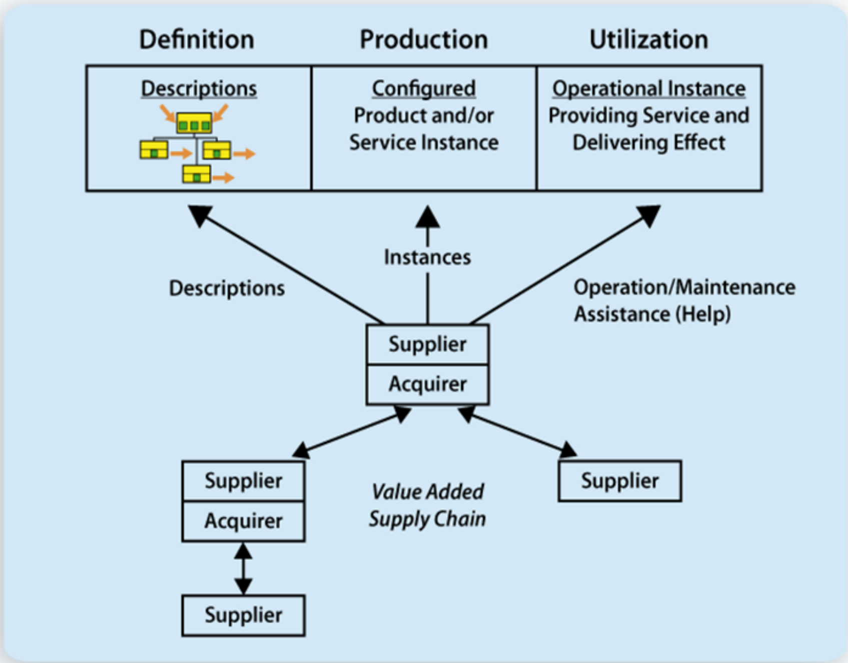


Figure 1. Supply Chain Structure (Lawson 2010). Reprinted with permission of Harold "Bud" Lawson.

In Figure 1, it is important to note that in an agreement with a supplier, the outsourcing can involve delivering complete system description solutions or portions thereof. For example, a supplier could, given a set of stakeholder requirements developed by the acquirer, develop and supply a system that conforms to the architectural solution. The supplier in turn can be an acquirer of portions of their delivered results by outsourcing to other suppliers.

In regards to production, the outsourcing agreement with a supplier can vary from total production responsibility to merely supplying instances of system elements to be integrated by the acquirer. Once again, these suppliers can be acquirers of portions of their delivery from outsourcing to other suppliers.

In regards to utilization, for non-trivial systems, outsourcing agreements can be made with a supplier to provide for operational services, for example, operating a health care information system. Further agreements with suppliers can involve various forms of logistics aimed at sustaining a system product or service or for supplying assistance in the form of help desks. Once again, suppliers that agree to provide services related to utilization can be acquirers of the services of other suppliers.

Important to all supply chains is the concept that supplying parties contribute some form of added value to the life cycle of a system-of-interest. The proper management of a supply chain system asset is a vital part of the operations of an enterprise. In fact, the supply chain itself is an enterprise system-of-interest that is composed of acquirers and suppliers as system elements. There is definitely a structure tied together by agreement relationships. Further, the operation of the supply chain results in an emergent behavior. The supply chain system becomes a vital infrastructure asset in the system portfolios of enterprises and forms the basis for extended enterprises.

Similar to a supply chain, the distribution channels for a product system can be a complex web of relationships between the product supplier and various distributors, for example, package delivery companies, warehouses, service depots, wholesale outlets, retail sales establishments, operator training and certification organizations, and so on. The nature of the distribution channels could have a significant impact on the architecture or design of a product system.

PSE may need to include special features in the product design to accommodate for the needs of distribution channel elements, for example, heavy load tie down or lifting brackets, protective shipping packages, retail marketing displays, product brochures, installation manuals, operator certification packages, training materials, and so on. Sometimes it may be necessary to create special versions (or instances) of the product for the training of operators and users for certifying safe or secure operations, for environmental testing and qualification, for product demonstration and user testing, for patent application, for load testing and scalability demonstrations, and for interface fit checking and mass balance certification, to name some examples.

Product Lifecycle and Product Adoption Rates

The life cycle of each product follows the typical incremental development phases shown below (Wasson 2006, 59-65). A particular product to be engineered could be preceded by a previous "model" of that product as shown in the product model life cycle below, and could be superseded later by a newer model of that product. It is worth noting that there is no standard set of life cycle phases. The example below is one of many ways that the phases can be structured.

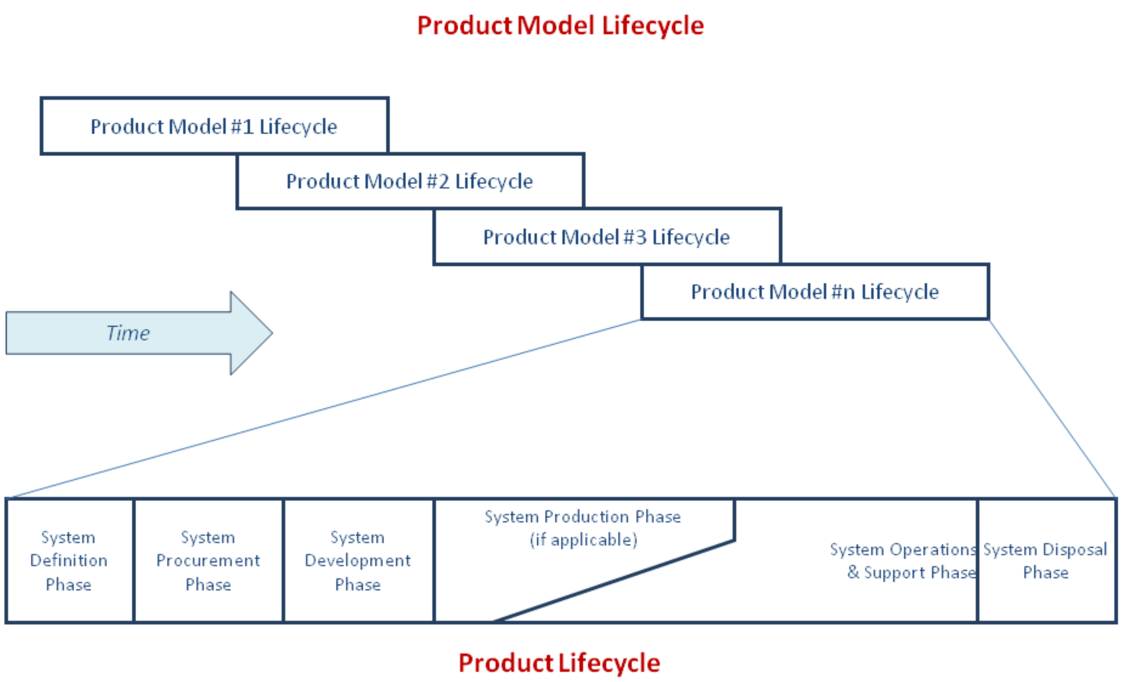


Figure 2. Product Lifecycle as Related to the Product Model Lifecycle (Wasson 2006). Reprinted with permission of John Wiley & Sons, Inc. All other rights are reserved by the copyright owner.

From an industry perspective, managing a product's life cycle involves more than just the engineering aspects:

> *Product lifecycle management (PLM) is the process of managing the entire lifecycle of a product from its conception through design and manufacture to service and disposal. PLM integrates people, data, processes and business systems, and provides a product information backbone for companies and their extended enterprise.* (CIMdata 2012)

There are many PLM tools and services available for facilitating the development and management of complicated product life cycles and especially for product line management (insert link to product line mgmt section here).

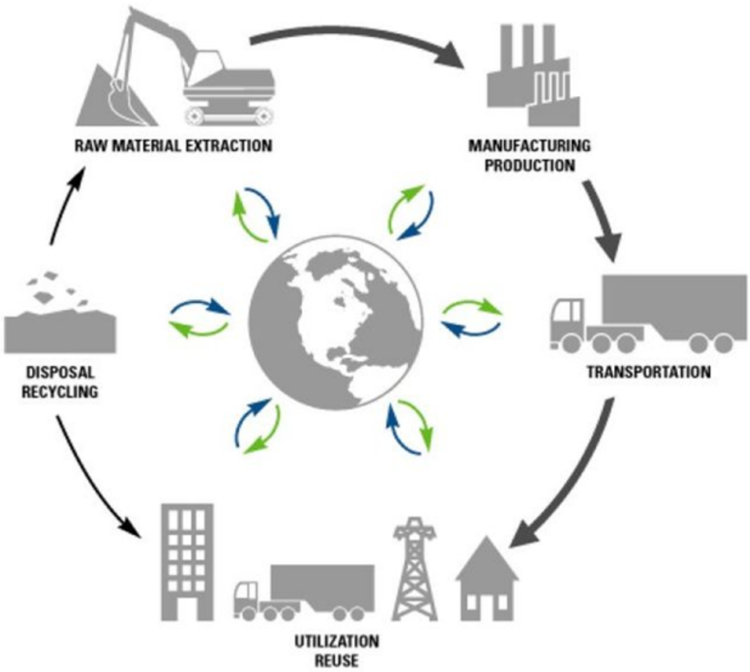


Figure 3. Product Lifecycle from an Industry Perspective. (Source: http://commons.wikimedia.org/wiki/File:Product%E2%80%99s_lifecycle.jpg#filelinks Accessed February 6, 2012. NIST Programs of the Manufacturing Engineering Laboratory, Released by US Federal Government, Public Domain)

The product and product model life cycles are driven by the product adoption rate, illustrated below, that is commonly experienced by most engineered products (Rogers 2003). As products reach market saturation (i.e., on the down slope of the curve below) then there would typically be a new, upgraded version of the product ready for delivery to the marketplace. PSE serves a critical role in determining the best timing for delivery of this new version and the set of features and functions that would be of the greatest value at that time.

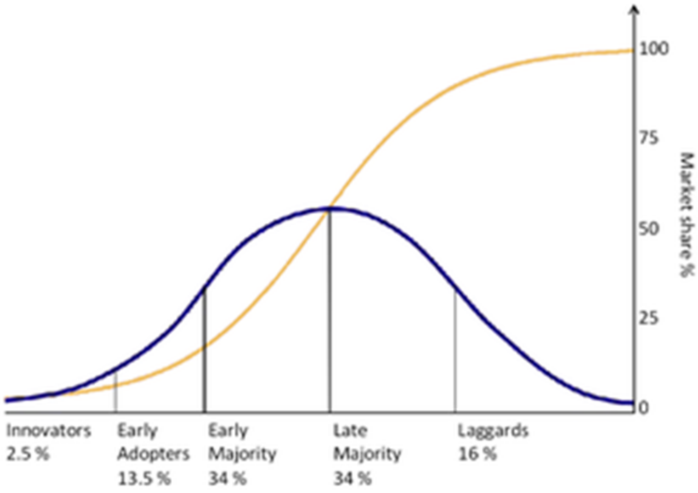


Figure 4. Rogers Innovation Adoption Curve. (Source: http://en.wikipedia.org/wiki/File:Diffusionofideas.PNG Accessed February 6, 2012, Released by Tungsten, Public Domain)

Integrated Product Teams and Integrated Product Development

Product systems as discussed throughout this KA mandate the participation of different disciplines for their success during their entire lifecycle from concept to product disposal or retirement. Rapid technology innovations and market pressures in the mid '90s demanded development process (mostly input-output serial) to shorten their development time and development cost, and to improve product quality to remain competitive. For commercial enterprises, the typical development times of 18-24 months to deploy new products into markets of the '90s have in many cases been reduced to 6-12 months and even 3-6 months for the highly competitive leading edge information technology products.

An initial response to these pressures was concurrent engineering. Concurrent engineering is "... a systematic approach to the integrated, concurrent design of products and their related processes, including manufacturing and support to cause developers, from the outset to consider all elements of the product lifecycle from conception through disposal, including quality, cost, schedule and end user requirements." This definition has evolved into the integrated product development (IPD) as more descriptive of this concurrency to describe the continuous integration of the entire product team, including engineering, manufacturing, test, and support through the life cycle. Later, as the importance of the process was recognized, the terminology was modified to integrated product and process development or IPPD (INCOSE 2012).

The INCOSE *Systems Engineering Handbook* v. 3.2.2 provides a good description of the IPT and IPDT process; the different types of IPDT; the steps in organizing and running an IPDT; good examples of IPDT, particularly for acquired systems; and a good discussion on IPDT pitfalls to avoid. (INCOSE 2012)

IPD/IPPD helps plan, capture, execute, and evaluate programs to help design, test, build, deliver, and support products that satisfy diverse stakeholder requirements. IPD/IPPD outlines the necessary infrastructure needed to deploy, maintain, evaluate and continuously improve processes and tools by aligning people (IPTs) and processes to realize product goals (customer satisfaction). The implementation of Integrated Product Development Processes (IPDP) requires an integrated approach for program planning and generally includes the following: Business Strategy, Program Management and Control, Project Planning, Product Requirements and Architecture Development, Product Design and Development, Production and Deployment, Product Verification and Validation, and Operations and Maintenance Support.

At each development stage, there is a decision gate that helps decide if the IPDP is feasible to enter the next stage of product development. IPD utilizes multi-functional IPTs to optimize the individual product and processes to meet overall cost and performance objectives. IPTs are a cross-functional group of people typically including representatives of all the relevant stakeholders in the project, who are brought together for delivering an integrated product to an external or internal customer using relevant IPDP. The main function of the IPTs is to ensure the business, technical and economical integrity and overall success of the product that is delivered to its eventual customer. IPTs carry out tailored IPDPs and follow relevant SE processes to deliver products that satisfy customer needs, overcomes external constraints, and adheres to the overall program strategy.

In the case of commercial enterprises, product development is tightly coupled with business strategies (short and long term), stakeholder value added measured in terms of return on investments (ROI), market presence/coverage, and other strategies as defined by the business objectives. Thus, product integration teams include strategic planners, business managers, financial managers, market managers, quality assurance managers, customer representatives, and end-users, as well as other disciplines required for acquired products. Phillips (2001), Annachino (2003), and Morse (2007) provide good discussions on this topic.

Role of Architectures, Requirements, and Standards

The architectural properties of a product system are influenced by the concerns of the various stakeholders as indicated in the ISO/IEC 42010 standard (ISO/IEC 2011). The stakeholders have various views that they express based on their specific perspective. These views are vital in establishing requirements and are inputs to those responsible for defining the functions, structures, and relationships needed to achieve the desired product or service.

A number of stakeholders have been identified in the discussions of product systems. It would be possible to identify a set of important stakeholders based on the life cycle thinking provided by the ISO/IEC 15288 standard (2015), for example, one such set could consist of owners, conceivers, developers, producers, users, and maintainers as discussed by Lawson (2010). As mentioned earlier, these stakeholders should cooperate at all stages of the life cycle in specifying requirements, verifying that the requirements are met, and validating that the products produced provide needed capabilities.

In addition to the two standards that have been identified, there are a variety of standards related to specialty aspects of products, such as safety and security, as well as standards that are applicable for project management and life cycle considerations, such as requirements and quality management.

Role of Modeling, Simulation, Prototyping, and Experimentation

Modeling, simulation, prototyping, and experimentation are techniques that have the purpose of improving stakeholder knowledge and shared understanding about aspects of the system to de-risk system development and operation before heavy commitment of time and funds. Examples of this are found below:

- Understanding future needs: “Warfighting experiments are the heart of the Army's warfighting requirements determination process. Progressive and iterative mixes of high fidelity constructive, virtual, and live simulations using real soldiers and units in relevant, tactically competitive scenarios provide Army leaders with future operational capability insights" (US Army 2012),
- Simulation is used to predict and optimize aspects of system performance for which there are good mathematical or logical models before committing the final physical design, and also to verify and validate the system design in scenarios where physical testing is too difficult, dangerous, or expensive, for example, checking the performance envelope of military systems in a wide range of engagement scenarios where test firing thousands of rounds to get statistically valid data is clearly unaffordable, ensuring that the safety features in a nuclear power station will operate correctly in a wide range of stressing scenarios, etc.,
- Prototyping (physical and virtual) is used in a wide variety of ways to check out aspects of system performance, usability, utility, and to validate models and simulations as part of the iterative process of converging on a final design,
- In a manufacturing context, the first units produced are often “prototypes” intended to make sure the production process is working properly before committing to high rate production, and are often not shipped to end users, but used for intensive testing to qualify the design, and
- Simulation is also used extensively for training and marketing purposes. For training, an accurate model of the human machine interface and representation of the operational context allows operators to do most of their training without putting operational hours on the real system enabling them to learn emergency procedures for combat and accident scenarios in a safe and repeatable environment; for example, airline and military pilots now train mainly on simulators. System simulators of various levels of fidelity are used to familiarize customers and end users with the potential characteristics and benefits of the system, available options and trade-offs, and integration issues early in the development and acquisition process.

All of these methods use a variety of physical and mathematical representations of the system and its environment so modeling is an enabler for simulation, prototyping, and experimentation.

Increasing Role of Software in Product Functionality

An important trend in commercial products is the increasing importance of software in an increasingly wide range of products. Everything from phones, cameras, cars, test gear, and medical equipment now has essential functionality implemented in software. Software has had an increasing role in providing the desired functionality in many products. The embedding of software in many types of products accounts for increasing portions of product functionality. In tangible products such as cars, software helps improve functionality and usability (cruise control, climate control, etc.). In intangible products such as insurance, software helps in improving operational efficiency, data accessibility, etc.

The movement toward the internet of “things” where sensing and activating functions are incorporated is now starting to permeate. The use of various software products in proving service is also described in the Service Systems Engineering article.

Recent advancements in IT and software have assisted in their increased use in PSE. Although software development is already a very complex field, the role of software in the development and functionality of products is growing larger each day.

There is a need to broaden the horizons of software engineers to think of problem solving not only in software terms, but also using the systems thinking approach. For this purpose, software engineers need to be able to think critically about the problem and also the possible solutions to the problem or opportunity and its implication for business objectives.

Product Integration and Interface Control

Integration is "the set of activities that bring together smaller units of a system into larger units" (Eisner 2008). Products may consist of several systems, subsystems, assemblies, parts, etc., which have to work together as a whole to deliver the offered product’s functionalities at specified performance levels in the intended operations environment. Product integration entails not only the working together of hardware and software components, but also the organization, processes, people, facilities, and the resources for the manufacturing, distribution, maintenance, customer support, sales channels, etc. Grady (2010) groups the above information into three fundamental integration components: functional organization, product integration, and process integration.

PSE plays an important role to ensure well defined interfaces, interactions, relationships, information exchange, and processes requirements between product components. These requirements are baseline, documented, traced, verified, and validated for the end-to-end Product integration and to maintain and ensure product offering integrity during its life cycle. The systems engineering hierarchical decomposition level allows requirement definition and allocations at different levels of abstraction to define the building blocks of the product architecture; these building blocks are assigned to integrated product development teams (IPDTs) for detailed design and development. The IPDTs or the systems engineering integration team (SEIT) must interact with all involved players to generate appropriate architectural block specifications at the lower tier of development for a product’s architectural configuration and configuration tracking. As the building blocks are put together, interface requirements, information exchange, and interaction and relationships among entities are verified against the baseline. Once a configuration item has been built and tested against the baseline, test and verification at higher levels are conducted to obtain the final product configuration; the final product configuration can only be changed by a formal approval from a configuration control board (CCB). Note: the acronym CCB is often used to mean the change control board that, in addition to configuration control, makes decisions of any aspect of a project or an enterprise.

Interface agreements, specifications, and interface designs are usually documented through the interface control documents (ICD) and the interface design descriptions (IDD); in some instances, depending on the complexity of the product and the type of internal and/or external interfaces, an interface control working group (ICWG) is created to analyze and baseline changes to an interface for further recommendation to the CCB.

A configuration item (CI) may be hardware (HWCI), software (SWCI), firmware, subsystems, assemblies, non-development items, commercial off-the-shelf (COTS) items, acquirer furnished equipment, and/or processes. Please see Wasson (2006), Grady (2006), and INCOSE *SE Handbook* v. 3.2.2 for a more detailed description of configuration and interface control.

A product may experience hundreds of changes during its life cycle due to new product releases/enhancements, repair/replacement of parts, upgrades/updates in operating systems, computer infrastructure, software modules, organizational changes, changes in processes and/or methods and procedures, etc. Thus, strong mechanisms for bookkeeping and activity control need to be in place to identify, control, audit, account and trace interfaces, interactions, and relationships between entities that are required to maintain product configuration status (Eisner 2008). The product configuration and CI’s are then controlled through the configuration management process.

Configuration Management and Risk Management

Configuration management (CM) deals with the identification, control, auditing, status accounting, and traceability aspects of the product, and broadly covers the book-keeping and control activities of the systems engineering process (Eisner 2001). Any product configuration changes to the baseline (configuration item, operational baseline, functional baseline, behavior baseline) or product baseline are submitted to a configuration control board (CCB) through an engineering change request (ECR) and/or a configuration change request (CCR). The CCB then analyzes the request to understand CI impacts and the feasibility (time and cost) of authorization or rejection of change request(s). The lack of proper control and tracking of CI and product baselines may result in a loss of features, functionality, data, interfaces, etc., leading to backtracking and CI version losses which may affect the offered product. All approved changes will have to be baselined, documented, and tested for backward compatibility and to ensure compliance with the integrated product functionality. Thus, successful implementation and life cycle management of the product mandates a highly disciplined CM process that maintains proper control over the product and its components. Please see the INCOSE *Systems Engineering Handbook* v. 3.2.2 (2012) for a detailed description of the CM Process.

Risk management deals with the identification, assessment, and prioritization of technical, cost, schedule, and programmatic risks in any system. Almost all engineered systems are designed, constructed, and operated under some level of risks and uncertainty while achieving multiple, and often conflicting, objectives. As greater complexities and new technologies are introduced in modern systems, the potential of risks have significantly increased. Thus, the overall managerial decision-making process should involve an extensive cost-benefit analysis of all identified, qualified, and evaluated risks (Haimes 2008). Risk management involves the coordinated and most cost-effective application of resources to minimize, monitor, and control the probability and/or impact of all identified risks within the systems engineering process. The risk management process requires the involvement of several disciplines and encompasses empirical, quantitative, and normative judgmental aspects of decision-making. Furthermore, risk assessment and management should be integrated and incorporated within the broader holistic approach so technology management can help align the risk management requirements to the overall systems engineering requirements. Thus, the inclusion of a well defined risk management plan that deals with the analysis of risks, within the systems engineering master plan is vital for the long term and sustained success of any system (Blanchard and Fabrycky 2011).

References

Works Cited

Annachino, M. 2003. *New Product Development: From Initial Idea to Product Management*. Amsterdam, Netherlands: Elsevier.

Blanchard, B.S., and W.J. Fabrycky. 2011. *Systems Engineering and Analysis*, 5th ed. Prentice Hall International Series in Industrial and Systems Engineering. Englewood Cliffs, NJ, USA: Prentice Hall.

Eisner, H. 2008. *Essentials of Project and Systems Engineering Management*, 3rd ed. New York, NY, USA: John Wiley & Sons.

Grady, J. 2006. *System Requirements Analysis*. New York, NY, USA: Academic Press.

Haimes, Y. 2008. *Risk Modeling, Assessment, and Management*, 3rd ed. New York, NY, USA: John Wiley & Sons.

INCOSE. 2012. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, version 3.2.2. San Diego, CA, USA: International Council on Systems Engineering (INCOSE), INCOSE-TP-2003-002-03.2.2.

ISO/IEC/IEEE. 2015.*Systems and software engineering - system life cycle processes*.Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC), Institute of

Electrical and Electronics Engineers.ISO/IEC 15288:2015.

ISO/IEC/IEEE. 2011. *Systems and software engineering - Architecture description*. Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC)/Institute of Electrical and Electronics Engineers (IEEE), ISO/IEC/IEEE 42010.

Kass, R. 2006. "The logic of warfighting experiments." DOD Command and Control Research Program (CCRP). August 2006. Accessed 23 April 2013 at http://www.dodccrp.org/files/Kass_Logic.pdf.

Lawson, H. 2010. *A Journey Through the Systems Landscape*. London, UK: College Publications.

Morse, L., and D. Babcock. 2007. *Managing Engineering and Technology*. International Series in Industrial and Systems Engineering. Upper Saddle River, NJ, USA: Prentice Hall.

Phillips, F. 2001. *Market Oriented Technology Management: Innovating for Profit in Entrepreneurial Times*. New York, NY, USA: Springer.

Pugh, S. 1990. *Total Design: Integrated Methods for Successful Product Engineering*. Englewood Cliffs, NJ, USA: Prentice Hall.

Reinertsen, D. 1997. *Managing the Design Factory: A Product Developers Tool Kit*. New York, NY, USA: Simon & Schuster Ltd.

Rogers, E.M. 2003. *Diffusion of Innovations*, 5th ed. New York, NY, USA: Free Press.

Smith, P., and D. Reinertsen. 1997. *Developing products in half the time – new rules, new tools*, 2nd ed. Hoboken, NJ, USA: John Wiley & Sons.

US Army. 2012. "Chapter 2, section A.4" in *Army Science and Technology Master Plan*. Accessed January 12, 2012. Available at: http://www.fas.org/man/dod-101/army/docs/astmp/c2/P2A4.htm.

Wasson, C.S. 2006. *System Analysis, Design, and Development*. Hoboken, NJ, USA: John Wiley & Sons.

Primary References

Blanchard, B.S., and W.J. Fabrycky. 2011. *Systems Engineering and Analysis*, 5th ed. Prentice Hall International Series in Industrial and Systems Engineering. Englewood Cliffs, NJ, USA: Prentice Hall.

Eisner, H. 2008. *Essentials of Project and Systems Engineering Management*, 3rd ed. New York, NY, USA: John Wiley & Sons.

Wasson, C.S. 2006. *System Analysis, Design, and Development*. Hoboken, NJ, USA: John Wiley & Sons.

Additional References

Annachino, M. 2003. *New Product Development: From Initial Idea to Product Management*. Amsterdam, Netherlands: Elsevier.

Haimes, Y. 2008. *Risk Modeling, Assessment, and Management*, 3rd ed. New York, NY, USA: John Wiley & Sons.

Kass, R. 2006. "The logic of warfighting experiments." DOD Command and Control Research Program (CCRP). August 2006. Accessed 23 April 2013 at http://www.dodccrp.org/files/Kass_Logic.pdf.

Lawson, H. 2010. *A Journey Through the Systems Landscape*. London, UK: College Publications.

Morse, L., and D. Babcock. 2007. *Managing Engineering and Technology*. International Series in Industrial and Systems Engineering. Upper Saddle River, NJ, USA: Prentice Hall.

Phillips, F. 2001. *Market Oriented Technology Management: Innovating for Profit in Entrepreneurial Times*. New York, NY, USA: Springer.

Pugh, S. 1990. *Total Design: Integrated Methods for Successful Product Engineering*. Englewood Cliffs, NJ, USA: Prentice Hall.

Reinertsen, D. 1997. *Managing the Design Factory: A Product Developers Tool Kit*. New York, NY, USA: Simon & Schuster Ltd.

Rogers, E.M. 2003. *Diffusion of innovations,* 5th ed. New York, NY: Free Press.

Smith, P., and D. Reinertsen. 1997. *Developing products in half the time – new rules, new tools*, 2nd ed. Hoboken, NJ, USA: John Wiley & Sons.

US Army. 2012. "Chapter 2, section A.4" in *Army Science and Technology Master Plan*. Accessed January 12, 2012. Available at: http://www.fas.org/man/dod-101/army/docs/astmp/c2/P2A4.htm.

< Previous Article | Parent Article | Next Article >

SEBoK v. 2.1, released 31 October 2019

Product Systems Engineering Special Activities

Lead Author: *Ricardo Pineda*

Product systems engineering has activities that are unique to products. This article discusses many of them.

Readiness Level Assessments

As a new system is developed, it is essential to verify and validate that the developed system is mature enough to be released as an operational product or service. Technology readiness assessments (TRA) are established tools used to qualify technology development and help make investment decisions within complex development programs in order to deploy systems or elements of technology to an end user in a timely fashion.

This notion of maturity was formalized by the US National Aeronautics and Space Administration (NASA) (Mankins 1995) and later modified for use by the Department of Defense (DoD), the Air Force Research Laboratory (AFRL), and the US Department of Energy (DoE), as well as a growing number of non-governmental organizations. Technology readiness levels (TRL) are a metric developed to summarize the degree of maturity of a technology. The original NASA TRL scale has nine different levels from the *basic principles observed and reported* (TRL 1) to *actual systems "flight proven" through successful mission operations* (TRL 9). The TRL scale utilized by the DoD is portrayed in Table 1.

Table 1. Technology Readiness Levels for Assessing Critical Technologies (Mankins 1995). Released by the Advanced Concept Office, Office of Space Access and Technology, NASA.

Technology Readiness Level	+ 1. Basic principles observed and reported.	+ 2. Technology concept and/or application formulated.	+ 3. An analytical and experimental critical function and/or characteristic proof of concept.	+ 4. Component validation in laboratory environment.	+ 5. Component validation in relevant environment.	+ 6. Prototype demonstration in a relevant environment.	+ 7. Prototype demonstration in an operational environment.	+ 8. System qualified through test and demonstration.	+ 9. System proven through successful mission operations.	Actual application of the technology in its final form and under mission conditions, such as those encountered in operational test and evaluation.

The utilization of TRLs has an impact on the structure and operation of life cycles as described in Part 3; they allow better management and control of risks inherent with technology, as well as better control of costs and the schedule of program development. However, TRLs do not provide an assessment of the programmatic influence on a TRL, technology criticality and priority, software aging and readiness context, as pointed out by Smith (2005). While TRLs have proven to be useful in evaluating a technology’s performance, as demonstrated in the laboratory or in a test environment, they do not inform one whether or not the technology product can actually be produced in an affordable manner. The concept of manufacturing readiness levels (MRL) has been incorporated to expand the TRL idea so that it can incorporate producibility concerns. The MRL approach addresses questions such as the level of technology reproducibility, the cost of production, and technology manufacturing production environment early in the development phase (GAO 2003, DoD 2011).

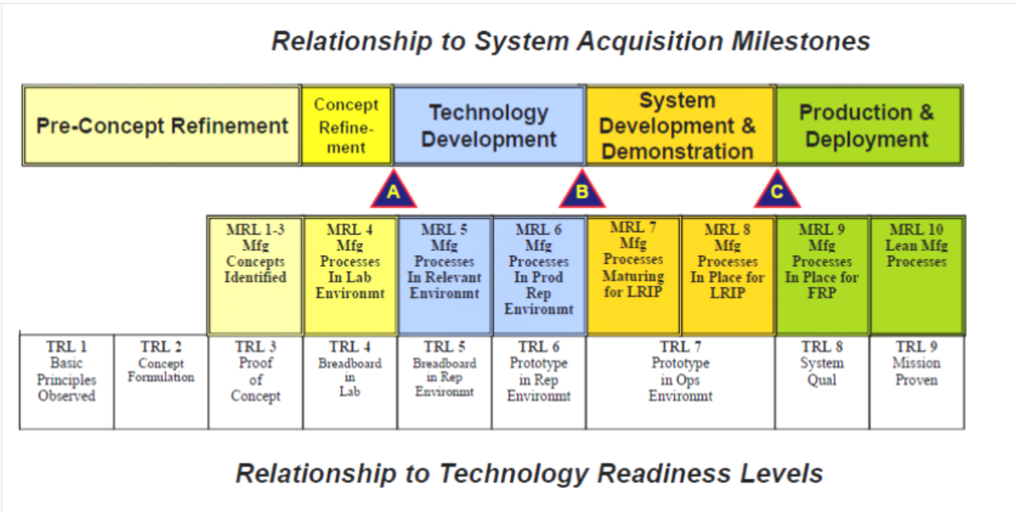


Figure 1. Technology Readiness Levels and Their Relationship to System Acquisition Milestones (Morgan 2008). Released by the Manufacturing Technology Division of the United States Air Force.

Readiness levels are an active research area within academia and government agencies in regards to the integration of technology components into complex systems (integration readiness levels (IRLs)) to address interface maturity among existing and maturing technology developments. TRLs apply to the critical enabling technologies, which are usually embodied at the subsystem, assembly level, or system component level. Systems readiness levels (SRL) are used when going from individual technologies to the whole system. The SRL model is a function of the individual TRLs in a system and their subsequent integration points with other technologies, the IRL (Sauser 2006).

Another maturity aspect is related to the provisioning of products that are readily available and referred to as commercial off-the-shelf (COTS). Such products, be they hardware, software, or a mixture of both, have hopefully achieved the degree of maturity so that those acquiring them can rely upon their operational properties and that the documentation of the COTS products is sufficient to provide the proper guidance in their use.

The PSE should realize that the TRL assessment for COTS changes dramatically if the operational environment or other requirements are imposed that exceed the design limits of the COTS product (e.g., operations at very high or very cold temperatures, high shock, or vibration levels).

Product Certification

Product certifications are both domain and product specific, and typically relate to human safety and health, the need to meet a specific government regulation, or are required by underwriters for insurance purposes. Certifications are performed by a third party (independent of the developer) who provides a guarantee of the quality, safety, and reliability of the product to the customer or user.

The INCOSE *SE Handbook* defines product certification as "the process of certifying that a certain product has passed performance or quality assurance tests or qualification requirements stipulated in regulations such as a building code or nationally accredited test standards, or that it complies with a set of regulations governing quality or minimum performance requirements." (INCOSE 2012)

The INCOSE *SE Handbook* also defines four methods for verification: inspection, analysis, demonstration, and testing (INCOSE 2012). In addition, it defines certification as a fifth verification method, which is defined as verification against legal or industrial standards by an outside authority without direction to that authority as to how the requirements are to be verified. For example, electronic devices require a CE certification in Europe, and a UL certification in the US and Canada (INCOSE 2012).

The best known certification is the airworthiness certification, which relates to the safety of flight for aircraft. In the US, the test for this certification is performed by the Federal Aviation Administration (FAA). Government certifications are also common in the medical systems field where the Federal Drug Administration (FDA) is the primary certification agency. Some certifications are based on standards defined by technical societies, such as the American Society of Mechanical Engineers (ASME). The combination of the technical standards and a certification allows product developers to perform certifications that meet government standards without having the government directly involved in the process.

There are equivalent government organizations in other countries and for other regulated areas, such as communications, building safety, nuclear systems, transportation systems to include ships, trains and automobiles, environmental impact, and energy use. Systems engineers must be aware of the certifications that are required for the domain and product being developed. Certification agencies must be involved early in the development effort to ensure the necessary certifications are included in the system requirements, the system development plan, and the funding provided to accomplish the development. When system changes and upgrades are necessary, the systems engineers must determine if product re-certification is necessary and include it in the plans and funding for the system upgrade.

Enabling Product Certifications

There may be other certifications for enabling products that must be considered and appreciated by PSE, such as an operator certification of airplane pilots to ensure flight safety, and certification of nuclear plant operators to ensure prevention or mitigation of nuclear radiation effects. An example of this is shown in the certification program by the North American Electric Reliability Corporation (NERC):

> *In support of NERC's mission, the System Operator Certification Program's mission is to ensure that employers have a workforce of system operators that meet minimum qualifications. These minimum qualifications are set through internationally recognized processes and procedures for agencies that certify persons. The Certification Program promotes excellence in the area of system operator performance and encourages system operators to be inquisitive and informed.* (NERC 2012)

Production qualification testing (PQT) is another type of certification which DAU (2005) describes as:

A technical test completed prior to the full-rate production (FRP) decision to ensure the effectiveness of the manufacturing process, equipment, and procedures. This testing also serves the purpose of providing data for the independent evaluation required for materiel release so that the evaluator can address the adequacy of the materiel with respect to the stated requirements. These tests are conducted on a number of samples taken at random from the first production lot, and are repeated if the process or design is changed significantly and when a second or alternative source is brought online.

Security certification and accreditation (C&A) is often required for the deployment of computing and networking equipment in a classified environment. Facility certification may be required to ensure that a building housing the equipment can provide the proper environment for safe and efficient operation of the equipment. High-altitude electromagnetic pulse (HEMP) certification may be required to ensure that a building and its equipment can withstand the effects of HEMP from nuclear weapons. A similar type of certification to HEMP is TEMPEST testing to ensure that sensitive electronic emissions are not allowed to leave high security facilities. TEMPEST is a code name referring to investigations and studies of compromising emission, and is not an acronym.

Technology Planning and Insertion

Technology planning can be an enterprise function or a program function. Technology planning as an enterprise function typically occurs on an annual basis to determine the funding necessary for independent research and development in the coming year. Technology planning as a program function occurs early in the program and often continues throughout the life of the system. The design of the product system is highly dependent on the availability of technologies that have acceptable risks and that meet the customer's cost, schedule, and performance requirements. These critical technologies will only be available when necessary if the systems engineers perform concept designs, technology assessments, and trade studies that define the critical technologies and the capabilities necessary before the system development activities that will use the critical technologies begin.

The MITRE *Systems Engineering Guide* (MITRE 2011) provides the following definition for technology planning:

Technology Planning is the process of planning the technical evolution of a program or system to achieve its future vision or end-state. Technology planning may include desired customer outcomes, technology forecasting and schedule projections, technology maturation requirements and planning, and technology insertion points. The goal is a defined technical end-state enabled by technology insertion over time.

Systems engineers who participate in technical planning must understand the future vision and system requirements, and relate these to the current and expected future technologies that can be applied to the system design during current development stages, as well as for potential future upgrades to the system. To do this, systems engineers must acquire and maintain knowledge of the existing and developing technology in their design domain. The systems engineer will also provide the essential connection between the system user and research communities to provide alignment between the technology developers and the system designers.

Technology planning and insertion usually requires that the systems engineer perform technology readiness assessments that rate the maturity levels and the risks associated with the planned technologies. Immature, risky technologies require risk reduction activities that include prototyping and product development and test activities that provide quantification of the capabilities and risks. The risk reduction activities provide the data necessary to assess and update the design to reduce its risk.

Product Road Mapping and Release Planning

Product road maps provide an outline that shows when products are scheduled for release and include an overview of the product's primary and secondary features. Both internal and external product road maps should be created. The form of the road map will depend on the development methodology being used. Waterfall, iterative, and spiral development models result in different road maps and release plans. The systems engineer must be an integral member of the team that creates road maps. Requirements should be mapped onto each of the planned releases. Test plans must be adapted to the development model and the release plans.

Product road maps should be aligned with the technology road maps that are applicable to the product. Technology maturity should be accomplished before the technologies are included in the product development plans and the road map for the product release that includes those technologies.

Product road maps are essential for software intensive systems that have many releases of software and capability upgrades. The identification of the requirements, the test plans, and the features provided for each release are an essential driver of the product development process. Clear definition of these items can make the difference between delivering the capabilities the customer is looking for and will support, or a product that fails to meet the needs of the customer and is abandoned.

Intellectual Property Management

Systems engineers must also manage intellectual property as part of their job. Existing systems engineering literature rarely covers this topic. However, there are many textbooks and management related literature that provide additional information, such as “Intellectual Property Rights for Engineers” (Irish 2005). Intellectual property may be considered as *intangible output of the rational thought process that has some intellectual or informational value and is normally protected via using copyrights, patents, and/or trade secrets* (Irish 2005). Listed below are some of the more important intellectual property types with brief explanations:

- Proprietary Information: Any information which gives a company (or enterprise) an advantage over its competitors is usually proprietary.
- Patents: A patent is the principle mechanism for protecting rights for an invention or discovery. In exchange for a full disclosure of how to practice it, the issuing government will grant the right to exclude others from practicing the invention for a limited amount of time, usually 15 to 20 years (in the US, a patent usually lasts for 17 years from the date of issue).
- Design Patents: In some countries, these are referred to by the more appropriate term *design registrations* or some other name. They protect rights in ornamental designs, provided the designs are new and inventive, i.e., *non-obvious* at the time they are made. In the US, the maximum length of a design patent is 14 years.
- Trademarks: A trademark identifies the source of origin for goods in commerce, and is not stronger than the actual use to which it has been put to and the diligence with which it has been protected from infringement, encroachment, or dilution. Under some circumstances, a trademark may be registered with governmental agencies. Among a company's most valuable assets is the corporate name, which also is the company's primary trademark.
- Copyrights: A claim of copyright protects such works as writings, musical compositions, and works of art from being copied by others, i.e., from plagiarism. A notice of claim of copyright must be made in the manner prescribed by law at the time of a protected work’s first publication.

Parts, Materials, and Process Management

The consequences of mission failure or an inability to deploy the system on time due to parts, materials, and process (PM&P) issues needs to be clearly understood by the systems engineer since these elements are fundamental to the overall mission reliability and program success. PM&P management is especially important in harsh environments (like outer space and underwater) and in situations where system failure can have catastrophic impacts on public safety (like nuclear power, bridges and tunnels, and chemical processing plants).

Generally, original equipment manufacturers (OEMs) engaged in the design and fabrication of electronic systems have a documented policy that deals with PM&P, sometimes in the form of a PM&P Management Manual. The elements of a PM&P control program include things such as

- PM&P requirements that apply to a system;
- the generation number of a program or project approved parts list (PAPL);
- the appointment of a PM&P control board (PMPCB);
- the development of a part stress derating policy and a part parameter derating policy for end of life use; and
- a definition of the minimum qualifications, quality controls, and screening requirements for parts.

PM&P management guidance is provided by MIL-HDBK-512 (DoD 2001) and ANSI/AIAA R-100 (2001), which identify the overall management process elements of a PM&P program. Additional issues to be addressed by PM&P include the following: hazardous materials, rare earth elements, conflict materials, and counterfeit materials.

References

Works Cited

ANSI/AIAA. 2001. *Recommended Practice for Parts Management*. Philadelphia, PA, USA: American National Standards Institute (ANSI)/American Institute of Aeronautics and Astronautics (AIAA), ANSI/AIAA R-100A-2001.

DAU. 2005. *Glossary of Defense Acquisition Acronyms & Terms*, 12th ed. Ft. Belvoir, VA, USA: Defense Acquisition University (DAU)/US Department of Defense (DoD). Available at: http://www.dau.mil/pubscats/PubsCats/13th_Edition_Glossary.pdf.

DoD. 2000. *Department of Defense Directive (DoD-D) 5000.01: The Defense Acquisition System*. Arlington, VA, USA: US Department of Defense, Acquisition, Technology, and Logistics (AT&L). Available at: http://www.dtic.mil/whs/directives/corres/pdf/500001p.pdf.

DoD. 2001. *Department of Defense Handbook: Parts Management*. Arlington, VA, USA: Department of Defense (DoD). MIL-HDBK-512A.

DoD. 2011 *Department of Defense Technology Readiness Assessment (TRA) Guidance*, Assistant Secretary of Defense for Research and Engineering (ASD(R&E)), May 2011.

FAA. 2011. *Airworthiness Certificates Overview.* Washington, DC, USA: Federal Aviation Administration (FAA). Available at: http://www.faa.gov/aircraft/air_cert/airworthiness_certification/aw_overview/.

GAO. 2003. *Defense acquisitions: Assessments of Major Weapon Programs*, GAO-03-476, US Government Accountability Office, May 2003.

INCOSE. 2012. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, version 3.2.2. San Diego, CA, USA: International Council on Systems Engineering (INCOSE), INCOSE-TP-2003-002-03.2.2.

Irish, V. 2005. *Intellectual Property Rights for Engineers*, 2nd ed. Herts, UK: Institution of Engineering and Technology (IET).

Mankins, J. 1995. *Technology Readiness Levels—A White Paper*. Washington, DC, USA: Advanced Concepts Office, Office of Space Access and Technology, National Aeronautics and Space Administration (NASA).

MITRE. 2011. "Systems Engineering Guide." Accessed September 11, 2012. Available at: http://www.mitre.org/work/systems_engineering/guide/.

Morgan, J 2007. *Manufacturing Readiness Levels (MRLs) and Manufacturing Readiness Assessments (MRAs)*. ADA510027 Air Force Research Lab Wright-patterson Afb Oh Manufacturing Technology Directorate. September 2007. Accessed 06 November 2014 at Defense Technical Information Center http://www.dtic.mil/get-tr-doc/pdf?AD=ADA510027

NERC. 2012. "North American Electric Reliability Corporation (NERC)." Accessed September 11, 2012. Available at: http://www.nerc.com.

Sauser, B., D. Verma, J. Ramirez-Marquez, and R. Gove. 2006. *From TRL to SRL: The Concept of System Readiness Levels*. Proceedings of the Conference on Systems Engineering Research (CSER), April 7-8, 2006, Los Angeles, CA, USA.

Smith, J. 2005. *An Alternative to Technology Readiness Levels for Non-Developmental Item (NDI) Software*. Proceedings of the 38th Hawaii International Conference on Systems Sciences, January 3-6, 2005, Island of Hawaii, USA.

Primary References

Mankins, J. 1995. *Technology Readiness Levels—A White Paper*. Washington, DC, USA: Advanced Concepts Office, Office of Space Access and Technology, National Aeronautics and Space Administration (NASA).

MITRE. "Systems Engineering Guide." Available at http://www.mitre.org/work/systems_engineering/guide/

Sauser, B., D. Verma, J. Ramirez-Marquez, and R. Gove. 2006. *From TRL to SRL: The Concept of System Readiness Levels*. Proceedings of the Conference on Systems Engineering Research (CSER), Los Angeles, CA, April 7-8, 2006.

Additional References

ANSI/AIAA. 2001. *Recommended Practice for Parts Management*. Philadelphia, PA, USA: American National Standards Institute (ANSI)/American Institute of Aeronautics and Astronautics (AIAA), ANSI/AIAA R-100A-2001.

DoD. 2000. *Department of Defense Directive (DoD-D) 5000.01: The Defense Acquisition System*. Arlington, VA, USA: US Department of Defense, Acquisition, Technology, and Logistics (AT&L). Available at: http://www.dtic.mil/whs/directives/corres/pdf/500001p.pdf.

DoD. 2001. *Department of Defense Handbook: Parts Management*. Arlington, VA, USA: Department of Defense (DoD). MIL-HDBK-512A.

FAA. 2011. "Airworthiness Certificates Overview." Washington, DC, USA: Federal Aviation Administration (FAA). Available at: http://www.faa.gov/aircraft/air_cert/airworthiness_certification/aw_overview/.

Irish, V. 2005. *Intellectual Property Rights for Engineers*, 2nd ed. Herts, UK: Institution of Engineering and Technology (IET).

Smith, J. 2005. *An Alternative to Technology Readiness Levels for Non-Developmental Item (NDI) Software*. Proceedings of the 38th Hawaii International Conference on Systems Sciences, January 3-6, 2005, Island of Hawaii, USA.

< Previous Article | Parent Article | Next Article >

SEBoK v. 2.1, released 31 October 2019

Knowledge Area: Service Systems Engineering

Service Systems Engineering

***Lead Authors:** Ricardo Pineda, Bud Lawson, Richard Turner*

The growth of services in the ever-evolving global economy has brought much needed attention to service science and service systems engineering (SSE). Research focuses on developing formal methodologies to understand enterprise-end-user (customer) interactions from both socio-economic and technological perspectives, and to enable value co-creation and productivity improvements. Service systems require trans-disciplinary collaborations between society, science, enterprises, and engineering. Service transactions are customized and personalized to meet a particular customer need. This requires a disciplined and systemic approach among stakeholders and resources to emphasize end-user satisfaction in the design and delivery of the service (Hipel et al. 2007; Tien and Berg 2003; Vargo and Akaka 2009; Maglio and Spohrer 2008; Maglio et al. 2010).

Topics

Each part of the SEBoK is divided into knowledge areas (KAs), which are groupings of information with a related theme. The KAs in turn are divided into topics. This KA contains the following topics:

- Service Systems Background
- Fundamentals of Services
- Properties of Services
- Scope of Service Systems Engineering
- Value of Service Systems Engineering
- Service Systems Engineering Stages

Introduction

New Service Development (NSD) has usually been a proprietary process closely guarded by product businesses and service businesses for their competitive advantage. Traditional systems engineering practices have been primarily applied in aerospace and defense sectors while SSE practices have been applied by information and communications technologies (ICT) service providers (Booz, Allen, and Hamilton 1982; Johnson et al. 2000; Eppinger 2001; Freeman 2004; Whitten and Bentley 2007; AT&T SRP 2008; Lin and Hsieh 2011).

These early efforts were, and in some instances remain, very important for product and service businesses. However, the growth and ubiquity of the World Wide Web, advances in computer science and ICT, and business process management through "social networking," support the realization of closely interrelated service systems. Product business (manufacturing, agriculture, etc.) and service business distinctions are going away (Spohrer 2011).

These services, or service innovations, must take into account social aspects, governance processes, business processes, operational processes, as well as design and development processes. The customer, service provider, product provider, and intermediaries need to collaborate toward the optimization of customer experiences and customer provided value (through co-creation). The interrelations among different stakeholders and resources require that methodologies, processes, and tools be dynamically tailored and delivered for either foreseen or newly discovered services to rapidly adapt to changing enterprise and end-user environments.

Even in the case of static, predetermined, interaction rules, the major problems faced in the definition, design, and implementation of services have been in understanding the integration needs among different systems, system entities, stakeholders, and in defining the information flows required for the governance, operations, administration, management and provisioning (OAM&P) of the service. (Maier 1998; Jamshidi 2008; Pineda 2010; Luzeaux and Ruault 2013). Thus, the 21st century technology-intensive services are “information-driven, customer centric, e-oriented, and productivity-focused" as discussed by Chesbrough (2011), Chang (2010), Tien and Berg (2003), and Hipel et al. (2007). A detailed discussion of these characteristics is given in the Value of Service Systems Engineering article within this KA.

Service Systems Engineering Knowledge Area Topics

This knowledge area (KA) describes best practices in SSE during the service design process and outlines current research on methods, processes, and tools. It does not attempt to describe the initial efforts and research in service science that were proposed and introduced by International Business Machines (IBM) (Maglio and Spohrer 2008), but it does recognize their leadership in championing these concepts in undergraduate and graduate curricula.

The rest of the KA is organized in the following way:

The Service Systems Background article presents some background on the transition from a manufacturing economy toward the service economy brought by the World Wide Web through co-creation of end-user value. It describes how this transformation is impacting industries, such as healthcare, agriculture, manufacturing, transportation, supply chain, environmental, etc. The article also describes the scope of the SSE discipline's contributions to meeting the needs of the service sector companies in strategic differentiation and operational excellence (Chang 2010) by pointing out some differences between product-oriented systems engineering and SSE.

The Fundamentals of Services and Properties of Services articles take the reader through a general discussion of services and current attempts to classify different types of services, in particular, attention is paid to the properties of service systems for the service sector, such us transportation, environmental and energy services, consulting services, healthcare, etc.

The Scope of Service Systems Engineering and Value of Service Systems Engineering articles cover the value of SSE, defining (or using when available) service architecture frameworks, and the stages of the service development process from concept to life cycle management.

The Service Systems Engineering Stages article summarizes the major SSE process activities that need to be carried out during the service design process and the needed output (work products) in each of the service design process stages.

Service Innovation and Value-Co-creation

Service innovation has several dimensions. Service innovation can come about through the creation of a service concept which is sufficiently different that it is not merely an improved service, but in reality is a new service concept. To maintain the rigor and value of innovation, it is necessary to distinguish between an improved service, which may generate some additional value, and a truly new and innovative service concept, which may generate a great deal of value. Dr. Noriaki Kano, a renowned quality management guru, has suggested that every service concept has its inherent attributes and we should strive to continuously improve upon these; but this is not innovation (Kano 1996).

To be innovative, the change in a value proposition cannot be incremental, but it must be enough to significantly impact customer and competitor behavior (e.g., new market creation). Value innovation involves a shift in perspective of customer needs that requires a rethinking of what service value proposition is delivered (Kano 1996).

Innovation can also come through a significant change in the way or the reason the customer is engaged or connected. In a service value chain the customer may well change from being just a receiver of service value to becoming a co-creator, or an active participant in the design and delivery, i.e., service transaction of service value. At the retail level, when a customer designs the time, route, and price selection for a plane ticket purchased online, he is co-creating the service. Value innovation involves a shift in perspective of customer needs that requires a rethinking of how a service value proposition is delivered (Bettencourt 2010).

Finally, service innovation can come through significant changes in the way the enterprise is organized to create a service value proposition from concept through delivery. A considerable improvement in the enterprise structure and/or governance can be seen as innovation. Value innovation involves a shift in perspective of customer needs that requires a rethinking of how an enterprise organizes to support a service value proposition.

Continuous improvement can be reasonably planned and predicted while innovation and breakthroughs cannot. The most effective way to obtain innovation and breakthroughs is to encourage the culture, environment, and atmosphere that are conducive to innovation and breakthroughs. Innovative co-creation requires the integration of people, ideas, and technology for the purpose of creating value for themselves, their customers, companies, and society.

The lone inventor sees a problem and must work to create the solutions to all dimensions of the problem. Co-creators see the problem and realize that there may already be several creators, each already having a piece of the solution. Co-creation embraces the value of things “not invented here” because of the velocity they can bring to ideation and time to market. This service innovation process is facilitated by modern mass (and at the same time, personal) communication technology evident in social networking platforms.

Towards a Discipline of Service Systems Engineering

Mindful of the evolution taking place in the global economy and the world markets, it would be futile to attempt covering all the major advances and the boundless possibilities in the services sector for the rest of the century. The services sector covers wide areas of application studied in many different fields (e.g., business science, social science, cognitive science, political science, etc.). The field of service systems, a trans-disciplinary analysis and study of services, was only introduced 10 to 15 years ago. As a consequence, much of the existing literature on services and service-innovation is scattered. The main objective of this KA is to document the systems engineering processes, methodologies, and existing tools as applied to the service design process, and to introduce critical SSE challenges and research areas.

References

Works Cited

AT&T SRP. 2008. *Technical Approach to Service Delivery*. General Services Administration, AT&T Bridge Contract No. GS00Q09NSD0003. Accessed on June 1, 2011. Available at: http://www.corp.att.com/gov/contracts/fts_bridge/technical/07_vol_I_section_1.pdf.

Bettencourt, L. 2010. *Service Innovation: How to Go from Customer Needs to Breakthrough Services*. New York, McGraw-Hill Professional. July 2010.

Booz, Allen, and Hamilton. 1982. *New Products Management for the 1980s*. New York, NY, USA: Booz, Allen, and Hamilton Inc.

Chang, C.M. 2010. *Service Systems Management and Engineering: Creating Strategic Differentiation and Operational Excellence*. New York, NY, USA: John Wiley & Sons, Inc.

Chesbrough, H. 2011. *Open Services Innovation: Rethinking Your Business to Grow and Compete in a New Era*. San Francisco, CA, USA: Jossey-Bass.

Eppinger, S. 2001. "Innovation at the Speed of Information" *Harvard Business Review*. 79 (1): 149-158.

Freeman, R.L. 2004. *Telecommunication Systems Engineering*, 4th ed. New York, NY, USA: John Wiley & Sons.

Hipel, K.W., M.M. Jamshidi, J.M. Tien, and C.C. White. 2007. *"The Future of Systems, Man, and Cybernetics: Application Domains and research Methods*. IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews. 37 (5): 726-743.

Jamshidi M, *System of Systems Engineering: Innovations for the Twenty-First Century*. New York, NY, USA: John Wiley & Sons. November 2008.

Johnson, S.P., L.J. Menor, A.V. Roth, and R.B. Chase. 2000. "A critical evaluation of the new service development process: integrating service innovation and service design," in Fitzsimmons, J.A., and M.J. Fitzsimmons (eds.). *New Service Development - Creating Memorable Experiences*. Thousand Oaks, CA, USA: Sage Publications. p. 1-32.

Kano, N. 1996. *Guide to TQM in Service Industry*. Tokyo, Japan: Asian Productivity Organization.

Lin, F.R., and P.S Hsieh. 2011. A SAT View on New Service Development." *Service Science*. 3 (2): 141-157.

Luzeaux, D. and Ruault. J,. *System of Systems*. New York, NY, USA: John Wiley & Sons. March 2013.

Maglio, P., C. Kieliszewski, and J. Spohrer. 2010. *Handbook of Service Science*, 1st ed. New York, NY, USA: Springer Science + Business Media.

Maglio, P., and J. Spohrer. 2008. "Fundamentals of Service Science." *Journal of the Academy of Marketing Science*. 36 (1): 18-20.

Maier, M.W., 1998. "Architecting Principles for System of Systems." *Systems Engineering*. 1 (4): 267-284.

Pineda, R. 2010. "Understanding Complex Systems of Systems Engineering." Presented at Fourth General Assembly, Cartagena Network of Engineering, September 21-24, 2010, Metz, France.

Spohrer, J.C. 2011. "Service Science: Progress & Directions." Presented at International Joint Conference on Service Science, May 25-27, 2011, Taipei, Taiwan.

Tien, J.M., and D. Berg. 2003. "A Case for Service Systems Engineering." *Journal of Systems Science and Systems Engineering*. 12 (1): 13-38.

Vargo, S.L., and R.F. Lusch. 2004. "The Four Service Marketing Myths – Remnants of a Goods-Based Manufacturing Model." *Journal of Service Research*. 6 (4): 324-335.

Vargo, S.L., and M.A. Akaka. 2009. "Service-Dominant Logic as a Foundation for Service Science: Clarifications." *Service Science*. 1 (1): 32-41.

Whitten, J., and L. Bentley. 2007. *Systems Analysis and Design Methods*. New York, NY, USA: McGraw-Hill Higher Education.

Primary References

Maglio, P., C. Kieliszewski, and J. Spohrer. 2010. *Handbook of Service Science*, 1st ed. New York, NY, USA: Springer Science + Business Media.

Tien, J.M., and D. Berg. 2003. "A Case for Service Systems Engineering." *Journal of Systems Science and Systems Engineering*. 12 (1): 13-38.

Vargo, S.L., and R.F. Lusch. 2004. "The Four Service Marketing Myths – Remnants of a Goods-Based Manufacturing Model." *Journal of Service Research*. 6 (4): 324-335.

Additional References

AT&T SRP. 2008. *Technical Approach to Service Delivery*. General Services Administration, AT&T Bridge Contract No. GS00Q09NSD0003. Accessed on June 1, 2011. Available at: http://www.corp.att.com/gov/contracts/fts_bridge/technical/07_vol_I_section_1.pdf.

Booz, Allen, and Hamilton. 1982. *New Products Management for the 1980s*. New York, NY, USA: Booz, Allen, and Hamilton Inc.

Chang, C.M. 2010. *Service Systems Management and Engineering: Creating Strategic Differentiation and Operational Excellence*. New York, NY, USA: John Wiley & Sons, Inc.

Eppinger, S. 2001. "Innovation at the Speed of Information." *Harvard Business Review*. 79 (1): 149-158.

Freeman, R.L. 2004. *Telecommunication Systems Engineering*, 4th ed. New York, NY, USA: John Wiley & Sons.

Hipel, K.W., M.M. Jamshidi, J.M. Tien, and C.C. White. 2007. "The Future of Systems, Man, and Cybernetics: Application Domains and research Methods." *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*. 37 (5): 726-743.

Johnson, S.P., L.J Menor, A.V. Roth, and R.B. Chase. 2000. "A critical evaluation of the new service development process: integrating service innovation and service design," in Fitzsimmons, J.A., and M.J. Fitzsimmons (eds.). *New Service Development - Creating Memorable Experiences*. Thousand Oaks, CA, USA: Sage Publications. p. 1-32.

Kano, N. 1996. *Guide to TQM in Service Industry*. Tokyo, Japan: Asian Productivity Organization.

Lin, F.R., and P.S Hsieh. 2011. A SAT View on New Service Development." *Service Science*. 3 (2): 141-157.

Luzeaux, D. and Ruault, J. 2013. *Systems of Systems*, Wiley.

Maier, M.W. 1998. "Architecting Principles for System of Systems." *Systems Engineering*. 1 (4): 267-284.

Maglio, P., and J. Spohrer. 2008. "Fundamentals of Service Science." *Journal of the Academy of Marketing Science*. 36 (1): 18-20.

Pineda, R. 2010. "Understanding Complex Systems of Systems Engineering." Presented at Fourth General Assembly, Cartagena Network of Engineering, September 21-24, 2010, Metz, France.

Spohrer, J.C. 2011. "Service Science: Progress & Directions." Presented at International Joint Conference on Service Science, May 25-27, 2011, Taipei, Taiwan.

Vargo, S.L., and M.A. Akaka. 2009. "Service-Dominant Logic as a Foundation for Service Science: Clarifications." *Service Science*. 1 (1): 32-41.

Whitten, J., and L. Bentley. 2007. *Systems Analysis and Design Methods*. New York, NY, USA: McGraw-Hill Higher Education.

< Previous Article | Parent Article | Next Article >

SEBoK v. 2.1, released 31 October 2019

Service Systems Background

Economies are pre-disposed to follow a developmental progression that moves them from heavy proportional reliance on agriculture and mining toward the development of manufacturing, and finally toward more service-based economic activity. As reported by the Organization for Economic Co-Operation and Development (OECD) in its “Science, Technology, and Industry (STI) Forum on The Service Economy":

> *The reason that we see a services economy today, and gather to talk about it and recognize its importance is because technology has allowed service industries to gain the operational leverage that manufacturing achieved 100 years ago. In addition to banks, health systems, telephone and telecommunications networks, and distribution and retailing firms are further examples of sectors that have been able to benefit from economies of scale. As a result, we are now living in a world where global-scale service companies exist for the first time, whereas we have seen global manufacturing companies for 50 years or more.* (OECD 2000, 8)

Evolution Toward Service-Based Economies

The typical industry example given of this progression toward services is the company International Business Machines (IBM). Even though IBM still produces hardware, they view their business as overwhelmingly service-oriented wherein hardware plays only an incidental role in their business solutions services; the fastest line of business growth within IBM has been the business-to-business (B2B) services: information technology (IT); for example, data centers and call centers; business process outsourcing/re-engineering; systems integration; and organizational change.

Business to government (B2G) is forecasted to have the fastest growth in the years to come (Spohrer 2011). For IBM, this trend started in 1989 with the launch of business recovery services; it accelerated with the acquisition of Price-Waterhouse Coopers Consultants in 2002 and culminated with the 2005 sale of the laptop (ThinkPad) manufacturing, their last major hardware operation.

IBM exemplifies the services trend which has accelerated in the last 25-30 years and as of 2006, the services produced by private industry accounted for 67.8% of U.S. gross domestic product (GDP). The top sub-sectors included real estate, financial, healthcare, education, legal, banking, insurance, and investment. Production of goods accounted for 19.8% of GDP. The top product sub-sectors included manufacturing, construction, oil and gas, mining, and agriculture (Moran 2006).

Beginning in the mid-1990s, the concept of a product-service system (PSS) started to evolve. PSSs have been adopted by businesses interested in using the model to bring not only added value to their existing offerings, but capital-intensive, environmentally favorable products to market (Mont and Tukker 2006).

There are some definitional issues in any discussion of PSS, including the fact that services can sometimes be considered as products, and services invariably need physical products to support their provisioning or delivery (2006). A PSS is comprised of tangibles and intangibles (activities) in combination to fulfill specific customer requirements, or ideally, to allow applications to be co-created flexibly by linking loosely coupled agents, typically over a network (Domingue et al. 2009). Research has shown that manufacturing firms are more amenable to producing "results" rather than solely products as specific artifacts and that end users are more amenable to consuming such results (Cook 2004; Wild et al. 2007).

The popularity of wikis, blogs, and social networking tools is strong evidence that "Enterprise 2.0" is already well under way; Andrew McAfee describes Enterprise 2.0 as "the use of emergent social software platforms within companies, or between companies and their partners or customers" (McAfee 2009). However, the integrated access to people, media, services, and things, provided by the Future Internet, will enable new styles of societal and economic interactions at unprecedented scales, flexibility, and quality. These applications will exploit the wisdom of crowds and allow for mass collaboration and value co-creation.

The future internet will provide location independent, interoperable, scalable, secure, and efficient access to a coordinated set of services (Tselentis et al. 2009), but such a broad vision demands a sound and well-defined approach for management and governance.

Current application service providers like Amazon, Facebook, Twitter, eBay, and Google must mediate between the business challenges enabled by network and IT convergence and customers (enterprise or consumer) demanding new and more value-adding services enabled by social networks (TMFORUM 2008). The differences between IT and communications technologies are disappearing; internally-focused processes (back-stage processes) for operations optimization are now being strongly tied to the customer facing (front-stage) processes for value co-creation and delivery. In this scenario, the enterprise's internal organization and employees are embedded in the service value chain to benefit customers and stakeholders. In the service-dominant logic (S-DL) for marketing (Vargo and Lusch 2004), service is the application (through deeds, processes, and performances) of specialized operant resources (knowledge and skills) for the benefit of another entity or the entity itself. The emphasis is on the process of doing something for, and with, another entity in order to create value; a service system is thus a system of interacting and interdependent parts (people, technologies, and organizations) that is externally oriented to achieve and maintain a sustainable competitive advantage (IFM 2008; Maglio and Spohrer 2008).

The future internet is expected to be more agile, scalable, secure, and reliable, demanding rapidly emerging applications/services with different requirements and implications for the Future Internet design that pose a significant set of problems and challenges, in particular, "the fragmentation of knowledge and the isolation of the specialist as well as the need to find new approaches to problems created by earlier 'solution of problems,'" (Skyttner 2006). The service systems engineering discipline may inform the discussion and offer potential multidisciplinary environments and trans-disciplinary solutions.

The internet has been successfully deployed for several decades due to its high flexibility in running over different kinds of physical media and in supporting different high-layer protocols and applications, including traditional file transfer, email, and client-server-based Web applications, among others.

Business Dependence on Service Systems

Most people and enterprises are heavily dependent on service interactions, including entertainment, communications, retail, education, healthcare, etc., brought about by emerging services, such as video on demand, web conferencing, time-shift services, place-shift and device-shift services, enterprise applications (e.g., enterprise resource planning (ERP), customer relationship management (CRM), manufacturing resource management (MRM), software configuration management (SCM), etc.), software as a service (SaaS), platform as a service (PaaS), cloud services, peer-to-peer (P2P) services, etc. A common denominator in the set of services mentioned is that applications are offered as services by the interaction of service system entities and thus they are service based applications (SBA).

Thus, "A service based application is obtained by composing various service system entities to satisfy the desired functionality" (Andrikopoulos et al. 2010). SBAs are heavily dependent on web services development, such as Web services 2.0 (WS). Software systems engineering (SwSE) plays a very important role in a business dependent on a service system. However, another important role is played by human interfaces, organizational development and technology development; for instance, governance (rules & regulations) and technology research and development are required for future services in healthcare services, intelligent transportation services, environmental services, energy services, etc. to address societal challenges of the 21st century (sustainability, energy, etc.) as presented by (Vest 2010) if we were to face those challenges as an ecosystem.

Service System Example

In an intelligent transport system-emergency transportation operation (ITS-ETO), the service goal is to provide safe evacuation, prompt medical care, and improved emergency management service. Typically, a traveler can request service through an emergency call or automated crash report feature, or a public safety officer on location can request service based on customer features and access rights.

The ITS-ETO service system utilizes advances in communication and information systems (technology and information enabler) to access essential, real-time data about conditions on routes throughout the affected area and coordinate operational and logistical strategies in cooperation within all service entities (organization processes). In a critical emergency situation, when patient conditions are continuously changing, ITS can help identify the appropriate response and get the correct equipment (infrastructure enabler), such as a helicopter and emergency personnel (people enabler), to and from the scene quickly and safely.

Efficient and reliable voice, data, and video communications (application enabler) further provide agencies with the ability to share information related to the status of the emergency, the operational conditions of the transportation facilities, and the location of emergency response resources to help communicate and coordinate operations and resources in real time. Advances in logistical and decision-making tools can enable commanders and dispatchers to implement strategies as conditions change (decision making).

It is also critical to receive information on the environmental conditions (storm, hazardous materials, multi-vehicle crashes, etc.) and/or road closures when coordinating evacuations. The availability of real-time data about transportation conditions, coupled with decision-making tools, enables more effective responses and coordination of resources during emergencies. ITS-ETO also enhances the ability of transportation agencies to coordinate responses with other stakeholders/entities.

As a result, increased data accuracy, timeliness, and automation leads to better use of resources, and reuse of exchanges, resulting in time and cost savings. Enhanced response and management leads to greater situational awareness and more effective reactions with the ability to identify and utilize the appropriate equipment, resulting in a more efficient response at the right time (output) (US DOT 2011). Figure 1 below lists the possible stakeholders in a service system.

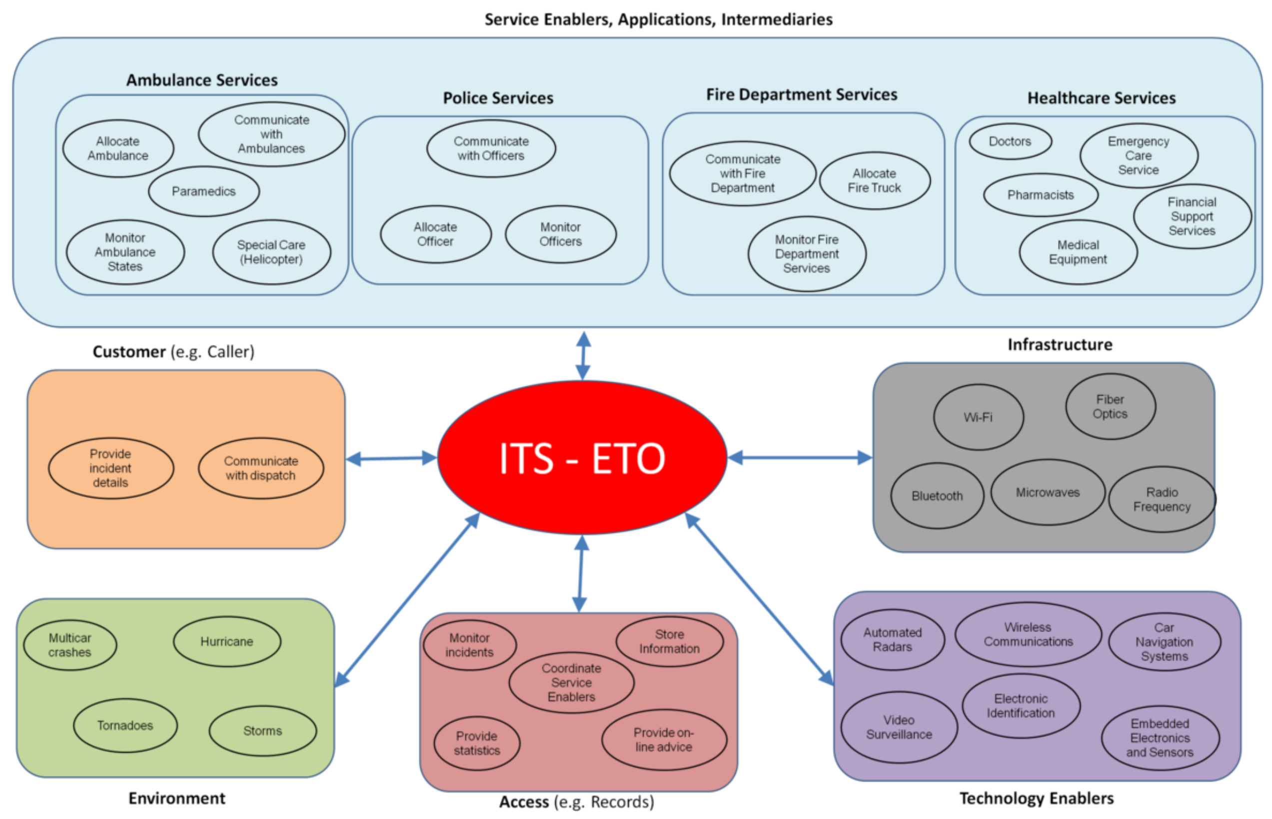


Figure 1. Service System Context Diagram. (SEBoK Original)

As seen in the above example, the service activities are knowledge-intensive; well defined linkages (including access rights) and relationships among different entities give rise to the needed service systems interactions for the service system to be successful. As the world becomes more widely interconnected, and people become better educated, the services networks created by the interaction of the service systems will be accessible from anywhere, at any time, by anyone with the proper access rights.

Knowledge agents are then humans creating new linkages of information to create new knowledge which “can later be embedded in other people, technology, shared information, and organizations." Thus, people can be considered as individual service systems with “finite life cycles, identities (with associated histories and expectations), legal rights and authority to perform certain functions, perform multitasking as a way to increase individual productivity output in a finite time, and engage in division-of-labor with others to increase collective productive output in finite time” through service transactions enabled by their access rights (Spohrer and Kwan 2008).

References

Works Cited

Andrikopoulos, V., A. Bucchiarone, E. Di Nitto, R. Kazhamiakin, S. Lane, V. Mazza, and I. Richardson. 2010. "Chapter 8: Service Engineering," in *Service Research Challenges and Solutions for the Future Internet S-Cube – Towards Engineering, Managing and Adapting Service-Based Systems*, edited by M. Papazoglou, K. Pohl, M. Parkin, and A. Metzger. Berlin Heidelberg, Germany: Springer-Verlag. p. 271-337.

Cook, M. 2004. "Understanding The Potential Opportunities Provided by Service-Orientated Concepts to Improve Resource Productivity," in *Design and Manufacture for Sustainable Development 2004*, edited by T. Bhamra and B. Hon. Bury St. Edmonds, Suffolk, UK: Professional Engineering Publishing Limited. p. 123-134.

Domingue, J., D. Fensel, J. Davies, R. González-Cabero, and C. Pedrinaci. 2009. "The Service Web: a Web of Billions of Services," in *Towards the Future Internet - A European Research Perspective*, edited by G. Tselentis, J. Domingue, A. Galis, A. Gavras, D. Hausheer, S. Krco, V. Lotz, and T. Zehariadis. Amsterdam, The Netherlands: IOS Press.

IFM. 2008. *Succeeding through Service Innovation: A service perspective for education, research, business and government*. University of Cambridge Institute for Manufacturing (IfM) and International Business Machines Corporation (IBM) report. Cambridge Service Science, Management and Engineering Symposium, July 14-15, 2007, Cambridge, UK.

Maglio P., and J. Spohrer 2008. "Fundamentals of Service Science." *Journal of the Academy of Marketing Science*. 36 (1): 18-20. DOI: 10.1007/s11747-007-0058-9.

Maglio, P., Weske, M., Yang, J. and Fantinato, Marcelo. 2010. *8th International Conference on Service Oriented Computing (ICSOC 2010)*. Lecture Notes in Computer Science. Vol. 6470. Springer-Verlag, San Francisco, California. December 2010.

McAfee, A. 2009. *Enterprise 2.0: New Collaborative Tools for Your Organization's Toughest Challenges*. Boston, MA, USA: Harvard Business School Press.

Mont, O., and A. Tukker. 2006. "Product-Service Systems." *Journal of Cleaner Production*. 14 (17): 1451-1454.

Moran, M. 2006. *Servicizing Solar Panels.* Industry Course Report. Lund University International Master's Programme in Environmental Studies and Sustainability Science Department (LUMES), Lund University, Sweden.

Organization for Economic Co-operation and Development (OECD). 2000. *The Service Economy*. Science Technology Industry (STI) Business and Industry Policy Forum Series. Paris, France: OECD. Available: http://www.oecd.org/dataoecd/10/33/2090561.pdf.

Skyttner, L. 2006. *General Systems Theory: Perspectives, Problems, Practice*, 2nd ed. Singapore: World Scientific Publishing Company.

Spohrer, J., and S.K. Kwan. 2009. "Service Science, Management, Engineering, and Design (SSMED): An Emerging Discipline - Outline & References." *International Journal of Information Systems in the Service Sector*. 1 (3): 1-31.

Spohrer, J.C. 2011. "Service Science: Progress & Directions." Presented at the International Joint Conference on Service Science, May 25-27, 2011, Taipei, Taiwan.

TM Forum. 2008. *Service Delivery Framework (SDF) Overview, Release 2.0*. Morristown, NJ: TeleManagement Forum. Technical Report 139.

Tselentis, G., J. Domingue, A. Galis, A. Gavras, D. Hausheer, S. Krco, V. Lotz, and T. Zahariadis (eds.). 2009. *Towards the Future Internet - A European Research Perspective*. Amsterdam, The Netherlands: IOS Press.

US DOT. 2011. "Emergency Transportation Operations." Research and Innovative Technology Administration. Accessed June 23, 2011. Last updated June 16, 2011. Available: http://www.its.dot.gov/eto/index.htm.

Vargo, S.L., and R.F. Lusch. 2004. "The Four Service Marketing Myths – Remnants of a Goods-Based Manufacturing Model." *Journal of Service Research*. 6 (4): 324-335.

Vest, C.M., 2013. "Educating Engineers for 2020 and Beyond" .*The Bridge*. Washington DC, National Academy of Engineering.

Wild, P.J., J. Jupp, W. Kerley, C. Eckert, and P.J. Clarkson. 2007. "Towards A Framework for Profiling of Products and Services." Presented at 5th International Conference on Manufacturing Research (ICMR), September 11-13, 2007, Leicester, UK.

Primary References

IFM. 2008. *Succeeding through Service Innovation: A service perspective for education, research, business and government*. University of Cambridge Institute for Manufacturing (IfM) and International Business Machines Corporation (IBM) report. Cambridge Service Science, Management and Engineering Symposium, July 14-15, 2007, Cambridge, UK.

Organization for Economic Co-operation and Development (OECD). 2000. *The Service Economy*. Science Technology Industry (STI) Business and Industry Policy Forum Series. Paris, France: OECD. Available: http:// www.oecd.org/dataoecd/10/33/2090561.pdf.

Vargo, S.L., and R.F. Lusch. 2004. "The Four Service Marketing Myths – Remnants of a Goods-Based Manufacturing Model." *Journal of Service Research*. 6 (4): 324-335.

Additional References

Andrikopoulos, V., A. Bucchiarone, E. Di Nitto, R. Kazhamiakin, S. Lane, V. Mazza, and I. Richardson. 2010. "Chapter 8: Service Engineering," in *Service Research Challenges and Solutions for the Future Internet S-Cube – Towards Engineering, Managing and Adapting Service-Based Systems*, edited by M. Papazoglou, K. Pohl, M. Parkin, and A. Metzger. Berlin Heidelberg, Germany: Springer-Verlag. p. 271-337.

Cook, M. 2004. "Understanding The Potential Opportunities Provided by Service-Orientated Concepts to Improve Resource Productivity," in *Design and Manufacture for Sustainable Development 2004*, edited by T. Bhamra and B. Hon. Bury St. Edmonds, Suffolk, UK: Professional Engineering Publishing Limited. p. 123-134.

Domingue, J., D. Fensel, J. Davies, R. González-Cabero, and C. Pedrinaci. 2009. "The Service Web: a Web of Billions of Services," in *Towards the Future Internet - A European Research Perspective*, edited by G. Tselentis, J. Domingue, A. Galis, A. Gavras, D. Hausheer, S. Krco, V. Lotz, and T. Zehariadis. Amsterdam, The Netherlands: IOS Press.

Maglio P., and J. Spohrer 2008. "Fundamentals of Service Science." *Journal of the Academy of Marketing Science*. 36 (1): 18-20. DOI: 10.1007/s11747-007-0058-9.

McAfee, A. 2009. *Enterprise 2.0: New Collaborative Tools for Your Organization's Toughest Challenges*. Boston, MA, USA: Harvard Business School Press.

Mont, O., and A. Tukker. 2006. "Product-Service Systems." *Journal of Cleaner Production*. 14 (17): 1451-1454.

Moran, M. 2006. *Servicizing Solar Panels.* Industry Course Report. Lund University International Master's Programme in Environmental Studies and Sustainability Science Department (LUMES), Lund University, Sweden.

Skyttner, L. 2006. *General Systems Theory: Perspectives, Problems, Practice*, 2nd ed. Singapore: World Scientific Publishing Company.

Spohrer, J., and S.K. Kwan. 2009. "Service Science, Management, Engineering, and Design (SSMED): An Emerging Discipline - Outline & References." *International Journal of Information Systems in the Service Sector*. 1 (3): 1-31.

Spohrer, J.C. 2011. "Service Science: Progress & Directions." Presented at the International Joint Conference on Service Science, May 25-27, 2011, Taipei, Taiwan.

TM Forum. 2008. *Service Delivery Framework (SDF) Overview, Release 2.0*. Morristown, NJ: TeleManagement Forum. Technical Report 139.

Tselentis, G., J. Domingue, A. Galis, A. Gavras, D. Hausheer, S. Krco, V. Lotz, and T. Zahariadis (eds.). 2009. *Towards the Future Internet - A European Research Perspective*. Amsterdam, The Netherlands: IOS Press.

US DOT. 2011. "Emergency Transportation Operations." Research and Innovative Technology Administration. Accessed June 23, 2011. Last updated June 16, 2011. Available: http://www.its.dot.gov/eto/index.htm.

Wild, P.J., J. Jupp, W. Kerley, C. Eckert, and P.J. Clarkson. 2007. "Towards A Framework for Profiling of Products and Services." Presented at 5th International Conference on Manufacturing Research (ICMR), September 11-13, 2007, Leicester, UK.

< Previous Article | Parent Article | Next Article >

SEBoK v. 2.1, released 31 October 2019

Fundamentals of Services

Lead Authors: *Ricardo Pineda, Bud Lawson*

Services are activities that cause a transformation of the state of an entity (a person, product, business, region, or nation) by mutually agreed terms between the service provider and the customer. Individual services are relatively simple, although they may require customization and significant back-stage support (e.g., database, knowledge management, analysis, forecasting, etc.) to assure quality and timely delivery. Product services are also relatively straightforward as product specifications, performance standards, quality control, installation guidelines, and maintenance procedures require good communication and understanding between providers and users. Business services can be rather complex; some may involve intensive negotiations, work process alignment, quality assurance, team collaboration, and service coproduction. Moreover, Chang (2010) states that: “Regional and National services are even more complex, as they may affect policy, custom regulations, export permits, local business practices, logistics, distribution, and other such issues" (see also Complexity).

Service Systems

The service and/or set of services developed and accessible to the customer (individual consumer or enterprise) are enabled by a service system. Service system stakeholders may interact to create a particular service value chain to be delivered with a specific objective (Spohrer and Maglio 2010). Service system entities dynamically configure four types of resources: people, technology/environment infrastructure, organizations(glossary)/institutions, and shared information/symbolic knowledge. Service systems can be either formal or informal in nature. In the case of formal service systems, the interactions are contracted through service level agreements (SLA). Informal service systems can promise to reconfigure resources without a written contractual agreement; in the case of the emergency transports operations example discussed in the Service Systems Background article, there is no formal contractual agreement (i.e., SLA) between the user requesting the service and the agency providing the service other than a “promise” for a quick and efficient response. SLAs are written contracts between and among service system entities, as well as the legal system for enforcing the contracts. The study of informal service systems contains the study of relationships (communications, interactions, and promises) between service systems and social systems, cultural norms and beliefs, as well as political systems that can maintain those relationships (Spohrer and Kwan 2008). The resources are either physical or non-physical and have rights or no rights. See Figure 1 below:

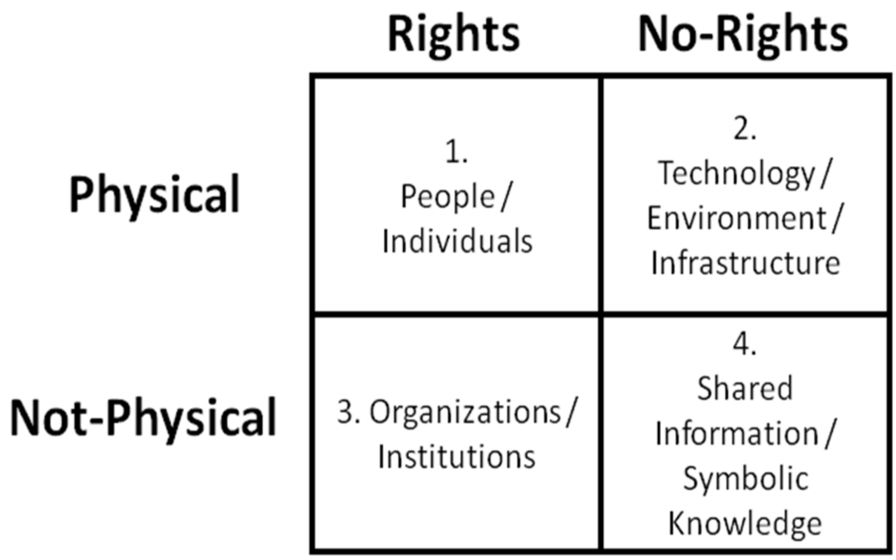


Figure 1. Service System Resources (Spohrer 2011). Reprinted with permission of Dr. James C. Spohrer. All other rights are reserved by the copyright owner.

Service Value Chain

SLAs and policies specify the conditions under which services system entities reconfigure access rights to resources by mutually agreed value propositions. Current management frameworks typically focus on single service system entity interfaces. They neither use SLAs for managing the implementation and delivery of services nor do they recognize/support the fact that many services may be composed of lower-level services, involve third-party providers, and rely on possibly complex relationships and processes among participating businesses, information communications, and technologies (CoreGRID 2007). While SLAs are mapped to the respective customer requirements, policies are provider-specific means to express constraints and rules for their internal operations. These rules may be independent of any particular customer (Theilmann 2009).

In service systems practice, we describe the service value chain in terms of links among the entities connected via the Network Centric operations of service systems. For instance, value could then be created and delivered in terms of e-services, such as business-to-business (B2B), business to consumer (B2C), business to government (B2G), government-to-business (G2B), government-to-government (G2G), government-to-consumer (G2C), etc. The emerging service in this case interacts or “co-produces” with their customer via the World Wide Web as compared to the physical environment in which the traditional, or brick and mortar, service enterprises interact with their customers.

The services sector requires information as input, involves the customer at the production/delivery stage, and employs mostly qualitative measures to assess its performance, i.e., technology-intensive services are “information-driven, customer centric, e-oriented, and productivity-focused" (Tien and Berg 2003; Hipel et al. 2007; Chesbrough 2011). Chang (2010) defines these features in this manner:

- **Information Driven:** The creation, management, and sharing of information is crucial to the design, production, and delivery of services.
- **Customer Centric:** Customers are generally the co-producer of the services, as in the case of self-service. Customers require a certain degree of self-adaptation or customization and customers must be satisfied with the rendered services.

- **E (electronics) Oriented:** Services are becoming increasingly e-oriented. Thus, e-access, e-commerce, and e-customer management are crucial to e-services.
- **Productivity Focused:** Both efficiency and effectiveness are important in the design, delivery, and support of services.
- **Value Adding:** Services need to provide some value for the target clients. For profit-seeking service companies, the value produced for customers assures the company's profitability. For non-profit service entities, the value produced for customers reinforces the quality of a service entity's policy.

A service system is defined by its value co-creation chain in which stakeholders work in open collaboration to deliver consistently high quality service according to business goals, service goals, and customer goals. A value proposition can be viewed as a request from one service system to another to run an algorithm (the value proposition) from the perspectives of multiple stakeholders according to culturally determined value principles. The four primary stakeholder's perspectives in regards to value are the customer, provider, authority, and the competitors. Figure 2 below depicts value calculations from multiple stakeholder perspectives.

Table 1. Value Calculation from Different Stakeholders' Perspectives (Spohrer 2011). Reprinted with permission of Dr. James C. Spohrer. All other rights are reserved by the copyright owner.

Stakeholder Perspective (the players)	Measure Impacted	Pricing Decision	Basic Questions	Value Proposition Reasoning
1. Customer	Quality (Revenue)	Value Based	Should we? (offer it)	Model of customer: Do customers want it? Is there a market? How large? Growth rate?
2. Provider	Productivity (Profit, Mission, Continuous Improvement, Sustainability)	Cost Plus	Can we? (deliver it)	Model of self: Does it play to our strengths? Can we deliver it profitability to customers? Can we continue to improve?
3. Authority	Compliance (Taxes and Fines, Quality of Fire)	Regulated	May we? (offer and deliver it)	Model of authority: Is it legal? Does it compromise our integrity in any way? Does it create a moral hazard?
4. Competitor (Substitute)	Sustainable Innovation (Market Share)	Strategic	Will we? (invest to make it so)	Model of competitor: Does it put us ahead? Can we stay ahead? Does it differentiate us from the competition?

From an engineering design point of view, the service and business goals are an entry point through which to analyze the business architectures (including organization and processes) needed, which in turn demand alignment between the information technology (IT) components and technology architecture to achieve the goals. From a systems engineering perspective, the next step is to identify service system entities that could participate in the service delivery (people, organizations, technologies, processes, etc.).

Service System Entities

Spath and Fahnrich (2007) defined a service meta-model comprised of nine types of **entities**:

1. **Customers**: customer features, customer attitudes, and customer preferences;
2. **Goals**: business goals, service goals, customer goals, and enterprise culture goals;
3. **Inputs**: physical, human beings, information, knowledge, currency, and constraints;
4. **Outputs**: physical, human beings, information, knowledge, currency, and waste;
5. **Processes**: service provision, service operations, service support, customer relationships, planning and control, and call center management;
6. **Human Enablers**: service providers, support providers, management, and owner organization (enterprise);
7. **Physical Enablers**: owner organization (physical), buildings, equipment, furnishings, and location;
8. **Informatics Enablers**: information, knowledge, procedures and processes, decision support, and skill acquisition; and
9. **Environment**: political factors, economic factors, social factors, technological factors, environmental factors, legal factors (PESTEL), and physical factors.

Thus, a service or service offering is created by the relationships among service system entities (including information flows) through business processes into strategic capabilities that consistently provide superior value to the customer. If we were to represent the service as a network diagram (as in Figure 3 below), then the entities represent the nodes and the links represent the relationships between nodes.

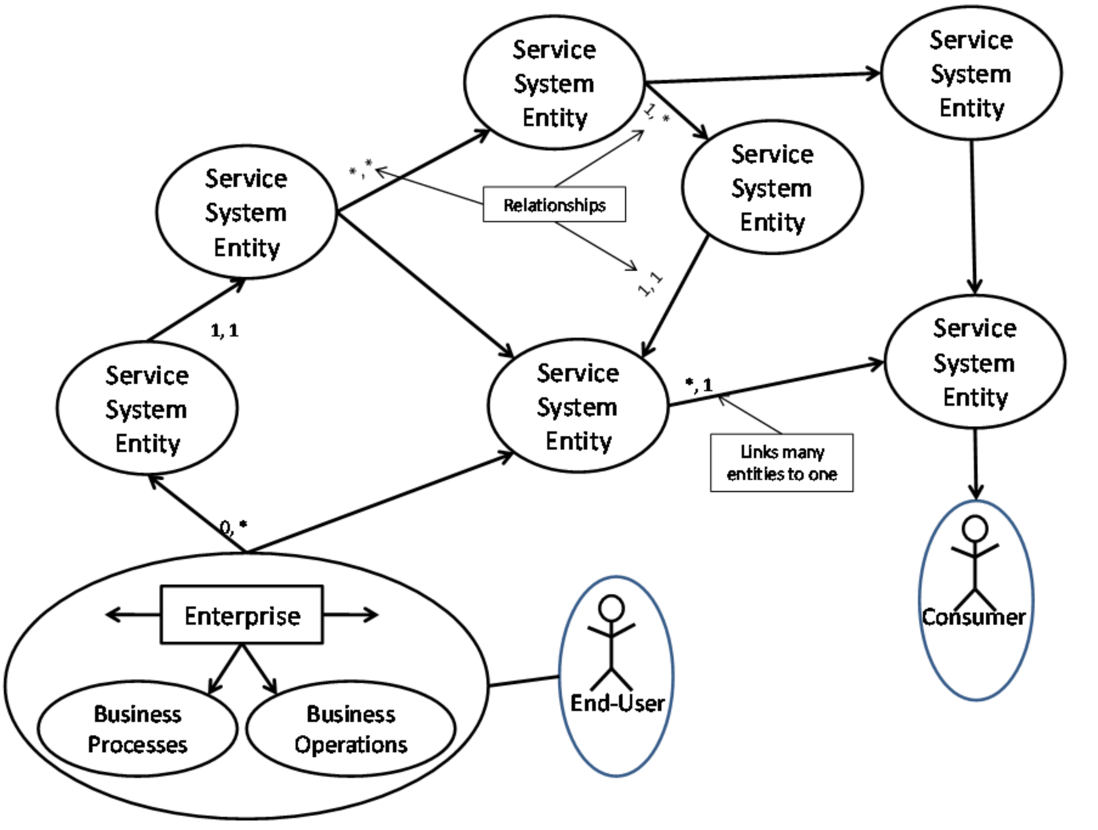


Figure 2. Service Systems Network Diagram. (SEBoK Original)

Service System Hierarchy

Systems are part of other systems which are often expressed by systems hierarchies (Skyttner 2010) to create a multilevel hierarchy, and thus the service system is composed of service system entities that interact through processes defined by governance and management rules to create different types of outcomes in the context of stakeholders with the purpose of providing improved customer interaction and value co-creation. Examples of service system entities are business enterprises, nations, or in the simplest form, a person (consumes and produces services).

Using the hierarchical approach, Spohrer conceptualizes an ecosystem at the highest level in which a service system is an entity of its own. This concept is extended to create the service system hierarchy as described in Figure 4 below (Spohrer 2011; Maglio and Spohrer 2008; Maglio et al. 2010).

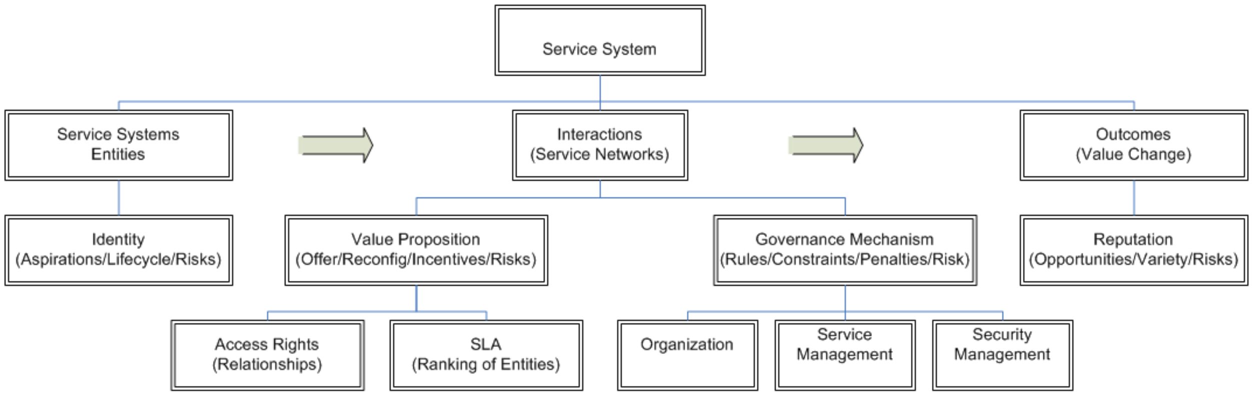


Figure 3. Service System Conceptual Framework (Spohrer 2011). Reprinted with permission of Dr. James C. Spohrer. All other rights are reserved by the copyright owner.

Service System Attributes

The fundamental attributes of a service system include togetherness, structure, behavior, and emergence. As mentioned earlier, today's global economy is very competitive and a service system may be very competitive in a given environment at a given time (the business space). The service system's trajectory should be well controlled as time goes by (Qiu 2009) since services are "real time in nature and are consumed at the time they are co-produced" (Tien and Berg 2003), that is, during service transactions.

The service system should evolve and adapt to the conditions within the business space in a manner which ensures that the customized service behaves as expected. This adaptive behavior of service systems implies that their design must be truly trans-disciplinary:

> *They must include techniques from social science (i.e., sociology, psychology, and philosophy) and management (i.e., organization, economics, and entrepreneurship). As a consequence, Systems, Man, and Cybernetics (SMC) must expand their systems (i.e., holistic oriented), man (i.e., decision-oriented), and cybernetics methods to include and be integrated with those techniques that are beyond science and engineering.* (Hipel et al. 2007)

References

Works Cited

Chang, C.M. 2010. *Service Systems Management and Engineering: Creating Strategic Differentiation and Operational Excellence*. Hoboken, NJ, USA: John Wiley & Sons, Inc.

Chesbrough, H. 2011. *Open Services Innovation: Rethinking Your Business to Grow and Compete in a New Era*. San Francisco, CA, USA: Jossey-Bass.

CoreGRID. 2007. *Using SLA for Resource Management and Scheduling - A Survey. Technical Report 0096*. Jülich & Dortmund, Germany: European Research Network on Foundations, Software Infrastructures and Applications for Large Scale Distributed, GRID and Peer-to-Peer Technologies, Institute on Resource Management and Scheduling. Accessed June 4, 2011. Available: http://www.coregrid.net/mambo/images/stories/TechnicalReports/tr-0096.pdf.

Hipel, K.W., M.M. Jamshidi, J.M. Tien, and C.C. White. 2007. "The Future of Systems, Man, and Cybernetics: Application Domains and Research Methods." *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*. 37 (5): 726-743.

Maglio, P., C. Kieliszewski, and J. Spohrer. 2010. *Handbook of Service Science.* New York, NY, USA: Springer Science and Business Media.

Maglio, P., and J. Spohrer. 2008. "Fundamentals of Service Science." *Journal of the Academy of Marketing Science*. 36 (1): 18-20.

Qiu, R. 2009. "Computational Thinking of Service Systems: Dynamics and Adaptiveness Modeling." *Service Science*. 1 (1): 42-55.

Skyttner, L. 2006. *General Systems Theory: Perspectives, Problems, Practice,* 2nd ed. Singapore: World Scientific Publishing Company.

Spath, D., and K.P. Fähnrich (eds.). 2007. *Advances in Services Innovations*. Berlin & Heidelberg, Germany: Springer-Verlag.

Spohrer, J., and S.K. Kwan. 2009. "Service Science, Management, Engineering, and Design (SSMED): An Emerging Discipline - Outline & References." *International Journal of Information Systems in the Service Sector*, 1 (3): 1-31.

Spohrer, J., and P.P Maglio. 2010. "Chapter 1: Service Science: Toward a Smarter Planet," in *Introduction to Service Engineering*, edited by G. Salvendy and W. Karwowski. Hoboken, NJ: John Wiley & Sons.

Spohrer, J.C. 2011. "Service Science: Progress & Directions." Presented at International Joint Conference on Service Science, May 25-27, 2011, Taipei, Taiwan.

Theilmann, W., and L. Baresi. 2009. "Multi-level SLAs for Harmonized Management in the Future Internet," in *Towards the Future Internet - A European Research Perspective*, edited by G. Tselentis, J. Domingue, A. Galis, A. Gavras, D. Hausheer, S. Krco, V. Lotz, and T. Zehariadis. Amsterdam, The Netherlands: IOS Press.

Tien, J.M., and D. Berg. 2003. "A Case for Service Systems Engineering." *Journal of Systems Science and Systems Engineering*. 12 (1): 13-38.

Vargo, S.L., and M.A. Akaka. 2009. "Service-Dominant Logic as a Foundation for Service Science: Clarifications." *Service Science*. 1 (1): 32-41.

Primary References

Chang, C.M. 2010. *Service Systems Management and Engineering: Creating Strategic Differentiation and Operational Excellence*. Hoboken, NJ, USA: John Wiley & Sons, Inc.

Hipel, K.W., M.M. Jamshidi, J.M. Tien, and C.C. White. 2007. "The Future of Systems, Man, and Cybernetics: Application Domains and Research Methods." *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*. 37 (5): 726-743.

Spath, D., and K.P. Fähnrich (eds.). 2007. *Advances in Services Innovations*. Berlin & Heidelberg, Germany: Springer-Verlag.

Spohrer, J.C. 2011. "Service Science: Progress & Directions." Presented at International Joint Conference on Service Science, May 25-27, 2011, Taipei, Taiwan.

Additional References

Chesbrough, H. 2011. *Open Services Innovation: Rethinking Your Business to Grow and Compete in a New Era*. San Francisco, CA, USA: Jossey-Bass.

CoreGRID. 2007. *Using SLA for Resource Management and Scheduling - A Survey. Technical Report 0096*. Jülich & Dortmund, Germany: European Research Network on Foundations, Software Infrastructures and Applications for Large Scale Distributed, GRID and Peer-to-Peer Technologies, Institute on Resource Management and Scheduling. Accessed June 4, 2011. Available: http://www.coregrid.net/mambo/images/stories/TechnicalReports/tr-0096.pdf.

Maglio, P., C. Kieliszewski, and J. Spohrer. 2010. *Handbook of Service Science.* New York, NY, USA: Springer Science and Business Media.

Maglio, P., and J. Spohrer. 2008. "Fundamentals of Service Science." *Journal of the Academy of Marketing Science*. 36 (1): 18-20.

Qiu, R. 2009. "Computational Thinking of Service Systems: Dynamics and Adaptiveness Modeling." *Service Science*. 1 (1): 42-55.

Skyttner, L. 2006. *General Systems Theory: Perspectives, Problems, Practice,* 2nd ed. Singapore: World Scientific Publishing Company.

Spohrer, J., and S.K. Kwan. 2009. "Service Science, Management, Engineering, and Design (SSMED): An Emerging Discipline - Outline & References." *International Journal of Information Systems in the Service Sector*, 1 (3): 1-31.

Spohrer, J., and P.P Maglio. 2010. "Chapter 1: Service Science: Toward a Smarter Planet," in *Introduction to Service Engineering*, edited by G. Salvendy and W. Karwowski. Hoboken, NJ: John Wiley & Sons.

Theilmann, W., and L. Baresi. 2009. "Multi-level SLAs for Harmonized Management in the Future Internet," in *Towards the Future Internet - A European Research Perspective*, edited by G. Tselentis, J. Domingue, A. Galis, A. Gavras, D. Hausheer, S. Krco, V. Lotz, and T. Zehariadis. Amsterdam, The Netherlands: IOS Press.

Tien, J.M., and D. Berg. 2003. "A Case for Service Systems Engineering." *Journal of Systems Science and Systems Engineering*. 12 (1): 13-38.

Vargo, S.L., and M.A. Akaka. 2009. "Service-Dominant Logic as a Foundation for Service Science: Clarifications." *Service Science*. 1 (1): 32-41.

< Previous Article | Parent Article | Next Article >

SEBoK v. 2.1, released 31 October 2019

Properties of Services

Lead Authors: *Ricardo Pineda, Bud Lawson, Brian Wells*

A service is realized by the service system through the relationships of service system entities that interact (or relate) in a particular way to deliver the specific service via a service level agreement (SLA). Current management frameworks typically only focus on the interfaces of single service system entities. Meanwhile, SLAs are mapped to the respective customer requirements. These policies are provider-specific means to express constraints and rules for their internal operations. These rules may be independent of any particular customer (Theilmann 2009).

Services not only involve the interaction between the service provider and the consumer to produce value, but have other attributes, like an intangible quality of service (e.g., an ambulance service's availability and response time to an emergency request). The demand for a service may have varying loads dependent on the time of day, day of week, season, or other unexpected needs (e.g., natural disasters, product promotion campaigns, etc.). In the US for instance, travel services have peak demands during Christmas week; Mother's day is usually the highest volume handling day for a telecommunications provider and tax services peak during extended periods (January through mid-April). Services cannot be inventoried; they are rendered at the time they are requested.

Additionally, for a business enterprise, delivering the service at the minimum cost while maximizing its profits may be the service objective. In contrast, for a non-profit organization the objective may be to maximize customer satisfaction while optimizing the resources required to render the service (e.g., during a natural disaster). Thus, the design and operations of service systems "is all about finding the appropriate balance between the resources devoted to the systems and the demands placed on the system so that the quality of service to the customer is as good as possible" (Daskin 2010).

Service Level Agreement

A SLA is a set of technical (functional) and non-technical (non-functional) parameters agreed among customers and service providers. SLAs can and do contain administrative level (non-functional) business related parameters, such as SLA duration, service availability for the SLA duration, consequences for variations, failure reporting, priorities, and provisions for modifications to the SLA. However, for service level management, the service level (technical) parameters need to be defined, monitored, and assessed; these parameters may include such things as throughput; quality; availability; security; performance; reliability, for example, mean time between failure (MTBF), maximum downtime, and time-to-repair; and resource allocation.

An SLA represents the negotiated service level requirements (SLR) of the customer and should establish valid and reliable service performance measures since it is usually the basis for effective service level management (SLM). The goal of SLM is to ensure that service providers meet and maintain the prescribed quality of service (QoS). However, care should be taken since in some domains the term QoS refers only to resource reservation control mechanisms rather than the achieved service quality (e.g., internet protocol (IP) networks). Some terms used to mean the "achieved service quality" include quality of experience (QoE), user-perceived performance, and degree of satisfaction of the user; these other terms are more generally used across service domains.

Non-functional properties fall into two basic categories: business properties, such as price and method of payment, and environmental properties, such as time and location. Business and environmental properties are classified as "context properties" by Youakim Badr (Badr et al. 2008). QoS properties are characteristics such as availability, resilience, security, reliability, scalability, agreement duration, response times, repair times, usability, etc. Therefore services evaluation measures are customer oriented and include not only traditional performance metrics (productivity, quality, etc.), but also require a comprehensive analysis of the service system from an end-to-end perspective. Service evaluation typically includes customer demand-supply to ensure economic viability across the lifecycle of the service system. Furthermore, the service delivery is evaluated using the key technical performance metrics listed above, adding also Service Process Measures (provisioning time, time-to-restore/repair, etc.) and Technical Performance Measures (end-to-end response times, latency, throughput, etc.). Finally, the service system's SLAs are then the composition of these categories evaluated on a systemic level to ensure consistency, equity, and sustainability of the service to assure that the desired/contracted SLA for customer satisfaction, value co-creation, and high system robustness are realized. (Spohrer 2011; Tien and Berg 2003; Theilmann and Baresi, 2009)

Service Key Performance Indicators

Service key performance indicators (KPI) are defined and agreed to in the SLA; the service KPIs are decomposed into service process measures (SPM) and technical performance measures (TPM) during the analysis stage of the service systems engineering (SSE) process. In the design process, the KPIs and TPM are allocated to service system entities and their components, as well as to the business processes and their components so as to ensure compliance with SLAs. The allocated measures generate derived requirements (SLR) for the system entities and their relationships, as well as for the service entities' components and the data and information flows required in the service systems to monitor, measure, and assess end-to-end SLA. These allocations ensure that the appropriate performance indicators apply to each of the links in the service value chain.

TPMs are typically categorized by the number of defective parts in a manufacturing service, data transmission latency and data throughput in an end-to-end application service, IP QoS expressed by latency, jitter delay, and throughput; SPMs are typically categorized by service provisioning time, end-to-end response times to a service request (a combination of data and objective feedback), and quality of experience (QoE verified by objective feedback). Together, the KPI (TPM combined with SPM) and perception measures make up the service level management function. A quality assurance system's (QAS) continuous service improvement (CSI), processes, and process quality management and improvement (PQMI) should be planned, designed, deployed, and managed for the capability to continuously improve the service system and to monitor compliance with SLAs (e.g., PQMI, capability maturity model integration (CMMI) (SEI 2007), International Organization for Standardization (ISO) Standards 9001 (ISO/IEC 2008), Telecom Quality Management System Standards (TL 9000) (QuEST Forum 2012), Information Technology Infrastructure Library (ITIL) v. 3 (OGC 2009), etc.).

As discussed earlier, QoS needs to correlate customer perceived quality (subjective measures) with objective SPM and TPM measures. There are several techniques available to help monitor, measure, and assess TPM's, but most are a variation on the theme of culling information from TPM's using, for example, perceptual speech quality measure (PSQM) and perceptual evaluation of video quality (PEVQ) and enhancing or verifying this information with customer or end-user perception of service by extending mean opinion score (MOS) techniques/customer opinion models (Ray 1984). Telecommunication systems engineering (TCSE) played an important role in finding methodologies for correlation between perception and objective measures for the services of the twentieth century; SSE should continue to encourage multidisciplinary participation to equally find methodologies, processes, and tools to correlate perceived service quality with TPM and with SPM for the services of the twenty-first century (Freeman 2004).

Subjective (qualitative) service quality is the customer's perceived conformity of the service with the expected objective. Word-of-mouth, personal needs, and past experiences create customer expectations regarding the service. The customers' perception of the service must be captured via surveys and interviews. The customers' perception of the service is then compared with their expectations for the service; this process captures the perceived service quality. Care should be taken to understand that subjective measures appear to measure customer attitudes, and attitudes may be the result of several encounters with the service, as well as numerous encounters with similar services.

In summary, the SLA documents the SLRs and establishes reliable and valid service performance measures, technical parameters, and the agreed performance levels for the technical parameters. The technical parameters are then monitored and continuously compared against both objective and subjective data culled from multiple internal and external sources (service level management). The goal is not to report the level of service in a given period, but to develop and implement a dynamic system capable of predicting and driving service level improvement over time (i.e., continual service improvement (CSI)).

Evolution of Services

The second, third, and fourth decades of the twenty-first century will almost certainly see similar, and probably accelerated, technology development as seen in the prior three decades. Mass collaboration will become an established mode of operation. The beginnings of mass collaboration have manifested in developments such as value co-creation where loosely entangled actors or entities come together to create value in unprecedented ways, but ways that meet mutual and broader market requirements. Further developments in the technology, use, and acceptance of social media will continue to fuel the acceleration of these developments.

The next decades will see the grounding of concepts, such as crowdsourcing, coined by Jeff Howe in a June 2006 *Wired* magazine article; open innovation, promoted by Henry Chesbrough, a professor and executive director at the Center for Open Innovation at Berkeley; and mass collaboration and open source innovation supported by Enterprise 2.0 tools, as conceived by Wikinomics consultant Don Tapscott.

Roberto Saracco, a telecommunications expert specializing in analyzing economical impacts of technology evolution, argues that: "Communications will be the invisible fabric connecting us and the world whenever and wherever we happen to be in a completely seamless way, connecting us so transparently, cheaply, and effortlessly that very seldom will we think about it." The ubiquity and invisibility of these communications will greatly facilitate the creation and destruction of ad hoc collectives (groups of entities that share or are motivated by at least one common issue or interest, or work together on a specific project(s) to achieve a common objective). This enterprise may engender the concept of the hive mind (the collective intelligence of many), which will be an intelligent version of real-life super organisms, such as ant or bee nests (Hölldobler and Wilson 2009).

These models will most certainly give rise to issues of property rights and liabilities; access rights for both the provider and the customer can be owned outright, contracted/leased, shared, or have privileged access (Spohrer 2011). For now, we are on the cusp of a management revolution that is likely to be as profound and unsettling as the one that gave birth to the modern industrial age. Driven by the emergence of powerful new collaborative technologies, this transformation will radically reshape the nature of work, the boundaries of the enterprise, and the responsibilities of business leaders (McAfee 2009).

The service-providing industry in the US is divided into thirteen sectors (Chang 2010):

1. professional and business services,
2. healthcare and social assistance,
3. state and local government,
4. leisure and hospitality,
5. other services,
6. educational services,
7. retail trade,
8. financial activities,
9. transportation and warehousing,
10. wholesale trade,
11. information,

12. federal government, and
13. utilities.

Spohrer (2011) goes beyond the service sectors to propose three types of service systems:

1. ***Systems that focus on flow of things***: transportation and supply chains, water and waste recycling, food and products, energy and electric Grid, information/ICT & cloud;
2. ***Systems that focus on Human Activities and Development***: buildings and construction, retail and hospitality / media and entertainment industries, banking and finance / business consulting industries, healthcare and family life systems, education and work life / jobs and entrepreneurship; and
3. ***Systems that focus on Governing***: cities, states, and nations.

Categorizing types and sectors of services is an important beginning because it can lead to a better understanding of the emerging rules and relationships in service value chains. This approach can further enhance the value co-creation capabilities of innovative service concepts that contribute to our quality of life. The classification also helps in identifying different objectives and constraints for the design and operations of the service system. Some examples include strategic policies under limited budget: education, strategic with readiness for quick response; national defense; business enterprise, maximizing profit while minimizing cost; etc.

In addition, this classification is being used to determine the overlap and synergies required among different science disciplines to enable trans-disciplinary collaboration and educational programs.

References

Works Cited

Badr, Y., A. Abraham, F. Biennier, and C. Grosan. 2008. "Enhancing Web Service Selection by User Preferences of Non-Functional Features." Presented at 4th International Conference on Next Generation Web Services Practices, October 20-22, 2008, Seoul, South Korea.

Chang, C.M. 2010. *Service Systems Management and Engineering: Creating Strategic Differentiation and Operational Excellence*. New York, NY, USA: John Wiley & Sons, Inc.

Daskin, M.S. 2010. *Service Science*. New York, NY, USA: John Wiley & Sons.

Freeman, R.L. 2004. *Telecommunication Systems Engineering*, 4th ed. New York, NY, USA: John Wiley & Sons.

Hölldobler, B., and E.O. Wilson. 2009. *The Super-organism: The Beauty, Elegance, and Strangeness of Insect Societies*. New York, NY, USA: W.W. Norton & Company.

ISO. 2008, ISO 9001:2008, *Quality management systems -- Requirements*. Geneva, Switzerland: International Organisation for Standardisation.

McAfee, A. 2009. *Enterprise 2.0: New Collaborative Tools for Your Organization's Toughest Challenges*. Boston, MA, USA: Harvard Business School Press.

OGC (Office of Government Commerce). 2009. *ITIL Lifecycle Publication Suite Books*. London, UK: The Stationery Office.

QuEST Forum. 2012. *Quality Management System (QMS) Measurements Handbook*, Release 5.0. Plano, TX, USA: Quest Forum.

Ray, R.F. (ed). 1984. *Engineering and Operations in Bell System*, 2nd ed. Florham Park, NJ, USA: AT&T Bell Labs.

SEI. 2007. *Capability Maturity Model Integrated (CMMI) for Development*, version 1.2. Pittsburgh, PA, USA: Software Engineering Institute (SEI)/Carnegie Mellon University (CMU).

Spohrer, J.C. 2011. "Service Science: Progress & Directions." Presented at the International Joint Conference on Service Science, 25-27 May 2011, Taipei, Taiwan.

Theilmann, W., and L. Baresi. 2009. "Multi-level SLAs for Harmonized Management in the Future Internet," in *Towards the Future Internet - A European Research Perspective*, edited by G. Tselentis, J. Domingue, A. Galis, A. Gavras, D. Hausheer, S. Krco, V. Lotz, and T. Zehariadis. Amsterdam, The Netherlands: IOS Press.

Tien, J.M., and D. Berg. 2003. "A Case for Service Systems Engineering." *Journal of Systems Science and Systems Engineering.* 12 (1): 13-38.

Primary References

Chang, C.M. 2010. *Service Systems Management and Engineering: Creating Strategic Differentiation and Operational Excellence.* New York, NY, USA: John Wiley & Sons, Inc.

Theilmann, W., and L. Baresi. 2009. "Multi-level SLAs for Harmonized Management in the Future Internet," in *Towards the Future Internet - A European Research Perspective*, edited by G. Tselentis, J. Domingue, A. Galis, A. Gavras, D. Hausheer, S. Krco, V. Lotz, and T. Zehariadis. Amsterdam, The Netherlands: IOS Press.

Additional References

None.

< Previous Article | Parent Article | Next Article >

SEBoK v. 2.1, released 31 October 2019

Scope of Service Systems Engineering

Lead Authors: *Ricardo Pineda, Bud Lawson*

Service systems engineering (SSE) involves all aspects of the enterprise. This topic discusses different aspects of the scope of SSE, from organizational strategy, to interoperability, to the life cycle of services, and then to their design.

SSE and the Enterprise

Enterprises plan, develop, and manage the enhancements of their infrastructure, products, and services, including marketing strategies for product and service offerings. These plans propose new products or service offerings based on new, unexplored, or unforeseen customer needs with clearly differentiated value propositions. Service strategies are the internal business processes required to design, operate, and deliver services. The mission of service strategies is to develop the capacity to achieve and maintain a strategic advantage (OGC 2009).

Taking the systems engineering (SE) approach to service systems, or (SSE), is imperative for the service-oriented, customer-centric holistic view to select and combine service system entities. The SSE approach can then define and discover relationships among service system entities to plan, design, adapt, or self-adapt to co-create value. The SSE approach should identify linkages, relationships, constraints, challenges/problems, new technologies, interoperability standards, interface agreements, or process development requirements among service entities required for the planned service or for potential future services (Lefever 2005).

SSE mandates participation not only from engineering, business operations, and customers, but also from various different domains, such as management science, behavioral science, social science, systems science, network science, computer science, decision informatics, etc.

Hipel et al. (2007) have presented a table for service science in terms of the domains and methods, including not only service systems, but also infrastructure and transportation systems, environmental and energy systems, and defense and space systems. The collaboration domains in Figure 1 below are a first approximation to the collaboration required from different disciplines for the SSE paradigm.

Table 1. Service Systems Engineering Domain Collaboration. (Hipel et al. 2007) Reprinted with permission of © Copyright IEEE - All rights reserved.

SEE	Collaboration Domains
SSE Management	• Management Science • Business Process Management • Cognitive Science • Decision Science
Service Realization Process (SRP)	• All engineering fields • Business Operations • Infrastructure Operations • Social Science • Computer Science • Management Science • Behavioral Science • Network Science • Computational Science • Systems Science • Decision Science
Methodologies, Processes, and Tools (MPT)	• Natural Science • Business Science (BPMN) • Mathematical • All engineering fields

Major challenges faced by SSE include the dynamic nature of service systems evolving and adapting to constantly changing operations and/or business environments, and the need to overcome silos of knowledge. Interoperability of service system entities through interface agreements must be at the forefront of the SSE design process for the harmonization of operations, administration, maintenance, and provisioning procedures of the individual service system entities (Pineda 2010).

In addition, service systems require open collaboration among all stakeholders, but recent research on mental models of multidisciplinary teams shows integration and collaboration into cohesive teams has proven to be a major challenge (Carpenter et al. 2010) (See also Team Dynamics). Thus, the emphasis on multidisciplinary (e.g., scientific, engineering, management, and social) education and training programs required to foster systems thinking helps bridge the gaps created by these silos of knowledge.

In the SSE approach, the social, governance, business, service, operations, and management activities are linked together through the service life cycle; service systems are by themselves a type of system of systems (SoS) where traditional systems engineering (TSE) practices need to be extended to include service systems entities' relationships (e.g., interface agreements among people, organizations, processes, and technologies) through information flows, technical interoperability, governance, and access rights within a system of systems.

Interoperability of Services

Interoperability among the different service system entities becomes highly relevant in SSE since the constituent entities are designed according to stakeholder needs; the entity is usually managed and operated to satisfy its own objectives independently of other system entities. The objectives of individual service system entities may not necessarily converge with the overall objectives of the service system. Thus, the need to include the following in the definition of a service system: analysis and design of the service system, governance frameworks to align political objectives, service strategies, business objectives, information and communications technologies (ICT) objectives, technology objectives and end-to-end operations, administration and maintenance procedures, and allocation of these procedures to individual entities (Luzeaux and Ruault 2010).

The previous discussion relates to a new service system development. There may be instances where a service is planned for delivery in phases of deployment (transition/deployment phase), or as presented earlier, if there is already a service system defined and deployed, then it's possible that the new request is for a service based application (SBA), in which case, the process is more focused on the adaptations needed to deploy the new application. For SBA, instances of advances in computer engineering, computer science, and software development already permit the adaptation and creation of SBA in a run-time environment for the discovery, development, and publishing of applications (Maglio et al. 2010).

The service design process (SDP) for new services is triggered by the market concept of the intended service and considers the stakeholder(s), service value chain(s), target market(s), target customer(s), proposed SLA, demand forecast, pricing strategy, and customer access privileges, which together comprise the service strategy. The SDP process then adapts the TSE as a life cycle approach (concept/definition, design/development, deployment/transition, operations, life cycle management/utilization/CSI, and retirement) as discussed in Life Cycle Models. A more detailed list of the SSE process activities is described in Value of Service Systems Engineering and Service Systems Engineering Stages.

Service Lifecycle Stages

The SDP stages and notation are depicted in Table 2 below; due to the complexity of service systems (see also Complexity) the documents generated are becoming more model-based electronic documents than written binders depending on the methodologies and tools used.

Table 2. Service Realization Process: Life Cycle Stages. (SEBoK Original)

<html>

Life Cycle Changes		**Purpose**	**Decision Gates**
Service Strategy/Concept	New Service identification	*Elicit enterprise needs* *Explore service concepts* *Identify service system entities* *Propose viable HL black box solutions* ***Output: Service Description***	Decision Options - Go, No-GO - Continue this stage - Go to preceding stage - Hold project activity - Terminate project - Test - Deploy
	Feasibility Phase		
	HL Analysis		
Service Design/Development	Service Requirement Analysis and Engineering	*Refine service system requirements* ***Output: Service Requirement Document*** *Create solution description* *Identify Interfaces among entities* ***Output: Preliminary Design*** *Develop service system detailed architecture and specs* ***Output: Service Specification Document*** *Verify and Validate system requirements* ***Output: service JV & V Plans***	
	Service Development		
	Service Integration, Verification, and Validation		
Service Transition/Deployment		*Service Insertion Plans* *Deploy service system* *Manage deployment activities* *Inspect and test (verify)* ***Output: Service Operation Plans, Operations Technical Plans, Operational Readiness Plans***	
Service Operations and / Continuous Service Improvement		*Operate a reliable service system to satisfy customer needs* *Monitor, Measure, & Assess* *Provide sustained system capability* *Troubleshoot potential issues* *Store, archive, or dispose of the service system*	

</html>

All the life cycle stages are included for completeness, but very often during the concept analysis phase it may be determined that not all of the stages are needed. In these cases, a recommendation should be made regarding which stages are specifically required for the realization of the service in question.

Service Design Management

Another important role of SSE is the management of the service design process. SSE utilizes TSE practices to manage the resource and asset allocation to perform the activities required to realize the service through the value chain for both the customer and the service provider. The main focus of the service design process management is to provide for the planning, organizational structure, collaboration environment, and program controls to ensure that stakeholder's needs are met from an end-to-end customer perspective.

The service design process management process aligns business objectives and business operational plans with end-to-end service objectives, including customer management plans, service management and operations plans, and operations technical plans. The main SSE management activities are

- planning;
- assessment and control;
- decision management;
- risk management;
- configuration management; and
- information management.

SSE plays a critical role in describing the needs of the intended service in terms of the service's day-to-day operations, including customer care center requirements, interface among service system entities, such as: manufacturing plant, smart grid, hospital, network infrastructure provider(s), content provider(s) and service provider(s), service based application provider(s), applications providers, and the customer management process for the service.

Current research in computer engineering and software systems engineering is looking at the development of run-time platforms to allow real time or near real time customer service discovery and publishing (Spark 2009). The service-centric systems engineering (SeCSE) consortium has a well-defined service design process that is being applied to SBA. In this approach, there are design time and run-time sub-processes for the composition, provisioning, orchestration, and testing for service publishing (Lefever 2005). There is particular interest from the research community to include human-computer interactions (HCI) and behavioral science to address current social networking services (Facebook, Twitter, Linkedin, Google+, etc.) used to share unverified information via audio, messaging, video, chats, etc.

This research is gaining relevance because of the thin line between the customer (consumer, enterprise) and content providers in regards to security, privacy, information authentication, and possible misuse of the user-generated content. Even as the research progresses, these networking services are examples of business models organizing communities of interest for innovation. Hsu says, “If we understand this networking, then we may be able to see through the business strategies and systems design laws that optimize connected value co-creation” (2009).

References

Works Cited

Carpenter, S., H. Delugach, L. Etzkorn, J. Fortune, D. Utley, and S. Virani. 2010. "The Effect of Shared Mental Models on Team Performance." Presented at Industrial Engineering Research Conference, Institute of Industrial Engineers, 2010, Cancun, Mexico.

Hipel, K.W., M.M. Jamshidi, J.M. Tien, and C.C. White. 2007. "The Future of Systems, Man, and Cybernetics: Application Domains and research Methods." *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*. 37 (5): 726-743.

Hsu, C. 2009. "Service Science and Network Science." *Service Science*, 1 (2): i-ii.

Lefever, B. 2005. *SeSCE Methodology*. Rome, Italy: SeCSE Service Centric Systems Engineering. SeCSE511680. Available: http://www.secse-project.eu/wp-content/uploads/2007/08/a5_d4-secse-methodology-v1_3.pdf.

Luzeaux, D., and J.R. Ruault (eds.). 2010. *Systems of Systems*. New York, NY, USA: John Wiley & Sons.

Maglio, P., M. Weske, J. Yang, and M. Fantinato (eds.). 2010. *Proceedings of the 8th International Conference on Service Oriented Computing: ICSOC 2010.* Berlin & Heidelberg, Germany: Springer-Verlag.

OGC (Office of Government Commerce). 2009. *ITIL Lifecycle Publication Suite Books*. London, UK: The Stationery Office.

Pineda, R. 2010. "Understanding Complex Systems of Systems Engineering." Presented at Fourth General Assembly Cartagena Network of Engineering, 2010, Metz, France.

Spark, D. 2009. "Real-Time Search and Discovery of the Social Web." Spark Media Solutions Report. Accessed September 2, 2011. Available: http://www.sparkminute.com/2009/12/07/free-report-real-time-search-and-discovery-of-the-social-web/.

Primary References

Lefever, B. 2005. *SeSCE Methodology*. Rome, Italy: SeCSE Service Centric Systems Engineering. SeCSE511680. Available: http://www.secse-project.eu/wp-content/uploads/2007/08/a5_d4-secse-methodology-v1_3.pdf.

Luzeaux, D., and J.R. Ruault (eds.). 2010. Systems of Systems. New York, NY, USA: John Wiley & Sons.

Additional References

None.

< Previous Article | Parent Article | Next Article >

SEBoK v. 2.1, released 31 October 2019

Value of Service Systems Engineering

Lead Authors: *Ricardo Pineda, Bud Lawson, Richard Turner*

Service systems engineering (SSE) is a multidisciplinary approach to manage and design value co-creation of a service system. It extends the holistic view of a system to a customer-centric, end-to-end view of service system design. Service systems engineers must play the role of an integrator by considering the interface requirements for the interoperability of service system entities, not only for technical integration, but also for the processes and organization required for optimal customer experience during service operations.

Service systems engineering uses disciplined approaches to minimize risk by coordinating/orchestrating social aspects, governance (including security), environmental, human behavior, business, customer care, service management, operations, and technology development processes. Therefore, systems engineers must have a good understanding of cross disciplinary issues to manage, communicate, plan, and organize service systems development and delivery of service. Service systems engineering also brings a customer focus to promote service excellence and to facilitate service innovation through the use of emerging technologies to propose creation of new service systems and value co-creation.

The service design process includes the definition of methods, processes, and procedures necessary to monitor and track service requirements verification and validation, in particular as they relate to the operations, administration, maintenance, and provisioning procedures of the whole service system and its entities. These procedures ensure that failures by any entity are detected and do not propagate and disturb the operations of the service (Luzeaux and Ruault 2010).

Research on service systems needs to fuse business process management, service innovation, and social networks for the modeling of service system value chain (Carroll et al. 2010). The systems engineering approach helps to better understand and manage conflict, thereby helping both private and public organizations optimize their strategic decision making. The use of a systemic approach reduces rework, overall time to market, and total cost of development.

Service SE Knowledge & Skills

The world’s economies continue to move toward the creation and delivery of more innovative services. To best prepare tomorrow’s leaders, new disciplines are needed that include and ingrain different skills and create the knowledge to support such global services. “In this evolving world, a new kind of engineer is needed, one who can think broadly across disciplines and consider the human dimensions that are at the heart of every design challenge” (Grasso and Martinelli 2007).

Service systems engineers fit the T-shaped model of professionals (Maglio and Spohrer 2008) who must have a deeply developed specialty area, as well as a broad set of skills and capabilities (See the Enabling Individuals article). Chang (2010) lists the following twelve service system management and engineering (SSME) skills:

1. Management of Service Systems. These skills include scheduling, budgeting and management of information systems/technologies, and leadership;
2. Operations of Service Systems. Engineers should be proficient in process evaluation and improvement, quality improvement, customer relationships, and uncertainty management;
3. Service Processes. These skills include performance measurements, flow charting, work task breakdown;
4. Business Management. Business skills include project costing, business planning, and change management;
5. Analytical Skills. These skills include problem solving, economic decision analysis, risk analysis, cost estimating, probability and statistics;

6. Interpersonal Skills. Increasingly, service systems engineers are expected to excel in professional responsibility, verbal skills, technical writing, facilitating, and team building;
7. Knowledge Management. Service systems engineers should be familiar with definition, strategies, success factors, hurdles, and best practices in industry;
8. Creativity and Innovation in Services. These skills include creative thinking methods, success factors, value chain, best practices, and future of innovation;
9. Financial and Cost Analysis and Management. Additional business skills include activity-based costing, cost estimation under uncertainty, T-account, financial statements, ratio analysis, balanced scoreboards, and capital formation;
10. Marketing Management. Market forecast, market segmentation, marketing mix- service, price, communications and distribution- are important marketing tools;
11. Ethics and Integrity. Service Systems Engineers must be held to high ethical standards. These include practicing ethics in workplace and clear guidelines for making tough ethical decisions, corporate ethics programs, affirmation action, and workforce diversity, as well as global issues related to ethics. (See Ethical Behavior); and
12. Global Orientation. Increasingly, engineers must be aware of emerging business trends and challenges with regards to globalization drivers, global opportunities, and global leadership qualities.

Service Architecture, Modeling & Views

Successful deployment of service value chains is highly dependent on the alignment of the service with the overall enterprise service strategy, customer expectations, and customer's service experience. The importance of service-oriented customer-centric design has been recognized for several years by traditional service providers (telecommunications, information technology (IT), business reengineering, web services, etc.) through the creation of process-driven architecture frameworks.

Architecture frameworks are important for creating a holistic system view. They promote a common understanding of the major building blocks and their interrelation in systems of systems or complex systems of systems (see also Complexity). An architecture is a model of the the system created to describe the entities, the interactions and interoperability among entities, as well as the expected behavior, utilization, and properties of the end-to-end system. The architectures become the main tool to guide stakeholders, developers, third-party providers, operations managers, service managers, and users in the understanding of the end-to-end service system, as well as to enable governance at the service management and the service development levels.

These architecture frameworks have been defined through standards bodies and/or by private enterprises that recognize their advantage—standard processes that integrate the business-strategic processes and operations with the information technology and technology infrastructure (See Systems Engineering Standards). Most architecture frameworks model different scopes and levels of detail of business strategies, product and service offerings, business operations, and organizational aspects. Unfortunately, there are currently no frameworks that cover all the aspects (views) required to model the service systems. Some frameworks focus on business strategies, others in business process management, others in business operations, still others in aligning IT strategy or technology strategy to business strategy. Thus, a combination of architecture frameworks is required to create the enterprise service system model. For instance, an enterprise may use an enterprise business architecture (EBA) model covering strategic goals and objectives, business organization, and business services and processes where driven by market evolution, technology evolution, and customer demands. However, a reference framework would be needed to model the IT strategy (e.g., Information Technology Infrastructure Library (ITIL) v. 3 (OGC 2009)) and the organizations and processes needed to deliver, maintain, and manage the IT services according to the business strategy.

Service Architecture Frameworks

Prime examples of Service Architecture Frameworks are listed below.

Standards:

- Zachmann Framework (Zachman 2003)
- Business Process Modeling (BPM) (Hantry et al. 2010)
- The Open Group Architecture Framework (TOGAF) (TOGAF 2009)
- Enhanced-Telecomm Operations Map (eTOM) by the TeleManagemnt Forum (eTOM 2009)
- Service Oriented Architecture (SOA) (Erl 2008)
- National Institute of Standards and Technology (NIST) Smart Grid Reference Model (NIST 2010)
- Web services business process execution language (WS-BPEL) (OASIS 2007)
- Department of Defense Architecture Framework (DoDAF) (DoD 2010)
- Others.

Proprietary Enterprise Architecture Frameworks:

- Hewlett - Packard IT Service Management Reference Model (HP ITSMRM 2000)
- International Business Machines Systems Management Solutions Life Cycle, IBM Rational Software.
- Microsoft Operations Framework

This list represents only a sample of the existing service architecture frameworks.

One great example of architecture frameworks applications for service systems, the “High Level Reference Model for the Smart Grid,” developed by NIST in 2010 under the “Energy Independence and Security Act of 2007” (EISA), is presented below:

> *EISA designated the development of a Smart Grid as a national policy goal, specifying that an interoperability framework should be “flexible, uniform and technology neutral. The law also instructed that the framework should accommodate “traditional, centralized generation and distribution resources” while also facilitating incorporation of new, innovative Smart Grid technologies, such as distributed renewable energy resources and energy storage.* (NIST 2010)

The NIST reference model was developed as “a tool for identifying the standards and protocols needed to ensure interoperability and cyber security, and defining and developing architectures for systems and subsystems within the smart grid." Figure 1 illustrates this model and the strategic (organizational), informational (business operations, data structures, and information exchanges required among system entities), and technical needs of the smart grid (data structures, entities specifications, interoperability requirements, etc.).

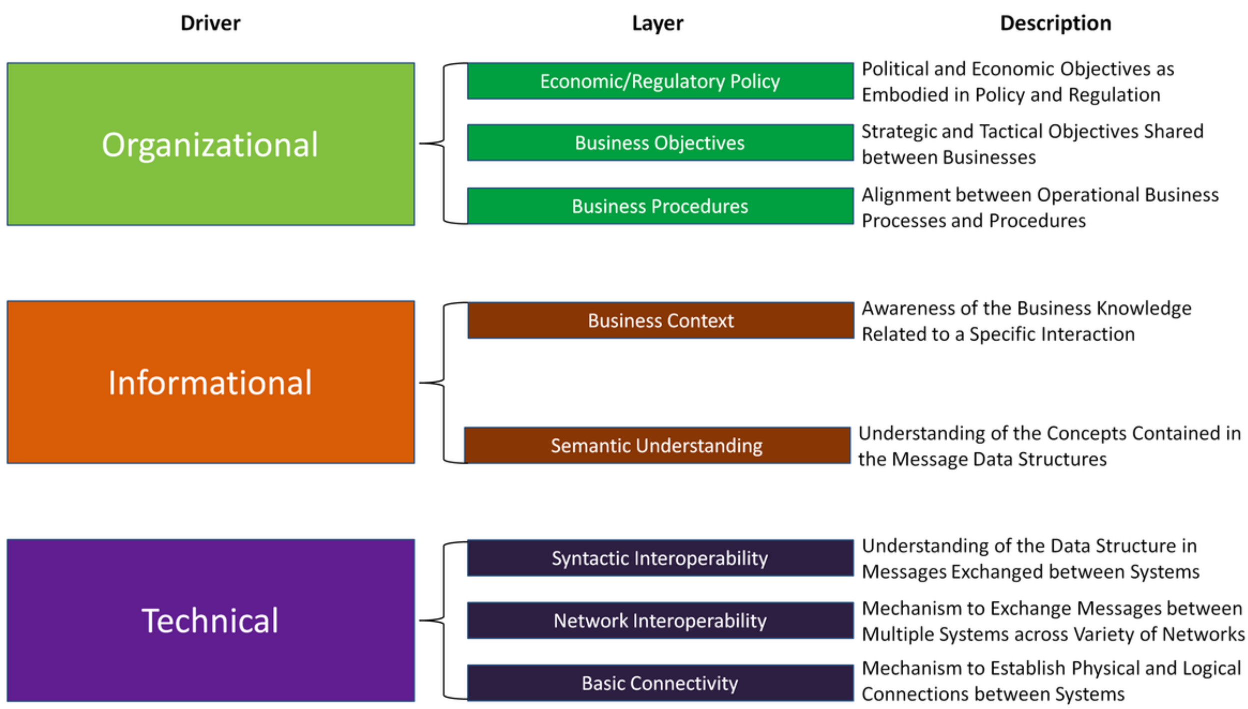


Figure 1. The Grid-Wide Architecture Council's Eight-Layered Stack (NIST and US Dept. of Commerce 2010). Released.

The NIST reference model uses this architecture framework to identify existing standards, identify new standards required for interoperability among interconnected networks, and to enable innovations where smart grid components (energy sources, bulk generation, storage, distribution, transmission, metering, cyber infrastructure, markets, service providers, customers, etc.) are supported by a broad range of interoperable options by well-defined interfaces useful across industries, including security. Emerging/innovative service development with massively scaled, well-managed, and secured networks will enable a dynamic market driven ecosystem representing new economic growth (NIST 2010).

This architecture framework is being used today by different standards organizations, such as the Smart Grid Interoperability Panel (SGIP), and several smart grid working groups. For details on priorities, working programs, and working group charters, see "High Level Reference Model for the Smart Grid" (NIST 2010).

For service systems, the application of any of these frameworks requires modifications/adaptations to create dynamic frameworks aware of environmental changes due to competitor's offerings, market demands, and customer co-creation. Most frameworks are static in nature; this requires business operations to manage changes through pre-defined (pre-programmed) processes for service configuration and change control. Dynamic frameworks would allow real-time, or near real-time, analysis of impacts of newly discovered service on business processes, organizations, and revenue for run-time environment deployment.

Automatic service configuration and change control are being incorporated into the management process via service oriented architecture (SOA) for service automation (Gu et al. 2010) and service oriented computing (Maglio et al. 2010). In particular, progress has been made over the last ten years on the standards for adaptation, orchestration and creation of web services (WS) for service based applications (SBA). A good summary of existing life cycle approaches for adaptable and evolvable SBA is presented in (Papazoglou et al. 2010). Some examples of this are

- web services development life cycle (SDLC);

- rational unified process (RUP) for SOA;
- service oriented modeling and architecture (SOMA); and
- service oriented analysis and design/decision Modeling (SOAD).

Further research is required to understand the architectural implications of dynamic service configuration, including research on human behavior, social aspects, governance processes, business processes, and implications of dynamic service level agreements (SLA) for an enterprise service system. New ways are needed to include adaptation requirements for new technologies that will exchange information with the service system entities and may have their own specifications. These technologies include robots, sensors, renewable energy, nanotechnologies, three dimensional printers, and implantable medical devices.

References

Works Cited

Carroll, N., E. Whelan, and I. Richardson. 2010. "Applying Social Network Analysis to Discover Service Innovation within Agile Service Networks." *Service Science*. 2 (4): 225-244.

Chang, C.M. 2010. *Service Systems Management and Engineering: Creating Strategic Differentiation and Operational Excellence*. New York, NY, USA: John Wiley & Sons, Inc.

DoD. 2010. *DoD Architecture Framework (DoDAF)*, version 2.0. Arlington, VA, USA: US Department of Defense (DoD).

Erl, T. 2008. *SOA Principles of Service Design*. Boston, MA, USA: Prentice Hall, Pearson Education.

eTOM. 2009. "Business Process Framework." Morristown, NJ: TeleManagement Forum. Accessed May 30, 2011 at http://www.tmforum.org/BusinessProcessFramework/1647/home.html.

Grasso, D., and D. Martinelli. 2007. "Section B: Holistic Engineering." *The Chronicle Review, The Chronicles of Higher Education*. Vol. 53, Issue 28. Page 8B. March 2007.

Gu. Q., Cuadrado, F., Lago, P. and Duenãs, J.C. 2010. "Architecture views illustrating the service automation aspect of SOA". *Service research challenges and solutions for the future internet*. 339-372.

Hantry, F., M.P. Papazoglou, W. van den Heuvel, R. Haque, E. Whelan, N. Carroll, D. Karastoyanova, F. Leymann, C. Nikolaou, W. Lamersdorf, and M. Hacid. 2010. "Business Process Management," in *Service Research Challenges and Solutions for the Future Internet S-Cube – Towards Engineering, Managing and Adapting Service-Based Systems*, edited by M. Papazoglou, K. Pohl, M. Parkin, and A. Metzger. Berlin and Heidelberg, Germany: Springer-Verlag. 27-54.

HP ITSMRM. 2000. "HP IT Service Management Reference Model. Technical White Paper." Palo Alto, California, USA: Hewlett – Packard Company. Accessed September 2, 2011. Available: ftp://ftp.hp.com/pub/services/itsm/info/itsm_rmwp.pdf.

Luzeaux, D., and J.R. Ruault (eds.). 2010. *Systems of Systems.* New York, NY, USA: John Wiley & Sons.

Maglio, P., and J. Spohrer. 2008. "Fundamentals of Service Science." *Journal of the Academy of Marketing Science.* 36 (1): 18-20.

National Institute of Standard and Technology (NIST). 2010. *NIST Framework and Roadmap for Smart Grid Interoperability Standards Release 1.0.* Gaithersburg, MD, USA: Office of the National Coordinator for Smart Grid Interoperability, US Department of Commerce. Accessed September 2, 2011. Available: http://www.nist.gov/smartgrid/upload/FinalSGDoc2010019-corr010411-2.pdf.

OASIS. 2007. "Web Services Business Process Execution Language Version 2.0." Organization for Advancement of Structured Information Standards (OASIS) Standard. Accessed September 2, 2011. Available: http://docs.oasis-open.org/webcgm/v2.0/OS/webcgm-v2.0.pdf.

OGC (Office of Government Commerce). 2009. *ITIL Lifecycle Publication Suite Books*. London, UK: The Stationery Office.

Papazoglou, M., K. Pohl, M. Parkin, and A. Metzger. 1998. "Service Research Challenges and Solutions for the Future Internet," in *Service Research Challenges and Solutions for the Future Internet S-Cube – Towards Engineering, Managing and Adapting Service-Based Systems*. Berlin and Heidelberg, Germany: Springer-Verlag.

TOGAF. 2009. "The Open Group Architecture Framework," version 9. The Open Architecture Group. Accessed September 2, 2011. Available: http://www.opengroup.org/togaf.

Zachman, J. 2003. "The Zachman Framework for Enterprise Architecture: Primer for Enterprise Engineering and Manufacturing." Accessed September 2, 2011. Available: http://www.zachmanframeworkassociates.com/index.php/ebook.

Primary References

Chang, C.M. 2010. *Service Systems Management and Engineering: Creating Strategic Differentiation and Operational Excellence.* New York, NY, USA: John Wiley & Sons, Inc.

Erl, T. 2008. *SOA Principles of Service Design*. Boston, MA, USA: Prentice Hall, Pearson Education.

Hantry, F., M.P. Papazoglou, W. van den Heuvel, R. Haque, E. Whelan, N. Carroll, D. Karastoyanova, F. Leymann, C. Nikolaou, W. Lamersdorf, and M. Hacid. 2010. "Business Process Management," in *Service Research Challenges and Solutions for the Future Internet S-Cube – Towards Engineering, Managing and Adapting Service-Based Systems*, edited by M. Papazoglou, K. Pohl, M. Parkin, and A. Metzger. Berlin and Heidelberg, Germany: Springer-Verlag. p. 27-54.

National Institute of Standard and Technology (NIST). 2010. *NIST Framework and Roadmap for Smart Grid Interoperability Standards Release 1.0.* Gaithersburg, MD, USA: Office of the National Coordinator for Smart Grid Interoperability, US Department of Commerce. Accessed September 2, 2011. Available: http://www.nist.gov/smartgrid/upload/FinalSGDoc2010019-corr010411-2.pdf.

Additional References

None.

< Previous Article | Parent Article | Next Article >

SEBoK v. 2.1, released 31 October 2019

Service Systems Engineering Stages

Lead Authors: *Ricardo Pineda, Bud Lawson, Richard Turner*

This article describes the stages of the service systems development process (SSDP) and expected outputs for each stage; for a closer alignment with the traditional systems engineering (TSE) process, the concept and feasibility phases have been combined into a single service strategy/concept as discussed in the SEBoK Systems Engineering and Management article. All of the stages of the SSDP take a similar iterative approach to fully understand the enterprise capabilities, enterprise process impact, information technology (IT), and technology impacts and customer expectations. Lin and Hsieh (2011) provide a good summary on New service Development processes. The Information Technology Infrastructure Library (ITIL) stage names have been purposely added to the SSDP to show the needed alignment between IT and technology. The reader should keep in mind that even though IT is crucial to the overall end-to-end system, service technology development needs must be taken into consideration in all the stages of SSDP.

Service Strategy/Concept

A service strategy/concept is the entry into the SSDP. The concept may be generated by an end-user (enterprise customer or consumer), a business manager, an engineering organization, new web service designers, new technology developments, and/or information technology trends. The service concept is the highest level of the service idea and it usually addresses what service is being proposed to what markets and to whom within these markets.

A high-level feasibility assessment of the concept is then carried out by the integrated service development team (ISDT) to assess the needs/impacts on enterprise process capabilities, operational capabilities, and/or new technology developments (access, infrastructure, operations support systems (OSS), service support systems (SSS), and business support systems (BSS). It should also consider any impacts on service governance, social, cultural, and human behaviors. The feasibility assessment also gives a plus or minus 30% estimate on the time to develop and the cost of development, which are entry points into the business case to evaluate whether the service is viable to develop and to market given the constraints and estimates. At this time, a decision (decision gate) determines if the service is to be developed.

If the business case is viable, then a detailed business description of the service is developed. This includes functions and features to be included, phases of development, markets to be addressed, customers within the markets to be targeted, and customer experiences expected from the service (i.e., defining the non-functional requirements of the service, such as the quality of service (QoS), availability, reliability, and security considerations and offerings within the service). This description allows detailed studies of expected human-computer interactions, social networking, technology requirements, and operations requirements. Governance and organizational process requirements should also be included to generate the “service description” as the main output from this stage.

Service systems engineering (SSE) takes an important role in understanding and eliciting the enterprise service concepts. Clearly, understood end-to-end business processes required for the intended service are fundamental to its successful development, deployment, and customer satisfaction. SSE works with business process management (BPM), social science, and cognitive science to elicit intended service operations, including target audiences, pre-sale, sale, and post-sale customer care processes.

Requirements Analysis and Engineering

A service requirements document is created that describes the service functions, the service entities, the intended interaction among entities, and the customer-facing and internal-facing functions/processes that are required to support the service. This description should conceptually include intended service level agreements (SLAs) and the obligations of the service provider process should there be any degree of non-compliance during service operation.

In addition to the TSE activities described earlier, the SSE requirements analysis and engineering process must develop a customer-centric view of the service to analyze SLA, QoS, value co-creation, monitoring, and assessment requirements to comply with the expected/planned SLA. This analysis will determine whether dynamic changes of the service are required during service operation to correct faults, reconfigure, administer, or to adapt/self-adapt for possible performance degradations.

Beyond the traditional service life cycle management (LCM) processes, the requirements must also be developed for service level management (SLM) processes and systems. These are needed to monitor, measure, and assess key performance indicators (KPIs), technical performance measures (TPMs), and service performance measures (SPMs) according to the SLA.

The SSE requirements analysis addresses the support systems for the governance, business, service, operations, and support processes to derive requirements for technologies, information systems, processes, and enterprise organizations. Interface requirements, information flows, and data requirements are also within the scope of requirements analysis. The main output is the service requirements document (SRD).

SSE plays a critical role in describing the services needs for day-to-day operations. These include customer care centers requirements and interfaces between network infrastructure provider(s), content provider(s), service provider(s), service based application provider(s), and the customer management process for the service. All of these are described in detail in the service operations plans (SOPs) and the operations technical plans (OTPs).

Systems Design/Development

The SRD, SOP, and OTP have enough detail regarding the service functions, operations, interfaces, and information flows required among the different service system entities to analyze, identify, and recommend end-to-end applicable architecture frameworks; to carry out trade-off analyses for the alternatives among service system entities; and to describe and allocate relationships (interactions) among entities at all levels of the service architecture. Detailed requirements are worked at lower levels to generate specifications for entity developers including data structures, data flow diagrams, and allocated performance requirements.

ITIL v. 3 (OGC 2007) recommends inclusion of the following service design processes:

- service catalog management,
- service level management,
- capacity management,
- availability management,
- service continuity management,
- security management, and
- supplier/provider management.

Service Integration, Verification & Validation

SSE defines integration and interface requirements for the seamless operation of the service. In this regard, the system engineer takes an integrator role to ensure proper data generation and flow through all the different systems composing the service offered. The goal is to ensure customers (consumer or internal) are getting the information required to carry out the tasks required in the business, operations, service, and customer processes. The service integration, verification, and validation plans need to include end-to-end verification and validation procedures for any new development or adaptations required for planned dynamic configuration/re-configuration of previously tested service systems. (See also System Verification and System Validation.)

The systems engineer creates these plans using a number of different perspectives. These include:

- end-to-end service (service validation test plans),
- customer care (operational readiness test plans),
- service provider (network validation test plans),
- service system entities interoperability/interface test plans,
- content provider (content validation test plans), and
- application (user acceptance test plans).

Service Transition/Deployment

Service systems may change very rapidly and new enhancements, new features, or new applications can be added as incremental developments, new developments, or adaptation to service offerings. Service systems engineers review new requirements to assess the feasibility of the changes to the service system entities, technologies, processes, and organizations, as well as their impacts on the service offerings. The service transition/deployment stage takes input from service development to plan for service insertion, technology insertion, processes adaptations, and implementation with minimal impact to existing services. During this stage, special care is taken with integration, verification, and validation test plans and regression testing to ensure new developments work flawlessly with existing services.

ITIL v. 3 (OGC 2007) recommends the following processes in the transition/deployment stage:

- transition planning and support,
- change management,
- service asset and configuration management,
- release and deployment management,
- service validation and testing,
- evaluation, and
- knowledge management.

Service Operations/Continuous Service Improvement (CSI)

Service operation manages the day-to-day activities of all aspects of the end-to-end service delivery to the customer. It manages the operations, administration, maintenance, and provisioning of the service, technology, and infrastructure required to deliver the contracted service to the customer within the specified service levels. The main service operations processes in ITIL v. 3 are

- event management,
- incident management,
- problem management,
- request fulfillment, and
- access management.

A continuous service improvement (CSI) plan for the implementation of technologies and tools for the continuous improvement of the service, monitoring, measuring, and analyzing process and service metrics is essential.

Service Systems Engineering Tools & Technologies

Tools and technologies from a broad spectrum of fields are extensively used during the different stages of SSE. Not only are they used for the development of the hardware, software, information systems and technology components, but also for the modelling, definition, and design of the organization, processes, and data structures of the service system (See also Representing Systems with Models). These tools and technologies include modelling, simulation, development, test bed, and social environmental aspects of the intended or to be designed service. The tools fall into three main domains:

1. business process management (BPM),
2. service design process, and
3. service design management.

Business process management (BPM) generally deals with process management scenarios to coordinate people and systems, including sequential workflow, straight through processing, case management, content life cycle management, collaborative process work, and value chain participation. Systems engineers work with service managers to align the business architectures with the technology and IT architecture. The business process modeling notation (BPMN) is a graphic notation standard that is implemented to describe a process's realization within any given workflow. This notation is linked with web services business process execution language (WS-BPEL), a format used to perform an automated business process by implementing web services technology. For an extensive review of existing BPM tools and BPM suites, please see Hantry et al. (2010), Carroll et al. (2010), Andrikoupolous et al. (2010), Lin and Hsieh (2011), and Ward-Dutton (2010).

Service design process: Architecture frameworks (AF) and enterprise architectures (EAs) are standards that help split complex systems (see also Complexity) into an interrelated, structured form. They describe the different characteristics of the products and services. Systems engineering modeling tools, such as the unified modeling language (UML) (OMG 2010a) and system modeling language (SysML) (OMG 2010b), help develop the AF and EA and greatly impact the continued evolution and successful implementation of complex projects. Service oriented architecture (SOA) and systems and software engineering architecture (ISO/IEC/IEEE 2011) are standards that apply architecture principles for specialized applications. Successful implementation of the architecture tools helps identify critical interfaces and improves understanding of the allocations between components and functions.

Mode-based systems engineering (MBSE), model driven architectures (MDA), and model oriented systems engineering (MOSES) are examples of commonly used tools for logical (functional), behavioral (operational), and physical design of the IT. UML, UML 2.0, and SysML are extensively used to describe operational scenarios, modes of operations, use cases, and entity relationships. For an extensive review of MBSE, MDA, and MOSES, please see Friedenthal (1998), Estefan (2008), Pezuela (2005), Andrikopoulos et al. (2010), and Hybertson (2010).

In addition, trade-off and engineering analyses use different optimization methodologies. Since services exhibit a significant level of randomness, statistical analysis, demand forecasting, multi-objective optimization, queuing theory, and stochastic optimization methodologies are tools used to model and simulate the service system behavior. These methodologies support decision making in areas as diverse as resource allocation, number of facilities, facilities' geographical locations, fleet routing and optimization, service systems reliability and prognosis, and network optimization. A good overview of these methodologies can be found in Daskin (2010).

During the service design process (SDP), planning for the implementation of technologies and tools for the continuous improvement of the service is performed. These tools support monitoring, measuring, and analyzing process and service performance metrics. The Deming cycle (plan, do, check, and act (PDCA) is widely used as the foundation for quality improvements across the service. Lean manufacturing, six sigma, swim lanes, balanced scoreboard, benchmarking, and gap analysis methodologies are commonly used for service evaluation and continuous improvement.

Service design management: There are standards for implementing and managing systems engineering processes (IEEE 1220 (1998)) that help coordinate and synchronize all the service systems engineering processes leading to improved organizational collaboration and improved service delivery (see also Systems Engineering Standards). Standards have been developed in software engineering for product evaluation (ISO/IEC 14598 (1998)) and product quality (ISO/IEC 9126 series (2003a, 2003b, & 2004)), as well as information security management (ISO 27001 (2005)) and evaluation series (ISO 15408 (2008a, 2008b, & 2009)). The ITIL v. 3 describes best practices for IT service management, which can be extended to include service systems.

References

Works Cited

Adams, S., A. Cartlidge, A. Hanna, S. Rance, J. Sowerby, and J. Windebank. 2009. *ITIL V3 Foundation Handbook*. London, England, UK: The Stationary Office.

Andrikopoulos, V., A. Bucchiarone, E. Di Nitto, R. Kazhamiakin, S. Lane, V. Mazza, and I. Richardson. 2010. "Chapter 8: Service Engineering," in *Service Research Challenges and Solutions for the Future Internet S-Cube – Towards Engineering, Managing and Adapting Service-Based Systems*, edited by M. Papazoglou, K. Pohl, M. Parkin, and A. Metzger. Berlin and Heidelberg, Germany: Springer-Verlag. p. 271-337.

Carroll, N., E. Whelan, and I. Richardson. 2010. "Applying Social Network Analysis to Discover Service Innovation within Agile Service Networks." *Service Science.* 2 (4): 225-244.

Daskin, M.S. 2010. *Service Science*. New York, NY, USA: John Wiley & Sons.

Estefan, J. 2008. *A Survey of Model-Based Systems Engineering (MBSE) Methodologies*, rev B. Seattle, WA, USA: International Council on Systems Engineering. INCOSE-TD-2007-003-02.

Friedenthal, S. 1998. "Object Oriented System Engineering: Process Integration for 2000 and beyond." Presented at System Engineering & Software Symposium, Lockheed Martin Corporation, 1998, New Orleans, LA.

Hantry, F., M.P. Papazoglou, W. van den Heuvel, R. Haque, E. Whelan, N. Carroll, D. Karastoyanova, F. Leymann, C. Nikolaou, W. Lamersdorf, and M. Hacid. 2010. "Business Process Management," in *Service Research Challenges and Solutions for the Future Internet S-Cube – Towards Engineering, Managing and Adapting Service-Based Systems*, edited by M. Papazoglou, K. Pohl, M. Parkin, and A. Metzger. Berlin and Heidelberg, Germany: Springer-Verlag. p. 27-54.

Hybertson, D.W. 2009. *Model-Oriented Systems Engineering Science: A Unifying Framework for Traditional and Complex Systems*. Boston, MA, USA: Auerbach Publications.

IEEE. 1998. IEEE 1220-1998, *IEEE Standard for Application and Management of the Systems Engineering Process*. Washington, DC, USA: Institute of Electrical and Electronics Engineers.

ISO/IEC. 1998. ISO/IEC 14598-5:1998, *Information technology — Part 5: Process for evaluators*. Geneva, Switzerland: International Organization for Standardization / International Electrotechnical Commission.

ISO/IEC. 2003a. ISO/IEC TR 9126-2:2003, *Software engineering — Product quality — Part 2: External metrics*. Geneva, Switzerland: International Organization for Standardization / International Electrotechnical Commission.

ISO/IEC. 2003b. ISO/IEC TR 9126-3:2003, *Software engineering — Product quality — Part 3: Internal metrics*. Geneva, Switzerland: International Organization for Standardization / International Electrotechnical Commission.

ISO/IEC. 2004. ISO/IEC TR 9126-4:2004, *Software engineering — Product quality — Part 4: Quality in use metrics*. Geneva, Switzerland: International Organization for Standardization / International Electrotechnical Commission.

ISO/IEC. 2005. ISO/IEC 27001:2005, *Information technology — Security techniques — Information security management systems — Requirements*. Geneva, Switzerland: International Organization for Standardization / International Electrotechnical Commission.

ISO/IEC. 2008a. ISO/IEC 15408-2:2008, *Information technology — Security techniques — Evaluation criteria for IT security — Part 2: Security functional components*. Geneva, Switzerland: International Organization for Standardization / International Electrotechnical Commission.

ISO/IEC. 2008b. ISO/IEC 15408-3:2008, *Information technology — Security techniques — Evaluation criteria for IT security — Part 3: Security assurance components*. Geneva, Switzerland: International Organization for Standardization / International Electrotechnical Commission.

ISO/IEC. 2009. ISO/IEC 15408-1:2009, *Information technology — Security techniques — Evaluation criteria for IT security* — Part 1: Introduction and general model. *Geneva, Switzerland: International Organization for Standardization / International Electrotechnical Commission.*

ISO/IEC/IEEE. 2011. ISO/IEC/IEEE 42010:2011, *Systems and software engineering — Architecture description*. Geneva, Switzerland: International Organization for Standardization / International Electrotechnical Commission.

Lefever, B. 2005. "SeSCE Methodology." Rome, Italy: SeCSE Service Centric Systems Engineering. SeCSE511680. Available: http://www.secse-project.eu/wp-content/uploads/2007/08/a5_d4-secse-methodology-v1_3.pdf.

Lin, F., and P. Hsieh. 2011. "A SAT View on New Service Development." *Service Science.* 3 (2): 141-157.

OGC (Office of Government Commerce). 2007. *ITIL Lifecycle Publication Suite Books*. London, England, UK: The Stationery Office.

OMG. 2010a. *Unified Modeling Language™ (UML)*, version 2. Needham, MA, USA: Object Management Group. Available: http://www.omg.org/spec/UML/.

OMG. 2010b. *OMG Systems Modeling Language (SysML)*, version 1.2. Needham, MA, USA: Object Management Group. Available: http://www.sysml.org/docs/specs/OMGSysML-v1.2-10-06-02.pdf.

Pezuela, C. 2005. "Collection of Existing Service Centric Prototypes." Report A5.D1. Brussels, Belgium: European Union, Information Society Technology. Accessed September 5, 2011. Available: http://www.secse-project.eu/wp-content/uploads/2007/08/a5d1-collection-of-existing-service-centric-prototypes.pdf.

Ward-Dutton, N. 2010. "BPM Technology: Vendor Capability Comparison." MWD Premium Advisory Report. Horsham, West Sussex, UK: Macehiter Ward-Dutton (MWD) Limited. MWD Advisors. Accessed September 5, 2011. Available: http://www.mwdadvisors.com/library/detail.php?id=380.

Primary References

Estefan, J. 2008. *A Survey of Model-Based Systems Engineering (MBSE) Methodologies*, rev B. Seattle, WA, USA: International Council on Systems Engineering. INCOSE-TD-2007-003-02.

Hybertson, D.W. 2009. *Model-Oriented Systems Engineering Science: A Unifying Framework for Traditional and Complex Systems*. Boston, MA, USA: Auerbach Publications.

Lefever, B. 2005. "SeSCE Methodology." Rome, Italy: SeCSE Service Centric Systems Engineering. SeCSE511680. Available: http://www.secse-project.eu/wp-content/uploads/2007/08/a5_d4-secse-methodology-v1_3.pdf.

OGC (Office of Government Commerce). 2007. *ITIL Lifecycle Publication Suite Books*. London, England, UK: The Stationery Office.

Additional References

None.

< Previous Article | Parent Article | Next Article >

SEBoK v. 2.1, released 31 October 2019

Knowledge Area: Enterprise Systems Engineering

Enterprise Systems Engineering

Lead Authors: *James Martin, Dick Fairley, Bud Lawson*

Enterprise systems engineering (ESE) is the application of systems engineering principles, concepts, and methods to the planning, design, improvement, and operation of an enterprise.

Topics

Each part of the SEBoK is divided into knowledge areas (KAs), which are groupings of information with a related theme. The KAs in turn are divided into topics. This KA contains the following topics:

- Enterprise Systems Engineering Background
- The Enterprise as a System
- Related Business Activities
- Enterprise Systems Engineering Key Concepts
- Enterprise Systems Engineering Process Activities
- Enterprise Capability Management

Introduction

This knowledge area provides an introduction to systems engineering (SE) at the enterprise level in contrast to "traditional" SE (TSE) (sometimes called "conventional" or "classical" SE) performed in a development project or to "product" engineering (often called product development in the SE literature).

The concept of enterprise was instrumental in the great expansion of world trade in the 17th century (see note 1) and again during the Industrial Revolution of the 18th and 19th centuries. The world may be at the cusp of another global revolution enabled by the information age and the technologies and cultures of the Internet (see note 2). The discipline of SE now has the unique opportunity of providing the tools and methods for the next round of enterprise transformations.

> *Note 1. "The Dutch East India Company… was a chartered company established in 1602, when the States-General of the Netherlands granted it a 21-year monopoly to carry out colonial activities in Asia. It was the first multinational corporation in the world and the first company to issue stock. It was also arguably the* ***world's first mega-corporation****, possessing quasi-governmental powers, including the ability to wage war, negotiate treaties, coin money, and establish colonies." (emphasis added, National Library of the Netherlands 2010)*
>
> *Note 2. This new revolution is being enabled by cheap and easily usable technology, global availability of information and knowledge, and increased mobility and adaptability of human capital. The enterprise level of analysis is only feasible now because organizations can work together to form enterprises in a much more fluid manner.*

ESE is an emerging discipline that focuses on frameworks, tools, and problem-solving approaches for dealing with the inherent complexities of the enterprise. Furthermore, ESE addresses more than just solving problems; it also deals with the exploitation of opportunities for better ways to achieve the enterprise goals. A good overall description of ESE is provided by in the book by Rebovich and White (2011).

Key Terms

Enterprise

An enterprise consists of a purposeful combination (e.g., a network) of interdependent resources (e.g., people, processes, organizations, supporting technologies, and funding) that interact with

- each other to coordinate functions, share information, allocate funding, create workflows, and make decisions, etc.; and
- their environment(s) to achieve business and operational goals through a complex web of interactions distributed across geography and time (Rebovich and White 2011, 4-35).

The term enterprise has been defined as follows:

> (1) *One or more organizations sharing a definite mission, goals, and objectives to offer an output such as a product or service.* (ISO 2000);
>
> (2) *An organization (or cross organizational entity) supporting a defined business scope and mission that includes interdependent resources (people, organizations and technologies) that must coordinate their functions and share information in support of a common mission (or set of related missions).* (CIO Council 1999);
>
> (3) *The term enterprise can be defined in one of two ways. The first is when the entity being considered is tightly bounded and directed by a single executive function. The second is when organizational boundaries are less well defined and where there may be multiple owners in terms of direction of the resources being employed. The common factor is that both entities exist to achieve specified outcomes.* (MOD 2004); and
>
> (4) *A complex, (adaptive) socio-technical system that comprises interdependent resources of people, processes, information, and technology that must interact with each other and their environment in support of a common mission.* (Giachetti 2010)

An enterprise must do two things: (1) develop things within the enterprise to serve as either external offerings or as internal mechanisms to enable achievement of enterprise operations, and (2) transform the enterprise itself so that it can most effectively and efficiently perform its operations and survive in its competitive and constrained environment.

Enterprise vs Organization

It is worth noting that an enterprise is not equivalent to an "organization” according to the definition above. This is a frequent misuse of the term enterprise. The figure below shows that an enterprise includes not only the organizations that participate in it, but also people, knowledge, and other assets such as processes, principles, policies, practices, doctrine, theories, beliefs, facilities, land, intellectual property, and so on.

Some enterprises are organizations, but not all enterprises are organizations. Likewise, not all organizations are enterprises. Some enterprises have no readily identifiable "organizations" in them. Some enterprises are self-organizing (i.e., not organized by mandate) in that the sentient beings in the enterprise will find for themselves some way in which they can interact to produce greater results than can be done by the individuals alone. Self-organizing enterprises are often more flexible and agile than if they were organized from above (Dyer and Ericksen 2009; Stacey 2006).

> *One type of enterprise architecture that supports agility is a non-hierarchical organization without a single point of control. Individuals function autonomously, constantly interacting with each other to*

define the vision and aims, maintain a common understanding of requirements and monitor the work that needs to be done. Roles and responsibilities are not predetermined but rather emerge from individuals' self-organizing activities and are constantly in flux. Similarly, projects are generated everywhere in the enterprise, sometimes even from outside affiliates. Key decisions are made collaboratively, on the spot, and on the fly. Because of this, knowledge, power, and intelligence are spread through the enterprise, making it uniquely capable of quickly recovering and adapting to the loss of any key enterprise component. (http://en.wikipedia.org/wiki/Business_agility)

In spite of this lack of "organization" in some enterprises, SE can still contribute much in the engineering of the enterprise, as described in the articles below. However, SE must be prepared to apply some non-traditional approaches in doing so. Hence the need for embracing the new discipline called enterprise systems engineering (ESE).

Giachetti (2010) distinguishes between enterprise and organization by saying that an organization is a view of the enterprise. The organization view defines the structure and relationships of the organizational units, people, and other actors in an enterprise. Using this definition, we would say that all enterprises have some type of organization, whether formal, informal, hierarchical or self-organizing network.

Extended Enterprise

Sometimes it is prudent to consider a broader scope than merely the "boundaries" of the organizations involved in an enterprise. In some cases, it is necessary (and wise) to consider the "extended enterprise" in modeling, assessment, and decision making. This could include upstream suppliers, downstream consumers, and end user organizations, and perhaps even "sidestream" partners and key stakeholders. The extended enterprise can be defined as:

Wider organization representing all associated entities - customers, employees, suppliers, distributors, etc. - who directly or indirectly, formally or informally, collaborate in the design, development, production, and delivery of a product (or service) to the end user. (http://www.businessdictionary.com)

Enterprise Systems Engineering

Enterprise systems engineering (ESE), for the purpose of this article, is defined as the application of SE principles, concepts, and methods to the planning, design, improvement, and operation of an enterprise (see note 3). To enable more efficient and effective enterprise transformation, the enterprise needs to be looked at "as a system," rather than merely as a collection of functions connected solely by information systems and shared facilities (Rouse 2009). While a systems perspective is required for dealing with the enterprise, this is rarely the task or responsibility of people who call themselves systems engineers.

Note 3. This form of systems engineering (i.e., ESE) includes (1) those traditional principles, concepts, and methods that work well in an enterprise environment, plus (2) an evolving set of newer ideas, precepts, and initiatives derived from complexity theory and the behavior of complex systems (such as those observed in nature and human languages).

Creating Value

The primary purpose of an enterprise is to create value for society, other stakeholders, and for the organizations that participate in that enterprise. This is illustrated in Figure 1 that shows all the key elements that contribute to this value creation process.

There are three types of organizations of interest: businesses, projects, and teams (see note 4). A typical business participates in multiple enterprises through its portfolio of projects. Large SE projects can be enterprises in their own right, with participation by many different businesses, and may be organized as a number of sub-projects.

Note 4. The use of the word "business" is not intended to mean only for-profit commercial ventures. As used here, it also includes government agencies and not-for-profit organizations, as well as commercial ventures. Business is the activity of providing goods and services involving financial, commercial, and industrial aspects.

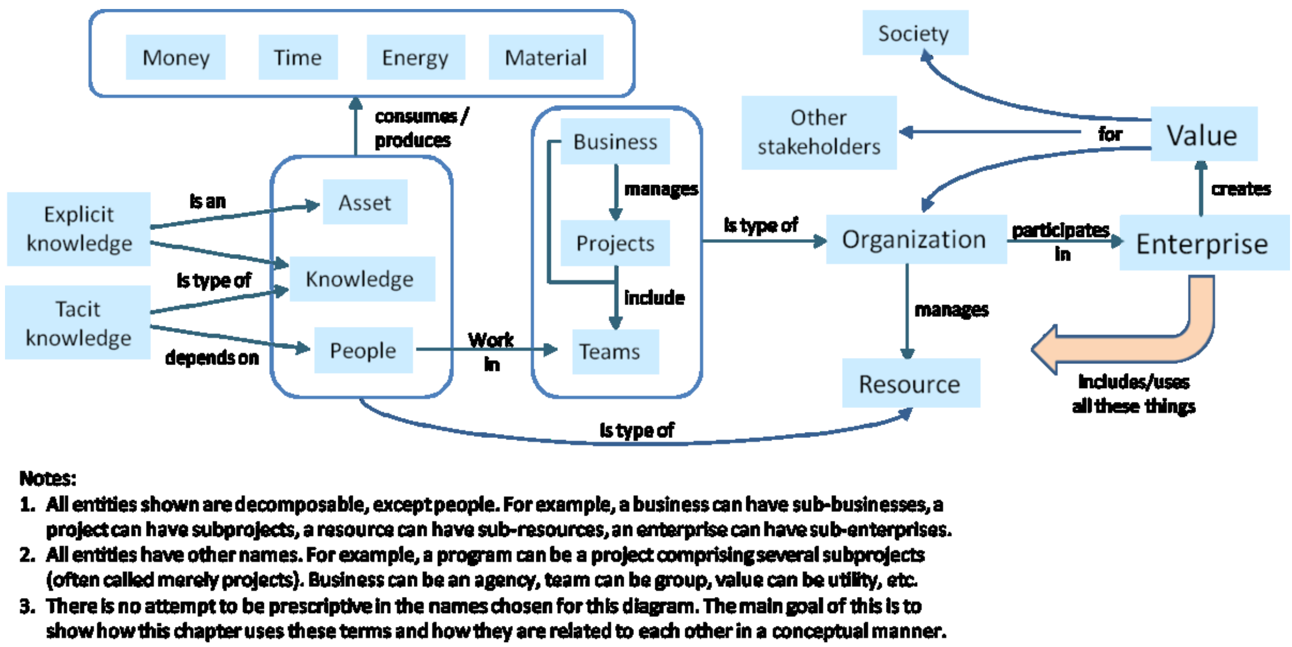


Figure 1. Organizations Manage Resources to Create Enterprise Value. (SEBoK Original)

Resource Optimization

A key choice for businesses that conduct SE is to what extent, if at all, they seek to optimize their use of resources (people, knowledge, assets) across teams, projects, and business units. Optimization of resources is not the goal in itself, but rather a means to achieve the goal of maximizing value for the enterprise and its stakeholders. At one extreme, in a product-oriented organization, projects may be responsible for hiring, training, and firing their own staff, as well as managing all assets required for their delivery of products or services. (The term "product-oriented organization" is not meant in the sense of product-oriented SE, but rather in the sense of this being one of the basic constructs available when formulating organizational strategy.)

At the other extreme, in a functional organization, the projects delegate almost all their work to functional groups. In between these two extremes is a matrix organization that is used to give functional specialists a "home" between project assignments. A full discussion of organizational approaches and situations along with their applicability in enabling SE for the organization is provided in the article called Systems Engineering Organizational Strategy.

The optimization debate can be handled as described in the book called "Enterprise Architecture as Strategy" (Ross, Weill, and Robertson 2006). In other words, an enterprise can choose (or not) to unify its operations and can choose (or not) to unify its information base. There are different strategies the enterprise might adopt to achieve and sustain value creation (and how ESE helps an enterprise to choose). This is further addressed in the section on Enterprise Architecture Formulation & Assessment in the article called Enterprise Capability Management.

Enabling Systems Engineering in the Organization

SE skills, techniques, and resources are relevant to many enterprise functions, and a well-founded SE capability can make a substantial contribution at the enterprise level, as well as at the project level. The article called Systems Engineering Organizational Strategy discusses enabling SE in the organization, while the article called Enabling Businesses and Enterprises focuses on the cross-organizational functions at the business and enterprise levels. The competence of individuals is discussed in the article called Enabling Individuals.

Kinds of Knowledge Used by the Enterprise

Knowledge is a key resource for ESE. There are generally two kinds of knowledge: explicit and tacit. Explicit knowledge can be written down or incorporated in computer codes. Much of the relevant knowledge, however, is “tacit knowledge” that only exists within the heads of people and in the context of relationships that people form with each other (e.g., team, project, and business level knowledge). The ability of an organization to create value is critically dependent on the people it employs, on what they know, how they work together, and how well they are organized and motivated to contribute to the organization’s purpose.

Projects, Programs & Businesses

The term “program” is used in various ways in different domains. In some domains a team can be called a program (e.g., a customer support team is their customer relationship "program"). In others, an entire business is called a program (e.g., a wireless communications business unit program), and in others the whole enterprise is called a program (e.g., the Joint Strike Fighter program and the Apollo Space program). And in many cases, the terms project and program are used interchangeably with no discernible distinction in their meaning or scope. Typically, but not always, there are program managers who have profit and loss (P&L) responsibility and are the ultimate program decision makers. A program manager may have a portfolio of items (services, products, facilities, intellectual property, etc.) that are usually provided, implemented, or acquired through projects.

The Office of Government Commerce provides a useful distinction between programs and projects:

> *The ultimate goal of a Programme is to realise outcomes and benefits of strategic relevance. To achieve this a programme is designed as a temporary flexible organisation structure created to coordinate, direct and oversee the implementation of a set of related projects and activities in order to deliver outcomes and benefits related to the organisation’s strategic objectives...*
>
> *A programme is likely to have a life that spans several years. A Project is usually of shorter duration (a few months perhaps) and will be focussed on the creation of a set of deliverables within agreed cost, time and quality parameters.* (OGC 2010)

Practical Considerations

When it comes to performing SE at the enterprise level, there are several good practices to keep in mind (Rebovich and White 2011):

- Set enterprise fitness as the key measure of system success. Leverage game theory and ecology, along with the practices of satisfying and governing the commons.
- Deal with uncertainty and conflict in the enterprise through adaptation: variety, selection, exploration, and experimentation.
- Leverage the practice of layered architectures with loose couplers and the theory of order and chaos in networks.

Enterprise governance involves shaping the political, operational, economic, and technical (POET) landscape. One should not try to control the enterprise like one would in a TSE effort at the project level.

References

Works Cited

BusinessDictionary.com, "Extended Enterprise." Accessed September 12, 2012. Available: http:/ / www. businessdictionary.com/definition/extended-enterprise.html.

CIO Council. 1999. *Federal Enterprise Architecture Framework (FEAF)*. Washington, DC, USA: Chief Information Officer (CIO) Council.

Dyer, L., and J. Ericksen. 2009. "Complexity-based Agile Enterprises: Putting Self-Organizing Emergence to Work," in *The Sage Handbook of Human Resource Management,* edited by A. Wilkinson et al. London, UK: Sage. p. 436–457.

Giachetti, R.E. 2010. *Design of Enterprise Systems: Theory, Architecture, and Methods*. Boca Raton, FL, USA: CRC Press, Taylor and Francis Group.

ISO. 2000. ISO 15704:2000, *Industrial Automation Systems -- Requirements for Enterprise-Reference Architectures and Methodologies*. Geneva, Switzerland: International Organization for Standardization (ISO).

MOD. 2004. *Ministry of Defence Architecture Framework (MODAF)*, version 2. London, UK: UK Ministry of Defence.

National Library of the the Netherlands. 2010. "Dossier VOC (1602-1799)." Accessed September 12, 2012. Available: http://www.kb.nl/dossiers/voc/voc.html (in Dutch).

OGC (Office of Government Commerce). 2010. *Guidelines for Managing Programmes: Understanding programmes and programme management*. London, UK: The Stationery Office.

Rebovich, G., and B.E. White (eds.). 2011. *Enterprise Systems Engineering: Advances in the Theory and Practice*. Boca Raton, FL, USA: CRC Press, Taylor & Francis Group, Auerbach.

Ross, J.W., P. Weill, and D. Robertson. 2006. *Enterprise Architecture As Strategy: Creating a Foundation for Business Execution*. Boston, MA, USA: Harvard Business Review Press.

Rouse, W.B. 2009. "Engineering the Enterprise as a System," in *Handbook of Systems Engineering and Management,* 2nd ed., edited by A.P. Sage and W. B. Rouse. New York, NY, USA: Wiley and Sons, Inc.

Stacey, R. 2006. "The Science of Complexity: An Alternative Perspective for Strategic Change Processes," in *Complexity and Organization: Readings and Conversations*, edited by R. MacIntosh et al. London, UK: Routledge. p. 74–100.

Wikipedia contributors, "Business agility," *Wikipedia, The Free Encyclopedia*. Accessed November 28, 2012. Available: http://en.wikipedia.org/w/index.php?title=Business_agility&oldid=503858042.

Primary References

Bernus, P., L. Nemes, and G. Schmidt (eds.). 2003. *Handbook on Enterprise Architecture*. Berlin and Heidelberg, Germany: Springer-Verlag.

Rebovich, G., and B.E. White (eds.). 2011. *Enterprise Systems Engineering: Advances in the Theory and Practice*. Boca Raton, FL, USA: CRC Press, Taylor & Francis Group, Auerbach.

Rouse, W.B. 2005. "Enterprise as Systems: Essential Challenges and Enterprise Transformation." *Systems Engineering, the Journal of the International Council on Systems Engineering (INCOSE)*. 8 (2): 138-50.

Rouse, W.B. 2009. "Engineering the Enterprise as a System," in *Handbook of Systems Engineering and Management*, 2nd ed., edited by A.P. Sage and W.B. Rouse. New York, NY, USA: Wiley and Sons, Inc.

Valerdi, R. and D.J. Nightingale. 2011. "An Introduction to the Journal of Enterprise Transformation." *Journal of Enterprise Transformation.* 1 (1): 1-6.

Additional References

Drucker, P.F. 1994. "The theory of business." *Harvard Business Review.* 72 (5): 95-104.

Fox, M., J.F. Chionglo, and F.G. Fadel. 1993. "A common sense model of the enterprise." Presented at the 3rd Industrial Engineering Research Conference, 1993, Norcross, GA, USA.

Joannou, P. 2007. "Enterprise, systems, and software—the need for integration." *Computer*. 40 (5): 103-105.

Journal of Enterprise Architecture. Available: http://www.globalaea.org/?page=JEAOverview.

MITRE. 2012. "Enterprise Engineering," in *Systems Engineering Guide*, MITRE Corporation. Accessed 8 July 2012. Available: http://www.mitre.org/work/systems_engineering/guide/enterprise_engineering/.

Nightingale, D., and J. Srinivasan. 2011. *Beyond the Lean Revolution: Achieving Successful and Sustainable Enterprise Transformation*. New York, NY, USA: AMACOM Press.

Nightingale, D., and R. Valerdi (eds). *Journal of Enterprise Transformation*. London, UK: Taylor & Francis. Available: http://www.tandf.co.uk/journals/UJET.

Saenz, O.A. 2005. "Framework for Enterprise Systems Engineering," in *FIU Electronic Theses and Dissertations*. Miami, FL, USA: Florida International University. Accessed September 12, 2012. Available: http:/ / digitalcommons.fiu.edu/cgi/viewcontent.cgi?article=1055&context=etd.

< Previous Article | Parent Article | Next Article >

SEBoK v. 2.1, released 31 October 2019

Enterprise Systems Engineering Background

Lead Authors: *James Martin, Dick Fairley, Bud Lawson*, ***Contributing Author:*** *Alan Faisandier*

This article provides a common context for the succeeding topics in the knowledge area.

Capabilities in the Enterprise

The enterprise acquires or develops systems or individual elements of a system. The enterprise can also create, supply, use, and operate systems or system elements. Since there could possibly be several organizations involved in this enterprise venture, each organization could be responsible for particular systems or perhaps for certain kinds of elements. Each organization brings their own organizational capability with them and the unique combination of these organizations leads to the overall operational capability of the whole enterprise. These concepts are illustrated below.

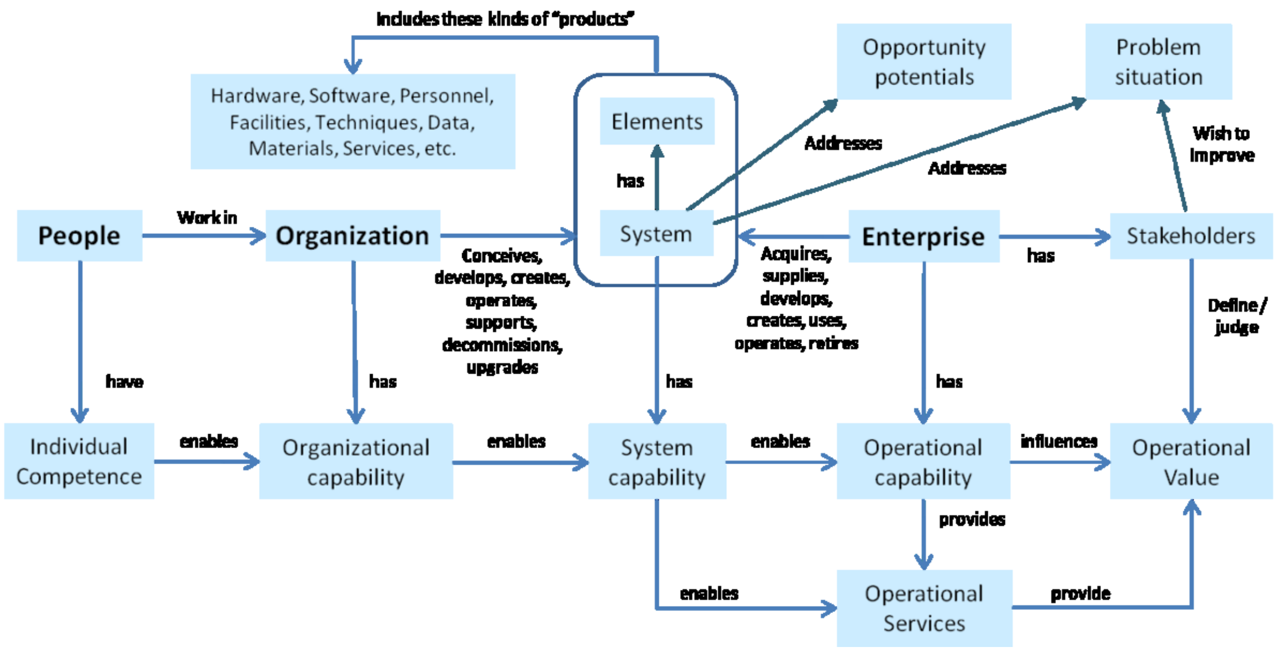


Figure 1. Individual Competence Leads to Organizational, System & Operational Capability. (SEBoK Original)

Organizational capabilities are addressed in the article on Systems Engineering Organizational Strategy, and individual competencies are addressed in the article on Enabling Individuals as they relate to the principles, theories, and practices of organizational behavior.

Organizational Capabilities and Competencies

The word "capability" is used in systems engineering (SE) in the sense of "the ability to do something useful under a particular set of conditions." This article discusses three different kinds of capabilities: organizational capability, system capability, and operational capability. It uses the word "competence" to refer to the ability of people relative to the SE task. Individual competence, (sometimes called "competency"), contributes to, but is not the sole determinant of, organizational capability. This competence is translated to organizational capabilities through the work practices that are adopted by the organizations. New systems (with new or enhanced system capabilities) are developed to enhance enterprise operational capability in response to stakeholder's concerns about a problem situation.

Enterprise stakeholders are the ultimate arbiters of value for the system to be delivered. Organizational, system, and operational capabilities cannot be designed, improved, and implemented independently. The key to understanding the dependencies between capabilities is through architecture modeling and analysis as part of the activities described in the article called Enterprise Capability Management. "Capability engineering" is an emerging discipline that could enhance the effectiveness of enterprise systems engineering (ESE), which is further discussed in the article on Systems of Systems (SoS).

Organizational Design

The competencies of individuals are important to the overall organizational capability as discussed in the article on Enabling Individuals. The organizational capability is also a function of how the people, teams, projects, and businesses are organized. The organizational design should specify the roles, authorities, responsibilities, and accountabilities (RARA) of the organizational units to ensure the most efficient and effective operations. Effectiveness of enterprise operations is certainly driven by management principles, concepts, and approaches, but it is also largely driven by its leadership principles, concepts, and approaches. These factors are discussed in the article on Systems Engineering Organizational Strategy that discusses how to organize for effective performance of SE.

Organizational structure is tightly tied to creating value for the enterprise's various stakeholders. Since the enterprise is made up of various elements including people, processes, technologies, and assets, the organizational structure of the people and the allocation of responsibilities for executing portions of the value stream is a "design decision" for the enterprise and hence is a key element of properly performing ESE. Organizational design is increasingly influenced by the portfolio of products and services and the degree of coupling between them. This organizational design will be based on organizational design patterns and their tradeoffs, as discussed in the article on Systems Engineering Organizational Strategy. Browning (2009) discusses one approach for modeling and analysis of an organization.

Operational Capabilities & Operational Services

As you can see in this figure, operational capabilities provide operational services that are enabled by system capabilities. These system capabilities are inherent in the system that is conceived, developed, created and/or operated by an enterprise. ESE concentrates its efforts on maximizing operational value for various stakeholders, some of whom may be interested in the improvement of some problem situation.

ESE, however, addresses more than just solving problems; it also deals with the exploitation of opportunities for better ways to achieve the enterprise goals. This opportunity might involve lowering of operating costs, increasing market share, decreasing deployment risk, reducing time to market, and any number of other enterprise goals. The importance of addressing opportunity potentials should not be underestimated in the execution of ESE practices.

This article focuses on the *operational capabilities* of an enterprise and the contribution of these capabilities to *operational value* (as perceived by the stakeholders). Notice that the organization or enterprise can deal with either the system as a whole or with only one (or a few) of its elements. These elements are not necessarily hard items, like hardware and software, but can also include "soft" items, like people, processes, principles, policies, practices, organizations, doctrine, theories, beliefs, and so on.

Services vs. Products vs. Enterprises

A service system is a collection of items (or entities) that perform the operations, administration, management and provisioning (OAM&P) of resources that together provide the opportunities to co-create value by both the service provider and the service consumer.

A collection of services is not necessarily a service system. In fact, this collection of services is often merely a product system that is one of the resources being OAM&P'ed by the service system. A product system can be composed of hardware, software, personnel (see note 1), facilities, data, materials, techniques, and even services. Each of these product system elements can be "engineered."

> *Note 1. Even personnel are engineered in the sense that their roles and responsibilities are specified precisely and trade-offs are made about which functions are performed by these people versus by hardware or software. People are "produced" in the sense that untrained people are trained to perform their allocated system functions, unknowledgeable people are educated to find or create the information they need to do their assigned task, and uninformed people are taught how to get access to the data they need, and how to extract relevant information from that data.*

It is important to understand the difference between the services "enabled" by a service system versus the services that are the elements of a service system entity. See the Service Systems Engineering article for more information about services and how they are engineered.

Likewise, a collection of services is not necessarily an enterprise system. An enterprise may be composed of service systems, along with product systems, as well as policies, procedures, properties, knowledge, financial capital, intellectual capital, and so on. An enterprise might even contain sub-enterprises. Enterprise SE must do the engineering not only across the enterprise itself, but may also get involved in the engineering of the service systems and products systems that the enterprise depends on in order to achieve its goals.

Enterprise Components

The above depictions of enterprise-related things do not show the components of an enterprise. The components of an enterprise when it is viewed as a “system” are different than the components of a product or service system (which is the focus of most literature on systems engineering). The figure below shows the typical kinds of components (shown here as “domains”) in an enterprise (Troux 2010) that could be utilized in achieving the desired enterprise *operational capability* as shown in Figure 1. It is this operational capability that drives ultimate value for the enterprise’s customers and other stakeholders. Further discussion on enterprise components is provided by Reese (2010) and Lawson (2010, chap. 8).

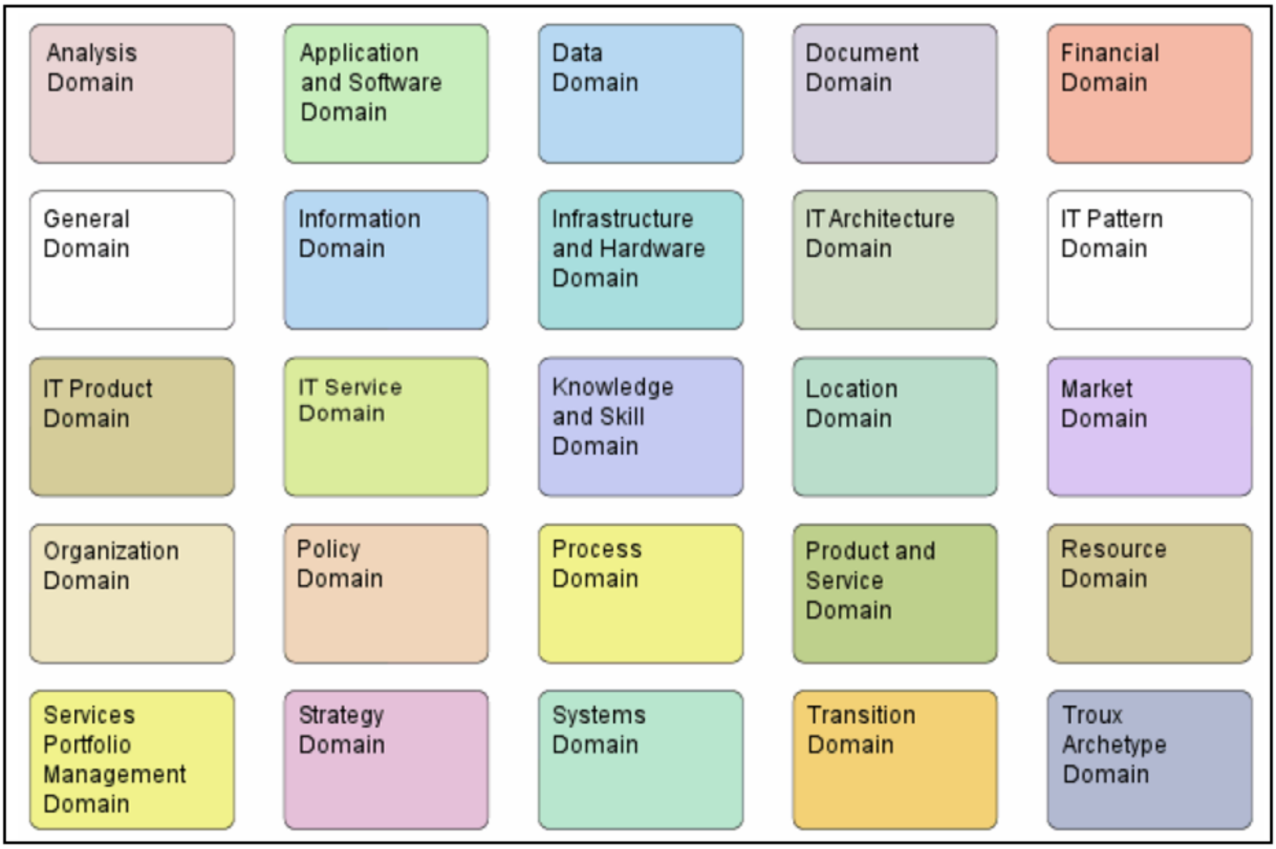


Figure 2. Categories of Enterprise Components (Troux Technologies, 2010). Reprinted with permission of Copyright © 2010 Troux Technologies. All other rights are reserved by the copyright owner.

The application/software and infrastructure/hardware domains (shown above) are likely the most familiar to systems engineers. The application/software domain contains things like the deployed software itself plus applications, modules, servers, patches, functions, and messages. The infrastructure/hardware domain contains things like the hardware itself plus networks and different kinds of hardware like computing hardware, cabinets, and network devices. There might different subtypes of computing hardware like computers, servers, desktops, laptops, and mainframes.

This particular "semantic model" had its origins in the area of information technology (IT) management but has been successfully expanded beyond the IT domain (Martin 2003 and 2005). You can see from this elaboration of these domains that an enterprise architecture "schema" can be quite extensive in the kinds of things it can model. The less technical domains would be things like policy, market, strategy, transition, financial, knowledge and skill, and analysis. In a typical enterprise architecture schema like this there could be over a hundred types of modeling objects grouped into these domains.

Various tools used in modeling the enterprise are described at http://www.enterprise-architecture.info/EA_Tools.htm (IEAD 2011). The TOGAF metamodel (http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap34.html) used in The Open Group Architecture Framework (TOGAF) is another useful depiction of the various modeling entities involved in modeling the enterprise (TOGAF 2009).

Scope of Enterprise SE

Computer and communications technologies make it easier to integrate activities across the enterprise, but this does not necessarily make the enterprise more effective and efficient. To enable this to happen, one needs to look at the whole enterprise as a system, rather than as a collection of functions connected solely by information systems and shared facilities.

Essential Challenges

Enterprises face strategic challenges that are essential to address in order to ensure that the enterprise will succeed (Rouse 2009):

- **Growth:** Increasing impact, perhaps in saturated/declining “markets”,
- **Value:** Enhancing relationships of processes to benefits and costs,
- **Focus:** Pursuing opportunities and avoiding diversions,
- **Change:** Competing creatively while maintaining continuity,
- **Future:** Investing in inherently unpredictable outcomes,
- **Knowledge:** Transforming information to insights to programs, and
- **Time:** Carefully allocating the organization’s scarcest resource.

To address these challenges, one recognizes that the central source of value in the enterprise is in its people. “Understanding and supporting the interests of an enterprise’s diverse stakeholders — and finding the ‘sweet spot’ among the many competing interests — is a central aspect of discerning the work of the enterprise as a system and creating mechanisms to enhance this work” (Rouse 2009).

Enterprise Transformation

Enterprises are constantly transforming, whether at the individual level (wherein individuals alter their work practices) or at the enterprise level (large-scale planned strategic changes) (Srinivasan 2010). These changes are a response on the part of the enterprise to evolving opportunities and emerging threats. It is not merely a matter of doing work better, but doing different work, which is often a more important result. Value is created through the execution of business processes. However, not all processes necessarily contribute to overall value (Rouse 2005, 138-150). It is important to focus on process and how they contribute to the overall value stream.

After gaining a good understanding of business processes, the next main concern is how best to deploy and manage the enterprise’s human, financial, and physical assets. The key challenge in transforming an enterprise is, in the midst of all this change, continuing to *satisfice* key stakeholders (see note 2).

> *Note 2. “Satisfice” means to decide on and pursue a course of action satisfying the minimum requirements to achieve a goal. For the enterprise as a whole, it is often impossible to completely satisfy all stakeholders given their competing and conflicting concerns and interests. Therefore, the concept of “satisficing” is a very important element in the execution of ESE practices. It has less stringent criteria than the concept of "satisfaction," which is commonly used in product/service systems engineering.*

Systems engineers have to respond to an increased recognition of the ‘connectedness’ of products and systems, brought about by a number of trends, for example: the capability of (mainly digital) technology, working across multiple systems, to transform businesses and operational systems; the need to create systems in families to increase product diversity and reuse technology, in order to reduce development and operating costs; and the need to build systems which can be brought together flexibly in operations, even if such co-operation was not foreseen at the time of development.

There has also been an increase in collaborative systems development activities, often spanning national boundaries. This has proceeded alongside a growth in the development of what might be called *meta-systems,* that is systems comprising parts which would previously have been considered as complex in their own right a generation ago, now conceived of and developed as a whole, and thus requiring fresh approaches, of the adaption of old ones.

Tackling these issues requires an approach that transcends the technical and process domain. ESE needs to address integration at the organizational and value chain level.

Transformation Context

Enterprise transformation occurs in the external context of the economy and markets as shown in the figure below (Rouse 2009). The “market” for the enterprise can be thought of as the context in which the enterprise operates. Of course, in the public sector, the enterprise’s “market” is commonly known as its “constituency.”

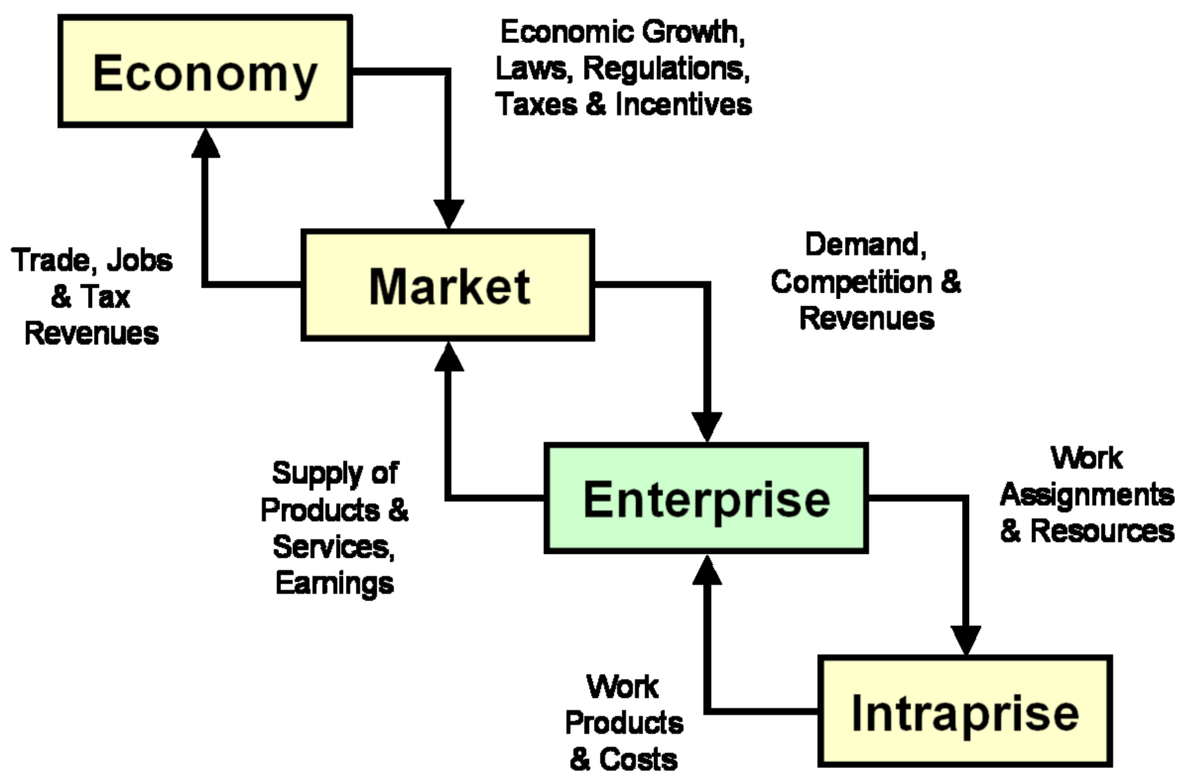


Figure 3. Context for Enterprise Transformation (Rouse 2009). Reprinted with permission of John Wiley & Sons Inc. All other rights are reserved by the copyright owner.

The term “intraprise” is used here to denote the many systems internal to the enterprise. This includes "information systems such as... ERP [enterprise resource planning] systems, as well as social and cultural systems. More specifically, work assignments are pursued via work processes and yield work products, incurring costs" (Rouse 2009). The social and cultural aspects of an enterprise are addressed further in the article called Enabling Businesses and Enterprises.

Modeling the Enterprise

Models of the enterprise can serve as the basis for understanding the enterprise in its context of markets and economies. The figure below shows the various drivers (or inputs) of an enterprise and its potential outcomes (or outputs) (Rouse 2009). Enterprise architecture can be a key enabler for modeling and can serve as a basis for transformation (Vernadat 1996; Bernus, Laszlo, and Schmidt 2003; Nightingale and Rhodes 2004). Enterprise architecture can be used to provide a model to understand how the parts of the enterprise fit together (or do not) (Giachetti 2010) (See also Representing Systems with Models). For a good review of the subject see Lillehagen and Krogstie (2008).

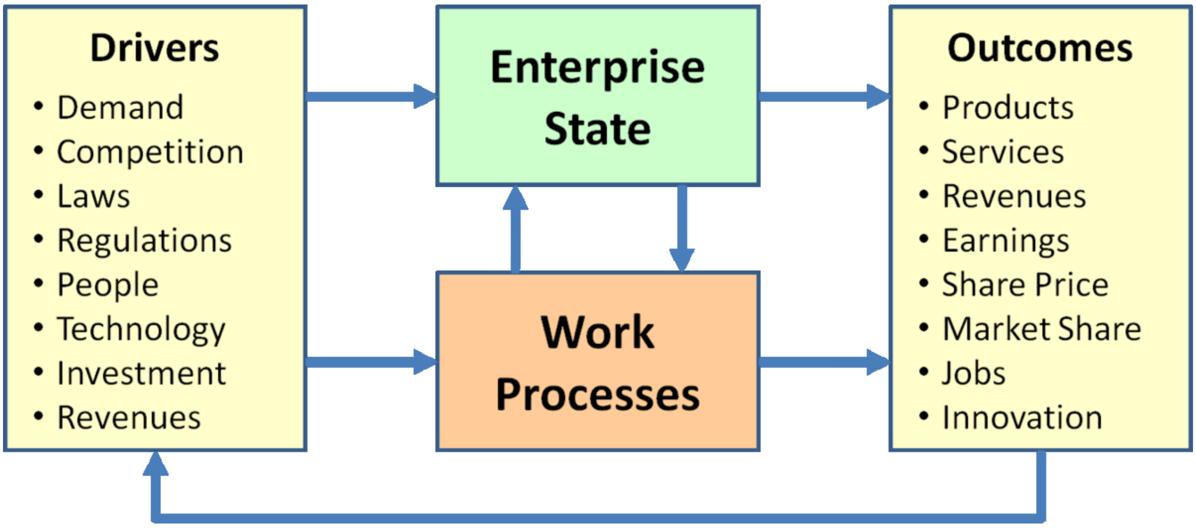


Figure 4. Drivers and Outcomes for the Enterprise (Rouse 2009). Reprinted with permission of John Wiley & Sons Inc. All other rights are reserved by the copyright owner.

In Pursuit of Value

Based on his theory of enterprise transformation, Rouse (2005, 279-295) has identified four alternative perspectives that tend to drive the need for transformation:

1. **Value Opportunities:** The lure of greater success via market and/or technology opportunities prompts transformation initiatives.
2. **Value Threats:** The danger of anticipated failure due to market and/or technology threats prompts transformation initiatives.
3. **Value Competition:** Other players' transformation initiatives prompt recognition that transformation is necessary to continued success.
4. **Value Crises:** Steadily declining market performance, cash flow problems, etc., prompt recognition that transformation is necessary for the enterprise to survive.

Work processes can be enhanced, streamlined, eliminated, and invented to help in the pursuit of enhanced value. These process changes should be aligned with enterprise strategy to maximize value produced by the enterprise (Hammer and Champy 1993). As shown below, there are many entities involved in helping the enterprise create value for society, participating organizations, and other stakeholders.

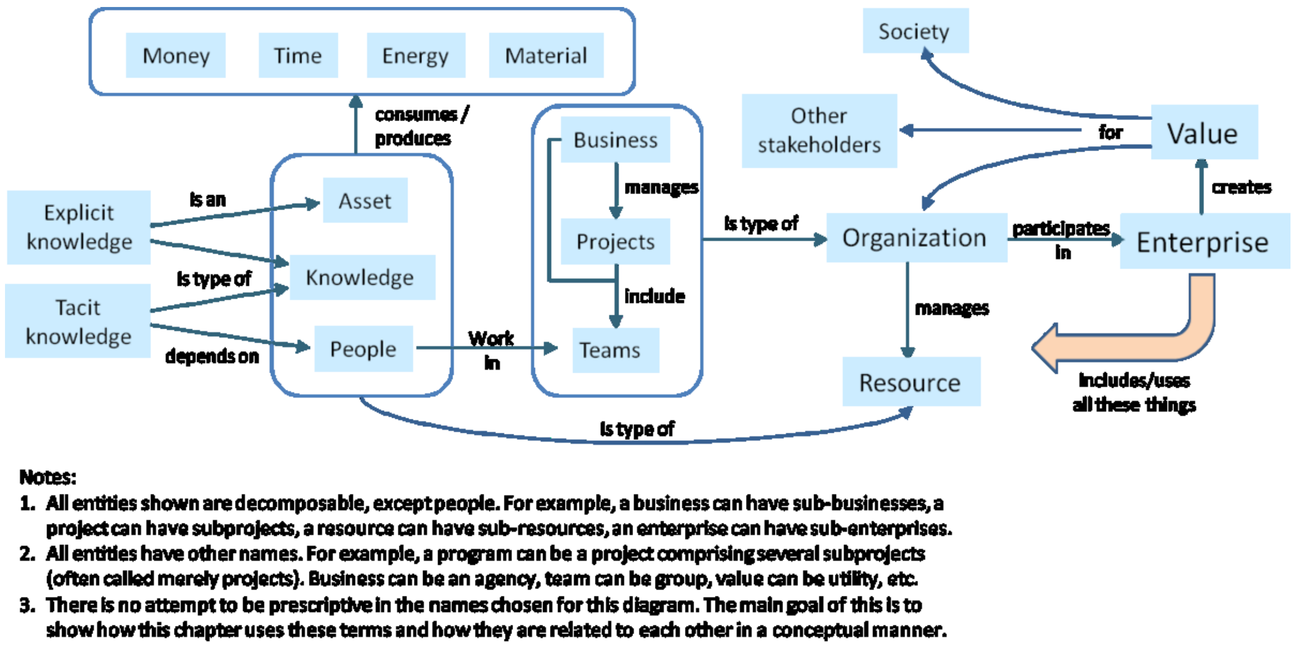


Figure 5. Organizations Manage Resources to Create Enterprise Value. (SEBoK Original)

References

Works Cited

Browning, T.R. 2009. "Using the Design Structure Matrix to Design Program Organizations," in *Handbook of Systems Engineering and Management*, 2nd ed., edited by A.P. Sage and W.B. Rouse. New York, NY: Wiley and Sons, Inc.

Giachetti, R.E. 2010. *Design of Enterprise Systems: Theory, Architecture, and Methods*. Boca Raton, FL, USA: CRC Press, Taylor & Francis Group.

Hammer, M., and J. Champy. 1993. *Reengineering the Corporation: A Manifesto for Business Revolution*. New York, NY: Harper Business, HarperCollins Publishers.

IEAD. 2011. "Enterprise Architecture Tools." Institute for Enterprise Architecture Developments. Accessed September 12, 2012. Available: http://www.enterprise-architecture.info/EA_Tools.htm.

Lawson, H. 2010. *A Journey Through the Systems Landscape*. Kings College, UK: College Publications.

Lillehagen, F., and J. Krogstie. 2008. "Chapter 4: State of the Art of Enterprise Modelling," in *Active Knowledge Management of Enterprises*. New York, NY, USA: Springer.

Martin, J.N. 2003. "On the Use of Knowledge Modeling Tools and Techniques to Characterize the NOAA Observing System Architecture." Presented at 13th Annual International Council on Systems Engineering (INCOSE) International Symposium, 2003, Arlington, VA, USA.

Martin, J.N. 2005. "Using an Enterprise Architecture to Assess the Societal Benefits of Earth Science Research." Presented at 15th Annual International Council on Systems Engineering (INCOSE) International Symposium, 2005, Rochester, NY, USA.

Miller, J., and S. Page. 2007. *Complex Adaptive Systems: An Introduction to Computational Models of Social Life*. Princeton, NJ, USA: Princeton University Press.

Reese, R.J. 2010. *Troux Enterprise Architecture Solutions*. Birmingham, UK: Packt Publishing Ltd.

Rouse, W.B. 2005. "Enterprise as Systems: Essential Challenges and Enterprise Transformation." *Systems Engineering, the Journal of the International Council on Systems Engineering (INCOSE)*. 8 (2): 138-150.

Rouse, W.B. 2009. "Engineering the Enterprise as a System," in *Handbook of Systems Engineering and Management*, 2nd ed., edited by A.P. Sage and W.B. Rouse. New York, NY: Wiley and Sons, Inc.

Srinivasan, J. 2010. "Towards a Theory Sensitive Approach to Planning Enterprise Transformation." Presented at 5th European Institute for Advanced Studies in Management (EIASM) Workshop on Organizational Change and Development, September 23-24, 2010, Vienna, Austria.

TOGAF 2009. "The Open Group Architecture Framework," version 9. The Open Architecture Group. Accessed September 2, 2011. Available: http://www.opengroup.org/togaf.

Troux. 2010. *Metamodeling and modeling with Troux Semantics,* version 9. Austin, TX, USA: Troux Technologies.

White, B.E. 2009. "Complex Adaptive Systems Engineering (CASE)." Presented at IEEE Systems Conference, March 23-26, 2009, Vancouver, Canada.

Primary References

Giachetti, R.E. 2010. *Design of Enterprise Systems: Theory, Architecture, and Methods*. Boca Raton, FL, USA: CRC Press, Taylor & Francis Group.

Rouse, W.B. 2009. "Engineering the Enterprise as a System," in *Handbook of Systems Engineering and Management*, 2nd ed., edited by A.P. Sage and W.B. Rouse. New York, NY: Wiley and Sons, Inc.

Srinivasan, J. 2010. "Towards a Theory Sensitive Approach to Planning Enterprise Transformation." Presented at 5th European Institute for Advanced Studies in Management (EIASM) Workshop on Organizational Change and Development, September 23-24, 2010, Vienna, Austria.

White, B.E. 2009. "Complex Adaptive Systems Engineering (CASE)." Presented at IEEE Systems Conference, March 23-26, 2009, Vancouver, Canada.

Additional References

McCarter, B.G., and B.E. White. 2009. "Emergence of SoS, sociocognitive aspects," in *Systems of systems engineering: Principles and applications*, edited by M. Jamshidi. Boca Raton, FL, USA: CRC Press, Taylor & Francis Group. p. 71-105.

Rouse, W.B. 2008. "Health Care as a Complex Adaptive System: Implications for design and management." *The Bridge, National Academy of Engineering*. 38 (1): 17-25.

Sage, A.P. 2000. "Transdisciplinarity Perspectives in Systems Engineering and Management," in *Transdiciplinarity: Recreating Integrated Knowledge*, edited by M.A. Somerville and D. Rappaport. Oxford, UK: EOLSS Publishers. p. 158-169.

von Bertalanffy, L. 1968. *General System Theory: Foundations, Development, Applications*, revised ed. New York, NY, USA: Braziller.

Weinberg, G., and D. Weinberg. 1988. *General Principles of Systems Design*. New York, NY, USA: Dorset House Publishing Company. White, B.E. 2007. "On Interpreting Scale (or View) and Emergence in Complex Systems Engineering." Presented at 1st Annual IEEE Systems Conference, April 9-12, 2007, Honolulu, HI, USA.

< Previous Article | Parent Article | Next Article >

SEBoK v. 2.1, released 31 October 2019

The Enterprise as a System

Lead Authors: *James Martin, Bud Lawson, Judith Dahmann*

To enable more efficient and effective enterprise transformation, the enterprise needs to be looked at “as a system,” rather than as a collection of functions connected solely by information systems and shared facilities (Rouse 2005 and 2009; Lawson 2010). What distinguishes the design of enterprise systems from product systems is the inclusion of people as a component of the system, not merely as a user/operator of the system.

> *The term 'enterprise system' has taken on a narrow meaning of only the information system an organization uses. Research and project experience has taught us that to design a good enterprise system, we need to adopt a much broader understanding of enterprise systems. The greater view of enterprise systems is inclusive of the processes the system supports, the people who work in the system, and the information [and knowledge] content of the system.* (Giachetti 2010)

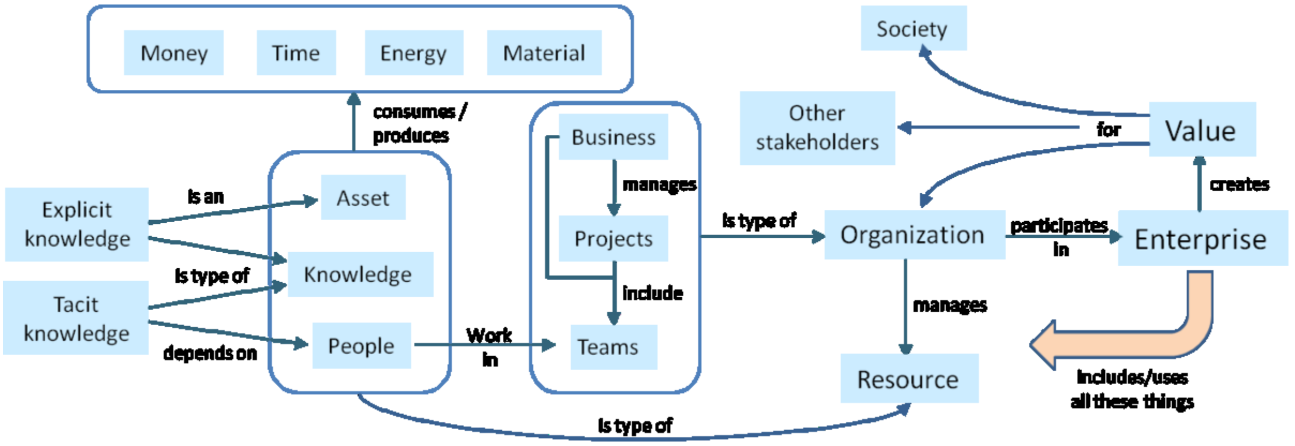
It is worth noting that the concept of "service" systems also includes people in the system. The thoughts above do not take this into account, primarily since their perspectives come mainly from a product system experience. The practice of service systems engineering is relatively new and is an emerging discipline. For more information on this, see the articles on Service Systems Engineering.

Creating Value

The primary purpose of an enterprise is to create value for society, other stakeholders, and for the organizations that participate in that enterprise. This is illustrated in Figure 1 that shows all the key elements that contribute to this value creation process. These elements in the enterprise can be treated as a "system" and the processes, methods, and tools ESE can be applied.

There are three types of organizations of interest: businesses, projects, and teams (see note 1). A typical business participates in multiple enterprises through its portfolio of projects. Large SE projects can be enterprises in their own right, with participation by many different businesses, and may be organized as a number of sub-projects.

> *Note 1. The use of the word “business” is not intended to mean only for-profit commercial ventures. As used here, it also includes government agencies and not-for-profit organizations, as well as commercial ventures. Business is the activity of providing goods and services involving financial, commercial, and industrial aspects.*



Notes:
1. **All entities shown are decomposable, except people. For example, a business can have sub-businesses, a project can have subprojects, a resource can have sub-resources, an enterprise can have sub-enterprises.**
2. **All entities have other names. For example, a program can be a project comprising several subprojects (often called merely projects). Business can be an agency, team can be group, value can be utility, etc.**
3. **There is no attempt to be prescriptive in the names chosen for this diagram. The main goal of this is to show how this chapter uses these terms and how they are related to each other in a conceptual manner.**

Figure 1. Organizations Manage Resources to Create Enterprise Value. (SEBoK Original)

Resource Optimization

A key choice for businesses that conduct SE is to what extent, if at all, they seek to optimize their use of resources (people, knowledge, assets) across teams, projects, and business units. Optimization of resources is not the goal in itself, but rather a means to achieve the goal of maximizing value for the enterprise and its stakeholders. At one extreme, in a product-oriented organization, projects may be responsible for hiring, training, and firing their own staff, as well as managing all assets required for their delivery of products or services. (The term "product-oriented organization" is not meant in the sense of product-oriented SE, but rather in the sense of this being one of the basic constructs available when formulating organizational strategy.)

At the other extreme, in a functional organization, the projects delegate almost all their work to functional groups. In between these two extremes is a matrix organization that is used to give functional specialists a “home” between project assignments. A full discussion of organizational approaches and situations along with their applicability in enabling SE for the organization is provided in the article called Systems Engineering Organizational Strategy.

The optimization debate can be handled as described in the book called "Enterprise Architecture as Strategy" (Ross, Weill, and Robertson 2006). In other words, an enterprise can choose (or not) to unify its operations and can choose (or not) to unify its information base. There are different strategies the enterprise might adopt to achieve and sustain value creation (and how ESE helps an enterprise to choose). This is further addressed in the section on Enterprise Architecture Formulation & Assessment in the article called Enterprise Capability Management.

Enabling Systems Engineering in the Organization

SE skills, techniques, and resources are relevant to many enterprise functions, and a well-founded SE capability can make a substantial contribution at the enterprise level, as well as at the project level. The article called Systems Engineering Organizational Strategy discusses enabling SE in the organization, while the article called Enabling Businesses and Enterprises focuses on the cross-organizational functions at the business and enterprise levels. The competence of individuals is discussed in the article called Enabling Individuals.

Kinds of Knowledge Used by the Enterprise

Knowledge is a key resource for ESE. There are generally two kinds of knowledge: explicit and tacit. Explicit knowledge can be written down or incorporated in computer codes. Much of the relevant knowledge, however, is "tacit knowledge" that only exists within the heads of people and in the context of relationships that people form with each other (e.g., team, project, and business level knowledge). The ability of an organization to create value is critically dependent on the people it employs, on what they know, how they work together, and how well they are organized and motivated to contribute to the organization's purpose.

Projects, Programs, and Businesses

The term "program" is used in various ways in different domains. In some domains a team can be called a program (e.g., a customer support team is their customer relationship "program"). In others, an entire business is called a program (e.g., a wireless communications business unit program), and in others the whole enterprise is called a program (e.g., the Joint Strike Fighter program and the Apollo Space program). And in many cases, the terms project and program are used interchangeably with no discernible distinction in their meaning or scope. Typically, but not always, there are program managers who have profit and loss (P&L) responsibility and are the ultimate program decision makers. A program manager may have a portfolio of items (services, products, facilities, intellectual property, etc.) that are usually provided, implemented, or acquired through projects.

The Office of Government Commerce provides a useful distinction between programs and projects:

> *The ultimate goal of a Programme is to realise outcomes and benefits of strategic relevance. To achieve this a programme is designed as a temporary flexible organisation structure created to coordinate, direct and oversee the implementation of a set of related projects and activities in order to deliver outcomes and benefits related to the organisation's strategic objectives...*
>
> *A programme is likely to have a life that spans several years. A Project is usually of shorter duration (a few months perhaps) and will be focussed on the creation of a set of deliverables within agreed cost, time and quality parameters.* (OGC 2010)

Enabling the Enterprise

ESE, by virtue of its inherent trans-disciplinarity (Sage 2000, 158-169) in dealing with problems that are large in scale and scope, can better enable the enterprise to become more effective and efficient. The complex nature of many enterprise problems and situations usually goes beyond the abilities of standard tools and techniques provided to business school graduates (See also Complexity). ESE can augment the standard business management methods using the tools and methods from the SE discipline to more robustly analyze and evaluate the enterprise as a holistic system. A more general viewpoint, or "view," for dealing with the enterprise consisting of scale, granularity, mindset, and time frame is provided by White (2007) and by McCarter and White (2009, 71-105).

ESE can provide the enablers to address the concerns of enterprise executives as shown in Table 1 (Rouse 2009). The methods for dealing with, and the special characteristics of, complex adaptive systems must be properly considered when adapting traditional systems engineering (TSE) practices for use at the enterprise level—many of which come out of the systems science and systems thinking domains (von Bertalanffy 1968; Weinberg and Weinberg 1988; Miller and Page 2007; Rouse 2008, 17-25). For an approach to complex adaptive systems (CAS) engineering, refer to White (2009, 1-16) and to McCarter and White (2009, 71-105).

Table 1. Executive Concerns and SE Enablers (Rouse 2009). Reprinted with permission of John Wiley & Sons Inc. All other rights are reserved by the copyright owner.

Executive Concerns	SE Enablers
Identifying ends, means, and scope and candidate changes	System complexity analysis to compare "as is" and "to be" enterprises
Evaluating changes in terms of process behaviors and performance	Organizational simulation of process flows and relationships
Assessing economics in terms of investments, operating costs, and returns	Economic modeling in terms of cash flows, volatility, and options
Defining the new enterprise in terms of processes and their integration	Enterprise architecting in terms of workflow, processes, and levels of maturity
Designing a strategy to change the culture for selected changes	Organizational and cultural change via leadership, vision, strategy, and incentives
Developing transformation action plans in terms of what, when, and who	Implementation planning in terms of tasks, schedule, people, and information

Enterprise Engineering

Another distinction is that "enterprise design does not occur at a single point in time like the design of most systems. Instead, enterprises evolve over time and are constantly changing, or are constantly *being designed*" (Giachetti 2010) [emphasis in original]. Giachetti calls this new discipline "enterprise engineering." We consider the enterprise engineering set of practices to be equivalent to what we call enterprise systems engineering (ESE) in this article.

> *The body of knowledge for enterprise engineering is evolving under such titles as enterprise engineering, business engineering, and enterprise architecture . . . . Many systems and software engineering principles are applicable to enterprise engineering, but enterprise engineering's unique complexities require additional principles…. Enterprise engineering's intent is to deliver a targeted level of enterprise performance in terms of shareholder value or customer satisfaction . . . . Enterprise engineering methods include modeling; simulation; total quality management; change management; and bottleneck, cost, workflow, and value-added analysis.* (Joannou 2007)

Supersystem Constructs

System of Systems (SoS)

The phrase "system of systems" (SoS) is commonly used, but there is no widespread agreement on its exact meaning, nor on how it can be distinguished from a conventional system. A system is generally understood to be a collection of elements that interact in such a manner that it exhibits behavior that the elements themselves cannot exhibit. Each element (or component) of the system can be regarded as a system in its own right. Therefore, the phrase "system of systems" can technically be used for any system and, as such, would be a superfluous term. However, the meaning of this phrase has been examined in detail by (Maier 1998, 267-284), and his definition has been adopted by some people (AFSAB 2005). Maier provides this definition:

> *A SoS is an assemblage of components which individually may be regarded as systems, and which possess two additional properties:*

- ***Operational Independence of the Components:*** *If the system-of-systems is disassembled into its component systems the component systems must be able to usefully operate independently. That is, the components fulfill customer-operator purposes on their own; and*
- ***Managerial Independence of the Components:*** *The component systems not only can operate independently, they do operate independently. The component systems are separately acquired and integrated but maintain a*

continuing operational existence independent of the system-of-systems. (Maier 1998, 267-284)

Maier goes on further saying that “the commonly cited characteristics of systems-of-systems (complexity of the component systems and geographic distribution) are not the appropriate taxonomic classifiers” (Maier 1998, 267-284). Four kinds of SoS have been defined (Dahmann, Lane, and Rebovich 2008).

For further details on SoS, see the *Systems Engineering Guide for SoS* developed by the US Department of Defense (DoD) (DUS(AT) 2008). Also, see the Systems of Systems (SoS) knowledge area.

Federation of Systems (FoS)

Different from the SoS concept, but related to it in several ways, is the concept called “federation of systems” (FoS). This concept might apply when there is a very limited amount of centralized control and authority (Sage and Cuppan 2001, 325-345; Sage and Rouse 2009). Each system in an FoS is very strongly in control of its own destiny, but “chooses” to participate in the FoS for its own good and the good of the “country,” so to speak. It is a coalition of the willing. An FoS is generally characterized by significant autonomy, heterogeneity, and geographic distribution or dispersion (Krygiel 1999). Krygiel defined a taxonomy of systems showing the relationships among conventional systems, SoSs, and FOSs.

This taxonomy has three dimensions: autonomy, heterogeneity, and dispersion. A FoS would have a larger value on each of these three dimensions than a non-federated SoS. An “Enterprise System,” as described above, could be considered to be an FoS if it rates highly on these three dimensions. However, it is possible for an enterprise to have components that are not highly autonomous, that are relatively homogeneous, and are geographically close together. Therefore, it would be incorrect to say that an enterprise is necessarily the same as an FoS.

Dove points out that in order for a large enterprise to survive in the twenty-first century, it must be more agile and robust (Dove 1999 and 2001). Handy (1992, 59-67) describes a federalist approach called “New Federalism” which identifies the need for structuring of loosely coupled organizations to help them adapt to the rapid changes inherent in the Information Age. This leads to the need for virtual organizations where alliances can be quickly formed to handle the challenges of newly identified threats and a rapidly changing marketplace (Handy 1995, 2-8). Handy sets out to define a number of federalist political principles that could be applicable to an FoS. Handy’s principles have been tailored to the domain of systems engineering (SE) and management by Sage and Cuppan (2001, 325-345):

- Subsidiarity,
- Interdependence,
- Uniform and standardized way of doing business,
- Separation of powers,
- Dual citizenship, and
- Scales of SE.

Scales of SE

According to Maier’s definition, not every enterprise would be called a SoS since the systems within the enterprise do not usually meet the criteria of operational and managerial independence. In fact, one of the key purposes of an enterprise is to explicitly establish operational dependence between systems that the enterprise owns and/or operates in order to maximize the efficiency and effectiveness of the enterprise as a whole. Therefore, it is more proper to treat an enterprise system and an SoS as different types of things, with different properties and characteristics. This distinction is illustrated in the figure below, where three corresponding categories of SE are shown (DeRosa 2005; Swarz et al. 2006).

It is true that an enterprise can be treated as a system itself and is comprised of many systems within the enterprise, but this discussion will reserve the term SoS to those systems that meet the criteria of operational and managerial independence. This distinction was also used within the MITRE Corporation in their ESE Office (Rebovich and White 2011).

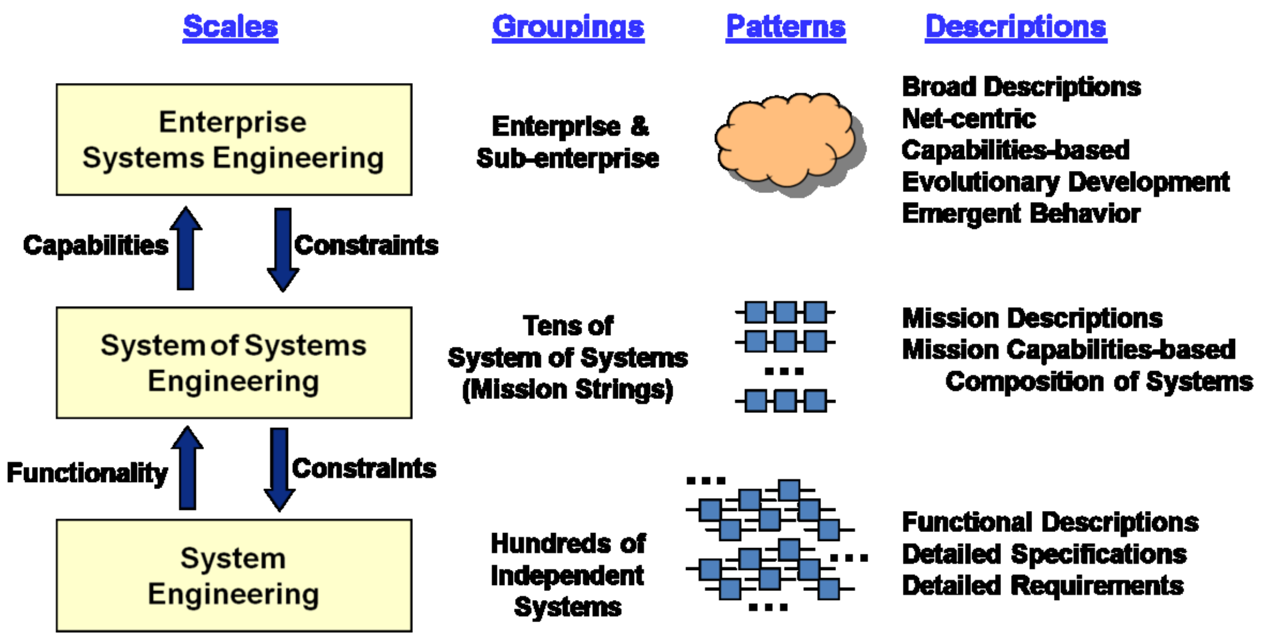


Figure 2. Different Groupings and Patterns Revealed at Different Scales (DeRosa 2005). Reprinted with permission of © 2011. The MITRE Corporation. All Rights Reserved. All other rights are reserved by the copyright owner.

Relationships between Enterprise and SoS

An enterprise may require a particular operational capability that is brought into being by connecting together a chain of systems that together achieve that capability. Any one of these systems in the chain cannot by itself provide this capability. The desired capability is the emergent property of this chain of systems. This chain of systems is sometimes called an SoS. However, the enterprise that requires this capability rarely has direct control over all the systems necessary to provide this full capability. This situation is illustrated in the figure below (Martin 2010).

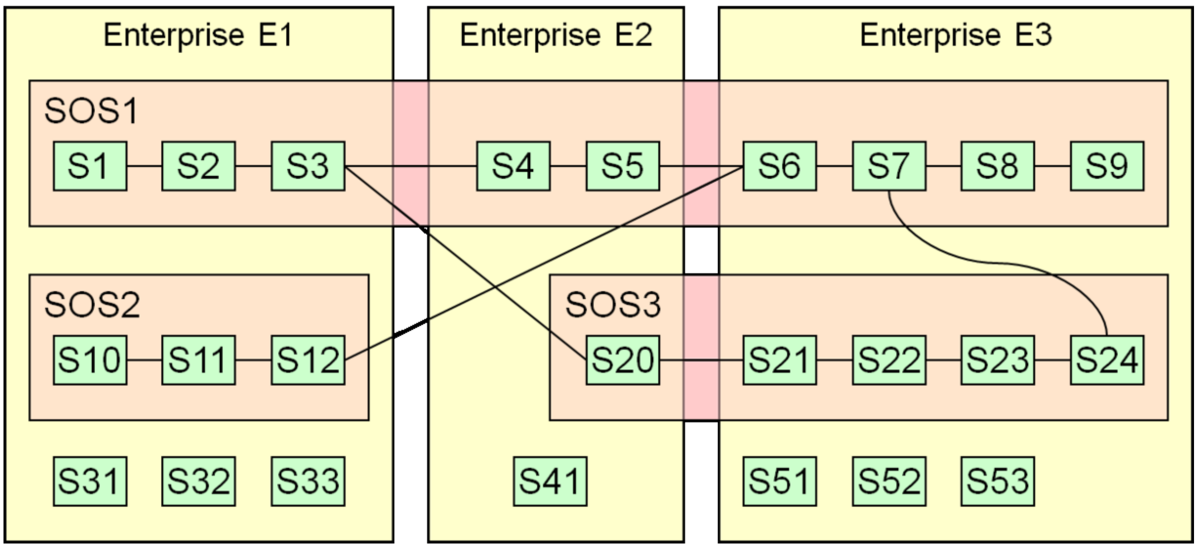


Figure 3. Relationships Between an Enterprise and SoSs (Martin 2010). Reprinted with permission of The Aerospace Corporation. All other rights are reserved by the copyright owner.

Enterprise E1 (in the example above) has full control over SoS2, but not full control over SoS1. TSE can be applied to the individual systems (S1, S2, …, S53) shown within each enterprise, but needs to be augmented with additional activities to handle SoS and enterprise kinds of issues.

There is a general issue regarding dealing with enterprises in this situation: there are at least two enterprises related to any particular SoS. First, there is the enterprise of builders/developers comprising projects and programs, which have to be organized appropriately and adopt special types of architectural principles. Second, there is the enterprise of users (those who use the products and service provided by the first enterprise), which has to exercise its own sort of agility. How the first enterprise designs systems to allow the second to operate is the core issue.

References

Works Cited

AFSAB. 2005. *Report on System-of-Systems Engineering for Air Force Capability Development*. Washington, DC: US Air Force Scientific Advisory Board (AFSAB), US Air Force. SAB-TR-05-04.

Dahmann, J.S., J.A. Lane, and G. Rebovich. 2008. "Systems Engineering for Capabilities." *CROSSTALK: The Journal of Defense Software Engineering*. 21 (11): 4–9.

DeRosa, J.K. 2005. "Enterprise Systems Engineering." Presented at Air Force Association, Industry Day, Day 1, August 4, 2005, Danvers, MA, USA.

Dove, R. 2001. *Response Ability: The Language, Structure, and Culture of the Agile Organization*. New York, NY, USA: John Wiley & Sons.

Dove, R. 1999. "Knowledge Management, Response Ability, and the Agile Enterprise," in Paradigm Shift International [database online]. Accessed September 6, 2011. Available: http:/ / www. parshift. com/ docs/ KmRaAeX.htm.

DUS(AT). 2008. *Systems Engineering Guide for Systems of Systems*, version 1.0. Washington, DC, USA: Deputy Under Secretary of Defense for Acquisition and Technology (DUS(AT)) / US Department of Defense (DoD). Accessed September 6, 2011. Available: http://www.acq.osd.mil/se/docs/SE-Guide-for-SoS.pdf.

Giachetti, R.E. 2010. *Design of Enterprise Systems: Theory, Architecture, and Methods*. Boca Raton, FL, USA: CRC Press, Taylor and Francis Group.

Handy, C. 1995. "Trust and the Virtual Organization." *Harvard Business Review*. 73 (3): 2-8.

Handy, C. 1992. "Balancing Corporate Power: A New Federalist Paper." *Harvard Business Review*. 70 (6): 59-67.

Joannou, P. 2007. "Enterprise, Systems, and Software—The Need for Integration." *Computer*. 40 (5): 103-105.

Krygiel, A.J. 1999. *Behind the Wizard's Curtain: An Integration Environment for a System of Systems*. Arlington, VA, USA: C4ISR Cooperative Research Program (CCRP).

Lawson, H. 2010. *A Journey Through the Systems Landscape*. Kings College, UK: College Publications.

Maier, M.W. 1998. "Architecting Principles for Systems-of-Systems." *Systems Engineering, the Journal of the International Council on Systems Engineering (INCOSE). 1 (4): 267-84.*

Martin, J.N. 2010. "An Enterprise Systems Engineering Framework." Presented at 20th Anniversary International Council on Systems Engineering (INCOSE) International Symposium, July 12-15, 2010, Chicago, IL, USA.

McCarter, B.G., and B.E. White. 2009. "Emergence of SoS, sociocognitive aspects," in *Systems of systems engineering: Principles and applications*, edited by M. Jamshidi. Boca Raton, FL, USA: CRC Press, Taylor & Francis Group. p. 71-105.

OGC (Office of Government Commerce). 2010. *Guidelines for Managing Programmes: Understanding programmes and programme management*. London, UK: The Stationery Office.

Ross, J.W., P. Weill, and D. Robertson. 2006. *Enterprise Architecture As Strategy: Creating a Foundation for Business Execution*. Boston, MA, USA: Harvard Business Review Press.

Rouse, W.B. 2009. "Engineering the Enterprise as a System," in *Handbook of systems engineering and management,* 2nd ed., edited by A.P. Sage and W.B. Rouse. New York, NY, USA: Wiley and Sons, Inc.

Rouse, W.B. 2008. "Health Care as a Complex Adaptive System: Implications for design and management." *The Bridge, National Academy of Engineering.* 38 (1): 17-25.

Rouse, W.B. 2005. "Enterprise as Systems: Essential Challenges and Enterprise Transformation." *Systems Engineering*. 8 (2): 138-50.

Sage, A.P. 2000. "Transdisciplinarity Perspectives in Systems Engineering and Management." in *Transdiciplinarity: Recreating Integrated Knowledge*, edited by M.A. Somerville and D. Rappaport. Oxford, UK: EOLSS Publishers. p. 158-169.

Sage, A., and C. Cuppan. 2001. "On the Systems Engineering and Management of Systems of Systems and Federations of Systems." *Information-Knowledge-Systems Management Journal*. 2 (4): 325-345.

Sage, A.P., and W.B. Rouse (eds). 2009. *Handbook of System Engineering and Management*, 2nd ed. New York, NY, USA: John Wiley & Sons.

Swarz, R.S., J.K. DeRosa, and G. Rebovich. 2006. "An Enterprise Systems Engineering Model." Proceedings of the INCOSE International Symposium, July 9-13, 2006, Orlando, FL, USA.

von Bertalanffy, L. 1968. *General System Theory: Foundations, Development, Applications*, revised ed. New York, NY, USA: Braziller.

Weinberg, G., and D. Weinberg. 1988. *General Principles of Systems Design*. New York, NY: Dorset House Publishing Company.

White, B.E. 2007. "On Interpreting Scale (or View) and Emergence in Complex Systems Engineering." Presented at 1st Annual IEEE Systems Conference, 9-12 April, 2007, Honolulu, HI, USA.

Primary References

Giachetti, R.E. 2010. *Design of Enterprise Systems: Theory, Architecture, and Methods*. Boca Raton, FL, USA: CRC Press, Taylor and Francis Group.

Joannou, P. 2007. "Enterprise, Systems, and Software—The Need for Integration." *Computer*. 40 (5): 103-105.

Rouse, W.B. 2009. "Engineering the Enterprise as a System," in *Handbook of systems engineering and management,* 2nd ed., edited by A.P. Sage and W.B. Rouse. New York, NY, USA: Wiley and Sons, Inc.

Rouse, W.B. 2005. "Enterprise as Systems: Essential Challenges and Enterprise Transformation." *Systems Engineering*. 8 (2): 138-50.

Additional References

Arnold, S., and H. Lawson. 2004. "Viewing Systems From a Business Management Perspective." *Systems Engineering*. 7 (3): 229.

Beimans, F.P.M., M.M. Lankhorst, W.B. Teeuw, and R.G. van de Wetering. 2001. "Dealing with the Complexity of Business Systems Architecting." *Systems Engineering*. 4 (2): 118-33.

Nightingale, D., and D. Rhodes. 2004. "Enterprise systems architecting: Emerging art and science within engineering systems." Presented at Engineering Systems Symposium, Massachusetts Institute of Technology (MIT), 29-31 March, 2004, Boston, MA, USA.

Rebovich, G. 2006. "Systems Thinking for the Enterprise: New & Emerging Perspectives." Presented at IEEE/SMC International Conference on System of Systems Engineering, April 2006, Los Angeles, CA, USA.

Rechtin, E. 1999. *Systems Architecting of Organizations: Why Eagles can't Swim*. Boca Raton, FL, USA: CRC Press, Taylor and Francis Group.

Ring, J. 2004. "Intelligent Enterprises." *INCOSE INSIGHT*. 6 (2).

Ring, J. 2004. "Seeing an Enterprise as a System." *INCOSE INSIGHT*. 6(2).

Valerdi, R., D. Nightingale, and C. Blackburn. 2009. "Enterprises as Systems: Context, Boundaries, and Practical Implications." *Information-Knowledge-Systems Management Journal*. 7 (4): 377-399.

< Previous Article | Parent Article | Next Article >

SEBoK v. 2.1, released 31 October 2019

Related Business Activities

Lead Authors: *James Martin, Dick Fairley, Bud Lawson*

The following business management activities can be supported by enterprise systems engineering (ESE) activities:

- mission and strategic planning,
- business processes and information Management,
- performance management,
- portfolio management,
- resource allocation and budgeting, and
- program and project management.

Introduction

The figure below shows how these business activities relate to each other as well as the relative scope of ESE and product systems engineering (PSE) (Martin 2010 and 2011). PSE is mainly involved at the project level and collaborates with project management activities, and gets somewhat involved in program management and the resource allocation and budgeting activities. On the other hand, ESE is heavily involved in the higher level activities from the program management level and up. Both ESE and PSE have key roles to play in the enterprise.

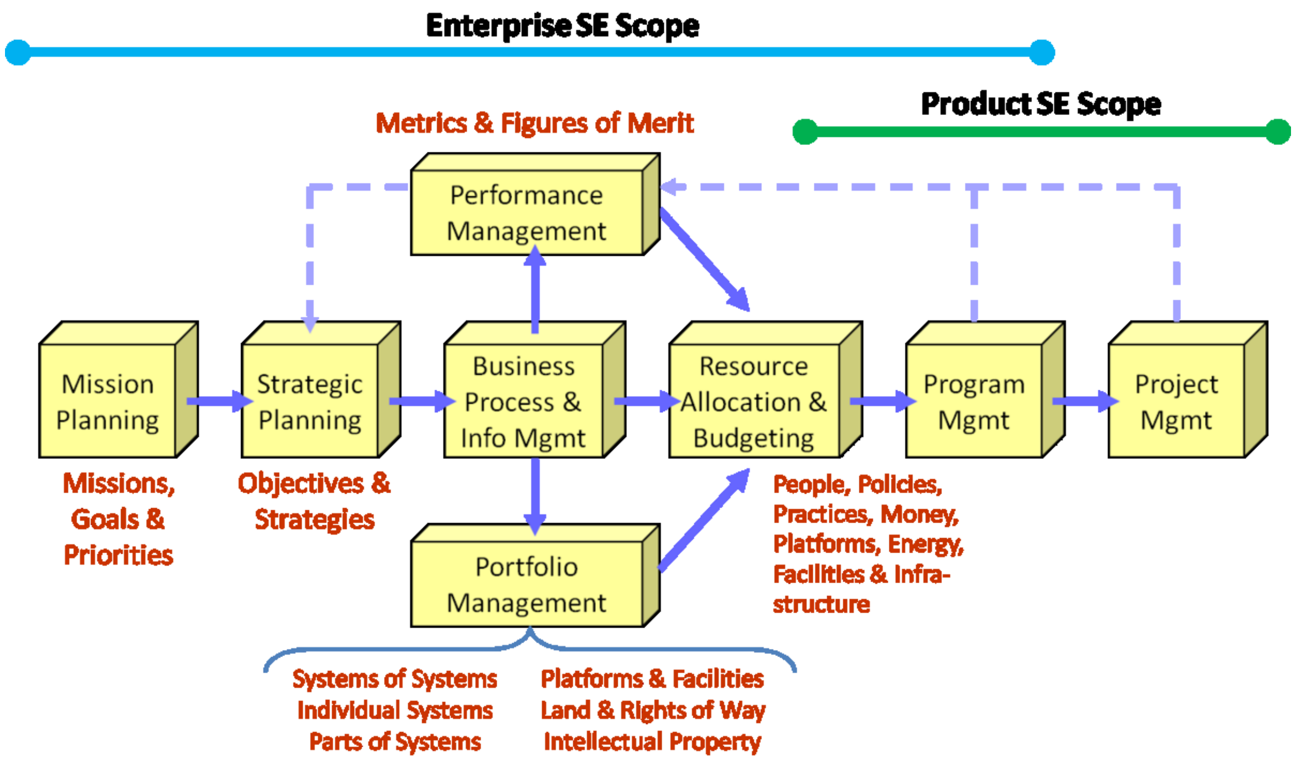


Figure 1. Related Business Activities (Martin 2010). Reprinted with permission of The Aerospace Corporation.

Shown in this manner, these business activities can be considered to be separate processes with a clear precedence in terms of which process drives other processes. TSE uses “requirements” to specify the essential features and functions of a system. An enterprise, on the other hand, typically uses goals and objectives to specify the fundamental characteristics of desired enterprise operational capabilities. The enterprise objectives and strategies are used in portfolio management to discriminate between options and to select the appropriate balanced portfolio of systems and other enterprise resources.

The first three activities listed above are covered in Enabling Businesses and Enterprises. The other business management activities are described in more detail below in regards to how they relate to ESE.

Business Management Cycles

PDCA stands for "plan-do-check-act" and is a commonly used iterative management process as seen in the figure below. It is also known as the Deming circle or the Shewhart cycle after its two key proponents (Deming 1986; Shewhart 1939). ESE should use the PDCA cycle as one of its fundamental tenets. For example, after ESE develops the enterprise transformation plan, execution of the planned improvements are monitored (i.e., “checked” in the PDCA cycle) to ensure they achieve the targeted performance levels. If not, then action needs to be taken (i.e., “act” in the PDCA cycle) to correct the situation and re-planning may be required. ESE can also use the PDCA cycle in its support of the 'business as usual' efforts, such as the annual budgeting and business development planning activities.

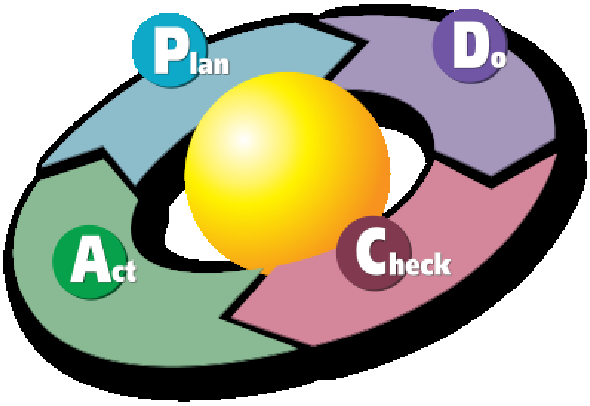


Figure 2. PDCA Cycle. (Source: http://en.wikipedia.org/wiki/PDCA.Accessed July 2010. Diagram by Karn G. Bulsuk (http://blog.bulsuk.com).Used through a CC BY 3.0 license (http://creativecommons.org/licenses/by/3.0/).

It is also worth mentioning the utility of using Boyd's OODA loop (observe, orient, decide, and act) to augment PDCA. This could be accomplished by first using the OODA loop (http://en.wikipedia.org/wiki/OODA_loop), which is continuous in situation awareness, and then followed by using the PDCA approach, which is discrete, having goals, resources, usually time limits, etc. (Lawson 2010).

Portfolio Management

Program and project managers direct their activities as they relate to the systems under their control. Enterprise management, on the other hand, is involved in directing the portfolio of items that are necessary to achieving the enterprise goals and objectives. This can be accomplished by using portfolio management:

> *Project Portfolio Management (PPM) is the centralized management of processes, methods, and technologies used by project managers and project management offices (PMOs) to analyze and collectively manage a group of current or proposed projects based on numerous key characteristics. The objectives of PPM are to determine the optimal resource mix for delivery and to schedule activities to best achieve an organization's operational and financial goals—while honoring constraints imposed by customers, strategic objectives, or external real-world factors.* (http://en.wikipedia.org/wiki/Project_portfolio_management)

The enterprise may not actually own these portfolio items. They could rent or lease these items, or they could have permission to use them through licensing or assignment. The enterprise may only need part of a system (e.g., one bank of switching circuits in a system) or may need an entire system of systems (SoS) (e.g., switching systems, distribution systems, billing systems, provisioning systems, etc.). Notice that the portfolio items are not just those items related to the systems that systems engineering (SE) deals with. These could also include platforms (like ships and oil drilling derricks), facilities (like warehouses and airports), land and rights of way (like railroad property easements and municipal covenants), and intellectual property (like patents and trademarks).

The investment community has been using portfolio management for a long time to manage a set of investments to maximize return for a given level of acceptable risk. These techniques have also been applied to a portfolio of "projects" within the enterprise (Kaplan 2009). However, it should be noted that an enterprise is not merely a portfolio of projects. The enterprise portfolio consists of whatever systems, organizations, facilities, intellectual property, and other resources that are needed to help the enterprise achieve its goals and objectives.

Portfolio management in the context of ESE is well addressed in the following article: http://www.mitre.org/work/systems_engineering/guide/enterprise_engineering/enterprise_planning_management/portfolio_management.html (MITRE 2010).

Resource Allocation and Budgeting

The resource allocation and budgeting (RA&B) activity is driven by the portfolio management definition of the optimal set of portfolio elements. Capability gaps are mapped to the elements of the portfolio, and resources are assigned to programs (or other organizational elements) based on the criticality of these gaps. Resources come in the form of people and facilities, policies and practices, money and energy, and platforms and infrastructure. Allocation of resources could also involve the distribution or assignment of corporate assets, like communication bandwidth, manufacturing floor space, computing power, intellectual property licenses, and so on. Resource allocation and budgeting is typically done on an annual basis, but more agile enterprises will make this a more continuous process. Some of the resource allocation decisions deal with base operational organizations that are not project related.

It is sometimes the case that RA&B is part of portfolio management (PfM). But as can be seen in Figure 1, it is sometimes useful and practical to separate these two activities. PfM usually recommends changes to the enterprise portfolio, but RA&B takes these PfM considerations into mind along with inputs from the business process and information management and the performance management activities. Furthermore, PfM is usually an annual or biannual activity whereas RA&B is often done more frequently. RA&B may need to execute *ad hoc* when perturbations happen, such as funding cuts, schedule slips, performance targets missed, strategic goals changed, and so on.

Program and Project Management

Within the enterprise, TSE is typically applied inside a project to engineer a single system (or perhaps a small number of related systems). If there is a SoS or a large, complex individual system to be engineered, then this might be handled at the program level, but is sometimes handled at the project level, depending on the size and complexity of the system-of-interest (See also Complexity).

There are commonly three basic types of projects in an enterprise. A development project takes a conceptual notion of a system and turns this into a realizable design. A production project takes the realizable design for a system and turns this into physical copies (or instantiations). An operations “project” directly operates each system or supports the operation by others. (Base operations are sometimes called "line organizations" and are not typically called projects per se, but should nonetheless be considered as key elements to be considered when adjusting the enterprise portfolio.) The operations project can also be involved in maintaining the system or supporting maintenance by others. A program can have all three types of projects active simultaneously for the same system, as in this example:

- Project A is developing System X version 3.
- Project B is operating and maintaining System X version 2.
- Project C is maintaining System X version 1 in a warehouse as a backup in case of emergencies.

Project management uses TSE as a tool to ensure a well-structured project and to help identify and mitigate cost, schedule, and technical risks involved with system development and implementation. The project level is where the TSE process is most often employed (Martin 1997; ISO/IEC/IEEE 2015; Wasson 2006; INCOSE 2010; Blanchard and Fabrycky 2010).

The Office of Government Commerce provides a useful distinction between programs and projects:

> *The ultimate goal of a Programme is to realise outcomes and benefits of strategic relevance. To achieve this, a programme is designed as a temporary flexible organisation structure created to coordinate, direct and oversee the implementation of a set of related projects and activities in order to deliver outcomes and benefits related to the organisation’s strategic objectives...*
>
> *A programme is likely to have a life that spans several years. A Project is usually of shorter duration (a few months perhaps) and will be focussed on the creation of a set of deliverables within agreed cost, time and quality parameters.* (OGC 2010)

Enterprise Governance

ESE is also concerned with the way in which organizations and embedded management and technical functions work together to achieve success at the enterprise level. Governance frameworks provide the essential additional structure and controls needed to both ‘steer a steady ship’ (during business as usual) and to ‘plot a course to a new place’ (during business transformation).

Such frameworks can be designed by recognizing that there are enduring management concerns that need to be addressed and by applying the principle of economy. For example, a particular concern for most organizations is linking the control of projects to business drivers and objectives. This leads to a requirement for a governance body to both approve the initiation of projects, and to regularly review their progress, continuing relevance, and if necessary, mutual coherence in the light of developments inside and outside the enterprise.

This might be achieved by delegating some or all of the roles; depending on circumstances, the enterprise might be driven towards top-down or a more collective, peer-to-peer approach—or even a combination of the two for different functions. Governance bodies and management roles can be engineered in this way against a common set of management concerns. Governance may also include the maintenance of common technical standards and their promulgation and use throughout relevant projects. See Bryant (2012) for more information on governance.

Multi-Level Enterprises

An enterprise does not always have full control over the ESE processes. In some cases, an enterprise may have no direct control over the resources necessary to make programs and projects successful. For example, the Internet Engineering Task Force (IETF) is responsible for the “smooth operation of the Internet,” yet it controls none of the requisite resources.

> *The Internet Engineering Task Force (IETF) is a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet. … The actual technical work of the IETF is done in its working groups, which are organized by topic into several areas (e.g., routing, transport, security, etc.). Much of the work is handled via mailing lists. The IETF holds meetings three times per year.* (IETF 2010a)

The IETF has “influence” over these resources even though it does not have direct control: "The IETF is unusual in that it exists as a collection of happenings, but is not a corporation and has no board of directors, no members, and no dues" (IETF 2010b).

The ESE processes might be allocated between a “parent” enterprise and “children” enterprises, as shown in the figure below (Martin 2010). The parent enterprise, in this case, has no resources. These resources are owned by the subordinate child enterprises. Therefore, the parent enterprise does not implement the processes of resource allocation and budgeting, program management, and project management.

The parent enterprise may have an explicit contract with the subordinate enterprises, or, as in some cases, there is merely a “working relationship” without the benefit of legal obligations. The parent enterprise will expect performance feedback from the lower level to ensure that it can meet its own objectives. Where the feedback indicates a deviation from the plan, the objectives can be adjusted or the portfolio is modified to compensate.

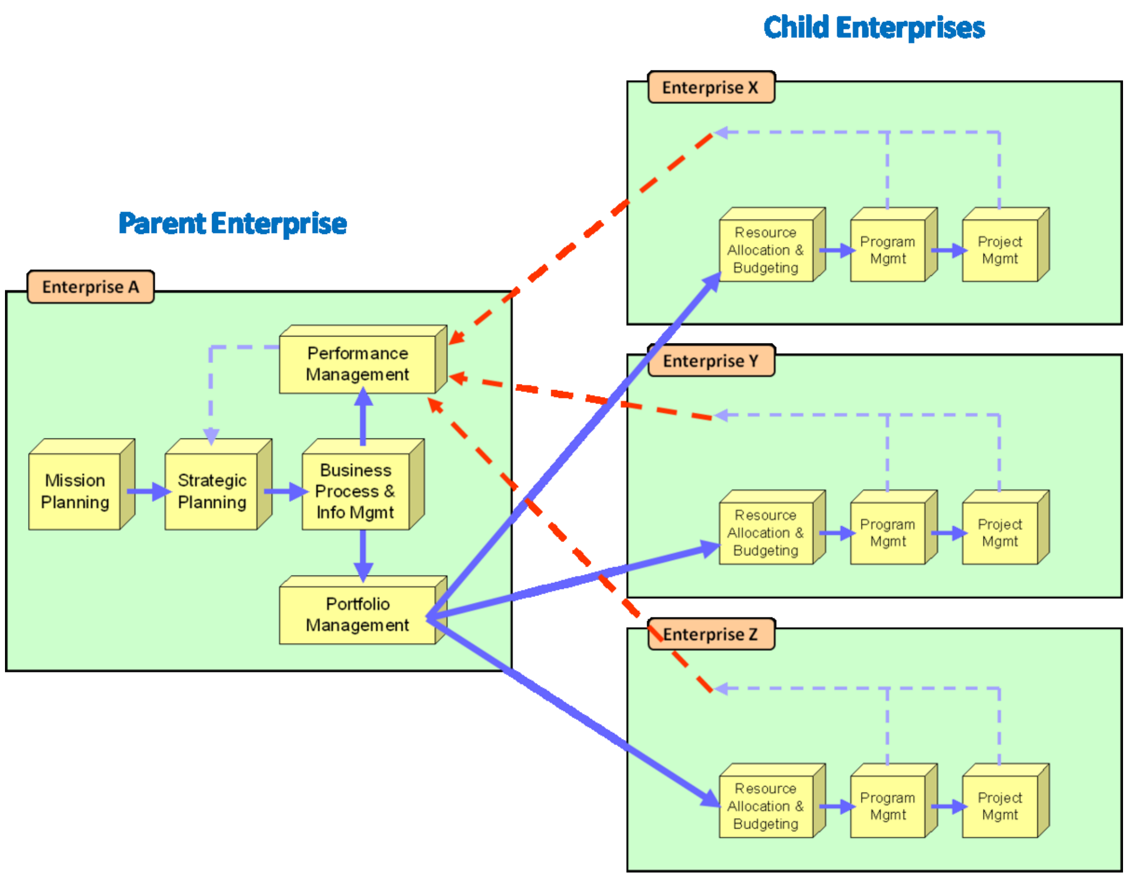


Figure 3. Parent and Child Enterprise Relationships (Martin 2010). Reprinted with permission of The Aerospace Corporation.

Enterprises X, Y, and Z in the situation shown above will cooperate with each other to the extent that they honor the direction and guidance from the parent enterprise. These enterprises may not even be aware of each other, and, in this case, would be unwittingly cooperating with each other. The situation becomes more complex if each enterprise has its own set of strategic goals and objectives as shown in the figure below.

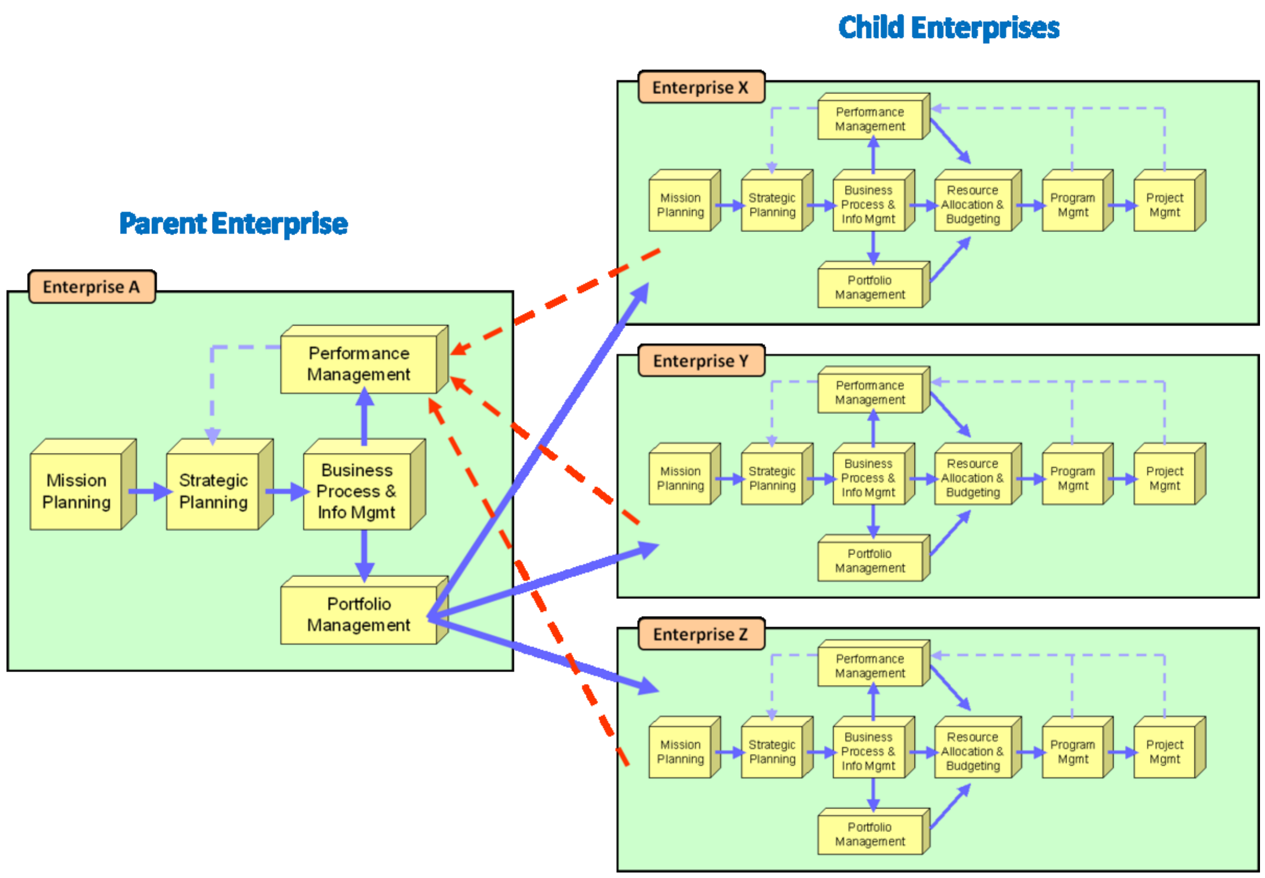


Figure 4. Mission and Strategic Planning at All Levels of Cooperating Enterprises (Martin 2010). Reprinted with permission of The Aerospace Corporation. All other rights are reserved by the copyright owner.

These separate, sub-enterprise objectives will sometimes conflict with the objectives of the parent enterprise. Furthermore, each subordinate enterprise has its own strategic objectives that might conflict with those of its siblings. The situation shown here is not uncommon, and illustrates an enterprise of enterprises, so to speak. This highlights the need for the application of SE at the enterprise level to handle the complex interactions and understand the overall behavior of the enterprise as a whole. TSE practices can be used, to a certain extent, but these need to be expanded to incorporate additional tools and techniques.

References

Works Cited

Bryant, P. 2012. "Modelling Governance within Business Architecture using Topic Mapping." Presented at 22nd Annual International Council on Systems Engineering (INCOSE) International Symposium, July 9-12, 2012, Rome, Italy.

Blanchard, B.S., and W.J. Fabrycky. 2011. *Systems Engineering and Analysis*, 5th ed. Englewood Cliffs, NJ, USA: Prentice-Hall.

Deming, W.E. 1986. *Out of the Crisis*. Cambridge, MA, USA: MIT Press, MIT Center for Advance Engineering Study.

IETF. 2010a. "Overview of the IETF," in Internet Engineering Task Force, Internet Society (ISOC) [database online]. Accessed September 6, 2011. Available: http://www.ietf.org/overview.html.

IETF. 2010b. "The Tao of IETF: A Novice's Guide to the Internet Engineering Task Force (draft-hoffman-tao4677bix-10)," in Internet Engineering Task Force, Internet Society (ISOC) [database online].

Accessed September 6, 2011. Available: http://www.ietf.org/tao.html#intro.

INCOSE. 2012. *Systems Engineering Handbook*, version 3.2.2. San Diego, CA, USA: International Council on Systems Engineering (INCOSE). INCOSE-TP-2003-002-03.2.2.

ISO/IEC/IEEE 2015. *Systems and Software Engineering -- System Life Cycle Processes.* Geneva, Switzerland: International Organisation for Standardisation / International Electrotechnical Commissions. ISO/IEC/IEEE 15288:2015.

Kaplan, J. 2009. *Strategic IT portfolio management: Governing enterprise transformation*. Waltham, Massachusetts, USA: Pittiglio, Rabin, Todd & McGrath, Inc. (PRTM).

Lawson, H. 2010. *A Journey Through the Systems Landscape*. Kings College, UK: College Publications.

Martin, J.N. 2011. "Transforming the Enterprise Using a Systems Approach." Presented at 21st Anniversary International Council on Systems Engineering (INCOSE) International Symposium, June 20-23, 2011, Denver, CO, USA.

Martin, J.N. 2010. "An Enterprise Systems Engineering Framework." Presented at 20th Anniversary International Council on Systems Engineering (INCOSE) International Symposium, July 12-15, 2010, Chicago, IL, USA.

Martin, J.N. 1997. *Systems Engineering Guidebook: A Process for Developing Systems and Products*, 1st ed. Boca Raton, FL, USA: CRC Press.

MITRE. 2012. "Enterprise Engineering," in *Systems Engineering Guide*. Bedford, MA, USA: MITRE Corporation. Accessed July 8, 2012. Available: http:/ / www. mitre. org/ work/ systems_engineering/ guide/ enterprise_engineering/.

OGC (Office of Government Commerce). 2010. *Guidelines for Managing Programmes: Understanding programmes and programme management*. London, UK: The Stationery Office.

Shewhart, W.A. 1939. *Statistical Method from the Viewpoint of Quality Control*. New York, NY, USA: Dover Publications.

Wasson, C.S. 2006. *System Analysis, Design and Development*. Hoboken, NJ, USA: John Wiley and Sons Ltd.

Primary References

Martin, J.N. 2011. "Transforming the Enterprise Using a Systems Approach." Presented at 21st Anniversary International Council on Systems Engineering (INCOSE) International Symposium, June 20-23, 2011, Denver, CO, USA.

Martin, J.N. 2010. "An Enterprise Systems Engineering Framework." Presented at 20th Anniversary International Council on Systems Engineering (INCOSE) International Symposium, July 12-15, 2010, Chicago, IL, USA.

Additional References

Arnold, S., and H. Lawson. 2004. "Viewing Systems from a Business Management Perspective." *Systems Engineering*. 7 (3): 229.

Beimans, F.P.M., M.M. Lankhorst, W.B. Teeuw, and R.G. van de Wetering. 2001. "Dealing with the Complexity of Business Systems Architecting." *Systems Engineering*. 4 (2): 118-133.

Drucker, P.F. 1994. "The Theory of Business." *Harvard Business Review*. 72 (5): 95-104.

Haeckel, S.H. 2003. "Leading on demand businesses–Executives as architects." *IBM Systems Journal*. 42 (3): 405-13.

Kaplan, R., and D. Norton. 1996. *The balanced scorecard: Translating strategy into action*. Cambridge, MA, USA: Harvard Business School Press.

Lissack, M.R. 2000. "Complexity Metaphors and the Management of a Knowledge Based Enterprise: An Exploration of Discovery." PhD Dissertation in Business Administration. Henley-on-Thames, UK: Henley Management College, University of Reading.

Rechtin, E. 1999. *Systems Architecting of Organizations: Why Eagles Can't Swim*. Boca Raton, FL, USA: CRC Press, Taylor and Francis Group.

< Previous Article | Parent Article | Next Article >

SEBoK v. 2.1, released 31 October 2019

Enterprise Systems Engineering Key Concepts

Lead Authors: *James Martin, Bud Lawson, Alan Faisandier*

The purpose of traditional systems engineering (TSE) is to bring together a diversity of discipline experts to address a wide range of problems inherent in the development of a large, complex “single” system (Blanchard and Fabrycky 2010; Hall 1989; Sage and Rouse 2009). Enterprise systems engineering (ESE) expands beyond this traditional basis to “consider the full range of SE services increasingly needed in a modern organization where information-intensive systems are becoming central elements of the organization’s business strategy” (Carlock and Fenton 2001, 242-261). The traditional role of systems engineering (SE) is heavily involved in system acquisition and implementation, especially in the context of government acquisition of very large, complex military and civil systems (e.g., F22 fighter jet and air traffic control systems).

ESE encompasses this traditional role in system acquisition, but also incorporates enterprise strategic planning and enterprise investment analysis (along with others as described below). These two additional roles for SE at the enterprise level are “shared with the organization’s senior line management, and tend to be more entrepreneurial, business-driven, and economic in nature in comparison to the more technical nature of classical systems engineering” (Carlock and Fenton 2001, 242-261).

Closing the Gap

ESE practices have undergone significant development recently.

> *Today the watchword is enterprise systems engineering, reflecting a growing recognition that an 'enterprise' may comprise many organizations from different parts of government, from the private and public sectors, and, in some cases, from other nations.* (MITRE 2004)

Rebovich (2006) says there are “new and emerging modes of thought that are increasingly being recognized as essential to successful systems engineering in enterprises.” For example, in addition to the TSE process areas, MITRE has included the following process areas in their ESE process (DeRosa 2005) to close the gap between ESE and PSE:

- strategic technical planning,
- enterprise architecture,
- capabilities-based planning analysis,
- technology planning, and
- enterprise analysis and assessment.

These ESE processes are shown in the context of the entire enterprise in the figure below (DeRosa 2006). The ESE processes are shown in the middle with business processes on the left and TSE processes on the right. These business processes are described in the article called Related Business Activities. The TSE processes are well documented in many sources, especially in the ISO/IEC/IEEE 15288 standard (2015).

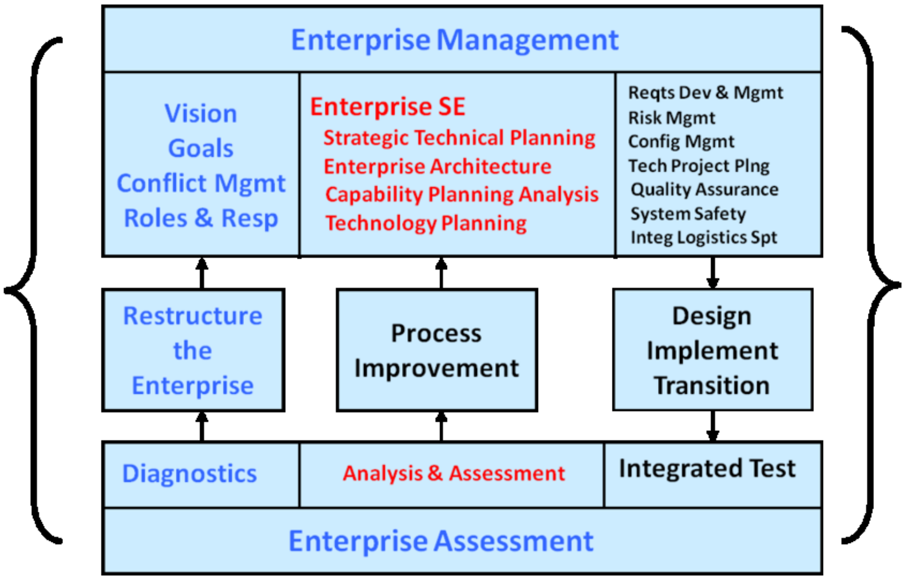


Figure 1. Enterprise SE Process Areas in the Context of the Entire Enterprise (DeRosa 2006). Reprinted with permission of © 2011. The MITRE Corporation. All Rights Reserved. All other rights are reserved by the copyright owner.

SE is viewed by many organizations and depicted in many process definitions as bounded by the beginning and end of a system development project. In MITRE, this restricted definition was referred to as TSE. Many have taken a wider view seeking to apply SE to the "whole system" and "whole life cycle." For example, Hitchins (1993) sets out a holistic, whole-life, wider system view of SE centered on operational purpose. Elliott and Deasley (2007) discuss the differences between development phase SE and in-service SE.

In contrast to TSE, the ESE discipline is more like a "regimen" (Kuras and White 2005) that is responsible for identifying "outcome spaces," shaping the development environment, coupling development to operations, and rewarding results rather than perceived promises (DeRosa 2005). ESE must continually characterize the operational environmental and the results of enterprise or SoS interventions to stimulate further actions within and among various systems in the enterprise portfolio. Outcome spaces are characterized by a set of desired capabilities that help meet enterprise objectives, as opposed to definitive "user requirements" based on near-term needs. Enterprise capabilities must be robust enough to handle unknown threats and situations in the future. A detailed description of previous MITRE views on ESE can be found in a work by Rebovich and White (2011).

Role of Requirements in ESE

TSE typically translates user needs into system requirements that drive the design of the system elements. The system requirements must be "frozen" long enough for the system components to be designed, developed, tested, built, and delivered to the end users (which can sometimes take years, and in the case of very large, complicated systems like spacecraft and fighter jets, more than a decade).

ESE, on the other hand, must account for the fact that the enterprise must be driven not by requirements (that rarely can even be defined, let alone made stable), but instead by continually changing organizational visions, goals, governance priorities, evolving technologies, and user expectations. An enterprise consists of people, processes, and technology where the people act as "agents" of the enterprise:

> *Ackoff has characterized an enterprise as a 'purposeful system' composed of agents who choose both their goals and the means for accomplishing those goals. The variety of people, organizations, and their strategies is what creates the inherent complexity and non-determinism in an enterprise. ESE must account for the concerns, interests and objectives of these agents.* (Swarz, DeRosa, and Rebovich 2006)
> (See also Complexity)

Whereas TSE focuses on output-based methodologies (e.g., functional analysis and object-oriented analysis), ESE is obligated to emphasize outcomes (e.g., business analysis and mission needs analysis), especially those related to the enterprise goals and key mission needs.

Enterprise Entities and Relationships

An enterprise “system” has different entities and relationships than you might find in a product/service system (see note 1). These can be usefully grouped into two categories: asset items and conceptual items. An example of an asset is hardware and software. Examples of conceptual items are things like analysis, financial elements, markets, policies, process, and strategy.

Note 1. An “enterprise system” should not be confused with the enterprise “perceived as a system.” An enterprise system is a product (or service) system used across the enterprise, such as payroll, financial accounting, or enterprise resource planning applications, and consolidated data center, data warehouse, and other such facilities and equipment used across one or more organizations.

Products and services are sometimes treated as “assets” as shown in the figure below (Troux 2010). This categorization of enterprise items comes from the semantic model (i.e., metamodel) used in the Troux Architect modeling tool for characterization and analysis of an enterprise architecture. Other enterprise entities of interest are things like information, knowledge, skills, finances, policies, process, strategy, markets, and resources, but these are categorized as "concept" items (in this particular schema). Further details on how to use this metamodel's entities and relationships are provided by Reese (2010).

Table 1. Asset Domain and Concept Domain Categories for Enterprise Entities. (Troux 2010) Reprinted with permission of Copyright © 2010 Troux Technologies. All other rights are reserved by the copyright owner.

Asset Domains	Concept Domains
Application and Software Domain Data Domain Document Domain Infrastructure and Hardware Domain IT Product Domain IT Service Domain Location Domain Organization Domain Product and Service Domain Services Portfolio Management Domain	Analysis Domain Financial Domain General Domain Information Domain IT Architecture Domain Knowledge and Skill Domain Market Domain Policy Domain Process Domain Resource Domain Strategy Domain Timeline Domain Transition Domain

The application/software and infrastructure/hardware domains are likely the most familiar to systems engineers (as illustrated in the figure below). The application/software domain contains things like the deployed software itself, plus applications, modules, servers, patches, functions, and messages. The infrastructure/hardware domain contains things like the hardware itself, plus networks and different kinds of hardware like computing hardware, cabinets, and network devices. There might be different subtypes of computing hardware like computers, servers, desktops, laptops, and mainframes. You can see from this elaboration of these domains that an enterprise architecture "schema" can be quite extensive in the kinds of things it can model.

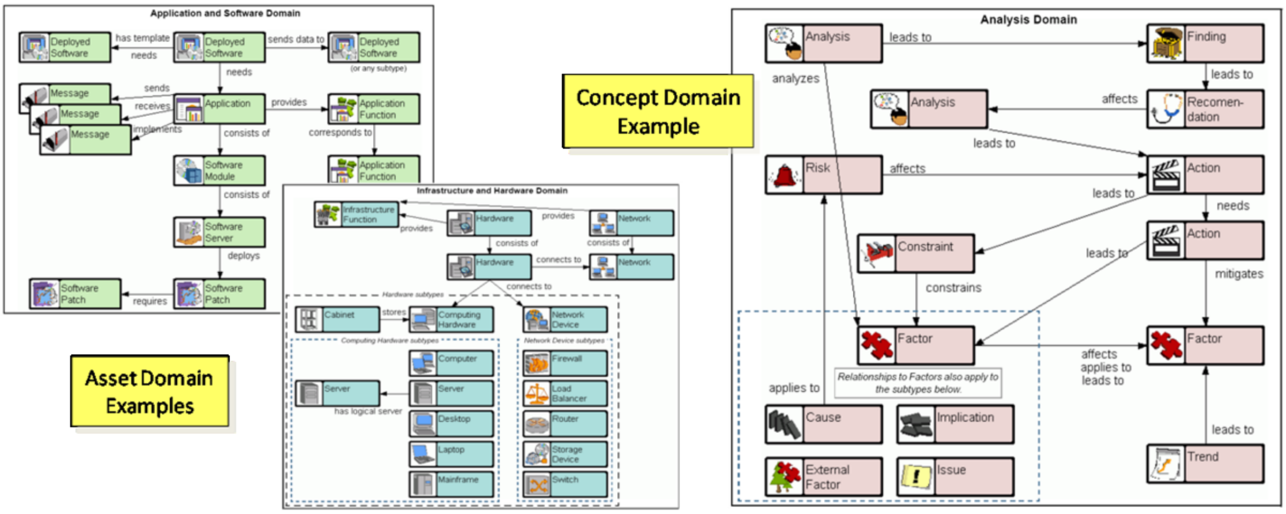


Figure 2. Example of Enterprise Entities & Relationships (Troux 2010). Reprinted with permission of Copyright © 2010 Troux Technologies. All other rights are reserved by the copyright owner.

The less technical domains would be things like policy, market, strategy, transition, financial, knowledge and skill, and analysis. In a typical enterprise architecture schema like this, there could be over a hundred types of modeling objects grouped into these domains. The examples give above are from the Troux Semantics metamodel used in the Troux Architect modeling tool for enterprise architecture activities. Other enterprise modeling tools have similar metamodels (sometimes called “schemas”). See Reese (2010) for more details on how to use the metamodel shown in the figure above.

Enterprise Architecture Frameworks & Methodologies

Enterprise architecture frameworks are collections of standardized viewpoints, views, and models that can be used when developing architectural descriptions of the enterprise. These architecture descriptions can be informal, based on simple graphics and tables, or formal, based on more rigorous modeling tools and methods. ISO/IEC 42010 (2011) specifies how to create architecture descriptions.

These frameworks relate to descriptive models of an enterprise, with conventions agreed in particular communities. There are various frameworks and methodologies available that assist in the development of an enterprise architecture.

Urbaczewski and Mrdalj (2006) provide an overview and comparison of five prominent architectural frameworks, including:

- the Zachman Framework for Enterprise Architecture (Zachman 1992),
- the Department of Defense Architecture Framework (DoDAF) (DoD 2010),
- the Federal Enterprise Architecture Framework (FEAF) (FEA 2001),
- the Treasury Enterprise Architecture Framework (TEAF) (US Treasury 2000),
- and The Open Group Architectural Framework (TOGAF) (TOGAF 2009).

References

Works Cited

Blanchard, B.S., and W.J. Fabrycky. 2011. *Systems Engineering and Analysis*, 5th ed. Englewood Cliffs, NJ, USA: Prentice-Hall.

Carlock, P., and R. Fenton. 2001. "System of Systems (SoS) Enterprise Systems Engineering for Information-Intensive Organizations." *Systems Engineering*. 4 (4): 242-261.

CIO Council. 1999. *Federal Enterprise Architecture Framework (FEAF)*, version 1.1. Washington, DC, USA: Federal Chief Information Officers Council.

DeRosa, J.K. 2005. "Enterprise Systems Engineering." Presented at Air Force Association, Industry Day, Day 1, August 4, 2005, Danvers, MA, USA.

DoD. 2010. *DoD Architecture Framework (DoDAF)*, version 2.0. Washington, DC: U.S. Department of Defense (DoD).

Elliott, C., and P. Deasley. 2007. *Creating Systems that Work--Principles of Engineering Systems for the 21st Century*. London, England, UK: Royal Academy of Engineering.

FEA. 2001. "Federal Enterprise Architecture – Practical Guide, version 1.0, February 2001." Available: https:// secure.cio.noaa.gov/hpcc/docita/files/a_practical_guide_to_federal_enterprise_architecture.pdf.

Friedman, G., and A.P. Sage. 2004. "Case Studies of Systems Engineering and Management in Systems Acquisition." *Systems Engineering*. 7 (1): 84-96.

Hall, A.D. 1989. *Metasystems Methodology: A New Synthesis and Unification*, 1st ed. Oxford, UK: Pergamon Press.

Hitchins, D. 1993. *Putting Systems to Work*. New York, NY, USA: John Wiley & Sons.

ISO/IEC/IEEE. 2015.*Systems and Software Engineering - System Life Cycle Processes.*Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC), Institute of Electrical and Electronics Engineers. ISO/IEC/IEEE 15288:2015.

Kuras, M.L., and B.E. White. 2005. "Engineering Enterprises Using Complex-Systems Engineering." Annotated presentation at 15th Annual International Council on Systems Engineering (INCOSE) International Symposium, July 10-15, 2005, Rochester, NY, USA.

MITRE. 2004. *MITRE 2004 Annual Report". McLean, VA, USA: MITRE Corporation.*

Rebovich, G. 2006. "Systems Thinking for the Enterprise: New & Emerging Perspectives." Presented at IEEE/SMC International Conference on System of Systems Engineering, April 2006, Los Angeles, CA, USA.

Rebovich, G., and B.E. White (eds.). 2011. *Enterprise Systems Engineering: Advances in the Theory and Practice*. Boca Raton, FL, USA: CRC Press, Taylor and Francis Group.

Reese, R.J. 2010. *Troux Enterprise Architecture Solutions*. Birmingham, UK: Packt Publishing Ltd.

Sage, A.P., and W.B. Rouse (eds). 2009. *Handbook of System Engineering and Management*, 2nd ed. New York, NY, USA: John Wiley & Sons.

Swarz, R.S., J.K. DeRosa, and G. Rebovich. 2006. "An Enterprise Systems Engineering Model." Proceedings of the 16th Annual International Council on Systems Engineering (INCOSE) International Symposium, July 9-13, 2006, Orlando, FL, USA.

TOGAF. 2009. "The Open Group Architecture Framework," version 9. Accessed September 7, 2011. Available: http://www.opengroup.org/togaf/.

Troux. 2010. *Metamodeling and modeling with Troux Semantics*, version 9. Austin, TX, USA: Troux Technologies.

Urbaczewski, L., and S. Mrdalj. 2006. "A Comparison of Enterprise Architecture Frameworks." *Issues in Information Systems*. 7 (2): 18-26.

US Treasury. 2000. *Treasury Enterprise Architecture Framework,* version 1. Washington, DC, USA: US Department of the Treasury Chief Information Officer Council.

Zachman, J.A. 1992. "Extending and Formalizing the Framework for Information Systems Architecture." *IBM Systems Journal*. 31 (3): 590-616.

Zachman, J.A. 1987. "A Framework for Information Systems Architectures." *IBM Systems Journal*. 26 (3): 276-92.

Primary References

Kuras, M.L., and B.E. White. 2005. "Engineering Enterprises Using Complex-Systems Engineering." Annotated presentation at 15th Annual International Council on Systems Engineering (INCOSE) International Symposium, July 10-15, 2005, Rochester, NY, USA.

Rebovich, G., and B.E. White (eds.). 2011. *Enterprise Systems Engineering: Advances in the Theory and Practice*. Boca Raton, FL, USA: CRC Press, Taylor and Francis Group.

Swarz, R.S., J.K. DeRosa, and G. Rebovich. 2006. "An Enterprise Systems Engineering Model." Proceedings of the 16th Annual International Council on Systems Engineering (INCOSE) International Symposium, July 9-13, 2006, Orlando, FL, USA.

Additional References

Journal of Enterprise Architecture. Available: http://www.globalaea.org/?page=JEAOverview.

Minoli, D. 2008. *Enterprise Architecture A to Z: Frameworks, Business Process Modeling, SOA, and Infrastructure Technology*. Boca Raton, FL, USA: CRC Press, Taylor and Francis Group, An Auerbach Book.

TRAK. 2011. "TRAK Enterprise Architecture Framework." Accessed September 7, 2011. Available: http://trak.sourceforge.net/index.html.

Vernadat, F.B. 1996. *Enterprise Modelling and Integration - Principles and Applications*. London, UK: Chapman and Hall.

< Previous Article | Parent Article | Next Article >

SEBoK v. 2.1, released 31 October 2019

Enterprise Systems Engineering Process Activities

Lead Authors: *James Martin, Bud Lawson, Alan Faisandier*

The application of the key concepts of Enterprise Systems Engineering requires processes. These processes span and can transform the enterprise.

Systems Engineering Role in Transforming the Enterprise

Enabling Systematic Enterprise Change

The systems engineering (SE) process as applied to the enterprise as a whole could be used as the "means for producing change in the enterprise … [where the] … Seven Levels of change in an organization [are defined] as effectiveness, efficiency, improving, cutting, copying, differentiating and achieving the impossible" (McCaughin and DeRosa 2006). The essential nature of enterprise systems engineering (ESE) is that it "determines the balance between complexity and order and in turn the balance between effectiveness and efficiency. When viewed as the fundamental mechanism for change, it goes beyond efficiency and drives adaptation of the enterprise" (McCaughin and DeRosa 2006). McCaughin and DeRosa (2006) provide a reasonably good definition for an enterprise that captures this notion of balance:

> *Enterprise: People, processes and technology interacting with other people, processes and technology, serving some combination of their own objectives, those of their individual organizations and those of the enterprise as a whole.*

Balancing Effectiveness versus Efficiency

Ackoff tells us that:

> *Data, information, knowledge and understanding enable us to increase efficiency, not effectiveness. The value of the objective pursued is not relevant in determining efficiency, but it is relevant in determining effectiveness. Effectiveness is evaluated efficiency. It is efficiency multiplied by value.* ***Intelligence*** *is the* ***ability to increase efficiency****;* ***wisdom*** *is the* ***ability to increase effectiveness****.*
>
> *The difference between efficiency and effectiveness is reflected in the difference between development and growth. Growth does not require an increase in value; development does. Therefore, development requires an increase in* ***wisdom as well as understanding, knowledge and information****.* ((Ackoff 1989, 3-9), emphasis added)

ESE has a key role to play in establishing the right balance between effectiveness and efficiency in enterprise operations and management. Value stream analysis is one technique, among others, that can help ESE determine where inefficiencies exist or ineffective results are being achieved.

Value Stream Analysis

Value stream analysis is one way of treating the enterprise as a system. It provides insights regarding where in the sequence of enterprise activities value is added as it moves towards the final delivery to customer or user (Rother and Shook 1999). It relates each step to the costs entailed in that step in terms of resource consumption (i.e., money, time, energy, and materials). In addition to direct costs, there may also be indirect costs due to overhead factors or infrastructure elements. This activity commonly involves drawing a flowchart of the value stream for the enterprise as illustrated in the figure below.

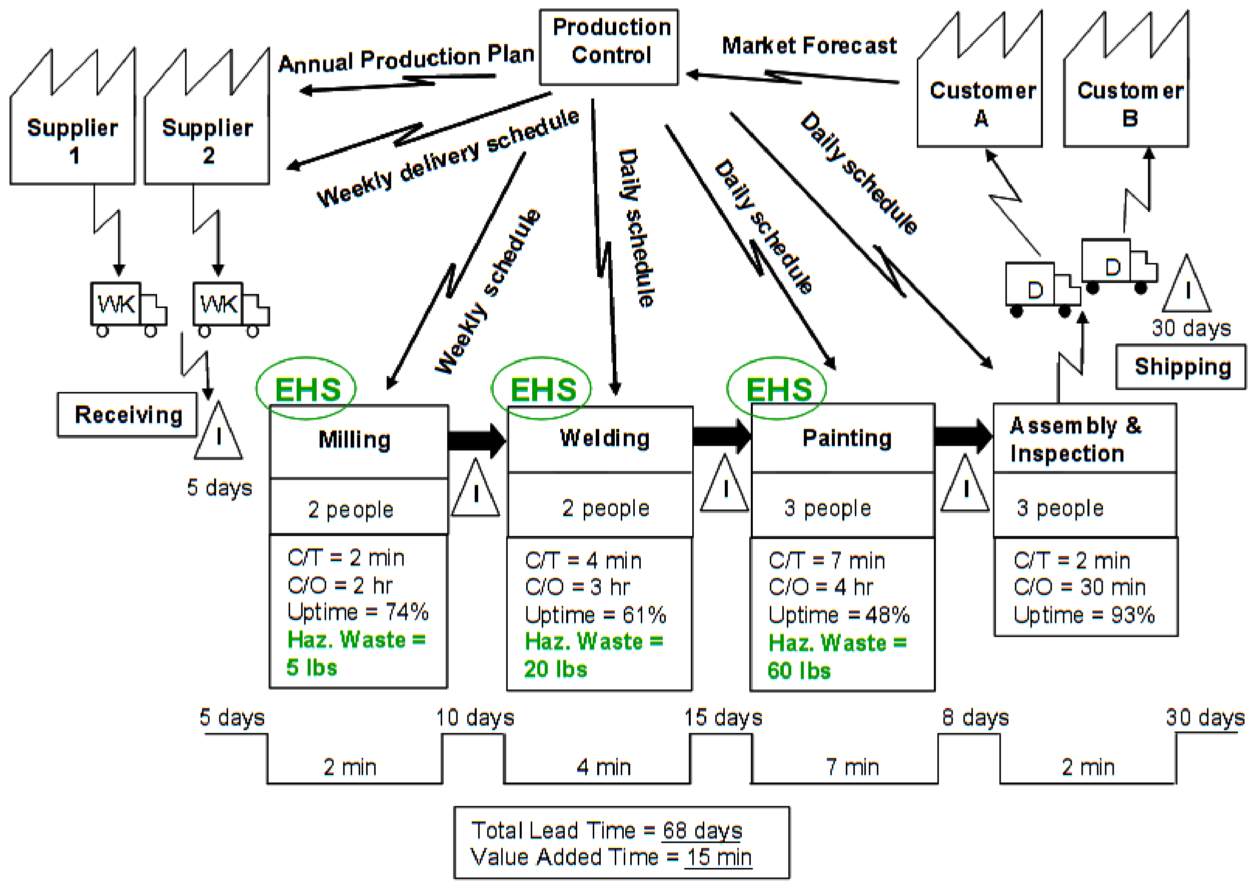


Figure 1. Value Stream Example. (Source: http://en.wikipedia.org/wiki/Value_stream_mapping Accessed September 6, 2010. US EPA Lean and Environment Toolkit, Public Domain.)

Analysis of this value stream diagram can highlight unnecessary space, excessive distance traveled, processing inefficiencies, and so on. Value stream mapping is associated with so-called "lean enterprise" initiatives. At Toyota, where the technique originated, it is known as "material and information mapping" (Rother 2009). Various value stream mapping tools are available (Hines and Rich 1997).

Enterprise Management Process Areas

Martin (2010) has determined that the following four processes are needed in ESE beyond the traditional SE processes in support of enterprise management activities:

1. Strategic technical planning,
2. Capability-based planning analysis,
3. Technology and standards planning, and
4. Enterprise evaluation and assessment.

The interactions between these four processes are illustrated below, along with their interactions with other processes that deal with architecture, requirements, risk, and opportunity.

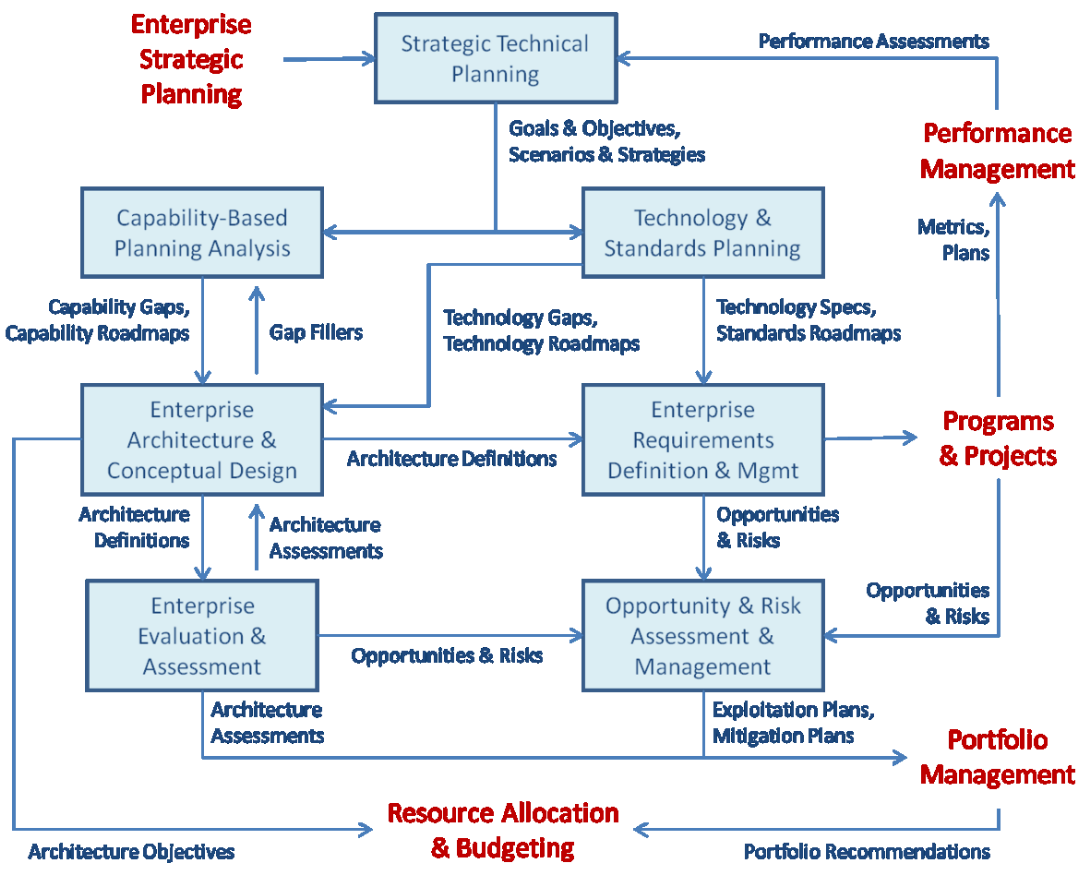


Figure 2. Enterprise Systems Engineering Process Activities. (SEBoK Original)

Strategic Technical Planning

The purpose of strategic technical planning (STP) is to establish the overall technical strategy for the enterprise. It creates the balance between the adoption of standards (see also Systems Engineering Standards) and the use of new technologies, along with consideration of the people aspects driven by the relevant trans-disciplinary technical principles and practices from psychology, sociology, organizational change management, etc.

This process uses the roadmaps developed during technology and standards planning (TSP). It then maps these technologies and standards against the capabilities roadmap to determine potential alignment and synergy. Furthermore, lack of alignment and synergy is identified as a risk to avoid or an opportunity to pursue in the technical strategy. The technical strategy is defined in terms of implementation guidance for the programs and projects.

One reason that STP and TSP are separate processes is that they are often done by different groups in the enterprise and they involve different skill sets. TSP is often done by the technology and science groups. TSP is done closer to (if not in) the chief architect and budget planning groups. Sometimes the great technology proposed by TSP just doesn’t line up with the capabilities needed in the requisite time frame. STP does this balancing between technology push and capability pull.

Capability-Based Planning Analysis

The purpose of *Capability-based Planning Analysis* is to translate the enterprise vision and goals into a set of current and future capabilities that helps achieve those goals. Current missions are analyzed to determine their suitability in supporting the enterprise goals. Potential future missions are examined to determine how they can help achieve the vision. Current and projected capabilities are assessed to identify capability gaps that prevent the vision and technical strategy from being achieved. These capability gaps are then used to assess program, project, and system opportunities that should be pursued by the enterprise. This is defined in terms of success criteria of what the enterprise is desired to achieve.

There are different types of capabilities, as shown in the figure below. It is common practice to describe capabilities in the form of capability hierarchies and capability roadmaps. Technology roadmaps (discussed below under Technology Planning) are usually related to the system capabilities while business capability roadmaps (BCRMs) are related to the operational capabilities of the enterprise as a whole (ref: Business-Capability Mapping: Staying Ahead of the Joneses, http://msdn.microsoft.com/en-us/library/bb402954.aspx). The BCRM development is usually done as part of enterprise strategic planning, which is one level higher than, and a key driver for, the strategic technical planning activity described above.

In some domains there may be competency roadmaps dealing with the organizational capabilities, with perhaps the desired competency levels of individuals mapped out in terms of the jobs or roles used in the enterprise or perhaps in terms of the knowledge and skills required for certain activities. (For more information on systems engineering competency, see the Enabling Individuals article.)

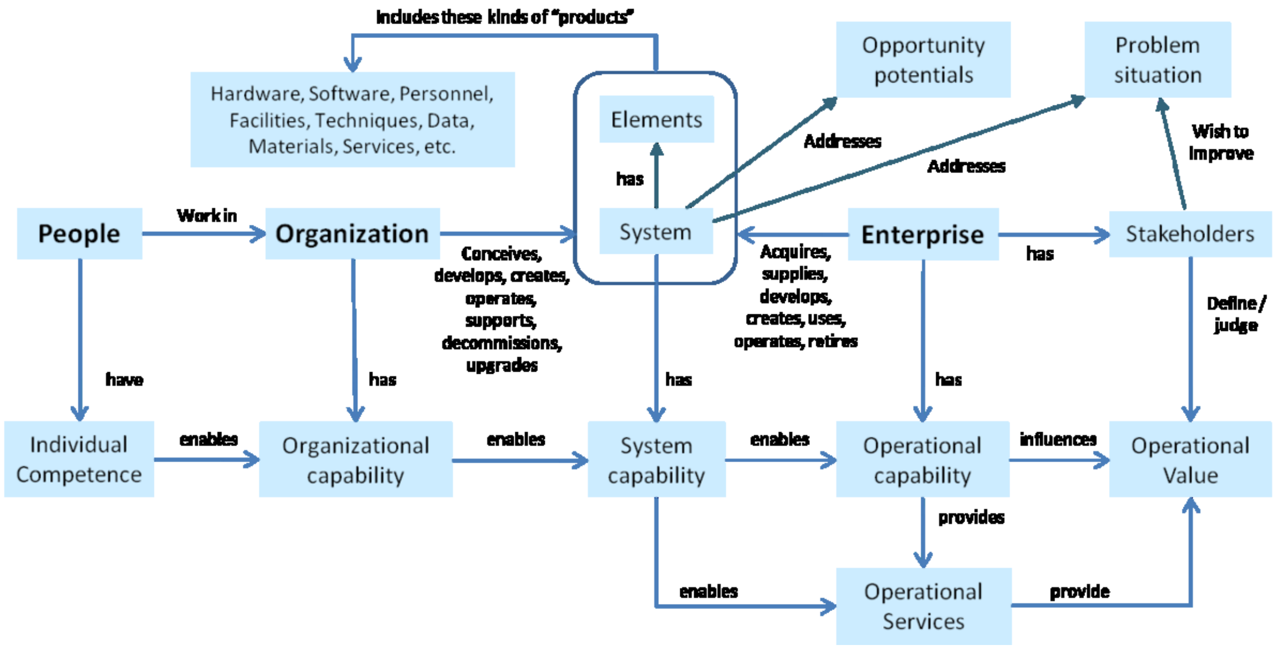


Figure 3. Organizational, System & Operational Capabilities. (SEBoK Original)

Technology and Standards Planning

The purpose of *Technology Planning* is to characterize technology trends in the commercial marketplace and the research community. This activity covers not just trend identification and analysis, but also technology development and transition of technology into programs and projects. It identifies current, and predicts future, technology readiness levels for the key technologies of interest. Using this information, it defines technology roadmaps. This activity helps establish the technical strategy and implementation guidance in the strategic technical plan. The business capabilities roadmap (BCRM) from the strategic planning activity is used to identify which technologies can contribute to achieved targeted levels of performance improvements.

The purpose of *Standards Planning* is to assess technical standards to determine how they inhibit or enhance the incorporation of new technologies into systems development projects. The future of key standards is forecast to determine where they are headed and the alignment of these new standards with the life cycles for the systems in the enterprise's current and projected future portfolios. The needs for new or updated standards are defined and resources are identified that can address these needs. Standardization activities that can support development of new or updated standards are identified (See also Systems Engineering Standards).

Enterprise Evaluation and Assessment

The purpose of enterprise evaluation and assessment (EE&A) is to determine if the enterprise is heading in the right direction. It does this by measuring progress towards realizing the enterprise vision. This process helps to "shape the environment" and to select among the program, project, and system opportunities. This is the primary means by which the technical dimensions of the enterprise are integrated into the business decisions.

This process establishes a measurement program as the means for collecting data for use in the evaluation and assessment of the enterprise. These measures help determine whether the strategy and its implementation are working as intended. Measures are projected into the future as the basis for determining discrepancies between what is observed and what had been predicted to occur. This process helps to identify risks and opportunities, diagnose problems, and prescribe appropriate actions. Sensitivity analysis is performed to determine the degree of robustness and agility of the enterprise.

Roberts states that EE&A must go beyond traditional system evaluation and assessment practices (Roberts 2006). He says that this process area:

> *must de-emphasize the utility of comparing detailed metrics against specific individual requirement values, whether the metrics are derived from measurement, simulation or estimation… [it] must instead look for break points where capabilities are either significantly enhanced or totally disabled.*

Key characteristics of this activity are the following:

- Multi-scale analysis,
- Early and continuous operational involvement,
- Lightweight command and control (C2) capability representations,
- Developmental versions available for assessment,
- Minimal infrastructure,
- Flexible modeling and simulation (M&S), operator-in-the-loop (OITL), and hardware-in-the-loop (HWIL) capabilities, and
- In-line, continuous performance monitoring and selective forensics. (Roberts 2006)

Enterprise architecture (EA) can be used as a primary tool in support of evaluation and assessment. EA can be used to provide a model to understand how the parts of the enterprise fit together (or do not) (Giachetti 2010). The structure and contents of the EA should be driven by the key business decisions (or, as shown in the six-step process presented by Martin (2005), the architecture should be driven by the "business questions" to be addressed by the architecture).

The evaluation and assessment success measures can be put into the EA models and views directly and mapped to the elements that are being measured. An example of this can be seen in the US National Oceanographic and Atmospheric Agency (NOAA) EA shown by Martin (2003a and 2003b). The measures are shown, in this example, as success factors, key performance indicators, and information needs in the business strategy layer of the architecture.

EA can be viewed as either the set of artifacts developed as "views" of the enterprise, or as a set of activities that create, use, and maintain these artifacts. The literature uses these terms in both senses and it is not always clear in each case which sense is intended.

Enterprise Portfolio Considerations

Opportunity Assessment and Management

The management activities dealing with opportunities (as opposed to just risk) are included in ESE. According to White (2006), the "greatest enterprise risk may be in not pursuing enterprise opportunities." Hillson believes there is:

> *a systemic weakness in risk management as undertaken on most projects. The standard risk process is limited to dealing only with uncertainties that might have negative impact (threats). This means that risk management as currently practiced is failing to address around half of the potential uncertainties—the ones with positive impact (opportunities).* (Hillson 2004)

White claims that "in systems engineering at an enterprise scale the focus should be on opportunity, and that enterprise risk should be viewed more as something that threatens the pursuit of enterprise opportunities" (White 2006). The figure below (Rebovich and White 2011, chapter 5) shows the relative importance of opportunity and risk at the different scales of an individual system, a system of systems (SoS), and an enterprise. The implication is that, at the enterprise level, there should be more focus on opportunity management than on risk management.

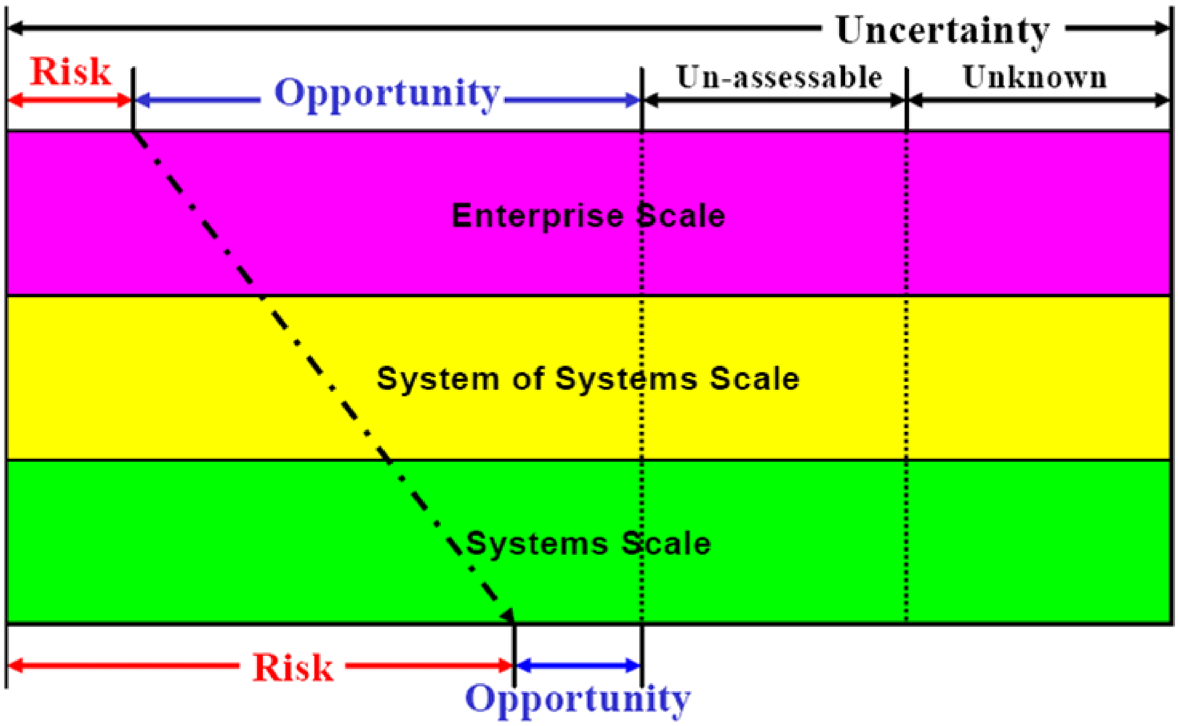


Figure 4. Risk & Opportunity at the Enterprise Scale versus the Systems Scale (White 2006). MITRE Approved for Public Release; Distribution Unlimited. Unique Tracking #05-1262.

Enterprise Architecture and Requirements

EA goes above and beyond the technical components of product systems to include additional items such as strategic goals and objectives, operators and users, organizations and other stakeholders, funding sources and methods, policies and practices, processes and procedures, facilities and platforms, infrastructure, and real estate. EA can be used to provide a model to understand how the parts of the enterprise fit together (or don't) (Giachetti 2010). The EA is not strictly the province of the chief information officer (CIO), and is not only concerned with information technology. Likewise, enterprise requirements need to focus on the cross-cutting measures necessary to ensure overall enterprise success. Some of these enterprise requirements will apply to product systems, but they may also apply to business processes, inter-organizational commitments, hiring practices, investment directions, and so on (Bernus, Nemes, and Schmidt 2003).

Architecture descriptions following the guidelines of an architecture framework have been used to standardize the views and models used in architecting efforts (Zachman 1987 and 1992; Spewak 1992). Architecture descriptions have also been developed using a business-question based approach (Martin 2003b; Martin 2006). The standard on Architecture Description Practices (ISO/IEC 42010) (ISO/IEC 2011) has expanded its scope to include requirements on architecture frameworks.

Government agencies have been increasingly turning to SE to solve some of their agency-level (i.e., enterprise) problems. This has sometimes led to the use of an architecture-based investment process, especially for information technology procurements. This approach imposes a requirement for linking business strategies to the development of EAs. The Federal Enterprise Architecture Framework (FEAF) (CIO Council 1999) and the DoD Architecture Framework (DoDAF) (DoD 2010) were developed to support such an architecture-based investment process. There have been several other architecture frameworks also developed for this purpose (ISO 2000; ISO/IEC 1998; NATO 2004; TOGAF 2009; MOD 2010; TRAK 2010).

ESE Process Elements

As a result of the synthesis outlined above, the ESE process elements to be used at the enterprise scale are as follows:

1. Strategic Technical Planning,
2. Capability-Based Planning Analysis,
3. Technology and Standards Planning,
4. Enterprise Evaluation and Assessment,
5. Opportunity and Risk Assessment and Management,
6. Enterprise Architecture and Conceptual Design,
7. Enterprise Requirements Definition and Management,
8. Program and Project Detailed Design and Implementation,
9. Program Integration and Interfaces,
10. Program Validation and Verification,
11. Portfolio and Program Deployment and Post Deployment, and
12. Portfolio and Program Life Cycle Support.

The first seven of these elements were described in some detail above. The others are more self-evident and are not discussed in this article.

References

Works Cited

Ackoff, R.L. 1989. "From Data to Wisdom." *Journal of Applied Systems Analysis*. 16 (1): 3-9.

Bernus, P., L. Nemes, and G. Schmidt (eds.). 2003. *Handbook on Enterprise Architecture*. Berlin and Heidelberg, Germany: Springer-Verlag.

CIO Council. 1999. *Federal Enterprise Architecture Framework (FEAF), Version 1.1*. Washington, DC, USA: Federal Chief Information Officers Council.

DoD. 2010. *DoD architecture framework (DoDAF)*, version 2.0. Washington, DC: US Department of Defense (DoD).

Giachetti, R.E. 2010. *Design of Enterprise Systems: Theory, Architecture, and Methods*. Boca Raton, FL, USA: CRC Press, Taylor & Francis Group.

Hillson, D. 2004. *Effective Opportunity Management for Projects: Exploiting Positive Risk*. Petersfield, Hampshire, UK; New York, NY, USA: Rick Doctor & Partners; Marcel Dekker, Inc.

Hines, P., and N. Rich. 1997. "The Seven Value Stream Mapping Tools." *International Journal of Operations & Production Management*. 1 (17): 46-64.

ISO. 2000. ISO 15704:2000, *Industrial Automation Systems — Requirements for Enterprise-Reference Architectures and Methodologies.* Geneva, Switzerland: International Organization for Standardization (ISO).

ISO/IEC/IEEE. 2011. *Systems and software engineering - Architecture description*. Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC)/Institute of Electrical and Electronics Engineers (IEEE), ISO/IEC/IEEE 42010.

ISO/IEC. 1998. ISO/IEC 10746:1998, *Information Technology — Open Distributed Processing — Reference Model: Architecture.* Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC).

Martin, J.N. 2010. "An Enterprise Systems Engineering Framework." Presented at 20th Anniversary International Council on Systems Engineering (INCOSE) International Symposium, July 12-15, 2010, Chicago, IL, USA.

Martin, J.N. 2006. "An Enterprise Architecture Process Incorporating Knowledge Modeling Methods." PhD dissertation. Fairfax, VA, USA: George Mason University.

Martin, J.N. 2005. "Using an Enterprise Architecture to Assess the Societal Benefits of Earth Science Research." Presented at 15th Annual International Council on Systems Engineering (INCOSE) International Symposium, 2005, Rochester, NY, USA.

Martin, J.N. 2003a. "An Integrated Tool Suite for the NOAA Observing System Architecture." Presented at 13th Annual International Council on Systems Engineering (INCOSE) International Symposium, 2003, Arlington, VA, USA.

Martin, J.N. 2003b. "On the Use of Knowledge Modeling Tools and Techniques to Characterize the NOAA Observing System Architecture." Presented at 13th Annual International Council on Systems Engineering (INCOSE) International Symposium, 2003, Arlington, VA, USA.

McCaughin, K., and J.K. DeRosa. 2006. "Process in Enterprise Systems Engineering." Presented at 16th Annual International Council on Systems Engineering (INCOSE) International Symposium, July 9-13, 2006, Orlando, FL, USA.

MOD. 2010. *Ministry of Defence Architecture Framework (MODAF)*, version 1.2.004. London, England, UK: UK Ministry of Defence. Accessed September 8, 2011. Available: http:/ / www. mod. uk/ NR/ rdonlyres/ 04B5FB3F-8BBC-4A39-96D8-AFA05E500E4A/0/20100602MODAFDownload12004.pdf.

NATO. 2010. *NATO Architecture Framework (NAF)*, version 3.1. Brussels, Belgium: North Atlantic Treaty Organization.

Rebovich, G., and B.E. White (eds.). 2011. *Enterprise Systems Engineering: Advances in the Theory and Practice*. Boca Raton, FL, USA: CRC Press, Taylor & Francis Group, Auerbach.

Roberts, J.L. 2006. "Enterprise Analysis and Assessment." Presented at 16th Annual International Council on Systems Engineering (INCOSE) International Symposium, July 9-13, 2006, Orlando, FL, USA.

Rother, M. 2009. *Toyota Kata: Managing People for Improvement, Adaptiveness, and Superior Results*. New York, NY, USA: McGraw-Hill.

Rother, M., and J. Shook. 1999. *Learning to See: Value-Stream Mapping to Create Value and Eliminate MUDA*. Cambridge, MA, USA: Lean Enterprise Institute.

Spewak, S.H. 1992. *Enterprise Architecture Planning: Developing a Blueprint for Data, Applications and Technology*. New York, NY, USA: Wiley and Sons, Inc.

TOGAF. 2009. "The Open Group Architecture Framework," version 9. Accessed September 2, 2011. Available: http://www.opengroup.org/togaf.

TRAK. 2011. "TRAK Enterprise Architecture Framework." Accessed September 7, 2011. Available: http://trak.sourceforge.net/index.html.

White, B.E. 2006. "Enterprise Opportunity and Risk." Presented at 16th Annual International Council on Systems Engineering (INCOSE) International Symposium, July 9-13, 2010, Orlando, FL, USA.

Zachman, J.A. 1992. "Extending and Formalizing the Framework for Information Systems Architecture." *IBM Systems Journal*. 31 (3): 590-616.

Zachman, J.A. 1987. "A Framework for Information Systems Architectures." *IBM Systems Journal*. 26 (3): 276-292.

Primary References

Giachetti, R.E. 2010. *Design of Enterprise Systems: Theory, Architecture, and Methods*. Boca Raton, FL, USA: CRC Press, Taylor & Francis Group.

Martin, J.N. 2010. "An Enterprise Systems Engineering Framework." Presented at 20th Anniversary International Council on Systems Engineering (INCOSE) International Symposium, July 12-15, 2010, Chicago, IL, USA.

Rebovich, G., and B.E. White (eds.). 2011. *Enterprise Systems Engineering: Advances in the Theory and Practice*. Boca Raton, FL, USA: CRC Press, Taylor & Francis Group, Auerbach.

Additional References

DeRosa, J.K. 2005. "Enterprise Systems Engineering." Presented at Air Force Association, Industry Day, Day 1, August 4, 2005, Danvers, MA, USA.

Holt, J., and S. Perry. 2010. *Modelling enterprise architectures*. Stevenage, England, UK: Institution of Engineering and Technology (IET).

Kaplan, R., and D. Norton. 1996. *The Balanced Scorecard: Translating Strategy into Action*. Cambridge, MA, USA: Harvard Business School Press.

McGovern, J., S. Ambler, M. Stevens, J. Linn, V. Sharan, and E. Jo. 2004. *A Practical Guide to Enterprise Architecture*. New York, NY, USA: Prentice Hall.

Swarz, R.S., J.K. DeRosa, and G. Rebovich. 2006. "An Enterprise Systems Engineering Model." INCOSE Symposium Proceedings, July 9-13, 2006, Orlando, FL, USA.

< Previous Article | Parent Article | Next Article >

SEBoK v. 2.1, released 31 October 2019

Enterprise Capability Management

Lead Authors: *James Martin, Bud Lawson, Alan Faisandier*

Introduction

There are three different kinds of capability: organizational capability, system capability, and operational capability. Management of organizational capability is addressed in the article called Enabling Businesses and Enterprises. Management of system capability is addressed by the Systems Engineering (SE) management activities described in the articles called Systems Engineering Management and Product and Service Life Management. Management of operational capability is described herein.

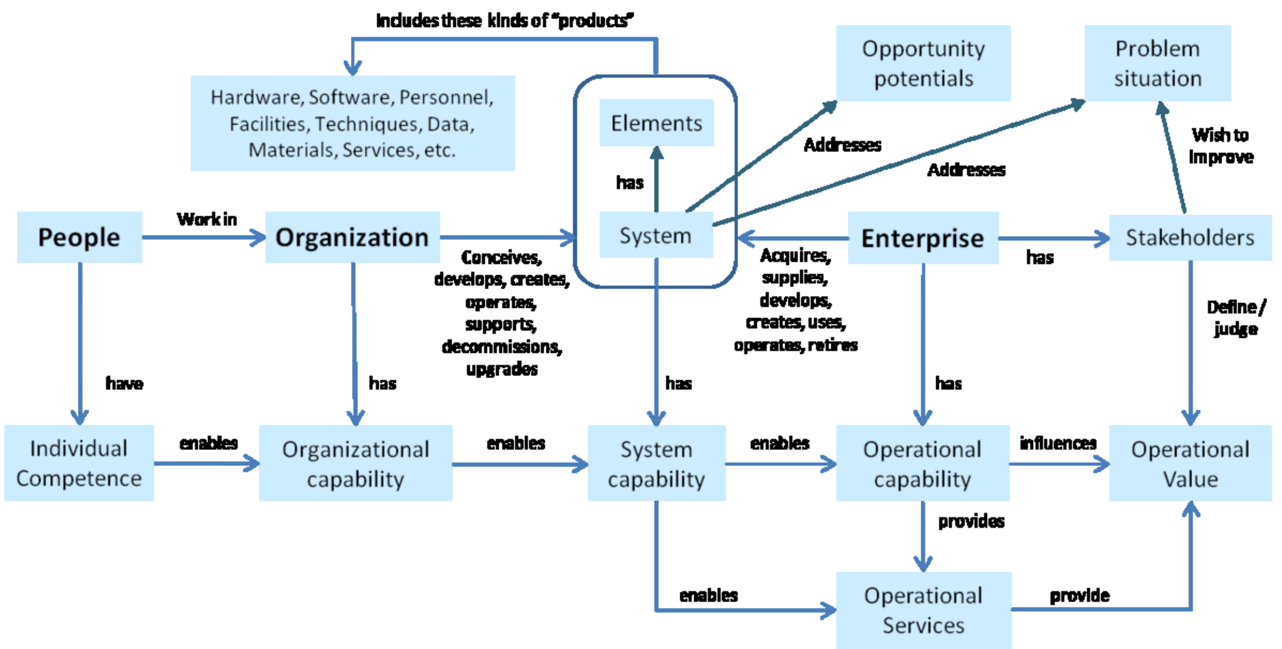


Figure 1. Three Kinds of Capability in the Enterprise: Organizational, System & Operational Capability. (SEBoK Original)

The enterprise has a current and planned (baseline) operational capability, based on its past activities and on its current plans for change. The purpose of the enterprise capability management function is to ensure the possibility of "vectoring" the enterprise away from the current baseline trajectory to a more desirable position where it can better meet its enterprise strategic goals and objectives, given all its resource constraints and other limitations.

Operational capability may need to include elements identified in the Information Technology Infrastructure Library (ITIL) best practices for operations management, starting with strategic operation planning (OGC 2009).

> *The ITIL is a set of practices for IT service management (ITSM) that focuses on aligning IT services with the needs of business. In its current form …, ITIL is published in a series of five core publications, each of which covers an ITSM lifecycle stage.*
>
> *ITIL describes procedures, tasks and checklists that are not organization-specific, used by an organization for establishing a minimum level of competency. It allows the organization to establish a baseline from which it can plan, implement, and measure. It is used to demonstrate compliance and to measure improvement.* (http://en.wikipedia.org/wiki/Information_Technology_Infrastructure_Library).

Needs Identification & Assessment

The enterprise has key stakeholders that have operational needs they would like the enterprise to address. These operational needs must be identified and assessed in terms of their relevance to the enterprise and the relative priorities of these needs compared to each other and to the priorities of the enterprise itself. The enterprise exists to meet these needs. An operational need is an expression of something desirable in direct support of the enterprise's end user activities. End user activities include such things as retail sales, entertainment, food services, and business travel. An example of an operational need is: "Provide transportation services to commuters in the metropolitan area of London."

Enterprise needs can be much more than eliminating waste, and the challenge for ESE might relate to any or all of the following: countering a perceived threat (business or military), meeting a policy goal (as in government), doing existing business more efficiently, taking advantage of technological opportunities, meeting new operational needs, replacing obsolete systems, creating integrated enterprises with others (on a temporary or permanent basis), and so on.

In addition to operational needs, there are enterprise needs that relate to enabling assets the enterprise has in place that allow the mission to be accomplished. Enabling assets are things such as personnel, facilities, communication networks, computing facilities, policies and practices, tools and methods, funding and partnerships, equipment and supplies, and so on. An enterprise need is an expression of something desirable in direct support of the enterprise's internal activities. Internal activities include such things as market forecast, business development, product development, manufacturing, and service delivery.

The purpose of the enterprise's enabling assets is to effect state changes to relevant elements of the enterprise necessary to achieve targeted levels of performance. The enterprise "state" shown in the figure below is a complex web of past, current and future states (Rouse 2009). The enterprise work processes use these enabling assets to accomplish their work objectives in order to achieve the desired future states. Enterprise architecture (EA) can be used to model these states and the relative impact each enabling asset has on the desired state changes.

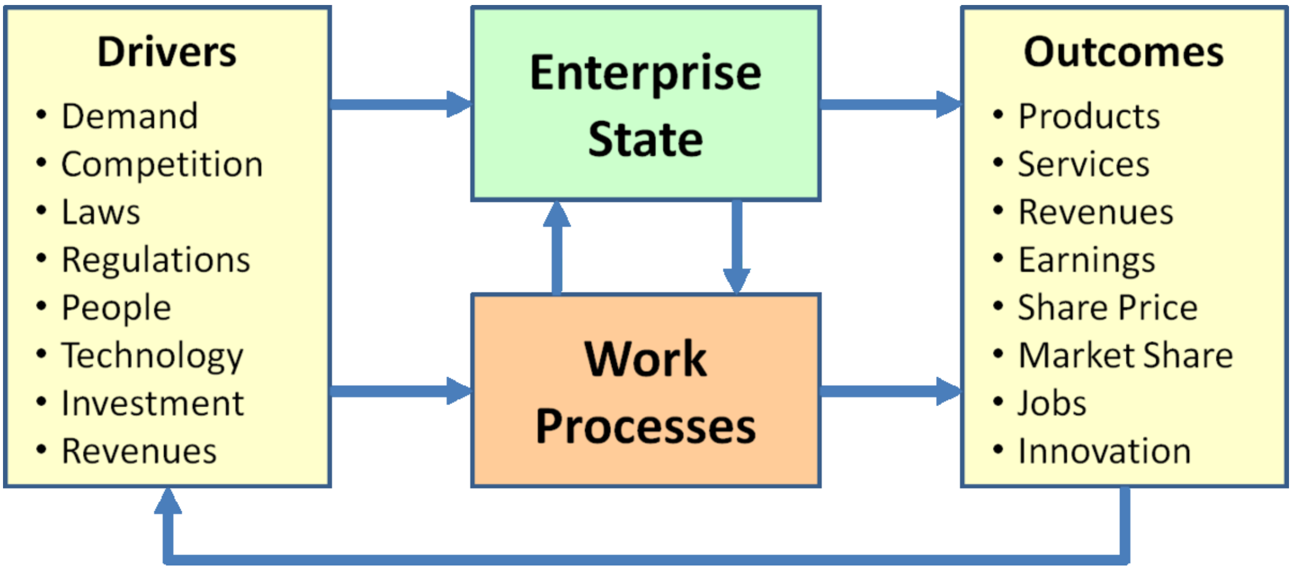


Figure 2. Enterprise State Changes Through Work Process Activities (Rouse 2009). Reprinted with permission of John Wiley & Sons Inc. All other rights are reserved by the copyright owner.

Enterprise needs are related to the enterprise efficiencies achieved through the performance of enterprise activities. The main goal of enterprise needs is to maximize the efficient utilization of enterprise assets, or in other words, enhance productivity, and find and eliminate waste. Waste represents that which does not contribute to the enterprise mission or that cannot reasonably be expected to be accomplished by the enterprise. An example of an enterprise need is: "Decrease power required for operation of enterprise data centers." (Power is a limited asset that consumes scarce enterprise funds that could be used for delivery of other more valuable services to its customers.)

Capability Identification & Assessment

The capabilities of an enterprise should exist for the sole purpose of meeting mission and enterprise needs. Hence, there will be both mission and enterprise capabilities to identify and assess how well they meet these needs. An example of an operational capability is: “Transport 150,000 passengers per hour among 27 nodes in the network.” A supporting enterprise capability might be: “Process 200,000 tickets per hour during peak loading.” There is a baseline capability due to capability development up to that point in time, plus any additional capability planned for the future. The desired levels of capability (based on needs assessment) are compared to the baseline capability to determine the capability gaps for the enterprise. This activity will also determine points of excess capability.

The gaps should be filled and the excesses should be eliminated. The projected gaps and excesses are sometimes mapped into several future timeframes to get a better understanding of the relative timing and intensity of change that might be required. It is typical to use time “buckets” like near-term, mid-term, and far-term, which, for some long-lasting capabilities, might correspond to five, ten, and twenty years out respectively. Of course, for fast-changing capabilities (like consumer products) these timeframes would necessarily be shorter in duration, for example, one, two and three years out.

Enterprise Architecture Formulation & Assessment

Enterprise architecture analysis can be used to determine how best to fill these capability gaps and minimize the excess capabilities (or “capacities”). Usually a baseline architecture is characterized for the purpose of understanding what one currently has and where the enterprise is headed under the current business plans. The needs and gaps are used to determine where in the architecture elements need to be added, dropped, or changed. Each modification represents a potential benefit to various stakeholders, along with associated costs and risks for introducing that modification. Enterprise architecture can be used to provide a model to understand how the parts of the enterprise fit together (or do not) (Giachetti 2010).

The enterprise architecture effort supports the entire capability management activity with enterprise-wide views of strategy, priorities, plans, resources, activities, locations, facilities, products, services, and so on (ISO/IEC/IEEE 15288 (ISO/IEC/IEEE 2015) and architectural design process: ISO/IEC 42010 (ISO/IEC 2011) and ISO 15704 (ISO 2000)).

Opportunity Identification & Assessment

The enterprise architecture is used to help identify opportunities for improvement. Usually these opportunities require the investment of time, money, facilities, personnel, and so on. There might also be opportunities for “divestment,” which could involve selling of assets, reducing capacity, canceling projects, and so on. Each opportunity can be assessed on its own merits, but usually these opportunities have dependencies and interfaces with other opportunities, with the current activities and operations of the enterprise, and with the enterprise's partners. Therefore, the opportunities may need to be assessed as a “portfolio,” or, at least, as sets of related opportunities. Typically, a business case assessment is required for each opportunity or set of opportunities.

Enterprise Portfolio Management

If the set of opportunities is large or has complicated relationships, it may be necessary to employ portfolio management techniques. The portfolio elements could be bids, projects, products, services, technologies, intellectual property, etc., or any combination of these items. Examples of an enterprise portfolio captured in an architecture modeling tool can be found in Martin (2005), Martin et al. (2004), and Martin (2003). See Kaplan's work (2009) for more information on portfolio management, and ISO/IEC (2008) for information on projects portfolio management process.

Enterprise Improvement Planning & Execution

The results of the opportunity assessment are compiled and laid out in an enterprise plan that considers all relevant factors, including system capabilities, organizational capabilities, funding constraints, legal commitments and obligations, partner arrangements, intellectual property ownership, personnel development and retention, and so on. The plan usually goes out to some long horizon, typically more than a decade, depending on the nature of the enterprise's business environment, technology volatility, market intensity, and so on. The enterprise plan needs to be in alignment with the enterprise's strategic goals and objectives and with leadership priorities.

The planned improvements are implemented across the enterprise and in parts of the extended enterprise (glossary) where appropriate, such as suppliers in the supply chain, distributors in the distribution chain, financiers in the investment arena, and so on. The planned changes should have associated performance targets and these metrics should be monitored to ensure that progress is being made against the plan and that the intended improvements are being implemented. As necessary, the plan is adjusted to account for unforeseen circumstances and outcomes. Performance management of enterprise personnel is a key element of the improvement efforts.

Enterprise Capability Change Management

In an operational context (particularly in defense) the term “capability management” is associated with developing and maintaining all aspects of the ability to conduct certain types of missions in a given threat environment. In an industrial context, capability refers to the ability to manage certain classes of product and service through those parts of their life cycle that are relevant to the business. Changes to enterprise capability should be carefully managed to ensure that current operations are not adversely affected (where possible) and that the long term viability of the enterprise is maintained. The following seven lenses can be used to facilitate change management: strategic objectives, stakeholders, processes, performance metrics, current state alignment, resources, and maturity assessment (Nightingale and Srinivasan 2011).

Capability management is becoming more often recognized as a key component of the business management tool suite:

> *Capability management aims to balance economy in meeting current operational requirements, with the sustainable use of current capabilities, and the development of future capabilities, to meet the sometimes competing strategic and current operational objectives of an enterprise. Accordingly, effective capability management assists organizations to better understand, and effectively integrate, re-align and apply the total enterprise ability or capacity to achieve strategic and current operational objectives; and develops and provides innovative solutions that focus on the holistic management of the defined array of interlinking functions and activities in the enterprise's strategic and current operational contexts.* (Saxena 2009, 1)

There is a widespread perception that capability management is only relevant to defense and aerospace domains. However, it is becoming more widely recognized as key to commercial and civil government efforts.

References

Works Cited

Giachetti, R.E. 2010. *Design of Enterprise Systems: Theory, Architecture, and Methods*. Boca Raton, FL, USA: CRC Press, Taylor & Francis Group.

ISO. 2000. ISO 15704:2000, *Industrial Automation Systems — Requirements for Enterprise — Reference Architectures and Methodologies.* Geneva, Switzerland: International Organization for Standardization (ISO).

ISO/IEC/IEEE. 2011. *Systems and software engineering - Architecture description*. Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC)/Institute of Electrical and Electronics Engineers (IEEE), ISO/IEC/IEEE 42010.

ISO/IEC/IEEE. 2015.*Systems and software engineering - system life cycle processes*.Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC), Institute of Electrical and Electronics Engineers.ISO/IEC 15288:2015.

Kaplan, J. 2009. *Strategic IT Portfolio Management: Governing Enterprise Transformation*. Waltham, MA, USA: Pittiglio, Rabin, Todd & McGrath, Inc. (PRTM).

Martin, J.N. 2005. "Using an Enterprise Architecture to Assess the Societal Benefits of Earth Science Research." Presented at 15th Annual International Council on Systems Engineering (INCOSE) International Symposium, 2005, Rochester, NY, USA.

Martin, J.N. 2003. "On the Use of Knowledge Modeling Tools and Techniques to Characterize the NOAA Observing System Architecture." Presented at 13th Annual International Council on Systems Engineering (INCOSE) International Symposium, 2003, Arlington, VA, USA.

Martin, J.N., J. Conklin, J. Evans, C. Robinson, L. Doggrell, and J. Diehl. 2004. "The Capability Integration Framework: A New Way of doing Enterprise Architecture." Presented at 14th Annual International Council on Systems Engineering (INCOSE) International Symposium, June 20-24, 2004, Toulouse, France.

Nightingale, D., and J. Srinivasan. 2011. *Beyond the Lean Revolution: Achieving Successful and Sustainable Enterprise Transformation*. New York, NY, USA: AMACOM Press.

OGC (Office of Government Commerce). 2009. *ITIL Lifecycle Publication Suite Books*. London, UK: The Stationery Office.

Rouse, W.B. 2009. "Engineering the Enterprise as a System," in *Handbook of Systems Engineering and Management*, 2nd ed., edited by A.P. Sage and W.B. Rouse. New York, NY, USA: Wiley and Sons, Inc.

Saxena, M.S. 2009. *Capability Management: Monitoring & Improving Capabilities*. New Dehli: Global India Publications Pvt Ltd.

Wikipedia contributors. "Information Technology Infrastructure Library." *Wikipedia, The Free Encyclopedia*. Accessed November 28, 2012. Available at: http:/ / en. wikipedia. org/ wiki/ Information_Technology_Infrastructure_Library.

Primary References

Kaplan, J. 2009. *Strategic IT Portfolio Management: Governing Enterprise Transformation*. Waltham, MA, USA: Pittiglio, Rabin, Todd & McGrath, Inc. (PRTM).

Nightingale, D., and J. Srinivasan. 2011. *Beyond the Lean Revolution: Achieving Successful and Sustainable Enterprise Transformation*. New York, NY, USA: AMACOM Press.

Rouse, W.B. 2009. "Engineering the Enterprise as a System," in *Handbook of Systems Engineering and Management*, 2nd ed., edited by A.P. Sage and W.B. Rouse. New York, NY, USA: Wiley and Sons, Inc.

Additional References

Dahmann, J.S., J.A. Lane, and G. Rebovich. 2008. "Systems Engineering for Capabilities." *CROSSTALK: The Journal of Defense Software Engineering* 21 (11): 4-9.

Hillson, D. 2004. *Effective Opportunity Management for Projects: Exploiting Positive Risk*. Petersfield, Hampshire, UK; New York, NY: Rick Doctor & Partners; Marcel Dekker, Inc.

Lillehagen, F., J. Kostie, S. Inella, H.G. Solheim, and D. Karlsen. 2003. "From enterprise modeling to enterprise visual scenes." Presented at International Society for Pharmaceutical Engineering (ISPE) Conference on Concurrent Engineering (CE), July 26-30, 2003, Madeira Island, Portugal.

McGovern, J., S. Ambler, M. Stevens, J. Linn, V. Sharan, and E. Jo. 2004. *A Practical Guide to Enterprise Architecture*. New York, NY: Prentice Hall.

Rechtin, E. 1999. *Systems Architecting of Organizations: Why Eagles Can't Swim*. Boca Raton, FL, USA: CRC Press, Taylor and Francis Group.

Roberts, J.L. 2006. "Enterprise Analysis and Assessment." Presented at 16th Annual International Council on Systems Engineering (INCOSE) International Symposium, July 9-13, 2006, Orlando, FL, USA.

< Previous Article | Parent Article | Next Article >

SEBoK v. 2.1, released 31 October 2019

Knowledge Area: Systems of Systems (SoS)

Systems of Systems (SoS)

Lead Authors: *Mike Henshaw, Judith Dahmann, Bud Lawson*

System of systems engineering (SoSE) is not a new discipline; however, this is an opportunity for the systems engineering community to define the complex systems of the twenty-first century (Jamshidi 2009). While systems engineering is a fairly established field, SoSE represents a challenge for the present systems engineers on a global level. In general, SoSE requires considerations beyond those usually associated with engineering to include socio-technical and sometimes socio-economic phenomena.

Topics

Each part of the SEBoK is divided into knowledge areas (KAs), which are groupings of information with a related theme. The KAs in turn are divided into topics. This KA contains the following topics:

- Architecting Approaches for Systems of Systems
- Socio-Technical Features of Systems of Systems
- Capability Engineering

Definition and Characteristics of Systems of Systems

There are several definitions of system(s) of systems (SoS), some of which are dependent on the particularity of an application area. Maier (1998) postulated five key characteristics (not criteria) of SoS: operational independence of component systems, managerial independence of component systems, geographical distribution, emergent behavior, and evolutionary development processes, and identified operational independence and managerial independence as the two principal distinguishing characteristics for applying the term 'systems-of-systems.' A system that does not exhibit these two characteristics is not considered a system-of-systems regardless of the complexity or geographic distribution of its components.

In the Maier characterization, emergence is noted as a common characteristic of SoS particularly in SoS composed of multiple large existing systems, based on the challenge (in time and resources) of subjecting all possible logical threads across the myriad functions, capabilities, and data of the systems in an SoS. As introduced in the article Emergence, there are risks associated with unexpected or unintended behavior resulting from combining systems that have individually complex behavior. These become serious in cases which safety, for example, is threatened through unintended interactions among the functions provided by multiple constituent systems in a SoS.

ISO/IEC/IEEE 21839 (ISO, 2019) provides a definition of SoS and constituent system:

> **System of Systems (SoS) —** *Set of systems or system elements that interact to provide a unique capability that none of the constituent systems can accomplish on its own. Note: Systems elements can be necessary to facilitate the interaction of the constituent systems in the system of systems*
>
> **Constituent Systems —** *Constituent systems can be part of one or more SoS. Note: Each constituent is a useful system by itself, having its own development, management goals and resources, but interacts within the SoS to provide the unique capability of the SoS.*

It should be noted that formation of a SoS is not necessarily a permanent phenomenon, but rather a matter of necessity for integrating and networking systems in a coordinated way for specific goals such as robustness, cost, efficiency, etc.

ISO/IEC/IEEE 15288 Annex G also describes the impact of these characteristics on the implementation of systems engineering processes. Because of the independence of the constituent systems, these processes are in most cases implemented for engineering both the systems and the system of systems and need to be tailored to support the characteristics of SoS. These processes are shown in the table below highlighting the fact that these processes are implemented at both the system and SoS levels, with SoSE often constrained by the systems.

Table 1. Differences Between Systems and Systems of Systems as They Apply to Systems Engineering.

SE Process	Implementation as Applied to SoS
Agreement processes	Because there is often no top level SoS authority, effective agreements among the systems in the SoS are key to successful SoSE.
Organizational project enabling processes	SoSE develops and maintains those processes which are critical for the SoS within the constraints of the system level processes.
Technical management processes	SoSE implements technical management processes applied to the particular considerations of SoS engineering - planning, analyzing, organizing, and integrating the capabilities of a mix of existing and new systems into a system-of-systems capability while systems continue to be responsible for technical management of their systems.
Technical processes	SoSE technical processes define the cross-cutting SoS capability, through SoS level business/mission analysis and stakeholder needs and requirements definition. SoS architecture and design frame the planning, organization and integration of the constituent systems, constrained by system architectures. Development, integration, verification, transition and validation are implemented by the systems. with SoSE monitoring and review. SoSE integration, verification, transition and validation applies when constituent systems are integrated into the SoS and performance is verified and validated.

Finally, based on work done by the INCOSE Systems of Systems Work Group (Dahmann, 2014), the major challenges facing SoSE have been catalogued in terms of seven pain points. These challenges are presented in the SoSE section of the INCOSE SE Handbook. (INCOSE 2015). These challenges include:

- **SoS Authorities**. In a 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 systems engineering under a single authority responsible for the entire system is absent from most SoS. In a SoS, SE relies on cross-cutting 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 towards collective objectives which may or may not coincide with those of the individual constituent systems.
- **Leadership**. Recognizing that the lack of common authorities and funding pose challenges for SoS, a related issue is the challenge of leadership in the multiple organizational environment of a SoS. This question of leadership is experienced where a lack of structured control normally present in SE of systems requires alternatives to provide coherence and direction, such as influence and incentives.
- **Constituent Systems' Perspectives.** Systems of systems are typically comprised, 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 which 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 the initial efforts at incorporating a system into a 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.
- **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 towards 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, the SoS SE needs to identify alternative approaches to meeting those needs through changes to the constituent systems or additions of other systems to the SoS. In effect this is asking the systems to take on new requirements with the SoS acting as the 'user'.

- **Autonomy, Interdependencies and Emergence.** The independence of constituent systems in a SoS is the source of a number of technical issues facing SE of SoS. The fact that a constituent system may continue to change independently of the SoS, along with interdependencies between that constituent system and other constituent systems, add to the complexity of the SoS and further challenges SE at the SoS level. In particular, these dynamics can lead to unanticipated effects at the SoS level leading to unexpected or unpredictable behavior in a SoS even if the behavior of constituent systems is well understood.
- **Testing, Validation, and Learning.** The fact that SoS are typically composed of constituent systems which are independent of the SoS poses challenges in conducting end-to-end SoS testing as is typically done with systems. Firstly, unless there is a clear understanding of the SoS-level expectations and measures of these expectations, it can be very difficult to assess level of performance as the basis for determining areas which need attention, or to assure users of the capabilities and limitations of the SoS. Even when there is a clear understanding of SoS objectives and metrics, testing in a traditional sense can be difficult. Depending on the SoS context, there may not be funding or authority for SoS testing. Often the development cycles of the constituent systems are tied to the needs of their owners and original ongoing user base. With multiple constituent systems subject to asynchronous development cycles, finding ways to conduct traditional end-to-end testing across the SoS can be difficult if not impossible. In addition, many SoS are large and diverse making traditional full end-to-end testing with every change in a constituent system prohibitively costly. Often the only way to get a good measure of SoS performance is from data collected from actual operations or through estimates based on modeling, simulation and analysis. Nonetheless the SoS SE team needs to enable continuity of operation and performance of the SoS despite these challenges.
- **SoS Principles.** SoS is a relatively new area, with the result that there has been limited attention given to ways to extend systems thinking to the issues particular to SoS. Work is needed to identify and articulate the cross cutting principles that apply to SoS in general, and to developing working examples of the application of these principles. There is a major learning curve for the average systems engineer moving to a SoS environment, and a problem with SoS knowledge transfer within or across organizations.

Types of SoS

In today's interconnected world, SoS occur in a broad range of circumstances. In those situations where the SoS is recognized and treated as a system in its right, an SoS can be described as one of four types (Maier 1998; Dahmann and Baldwin 2008):

- **Directed** - The SoS is created and managed to fulfill specific purposes and the constituent systems are subordinated to the SoS. The component systems maintain an ability to operate independently; however, their normal operational mode is subordinated to the central managed purpose;
- **Acknowledged** - The SoS has recognized objectives, a designated manager, and resources for the SoS; however, the constituent systems retain their independent ownership, objectives, funding, and development and sustainment approaches. Changes in the systems are based on cooperative agreements between the SoS and the system;
- **Collaborative** - The component systems interact more or less voluntarily to fulfill agreed upon central purposes. The central players collectively decide how to provide or deny service, thereby providing some means of enforcing and maintaining standards; and
- **Virtual** - The SoS lacks a central management authority and a centrally agreed upon purpose for the SoS. Large-scale behavior emerges—and may be desirable—but this type of SoS must rely on relatively invisible

mechanisms to maintain it.

This taxonomy is based on the degree of independence of constituents and it offers a framework for understanding SoS based on the origin of the SoS objectives and the relationships among the stakeholders for both the SoS and its constituent systems. In most actual cases, an SoS will reflect a combination of SoS types which may change over time. This taxonomy is in general use. It is presented in 15288 Annex G and in draft ISO 21840, "Taxonomy of Systems of systems". Other taxonomies may focus on nature/type of components, their heterogeneity, etc. (Cook 2014)

As noted above, many SoS exist in an unrecognized state; this is increasingly true as the levels of interconnectivity between modern systems keeps increasing. Kemp et al (2013) describe such systems as “accidental” but they can be described as “discovered” because it is only when they become significant for some reason that we recognize them, at which point they can usually fall into one of the above four categories, since their significance means they must now operate, with management, in some defined way.

From the SoSE point of view, another potential classification would consider the level of anticipation/preparation of SoSE with respect to SoS operations and level of stability of the SoS objectives; this is referred to as variability by kinder et. al. (2012). This could range from an SoS which responds to a particular trigger and is put immediately in place when needs are expressed. An example of such an SoS would be a crisis management SoS. This type of SoS is updated dynamically during the operation. At the other end of the spectrum there are well-specified and stable SoS developed to answer to specified ongoing needs. An example of such a persistent SoS is an air traffic management system. This type of SoS is acquired and qualified in a well-defined environment and any need for evolution will imply a formal SE evolution and re-qualification.

SoSE Application Domains

Application of SoSE is broad and is expanding into almost all walks of life. Originally identified in the defense environment, SoSE application is now much broader and still expanding. The early work in the defense sector has provided the initial basis for SoSE, including its intellectual foundation, technical approaches, and practical experience. In addition, parallel developments in information services and rail have helped to develop SoSE practice (Kemp and Daw, 2015). Now, SoSE concepts and principles apply across other governmental, civil and commercial domains.

Some examples include:

- **Transportation** - air traffic management, the European rail network, integrated ground transportation, cargo transport, highway management, and space systems,
- **Energy** - smart grid, smart houses, and integrated production/consumption,
- **Health Care** - regional facilities management, emergency services, and personal health management,
- **Defense** - Military missions such as missile defense, networked sensors,
- **Rail –** Urban, national, international rail systems,
- **Natural Resource Management** - global environment, regional water resources, forestry, and recreational resources,
- **Disaster Response** - responses to disaster events including forest fires, floods, and terrorist attacks,
- **Consumer Products** - integrated entertainment and household product integration,
- **Business-** banking and finance,and
- **Media** - film, radio, and television.

Increased networking and interconnectedness of systems today contributes to growth in the number and domains where SoS are becoming the norm, particularly with the considerable converge among systems of systems, cyber-physical systems and the internet of things. (Henshaw, 2016).

Difference between System of Systems Engineering and Systems Engineering

Observations regarding differences between individual or constituent systems and SoS are listed in Table 1. These differences are not as black and white as the table might suggest and in each case, the degree of difference varies in practice. Modern systems tend to be highly inter-connected, so that the assumptions that lead to the characteristics of Systems Engineering in Table 2 are less frequently met.

Table 2. Differences Between Systems and Systems of Systems as They Apply to Systems Engineering. (INCOSE, 2018)

Systems tend to ...	Systems of systems tend to ...
Have a clear set of stakeholders	Have multiple levels of stakeholders with mixed and possibly competing interests
Have clear objectives and purpose	Have multiple, and possibly contradictory, objectives and purpose
Have clear operational priorities, with escalation to resolve priorities	Have multiple, and sometimes different, operational priorities with no clear escalation routes
Have a single lifecycle	Have multiple lifecycles with elements being implemented asynchronously
Have clear ownership with the ability to move resources between elements	Have multiple owners making independent resourcing decisions

SoSE Standards

Standards for system of systems engineering are beginning to emerge as the practice of SoSE matures. As a recent report of an ISO SoS Standards study group (ISO, 2016) found that there is:

- A strong foundation for SoSE in fundamentals and continued growth in emerging areas
- SoSE activity across multiple domains and growing demand also across SoS domains
- Limited cross domain communication among practitioners
- Limited understanding and appreciation of current SoSE knowledge base, including current practice and research

In terms of SoS standards:

- The only SoS-specific standards were those in Annex G to ISO/IEC/IEEE 15288:2015
- There are no top level standards to aid in communication across SoSE practitioners and domains
- There is no guidance on how SoSE can employ existing standards (and not generate new duplicative standards).

Based on the study results, work has been initiated on three SoSE standards development activities to address this gap. These emerging standards are

- **ISO/IEC/IEEE 21839 –** ***System of Systems (SoS) Considerations in Life Cycle Stages of a System***

This standard provides a set of critical considerations to be addressed at key points in the life cycle of systems created by humans and refers to a constituent system that will interact in a system of systems as the system of interest (SOI). These considerations are aligned with ISO/IEC/IEEE 15288 and the ISO/IEC/IEEE 24748 framework for system life cycle stages and associated terminology.

- **ISO/IEC/IEEE 21840 –** ***Guidelines for the utilization of ISO/IEC/IEEE 15288 in the context of System of Systems (SoS) Engineering***

This standard provides guidance for the utilization of ISO/IEC/IEEE 15288 in the context of SoS. While ISO/IEC/IEEE 15288 applies to systems (including constituent systems), this document provides guidance on application of these processes to SoS. However, ISO/IEC/IEEE 21840 is not a self-contained SoS replacement for ISO/IEC/IEEE 15288. This document is intended to be used in conjunction with ISO/IEC/IEEE 15288, ISO/IEC/IEEE 21839 and ISO/IEC/IEEE 21841 and is not intended to be used without them.

- **ISO/IEC/IEEE 21841 –** ***Taxonomy of Systems of Systems***

The purpose of this standard is to define normalized taxonomies for systems of systems (SoS) to facilitate communications among stakeholders. It also briefly explains what a taxonomy is and how it applies to the SoS to aid in understanding and communication.

References

Works Cited

Boardman, J., and B. Sauser. 2006. "System of Systems - the Meaning of Of." IEEE Conference on Systems of Systems Engineering, April 24-26, 2006, Los Angeles, CA.

Cook, S. C. and Pratt, J. M., “Towards designing innovative SoSE approaches for the Australian defence force,” Proc. 9th Int. Conf. Syst. Syst. Eng. Socio-Technical Perspect. SoSE 2014, pp. 295–300, 2014.

Dahmann, J., and K. Baldwin. 2008. "Understanding the Current State of US Defense Systems of Systems and the Implications for Systems Engineering." Presented at IEEE Systems Conference, April 7-10, 2008, Montreal, Canada.

DoD. 2008. *Systems Engineering Guide for Systems of Systems.* Arlington, VA: US Department of Defense, Director, Systems and Software Engineering, Deputy Under Secretary of Defense (Acquisition and Technology), Office of the Under Secretary of Defense (Acquisition, Technology and Logistics). Accessed November 12, 2013. Available: http://www.acq.osd.mil/se/docs/SE-Guide-for-SoS.pdf.

Henshaw, M.J.d,, “Systems of Systems. Cyber-Physical Systems, the Internet of Things… Whatever Next?” INCOSE INSIGHT, October 2016, Volume 19, Issue 3, pages 51-54.

INCOSE, 2018. Systems of Systems Primer, INCOSE-TP-2018-003-01.0.

International Standards Organization, 2016. Report of the SC7 SG on Systems of Systems Engineering.

Kemp, D., et. al.. 2013. *Steampunk System of Systems Engineering: A case study of successful System of Systems engineering in 19th century Britain*." Presented at INCOSE International Symposium, June 24–27, 2013, Philadelphia, PA.

Kinder, A., Barot, V., Henshaw, M., & Siemieniuch, C. (2012). System of Systems: “Defining the system of interest.” In Proc. 7th Int. Conf. Systems of Systems Eng., Genoa, italy (pp. 463–468). Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6384211

Neaga, E.I., M.J.d. Henshaw, and Y. Yue. 2009. "The influence of the concept of capability-based management on the development of the systems engineering discipline." *Proceedings of the 7th Annual Conference on Systems Engineering Research,* April 20-23, 2009, Loughborough University, Loughborough, England, UK.

Maier, M.W. 1998. "Architecting Principles for Systems-of-Systems." *Systems Engineering.* 1 (4): 267-284.

Primary References

Dahmann, J., and K. Baldwin. 2008. "Understanding the Current State of US Defense Systems of Systems and the Implications for Systems Engineering." Presented at IEEE Systems Conference, April 7-10, 2008, Montreal, Canada.

DoD. 2008. Systems Engineering Guide for Systems of Systems, version 1.0. Washington, DC, USA: US Department of Defense (DoD). Available: http://www.acq.osd.mil/se/docs/SE-Guide-for-SoS.pdf.

INCOSE Systems Engineering Handbook, 4th Edition, John Wiley & Sons Inc., 2015

International Standards Organization, 2016. Report of the SC7 SG on Systems of Systems Engineering.

Jamshidi, M. (ed). 2009a. *Systems of Systems Engineering – Innovations for the 21st Century*. Hoboken, NJ, USA: Wiley.

Maier, M.W. 1998. "Architecting Principles for Systems-of-Systems." *Systems Engineering*. 1 (4): 267-284.

Additional References

Barot, V., S. Henson, M. Henshaw, C. Siemieniuch, M. Sinclair, S.L. Lim, M. Jamshidi, and D. DeLaurentis. 2012. *Trans-Atlantic Research and Education Agenda in Systems of Systems (T-AREA-SoS) SOA Report*. Longborough, England, UK: Longborough University. Ref. TAREA-RE-WP2-R-LU-7.

Carlock, P., and J.A. Lane. 2006. *System of Systems Enterprise Systems Engineering, the Enterprise Architecture Management Framework, and System of Systems Cost Estimation*. Los Angeles, CA, USA: Center for Systems and Software Engineering (CSSE), University of Southern California (USC). USC-CSE-2006-618.

Checkland, P.B. 1999. *Systems Thinking, Systems Practice*. Chichester, UK: John Wiley & Sons Ltd.

Dahmann, J., Rebovich, G., Lane, J., Lowry, R. & Baldwin, K. 2011. "An Implementer's View of Systems Engineering for Systems of Systems." IEEE Systems Conference, April 4-7, 2011, Montreal, Canada. p. 212-217.

Keating C.B., J.J. Padilla, and K. Adams. 2008. "System of systems engineering requirements: Challenges and guidelines". *EMJ - Engineering Management Journal.* 20 (4): 24-31.

Luzeaux, D., and J.R. Ruault. 2010. *Systems of Systems*. London, UK: ISTE.

Poza, A.S., S. Kovacic, and C. Keating. 2008. "System of Systems Engineering: An Emerging Multidiscipline". *International Journal of System of Systems Engineering.* 1 (1/2).

Rebovich Jr., G. 2009. "Chapter 6: Enterprise System of Systems," in *Systems of Systems Engineering - Principles and Applications*. Boca Raton, FL, USA: CRC Press.

Ring J. 2002. "Toward an ontology of systems engineering." *INSIGHT.* 5 (1): 19-22.

< Previous Article | Parent Article | Next Article >

SEBoK v. 2.1, released 31 October 2019

Architecting Approaches for Systems of Systems

Lead Authors: *Judith Dahmann, Bud Lawson, Mike Henshaw*

A key part of systems engineering (SE) for system of systems (SoS) is the composition of systems to meet SoS needs. This may include simply interfacing systems and leveraging their existing functionality or it may require changing the systems functionality, performance or interfaces. These changes occur incrementally, as a SoS evolves over time to meet changing SoS objectives. System of Systems Engineering (SoSE) supports these changes by developing and evolving a technical framework that acts as an overlay to the systems of which the SoS is composed. This framework provides the *architecture* for the SoS. The SoS architecture defines how the systems work together to meet SoS objectives and considers the details of the individual systems and their impact the SoS performance or functionality.

The Role of System of Systems Architecting

An architecture is the structure of components, their relationships, and the principles and guidelines governing their design evolution over time (IEEE 610.12-1990).

In a SoS, the architecture is the technical framework for the systems comprising the SoS which designates how the systems will be employed by the users in an operational setting (sometimes called the concept of operations (CONOPs or CONOPs), the internal and external relationships and dependencies among the constituent systems and their functions and, finally, the end-to-end functionality and data flow as well as communications among the systems.

Because SoS largely comprise extant independent systems, these place constraints on the SoS architecture and may require that migration to a SoS architecture be incremental. Developing a SoS architecture requires consideration of technical feasibility for the constituent systems as well as the needs of the SoS itself. Architecture data for the constituent systems can also be important data for architecting the SoS. There is some similarity here to the introduction of Commercial Off The Shelf (COTS) products into systems: the COTS product has been independently managed but sufficient data is required by the systems developer to ensure satisfactory integration. However, in this case the COTS product is not independently operated

Maier (1998) provides a conceptual discussion on the impact of SoS characteristics on SoS architecting. Additionally, the US DoD SE Guide for SoS (2008) describes practical considerations in developing and evolving a SoS architecture as a core element of SoSE.

Challenges in Architecting SoS

In the case of a new system development, the systems engineer can begin with a fresh, unencumbered approach to architecture. However, in a SoS, the systems contributing to the SoS objectives are typically in place when the SoS is established and the SoS engineer needs to consider the current state and plans of the individual systems as factors in developing an architecture for the SoS. This, along with the fact that constituent systems may be complex systems in their own right, leads to a set of challenges to architecting SoS. The approach to architecting must be determined by the type of SoS under consideration. Whereas a directed SoS can be architected in much the same way as a monolithic system, the other types are less straightforward, because the SOI may be less clearly defined and because the SoS architects knowledge of the constituent systems may be partial. Furthermore, whereas in a directed SoS the owner may have authority and funding to require architectural changes of constituent systems, in acknowledged and collaborative SoS re-architecting is at the discretion of the owners of the constituent systems. Maier (Maier 1998) has focused architecting attention on communication for SoS, arguing that this is the common feature of all types, and he partitions the communication into layers that have a similarity to the layers of interoperability (NCOIC, 2008).

The independence of the constituent systems means that these systems are typically not designed to optimize SoS objectives. It may even be the case that a constituent system should operate sub-optimally at the system level in order to achieve overall SoS effectiveness. (Rebovich 2009) has articulated this difficulty as a fundamental problem of SoS:

> *From the single-system community's perspective, its part of the SoS capability represents additional obligations, constraints and complexities. Rarely is participation in an (sic) SoS seen as a net gain from the viewpoint of single-system stakeholders.*

The development and implementation of a SoS architecture may be significantly constrained by a reluctance to make changes or invest in the constituent systems, which could be very mature (e.g. in sustainment) or currently productively supporting other uses. In this case, approaches such as gateways and wrapping may be used to incorporate these systems into the SoS without making significant changes in the other systems.

Architecture Analysis

Large-scale systems integration has grown in importance and correspondingly, there has been a growing interest in SoS concepts and strategies. The performance and functioning of groups of heterogeneous systems has become the focus of various applications including military, security, aerospace, distributed energy, healthcare, and disaster management systems (Lopez 2006; Wojcik and Hoffman 2006). There is an increasing interest in exploiting synergy between these independent systems to achieve the desired overall system performance (Azarnoush et al. 2006).

Modeling and simulation is conducted to analyze architecture effectiveness and to verify architectural features. In the literature, researchers have addressed the issues of coordination and interoperability in a SoS (Abel and Sukkarieh

2006; Sahin et al. 2007). In order to study SoS characteristics and parameters, one needs to have realistic simulation frameworks properly designed for system of systems architecture. There are some attempts to develop simulation frameworks for multi-agent systems using Discrete Event Simulation (DEVS) tools (Zeigler et al. 2000a). In these research efforts, the major focus is given to DEVS architecture with JAVA. In (Mittall 2000), DEVS state machine approach is introduced. Finally, DEVS Modeling Language (DEVSML) is developed by using XML based JAVA in order to simulate systems in a net-centric way with relative ease. Sahin et al. (2007) have recently introduced a discrete event XML based SoS simulation framework based on DEVS and JAVA.

The Open Approach to SoS Engineering

As noted above, one of the key challenges with SoS architecting is that the constituent systems of a SoS may not have been designed, developed and employed with regard to their role in the SoS, which constrains SoS architecture options. The degree to the architecture which *overlays* these constituent systems and supports the SoS end-to-end capabilities can be based on open standards; the SoS may be able to benefit from open architecture for future evolution.

The critical challenge of moving from SoS, as a concept to the engineering of SoS, is the significant technological, human, and organizational differences in consideration system of systems engineering and management approaches (Wells and Sage 2008). A potential approach to engineering a SoS can be the *open systems approach* to SoSE (Azani 2009). The following open systems principles are listed by Azani (2009):

- **Open interface principle** - Open systems have permeable boundaries that allow them to exchange mass, energy, and information with other systems;
- **Synergism principle –** The notion that designates that the co-operative interaction between constituent systems has a greater effect in their combined efforts than the sum of their individual parts. Essentially, this is what gives rise to emergence;
- **Self-government principle** - This implies that the SoS maintains and develops its internal order without interference from external sources. This could be through cybernetic control, homeostasis, or self-organization;
- **Emergence principle** - In this case, this refers to the occurrence of novel and coherent structures, patterns, and properties during the self-organization of the SoS;
- **Conservation principle –** This principle states that energy and mass (material) are conserved within the SoS;
- **Reconfiguration principle –** This refers to the SoS reconfiguring and adapting itself to sustain itself against changes in its environment;
- **Symbiosis principle** - The systems within the SoS have a symbiotic relationship to each other; more transparently, the successful development and sustainment of a SoS depends on symbiotic collaboration between the stakeholders of the systems of which it is comprised; and
- **Modularity principle** - This holds that each level and each system is to some extent independent of others. In SoS design, the development of independent modular systems that interoperate with each other through standardized interfaces enables greater flexibility to promote better evolution of the SoS.

Azani (2009) elaborates on the open systems development strategies and principles utilized by biotic SoS, discusses the implications of engineering of man-made SoS, and introduces an integrated SoS development methodology for the engineering and development of an adaptable, sustainable, and interoperable SoS based on open systems principles and strategies.

Hitchens (2003, 107), on the other hand, discusses the principles of open systems rather differently in terms of their systems life cycles, as the seven principles that he addresses are system reactions, system cohesion, system adaptation, connected variety, limited variety, preferred patterns, and cyclic progression. This description takes a systems dynamics approach to show how open systems evolve; the description is applicable to natural and man-made systems.

The enablers of openness include open standards and open specifications, which draw from consensus amongst a community of interest, and are published by, and freely available within, that community. An open specification must ensure that its detail-level is allows for it to be implementable by independent parties. Compliance with open standards is intended to ensure consistent results (Henshaw, et. al., 2011). This parallels the notion of open systems architecture, which is an open specification of the architecture of a system or system of systems for the purpose of acquiring specified capabilities. As a general feature of good design (for a system or system of systems), an open system architecture should allow for the quick and easy improvement and updating of the system capabilities, by adding or changing components. However, Henshaw et. al. (2011) also denote that open architecture represents a commercial challenge (rather than a technical one) and that establishing open architecture approaches to acquisition can be challenging, due to issues involving protection of intellectual property (IP) and commercial advantage.

Networks and Network Analysis

Because networks are such a common component of SoS, they warrant specific attention. In SoS that are based on an underlying network, communications and information exchange typically constitute a SoS in its own right. This enabling SoS requires architecting like any other SoS, which will be addressed in this section. In the case of an enabling network SoS, the 'user', the end-to-end functionality of the larger SoS and enabling network SoS is driven by these user needs. The relationship between SoSE concepts and network enablement, as well as the concepts of networks and network analysis that extend beyond information sharing, have been explored extensively by the defense community (Dickerson and Mavris 2009). For instance, during the U.S. Navy's work on command, control, communications, computers, intelligence, surveillance, and reconnaissance (C4ISR) as part of a SoS (Owens 1996), the term network included organizational aspects of command and control (C2) structure as well as communications.

Differences in the architecting of an enabling network SoS derive from the fact that these SoS are typically built on commercial technologies and architectures, which are changing rapidly in today's dynamic technological environment. In addition, these enabling networks are often shared among SoS and hence may further constrain the overall SoS architecture. For example, many SoS (for cost and convenience) expect to operate over the internet, and therefore must consider characteristics of the internet in the expectations for performance and security provided by use of a shared enabling infrastructure.

Enabling network SoS architecting is particularly well-served by the initial analysis that explores sensitivities through modeling, simulation, analysis, and/or laboratory experimentation and identifies scalability issues or divergent behavior (e.g., concerning requirements or usage assumptions, assumed network bandwidth, or others), beyond which performance starts to break down. This type of analysis provides a basis for network architecture decisions.

In directed SoS, because of the top-down control, there is the option for creating a specialized network for the particular SoS. In the other types of SoS, if the constituents are already supported by some type of a network then the overall SoS networking approach typically needs to accommodate these since the constituent systems are likely to need to continue to use their current approach to support their original users.

Interoperability

Interoperability within a SoS implies that each system can communicate and interact (control) with any other system regardless of their hardware and software characteristics or nature. This implies that each constituent member (and potential new members) of a SoS should be able to communicate with others without compatibility issues in the operating systems, communication hardware, and so on. For this purpose, a SoS needs a common language the SoS's systems can speak. Challenges here are to work towards a common language for exchange of information and data among systems of a SoS. Examples of such system are XML (eXtensible Markup Language), as one potential candidate (Jamshidi, 2009a).

However, interoperability must be achieved at many levels and not just at the data/network level. There are a number of frameworks that describe the levels of interoperability. From military applications, the NCOIC (Network Centric Operations Industry Consortium) Interoperability Framework (NCOIC 2008) covers three broad levels of interoperability, subdivided into further layers as indicated below:

- Network Transport:
  - Physical Interoperability and
  - Connectivity and Network Interoperability;
- Information Services:
  - Data/Object Model Interoperability,
  - Semantic/Information Interoperability, and
  - Knowledge/Awareness of Actions Interoperability; and
- People, Processes and Applications:
  - Aligned Procedures,
  - Aligned Operations,
  - Harmonized Strategy/Doctrine, and
  - Political or Business Objectives.

This spectrum of interoperability layers requires appropriate coherence at each layer consistent with the SoS shared goals.

There exist interoperability frameworks in other fields of activity. An example is the European Interoperability Framework (European Commission 2004), which focuses on enabling business (particularly e-business) interoperability and has four levels within a political context:

- Legal Interoperability,
- Organizational Interoperability,
- Semantic Interoperability, and
- Technical Interoperability.

The interoperability between the component systems of a SoS is a fundamental design consideration for SoS that may be managed through the application of standards.

References

Works Cited

Abel, A., and S. Sukkarieh. 2006. "The Coordination of Multiple Autonomous Systems using Information Theoretic Political Science Voting Models." Proceedings of the IEEE International Conference on System of Systems Engineering, April 24-26, 2006, Los Angeles, CA, USA.

Azani, C. 2009. *A Multi-criteria Decision Model for Migrating Legacy System Architectures into Open Systems and Systems-of-Systems Architectures.* Washington, DC, USA: Defense Acquisition University.

Azarnoush, H., B. Horan, P. Sridhar, A.M. Madni, and M. Jamshidi. 2006. "Towards optimization of a real-world robotic-sensor system of systems". Proceedings of the World Automation Congress (WAC), July 24-26, 2006, Budapest, Hungary.

Cloutier, R.M., J. DiMario, and H.W. Polzer. 2009. "Net-Centricity and System of Systems," in *Systems of Systems Engineering - Principles and Applications*, edited by M. Jamshidi. Boca Raton, FL, USA: CRC Press.

Dagli, C.H., and N. Kilicay-Ergin. 2009. "System of Systems Architecting," in *Systems of Systems Engineering - Principles and Applications*, edited by M. Jamshidi. Boca Raton, FL, USA: CRC Press.

Dickerson, C.E., and D. Mavris. (2009) *Architecture and Principles of Systems Engineering*. New York, NY, USA: CRC Press, Auerbach Publications.

DoD. 1998. *Levels of Information System Interoperability*. Washington, DC, USA: C4IST Interoperability Working Group, US Department of Defense.

Hall, J. 2007. "Openness – An Important Principle For The Stewardship of DoD IT Standards." *DSPO Journal*, 4-7. Available: http://www.dsp.dla.mil/app_uil/content/newsletters/journal/DSPJ-01-07.pdf.

Henshaw, M. (ed.). 2011. *Assessment of open architectures within defence procurement issue 1: systems of systems approach community forum working group 1 - open systems and architectures.* London, UK: SoSA Community Forum Working Group 1 (Crown owned copyright). Available: https://dspace.lboro.ac.uk/2134/8828.

Hitchins, D.K. 2003. *Advanced Systems Thinking, Engineering and Management*. Norwood, MA, USA: Artech House, Inc.

IEEE. 1990. IEEE 610.12-1990, *Standard Glossary of Software Engineering Terminology*. Washington, DC, USA: Institute of Electrical & Electronics Engineers (IEEE).

Jamshidi, M. (ed.) 2009. *Systems of Systems Engineering - Principles and Applications*. Boca Raton, FL, USA: CRC Press.

Johnson M. 2009. "System of Systems Standards," in *System of Systems Engineering - Innovations for the 21st Century*. Hoboken, NJ, USA: Wiley.

Lopez D. 2006. "Lessons Learned From the Front Lines of the Aerospace." Proceedings of the IEEE International Conference on System of Systems Engineering, April 24-26, 2006, Los Angeles, CA, US.

Mittal, S. 2000. "DEVS Unified Process for Integrated Development and Testing of Service Oriented Architectures." PhD Dissertation. Tucson, AZ, USA: University of Arizona.

NCOIC. 2008. "NCOIC Interoperability Framework (NIF(R))." Available: http://www.ncoic.org/technology/technical-products/frameworks/10-technology/33-tech-prod-framework-nif.

Owens, W.A. 1996. *The Emerging U.S. System-of-Systems*. Washington, DC, USA: The National Defense University, Institute of National Security Studies.

Rebovich Jr., G. 2009. "Chapter 6: Enterprise System of Systems," in *Systems of Systems Engineering - Principles and Applications.* Boca Raton, FL, USA: CRC Press. p. 169.

Sahin, F., M. Jamshidi, and P. Sridhar. 2007. "A Discrete Event XML based Simulation Framework for System of Systems Architectures." Proceedings of the IEEE International Conference on System of Systems, April 16-18, 2007, San Antonio, TX, USA.

US Department of Defense. 2008. *Systems Engineering Guide for Systems of Systems,* version 1.0. Washington, DC, USA: US Department of Defense.

Wells, G.D., and A.P. Sage. 2008. "Engineering of a System of Systems," in *Systems of Systems Engineering - Principles and Applications.* Boca Raton, FL, USA: CRC Press.

Wojcik, L.A., and K.C. Hoffman. 2006. "Systems of Systems Engineering in the Enterprise Context: A Unifying Framework for Dynamics." Proceedings of the IEEE International Conference on System of Systems Engineering, April 24-26, 2006, Los Angeles, CA, USA.

Zachmann, J. 1987. "A framework for information systems architecture." *IBM Systems Journal*. 26 (3).

Zeigler, B.P., T.G. Kim, and H. Praehofer. 2000. *Theory of Modeling and Simulation*. New York, NY, USA: Academic Press.

Primary References

Chen, D., G. Doumeingts, F. Vernadat. 2008. "Architectures for Enterprise Integration and Interoperability: Past, Present and Future." *Comput.Ind*. 59 (7):647-659.

Maier, M.W. 1998. "Architecting Principles for Systems-of-Systems." *Systems Engineering.* 1 (4): 267-284.

Additional References

Dickerson, C.E., S.M. Soules, M.R. Sabins, and P.H. Charles. 2004. *Using Architectures for Research, Development, and Acquisition*. Washington, DC, USA: Office of the Assistant Secretary of The Navy (Research Development And Acquisition). ADA427961. Available: http://handle.dtic.mil/100.2/ADA427961.

European Commission. 2010. "Annex to the Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of Region," in *Towards interoperability for European public services.* Available: http://ec.eupora.eu/isa/strategy/doc/annex_ii_eif_en.pdf

Giachetti, R.E. 2010. *Design of Enterprise Systems, Theory, Architecture, and Methods*. Boca Raton, FL, USA: CRC Press.

Rhodes, D.H., A.M. Ross, and D.J. Nightingale. 2010. "Architecting the System of Systems Enterprise: Enabling Constructs and Methods from the Field of Engineering Systems." IEEE International Systems Conference, March 23-26, 2009, Vancouver, Canada.

MITRE. 2012. "Architectures Federation," in *Systems Engineering Guide.* Bedford, MA, USA: MITRE Corporation. Accessed September 11, 2012. Available: http:/ / www. mitre. org/ work/ systems_engineering/ guide/ enterprise_engineering/engineering_info_intensive_enterprises/architectures_federation.html.

Mittal, S. 2000. "Extending DoDAF to Allow DEVS-Based Modeling and Simulation." *Journal of Defense Modeling and Simulation (JDMS).* 3 (2).

Valerdi R., E. Axelband, T. Baehren, B. Boehm, and D. Dorenbos. 2008. "A research agenda for systems of systems architecting." *International Journal of System of Systems Engineering*. 1 (1-2): 171-188.

Zeigler, B.P., D. Fulton, P. Hammonds, and J. Nutaro. 2000. "Framework for M&S–Based System Development and Testing in a Net-Centric Environment." *ITEA Journal of Test and Evaluation*. 26 (3): 21-34.

< Previous Article | Parent Article | Next Article >

SEBoK v. 2.1, released 31 October 2019

Socio-Technical Features of Systems of Systems

Lead Authors: *Judith Dawson, Mike Henshaw, Bud Lawson*, **Contributing Authors:** *Heidi Davidz, Alan Faisandier*

In perhaps the earliest reference to Systems of Systems (SoS), Ackoff (1971) describes a concept that is mostly concerned with organizations, i.e. social. However, this section is concerned with the socio-technical aspects of technical SoS, which are composed of interdependent resources, such as, people, processes, information, and technology that interact with each other and with their environment in support of a common mission (glossary).

The Socio-Technical Nature of Systems of Systems

Rebovich (2009) [has captured the essence of the SoS problem as:

"From a single-system community's perspective, its part of the SoS capability represents additional obligations, constraints and complexities. Rarely is participation in an (sic) SoS seen as a net gain from the viewpoint of single-system stakeholders."

Three of the persistent SoS challenges, or pain points, identified by Dahmann (2015) are directly related to this problem of stakeholder perspective and the local optimization of constituent system performance at the expense, or to the detriment of, the overall SoS performance. These are: SoS Authority, Leadership, and Autonomy, Interdependencies & Emergence. Thus, the sociological aspects affecting decision making and human behaviors must be given similar weight to the technical aspects of SoS.

Turning to views outside of Systems Engineering, Ergonomists regard socio-technical systems as having the following characteristics (Maguire, 2014):

- There are collective operational tasks,
- They contain social and technical sub-systems,
- They are open systems (i.e. strongly interacting with their environments), and
- The concept of the system being an unfinished system.

These are also characteristics of Systems of Systems. Klein (2014) has noted that approaches to socio-technical systems can take the two perspectives of "system affects people" or "people affect system", depending upon how the system boundary is drawn. It is generally true for systems that consideration of their context requires socio-technical aspects to be taken into account.

Although focused largely on IT systems, Baxter and Sommerville (2011) have noted that the introduction of new business SoS are generally carried out in conjunction with a change process. They argue that frequently the social and organizational aspects are disruptive and that inadequate attention is paid to the connection between change processes and systems development processes. They propose two types of Socio-Technical Systems Engineering activities:

- Sensitizing and awareness activities, designed to sensitize stakeholders to the concerns of other stakeholders.
- Constructive engagement activities, which are largely concerned with deriving requirements accurately and meaningfully.

The extent to which these activities can be effective may be challenged by independent management or operation of constituent systems in a SoS.

Although there are many matters concerning the socio-technical aspects of SoS, there are two important issues, that are dealt with her. The first is the need for appropriate governance structures, given that operational and/or managerial independence affects top-down direction of the SoS and may compromise achievement of the SoS goal(s). The second issue is a lack of situational awareness of managers, operators, or other stakeholders of the SoS, so that they may not understand the impact of their local decisions on the wider SoS.

SoS Governance

Generally, design and operation of complex systems is concerned with control, but the classification of SoS (Dahmann, et. al., 2008) is based on the notion of diminishing central control, as the types go from directed to virtual. Sauser, et. al. (2009) has described the 'control paradox of SoS' and asserted that for SoS, 'management' is replaced by 'governance'. 'Control is a function of rules, time, and bandwidth; whereas command is a function of trusts, influence, fidelity, and agility'.

Some practitioners have found the Cynefin framework, developed by David Snowden, helpful in understanding the nature of complexity that may arise in SoS. Developed from knowledge management considerations, Kurtz and Snowden (2003) propose three reasons why the behavior of systems involving people may be difficult to predict. Firstly, humans are not limited to one identity, and so modelling human behaviors using norms may not be reliable. Secondly, humans are not limited to acting in accordance with predetermined rules. Thirdly, humans are not limited to acting on local patterns. These reasons all undermine control, so that the sociological aspects of SoS make their behaviours hard to predict and, possibly indeterminate. The Cynefin framework considers systems to be classified in four domains:

- Known – simple systems with predictable and repeatable cause and effect
- Knowable – amenable to systems thinking and analytical/reductionist methods
- Complex – adaptive systems where cause and effect are only discernable in retrospect and do not repeat
- Chaotic – no cause and effect relationships are perceivable

The different types of SoS (directed, acknowledged, collaborative, and virtual) could all be described in any of the above domains, depending on many factors internal to the SoS, but in all cases it is the sociological element of the socio-technical SoS that is most likely to give rise to ambiguity in predicting behavior.

A major governance issue for SoS is understanding the ownership of, and making reliable estimates of risk (Fovino & Masera, 2007). High levels of connectivity, and the potential for emergent behavior due to the interactions of separately owned/operated constituent systems, means that significant risks may go unacknowledged and their mitigations unplanned.

In general, governance can be summed up by asking three connected questions (Siemieniuch and Sinclair, 2014):

- Are we doing the right things (leadership)?
- Are we doing those things right (management)?
- How do we know this (metrics and measurements)?

Currently, there is no accepted framework for addressing these questions in a SoS context, but Henshaw et. al. (2013) highlighted architectures as an important means through which governance may be clarified. They postulate that a SoS can be regarded as a set of trust and contract relationships between systems (i.e. including both informal and formal relationships). The systems architect of a constituent system must, therefore, address trust issues for each participating organization in the overall enterprise with which his/her system must interoperate. For SoS, technical engineering governance is concerned with defining and ensuring compliance with trust at the interface between constituent systems. An example of difficulty managing the interfaces in a SoS is provided in the Cassini-Huygens mission case study .

Situational Awareness

Situational awareness is a decision maker's understanding of the environment in which he/she takes a decision; it concerns information, awareness, perception, and cognition. Endsley (1995) emphasizes that situational awareness is a state of knowledge. There are numerous examples of SoS failure due to the operator of one constituent system making decisions based on inadequate knowledge of the overall SoS (big picture).

On the other hand, SoS development is also viewed as the means through which improved situational awareness may be achieved (Van der Laar, et. al., 2013). In the defense environment, Network Enabled Capability (NEC) was a system of systems approach motivated by the objective of making better use of information sharing to achieve military objectives. NEC was predicated on the ability to share useful information effectively among the stakeholders that need it. It is concluded that improving situational awareness will improve SoS performance, or at least reduce the risk of failures at the SoS level. Thus, the principles which govern the organization of the SoS should support sharing information effectively across the network; in essence, ensuring that every level of the interoperability spectrum is adequately serviced. Operators need insight into the effect that their own local decisions may have on the changing SoS or environment; similarly they need to understand how external changes will affect the systems that they own.

Increasingly, SoS include constituent systems with high levels of autonomous decision making ability, a class of system that can be described as cyber-physical systems (of systems). The relationship to SoS is described by Henshaw (2016). Issues arise because autonomy can degrade human situational awareness regarding the behavior of the SoS, and also the autonomous systems within the SoS have inadequate situational awareness due to a lack of competent models of humans (Sowe, 2016)

References

Works Cited

Ackoff, R.L.(1971) "Towards a Systems of Systems Concepts," *Manage. Sci.*, vol. 17, no. 11, pp. 661–671.

Baxter, G. and I. Sommerville, (2011) "Socio-technical systems: From design methods to systems engineering," Interact. Comput., vol. 23, no. 1, pp. 4–17.

Dahmann, J. S. & Baldwin, K. J. (2008) Understanding the Current State of US Defense Systems of Systems and the Implications for Systems Engineering, 2nd Annual IEEE Systems Conference, 1–7. http://doi.org/10.1109/SYSTEMS.2008.4518994

Dahmann, J.S. (2015) "Systems of Systems Characterization and Types," in Systems of Systems Engineering for NATO Defence Applications (STO-EN-SCI-276), pp. 1–14.]

Endsley, M. R. (1995) Toward a Theory of Situation Awareness in Dynamic Systems, J. Human Factors and Ergonomics Soc., 37(1), 32–64. http://doi.org/10.1518/001872095779049543

Fovino, I. N., & Masera, M. (2007) Emergent disservices in interdependent systems and system-of-systems, in Proc. IEEE International Conference on Systems, Man and Cybernetics, Vol. 1, pp. 590–595. http://doi.org/10.1109/ICSMC.2006.384449

Henshaw, M. J. de C., Siemieniuch, C. E., & Sinclair, M. A. (2013) Technical and Engineering Governance in the Context of Systems of Systems, in NATO SCI Symp. Architecture Assessment for NEC (pp. 1–10). Tallinn, Es. NATO STO.

Henshaw, M. (2014) A Socio-Technical Perspective on SoSE, in Lecture Series in Systems of Systems Engineering for NATO Defence Applications (SCI-276). NATO CSO.

Henshaw, M. (2016). Systems of Systems, Cyber-Physical Systems, The Internet-of-Things…Whatever Next? INSIGHT, 19(3), pp.51–54.

Klein, L. (2014) What do we actually mean by 'sociotechnical'? On values, boundaries and the problems of language, Appl. Ergon., vol. 45, no. 2 PA, pp. 137–142.

Kurtz, C.F. and D. J. Snowden (2003) "The New Dynamics of Strategy: Sense-making in a Complex-Complicated World," IBM Syst. J., vol. 42, no. 3, pp. 462–483.

Maguire, M. (2014) Socio-technical systems and interaction design - 21st century relevance, Appl. Ergon., vol. 45, no. 2 PA, pp. 162–170.:/mil/ [1]

Rebovich, G. (2009) "Enterprise systems of Systems," in Systems of Systems Engineering - Principles and Applications, M. Jamshidi, Ed. Boca Raton: CRC Press, pp. 165–191.

Sauser, B., Boardman, J., & Gorod, A. (2009) System of Systems Management, in System of Systems Engineering: Innovations for the 21st Century, M. Jamshidi (Ed.), (pp. 191–217) Wiley.

Siemieniuch, C.E. & Sinclair, M.A. (2014) Extending systems ergonomics thinking to accommodate the socio-technical issues of Systems of Systems, Appl. Ergon., V 45, Issue 1, Pages 85-98

Sowe, S.K. et al. (2016) Cyber-Physical-Human Systems - putting people in the loop. IT Professional, 18(February), pp.10–13.]

Van der Laar, P., Tretmans, J., & Borth, M. (2013) Situational Awareness with Systems of Systems. Springer.

Primary References

Checkland, P.B. 1981. Systems Thinking, Systems Practice. Chichester, West Sussex, England, UK: John Wiley & Sons, Ltd.

Additional References

Bruesburg, A., and G. Fletcher. 2009. *The Human View Handbook for MODAF*, draft version 2, second issue. Bristol, England, UK: Systems Engineering & Assessment Ltd. Available: http://www.hfidtc.com/research/process/reports/phase-2/hv-handbook-issue2-draft.pdf.

IFIP-IFAC Task Force. 1999. "The Generalised Enterprise Reference Architecture and Methodology," V1.6.3. Available: http://www.cit.gu.edu.au/~bernus/taskforce/geram/versions/geram1-6-3/v1.6.3.html.

ISO. 1998. ISO 14258:1998, *Industrial automation systems — Concepts and rules for enterprise models.* Geneva, Switzerland: International Organization for Standardization.

ISO. 2006. ISO 19439:2006, *Enterprise integration — Framework for enterprise modelling.* Geneva, Switzerland: International Organization for Standardization.

ISO. 2007. ISO 19440:2007, *Enterprise integration — Constructs for enterprise modelling.* Geneva, Switzerland: International Organization for Standardization.

Miller, F.P., A.F. Vandome, and J. McBrewster. 2009. *Enterprise Modelling.* Mauritius: Alphascript Publishing, VDM Verlag Dr. Müller GmbH & Co. KG.

< Previous Article | Parent Article | Next Article >

SEBoK v. 2.1, released 31 October 2019

References

[1] http://www.dtic.mil/dtic/tr/fulltext/u2/a468785.pdf

Capability Engineering

Lead Authors: *Judith Dahmann, Bud Lawson, Mike Henshaw*

Capability is increasingly being used to describe the Systems Engineering of Operational Capabilities. The INCOSE UK Capability System Engineering Guide (Kemp and Daw) built on this analysis and describes:

- That Capabilities are realised through a combination of people, processes, information as well as equipment;
- They are concerned with delivering outcomes, rather than outputs;
- They are enduring, with capabilities being upgraded rather than replaced;The term emerged in defence in the early 2000, however the concepts go back far earlier (Checkland, 1997).
- The concepts of Capability Systems Engineering have been used in Rail (Dogan, 2012) and Healthcare (Royal Academy of Engineering, 2017).

is widely used across many industrial sectors and has begun to take on various specific meanings across, and even within, those sectors. Terms such as capability-based acquisition, capability engineering and management, life capability management, capability sponsor, etc. are now ubiquitous in defense and elsewhere. Henshaw et al. (2011) have identified at least eight worldviews of capability and capability engineering and concluded that the task of capability engineering is not consistently defined across the different communities.

The aim of capability systems engineering is to ensure that the upgraded capability meets stakeholders needs. Good Capability Systems Engineering provides a clear line of sight from the purpose of the capability, through the operational concept and whole system design down to specific requirements and interfaces (Figure 1)

[Add Figure]

Capability engineering is concerned with the whole lifecycle (Figure 2); the “Fuzzy front end” of capability trade-offs, the conventional ‘V’ product lifecycle, and the “Messy in-service” support phase.

[Add figure]

Capability Systems Engineering uses standard SE tools, applied from the perspective of the asset owner-operators (i.e. the military user or rail transportation provider).

Kemp and Daw (2014) note several differences between Capability Systems Engineering and the more traditional product Systems Engineering:

- Using persuasion and influence as much as command and control to implement decisions
- Building in flexibility where possible, as the capability will change.
- Implementing the transition to the improved capability as both an engineering and cultural change.
- Recognising that capabilities are often Complex Adaptive Systems. As the capability improves, users or competitors change their behaviour, reducing the effectiveness of the capability
- Capability trade-offs are not about simple comparisons, between similar things – often they are choices between new equipment, better training or new processes.

There is a strong relationship between Capability Engineering and system of systems (SoS).. To some a Capability is a type of system/SoS, to others it is what the system/SoS does. This is explored in Henshaw et al. (2011), who describe at least eight worldviews of capability and capability engineering

Services View of SoSE

As it has been discussed throughout the Systems of Systems (SoS) knowledge area, a ‘system of systems’ is typically approached from the viewpoint of bringing together multiple systems to provide broader capability. As is discussed in Architecting Approaches for Systems of Systems, the networking of the constituent systems in a SoS is often a key part of an SoS. In some circumstances, the entire content of a SoS is information and the SoS brings together multiple information systems to support the information needs of a broader community. These ‘information technology

information technology

information technology

information technology (IT)-based’ SoSs have the same set of characteristics of other SoSs and face many of the same challenges. Currently, IT has adopted a ‘services’ view of this type of SoS and increasingly applies a International Organization for Standaradization (ISO) 20000 series (Information technology -- Service management) or Information Technology Infrastructure Library (ITIL) v. 3 (OGC 2009) based approach to the design and management of information-based SoS. A service perspective simplifies SoSE as it:

- is a more natural way for users to interact with and understand a SoS,
- allows designers to design specific services to meet defined performance and effectiveness targets, and
- enables specific service levels to be tested and monitored through life.

Although it has not been proven to be universally applicable, the services view works well in both IT and transportation SoS.

References

Works Cited

Henshaw, M., D. Kemp, P. Lister, A. Daw, A. Harding, A. Farncombe, and M. Touchin. 2011. "Capability Engineering - An Analysis of Perspectives." Presented at International Council on Systems Engineering (INCOSE) 21st International Symposium, June 20-23, 2011, Denver, CO, USA.

Checkland P. and Holwell, S, 1997, Information, Systems and Information Systems: Making Sense of the Field

Dogan, H., Henshaw, M. and Johnson, J., 2012. An incremental hybridisation of heterogeneous case studies to develop an ontology for Capability Engineering. In: INCOSE, ed. The 22nd Annual INCOSE International Symposium 9-12 July 2012 Rome, Italy.

Royal Academy of Engineering, 2017, Engineering better care a systems approach to health and care design and continuous improvement

Erl, T. 2008. *SOA Principles of Service Design.* Boston, MA, USA: Prentice Hall Pearson Education.

Hitchins, D.K. 2003. *Advanced Systems Thinking, Engineering and Management*. Norwood, MA, USA: Artech House, Inc.

OGC (Office of Government Commerce). 2009. *ITIL Lifecycle Publication Suite Books.* London, UK: The Stationery Office.

Primary References

Henshaw, M., D. Kemp, P. Lister, A. Daw, A. Harding, A. Farncombe, and M. Touchin. 2011. "Capability Engineering - An Analysis of Perspectives." Presented at International Council on Systems Engineering (INCOSE) 21st International Symposium, June 20-23, 2011, Denver, CO, USA.

Kemp D., Daw A. 2014, INCOSE UK Capability Systems Engineering Guide

Additional References

Davies, J.K. 2011. *Survey of Background Literature for Capability Engineering*. INCOSE UK Capability Working Group Report.

< Previous Article | Parent Article | Next Article >

SEBoK v. 2.1, released 31 October 2019

Knowledge Area: Healthcare Systems Engineering

Healthcare Systems Engineering

This article provides an overview of the role of systems engineering in the engineering or re-engineering of healthcare systems to meet a number of modern day challenges. The role of SE in medical devices, healthcare IT, pharmaceuticals, and public health systems are considered and contrasted to "traditional" SE practices discussed elsewhere in the SEBoK. See Overview of the Healthcare Sector for details of the stakeholders and constraints of the these different parts of the sector.

Topics

Each part of the SEBoK is divided into knowledge areas (KAs), which are groupings of information with a related theme. The KAs in turn are divided into topics. This KA contains the following topics:

- Overview of the Healthcare Sector
- Systems Engineering in Healthcare Delivery
- Systems Biology
- Lean in Healthcare

Healthcare and Systems Engineering

Healthcare today faces many challenges related to safety (e.g. Hospital Safety Score 2013, Andel et al. 2012, Institute of Medicine 1999), affordability, access, and the means for reliably producing positive outcomes for all patients of all ages and across all care environments. Furthermore, the health of individuals is challenged by many threats such as environmental and behavioral norms, emerging natural infectious diseases, and acute and chronic conditions that are becoming more prevalent because of longer lifespans. Re-engineering today's healthcare to address these challenges requires a systems approach – an approach that develops solutions to contend with the complexity of healthcare-related policy, economics, social dynamics, and technology. Systems Engineers are trained to grapple with this kind of complexity by thinking holistically and to work with trans-disciplinary teams to develop solutions making re-engineering healthcare a natural fit for systems engineers and the tools of systems engineering.

The disciplines involved in re-engineering healthcare are far reaching across academia, government, industry, private, and public sectors including the patients and families the healthcare field serves. Systems Engineers involved in this re-engineering draw on several tools when working with these stakeholders to develop solutions. In doing so they follow the general systems principles described in the Systems Approach Applied to Engineered Systems knowledge area in SEBoK Part 2. First, with so many diverse stakeholders involved in this field, it is vitally important for the Systems Engineer to help clarify the problem or opportunity and to conceive of the objective of the re-engineering. They need to "envision the solution" without being entirely prescriptive of the solution's specific implementation, see Identifying and Understanding Problems and Opportunities. Then, drawing from best practices, the Systems Engineer guides the stakeholders through the synthesis of possible solutions and the analysis and selection between alternatives. Systems engineers are also involved in the implementation and testing and the deployment, use and sustainment of healthcare systems to provide stakeholder value. The systems approach in healthcare must be particularly mindful to not exclusively focused on technical aspects of the effort since the solutions to healthcare's challenges exist not only in technical areas but the integration of culture, workflow and processes, with technology as a tool to support the delivery of safe, affordable, and accessible care.

To achieve this system approach healthcare projects follow a version of the SE life cycle described in SEBoK Part 3. This included the creation of Stakeholder and System Requirements, Systems Architecture and Design and System Integration, Verification and Validation. The SE life cycle extends to include System Deployment, Operation, Maintenance and Logistics. Healthcare project will also follow some of the Systems Engineering Management processes described in Part 3.

It is vitally important for the healthcare systems engineer to ensure socio-technical integration and interoperability among system components are part of any project – the last thing healthcare needs is another standalone innovation that perpetuates the silos that exist in the field today. Remaining focused on the objective and problem to solve, managing scope creep, disciplined design, implementation, and project management are key activities the Systems Engineer is responsible for in healthcare systems engineering.

Systems Engineering for Medical Device Development

Systems Engineering for medical device development is essentially an application of Product Systems Engineering as described in SEBoK Part 4 with a few customizations:

- The life cycle has to comply with specific healthcare regulations, which constrain aspects of the life cycle, as exemplified by FDA regulations in the US (21CFR 820.30)
- The products are market driven, with little customization allowed by the manufacturer at the customer site
- The markets are midsized, with the market for a given technology or product line often being in the $1-10B range
- Medical device development programs are mid-sized…many from 10-100 man-years of development, lasting 1-2 years
- Time to market is critical, with the first mover or first with a complete solution capturing the majority of the profits
- Most products are cyber-physical, with software becoming a larger part of the product. Many products include significant aspects of physiology or chemistry
- There is a special tension between “efficacy” and “safety”. Efficacy requires the vast majority to be helped. Safety is compromised if only a very small minority is adversely affected. Truly safe systems require a special approach to systems engineering . (Leveson 2011)
- Customer feedback may be constrained by safety issues as well as HIPAA regulations

Device Development in a Market Environment

One critical difference between many “traditional” systems engineering industries (defense and aerospace) and healthcare device development is that most healthcare device development is market driven, rather than contract driven. Some key differences between market and contract systems engineering:

- The program size (budget) and dates are not ‘fixed’, they are set by the business leadership designed to maximize return on investment across a portfolio of product programs
- Program scope and requirements are not fixed externally; they can be changed fairly rapidly by negotiation between functions and the executive committee.
- The goal for the product development isn’t necessarily a feature set, it is a market share and price premium relative to the competition…which can be a moving target. A competitive announcement will often force a change in the program scope
- In a contract based program there is an identified customer, with a set of applications and workflows. In a market driven program the workflow and use cases are defined by the developer, and the buyer needs to ‘own’ the integration of the offering into their specific systems and workflows.
- For specific medical products the FDA can require pre-market trials and post market studies . (FDA 2014)

- The different types of healthcare reimbursement across the world (universal coverage private insurance, national single provider, national single payer, private insurance, and out of pocket) creates dramatically different market dynamics (for individuals, healthcare providers, and product developers) . (Reid 2010)

Regulations for Medical Device Development

As with all regulated products, there are many regulations governing the development of medical devices. The medical device industry specific regulations are primarily driven by the US (FDA), Europe (European Commission), and Canada (Health Canada). Within the US, the FDA governs medical devices primarily through 21 CFR 820.30 (Quality Systems Regulation, Subpart C Design Controls) , which contains requirements similar to ISO 13485. The sections of the Quality Systems Regulation for Design Controls can be mapped fairly directly to ISO/IEC/IEEE 15288 (2015) and the INCOSE *SE Handbook* (INCOSE 2015).

Table 1. Comparison of Healthcare Safety Regulations with ISO/IEC/IEEE 15288 and the INCOSE *SE Handbook*.

21CFR820.30	ISO/IEC/IEEE 15288:2015	INCOSE *SE Handbook*
(b) Design and development planning	6.3.1 Project Planning Process	5.1 Project Planning Process
(c) Design input.	6.4.2 Stakeholder needs and requirements definition process 6.4.3 Systems requirements definition process	4.2 Stakeholder needs and requirements definition process 4.3 Systems requirements definition process
(d) Design output	6.4.5 Design definition process 6.4.7 Implementation process	4.5 Design definition process 4.7 Implementation process
(e) Design review	6.3.2 Project Assessment and Control process	5.2 Project Assessment and Control process
(f) Design verification	6.4.9 Verification Process	4.9 Verification Process
(g) Design validation	6.4.11 Validation Process	4.11 Validation Process
(h) Design transfer	6.4.10 Transition Process	4.10 Transition Process
(i) Design changes	6.3.5 Configuration Management Process 6.4.13 Maintenance Process	5.5 Configuration Management Process 4.13 Maintenance Process
(j) Design history file	6.2.6 Knowledge Management Process	5.6 Information Management Process

In the biomedical and healthcare environment, an important differentiator in Risk Management activities compared to other industries (see Risk Management) is that the users and patients are the center of risk analysis rather than technical or business risks. Risk management is an important element of the design control process, as preliminary hazard analysis drive initial design inputs. Traceability between identified risks, risk mitigations, design inputs, and design outputs is a key factor in product clearance through regulatory agencies. Most regulatory bodies have recognized ISO 14971: Medical devices -- Application of risk management to medical devices as a methodology for assessing and documenting product safety and effectiveness.

Usability Engineering is an important subset of risk management activities. ISO 62366-1 Medical devices – Part 1: Application of usability engineering to medical devices provides a “process for a manufacturer to analyze, specify, develop and evaluate the usability of a medical device as it relates to safety. This usability engineering (human factors engineering) process permits the manufacturer to assess and mitigate risks associated with correct use and use errors, i.e., normal use.“ . (IEC 62366-2015) For example, for a device designed for home care use, there are many complex interfaces that product designers must consider. Patients may be physically or cognitively affected (age, medication, injury, etc.); they may be untrained or cared for people who are untrained; they are not professionals used to technical systems, etc. Even in the hospital setting, untrained patients may have physical access to systems. This puts a critical focus on usability and human factors considerations and the complexity of the use environment.

Further, as medical devices incorporate more software and become cyber-physical devices, the regulators are also focusing on privacy and security (ISO 21827) and software life cycle management (ISO 62304).

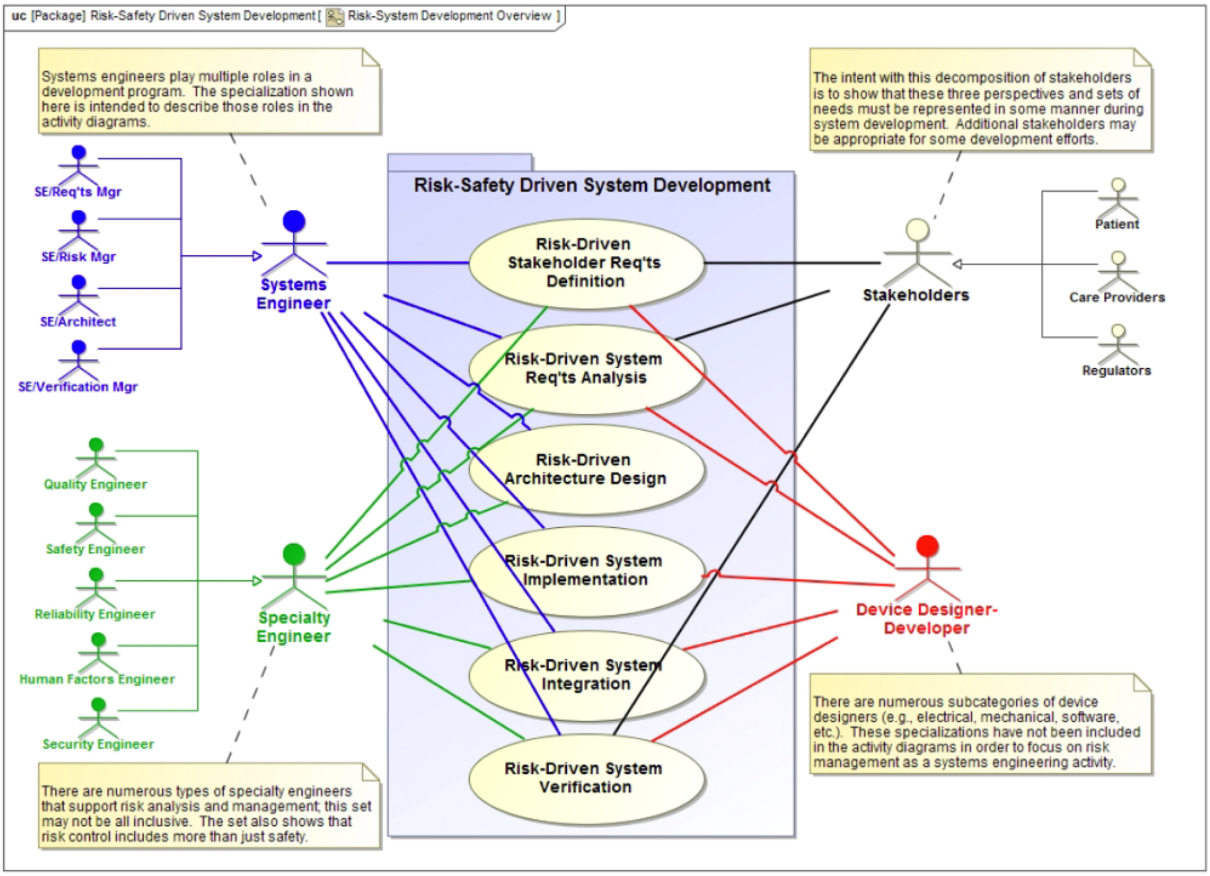


Figure 1 Overlap of Regulations and Standards for Medical Device Development. (Modified from (Malins et al. 2015). Used with Permission. All other rights are reserved by the copyright owner.)

Medical Device regulations, guidance, and technical standards are constantly changing, adding a complex dynamic to manage and incorporate throughout the product development life cycle.

Systems Engineering for Healthcare IT

Systems Engineering for Healthcare Information Technology is very similar to other IT developments, with the addition of medical regulations. Healthcare Information Technology is critical to efficient flow of information and delivery of services . (Presidents Council of Advisors on Science and Technology 2010) The product development is a mix of contract driven development (with a target customer, such as healthcare.gov), and market driven (where there are more standard products, with minimal customization). Much of the market, especially for hospitals and hospital chains, is a mix of standard products with large amounts of customization to the customer’s specific needs, terminology, and workflows.

Systems Engineering for Pharmaceuticals

The pharmaceutical industry leverages systems that include hardware, software and sometimes single-use components in different part of their value chain, for example complex analytical systems during drug discovery, complex bioreactors and downstream filtration and chromatography systems in manufacturing and drug delivery devices for the use of their drugs. These systems are subject to very different regulations, e.g. GMP or medical devices, depending on the use. One challenging aspect of these systems is that the users have different skill sets and working under different environments. And in all of the examples below, biological and/or chemical processes run on these systems, requiring deep domain knowledge of the system development teams.

The in-vitro diagnostic industry also uses many systems, small devices (e.g. self-testing blood glucose or coagulation monitoring systems) all the way to large, fully automated, high throughput systems for the use in centralized laboratories. Very often, these systems operate as a closed system, so that the reagents used for the diagnostics tests, are proprietary and the vendor of the system only guarantees high quality results only when using the proprietary chemistry. This enables the vendors to often 'place' the instruments as highly competitive prices when the actual profit is generated through the consumables.

For the chemistry part of pharma, understanding the scientific method, using a systems thinking approach, and using six sigma approaches to managing variation and interdependencies is critical. Once you create a product which includes software and physical parts (including manufacturing equipment), systems engineering of the functional design, design analysis, and integration and verification of the solution become critical.

Systems Challenges for Public Health

Summits and inquiries into problems or shortcomings in the public health space have consistently uncovered the same issues: systemic failures in the way that public health is approached that make it nearly impossible to adequately respond to major health events. Examples can be seen from the US response to Hurricane Katrina (e.g. The White House 2006), the 2011 Thoku tsunami (e.g. Carafano 2011, The Heritage Foundation 2012), or even the 2014-2015 Ebola outbreak in West Africa (e.g. GHTC 2015).

The White House report provides insights into just a few of potential challenges for the health aspects of disasters or large-scale emergencies (2006, Chapter 6):

- Tens of thousands of people may require medical care.
- Large portions of a population with chronic medical conditions may themselves without access to their usual medications and sources of medical care.
- Hospitals and other healthcare facilities may be totally destroyed or otherwise rendered inoperable and the area's health care infrastructure may sustain extraordinary damage.

The types of public health challenges will also change over time: Immediate challenges include the identification, triage and treatment of acutely sick and injured patients; the management of chronic medical conditions in large numbers of evacuees with special health care needs; the assessment, communication and mitigation of public health risk; and the provision of assistance to State and local health officials to quickly reestablish health care delivery systems and public health infrastructures. (The White House 2006) As time passes, longer-term infectious disease outbreaks may occur or environmental impacts may cause health risks (e.g. Fukushima nuclear meltdown after the 2010 tsunami). And over time, the public health and overall healthcare infrastructure must be re-established and repaired.

But the public health "system" in most countries, as currently structured, is not prepared to deal with these types of challenges. In talking about the US, Salinsky and Gursky state, "Despite recent attention to the biodefense role of public health, policymakers have not developed a clear, realistic vision for the structure and functionality of the governmental public health system. Lack of leadership and organizational disconnects across levels of government have prevented strategic alignment of resources and undermined momentum for meaningful change. A transformed public health system is needed to address the demands of emergency preparedness and health protection. … The future public health system cannot afford to be dictated by outmoded tools, unworkable structures, and outdated staffing models." (2006)

The framing of the challenge as a systems one requires the application of a systems approach, and the use of tools capable of supporting systems views, to enable better understanding of the challenges for public health and for creating ways to address these challenges. The SEBoK knowledge areas on Enterprise Systems Engineering and Systems of Systems (SoS) at least partially consider some of these challenges from a systems engineering perspective.

Systems Biology for Healthcare

As systems science is a foundation for system engineering, systems biology is becoming recognized as a foundational discipline for healthcare systems engineering. Systems biology is an emerging discipline and is recognized as strategically important when tackling complex healthcare problems. The development of systems biology is also an emerging environment for systems engineers.

According to Harvard University, "Systems biology is the study of systems of biological components, which may be molecules, cells, organisms or entire species. Living systems are dynamic and complex and their behavior may be hard to predict from the properties of individual parts. To study them, we use quantitative measurements of the behavior of groups of interacting components, systematic measurement technologies such as genomics, bioinformatics and proteomics, and mathematical and computational models to describe and predict dynamical behavior. Systems problems are emerging as central to all areas of biology and medicine." (Harvard University 2010)

As systems biology matures, its integration into healthcare approaches is expected to lead to advanced practices such as personalized and connected healthcare and the resolution of complex diseases.

Conclusion

While systems engineering practices apply to the healthcare domain, they face different challenges than other industries and need to be tailored. In fact, different segments of the healthcare industry can take significantly different approaches to effective systems engineering and systems thinking.

References

Works Cited

21 CFR 820.30. "Part 820 – Quality System Regulation: Subpart C – Design Controls." *Title 21 – Food and Drugs: Chapter I – Food and Drug Administration, Department of Health and Human Services: Subchapter H – Medical Devices*. Available at: https://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfCFR/CFRSearch.cfm?fr=820.30

Andel, C., S.L. Davidow, M. Hollander, D.A. Moreno. 2012. "The economics of health care quality and medical errors." *Journal of Health Care Finance*. 39(1):39:50. Abstract available at: http://www.ncbi.nlm.nih.gov/pubmed/23155743 200,000 Americans die from preventable medical errors including facility-acquired conditions and millions may experience errors. In 2008, medical errors cost the United States $19.5 billion.

Carafono, J.J. 2011. *The Great Eastern Japan Earthquake: Assessing Disaster Response and Lessons for the U.S.* Washington, DC: The Heritage Foundation. Special Report #94 for Japan. May 25, 2011.

FDA. 2014. "Premarket Approval (PMA)". Washington, DC: U.S. Food and Drug Administration (FDA). Accessed February 17, 2016. Available at: http://www.fda.gov/Medicaldevices/Deviceregulationandguidance/Howtomarketyourdevice/Premarketsubmissions/Premarketapprovalpma/Default.Htm

GHTC. 2015. "Will We Learn from the Lessons of the Ebola Outbreak?" 2015 Policy Report. Washington, DC: Global Health Technologies Coalition (GHTC).

Gursky, E. 2005. *Epidemic Proportions: Building National Public Health Capabilities to Meeting National Security Threats*. Arlington, VA: Analytic Services Inc. (ANSER).

Harvard University. 2010. "Department of Systems Biology." Cambridge, MA: Harvard University. Accessed February 17, 2016. Available at: https://sysbio.med.harvard.edu/

Hospital Safety Score. 2013. "Hospital Errors are the Third Leading Cause of Death in U.S., and New Hospital Safety Scores Show Improvements are Too Slow." Washington, DC: The LeapFrog Group. Accessed February 17, 2016. Available at: http://www.hospitalsafetyscore.org/newsroom/display/hospitalerrors-thirdleading-causeofdeathinus-improvementstooslow

IEC. 2015. *Medical devices – Part 1: Application of usability engineering to medical devices*. Geneva, Switzerland: International Electrotechnical Commissions. IEC 62366-1:2015. Available at: http:/ / www. iso. org/ iso/ catalogue_detail.htm?csnumber=63179

INCOSE. 2015. "Section 8.2.2." *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, version 4.0. Hoboken, NJ, USA: John Wiley and Sons, Inc, ISBN: 978-1-118-99940-0

Institute of Medicine. 1999. *To Err is Human: Building a Safe Health System*. Washington, DC: The National Academy Press, The National Academy of Sciences. Novermber 1999. Available at: https:/ / iom. nationalacademies. org/ ~/ media/ Files/ Report%20Files/ 1999/ To-Err-is-Human/ To%20Err%20is%20Human%201999%20%20report%20brief.pdf

ISO/IEC/IEEE. 2015. *Systems and Software Engineering -- System Life Cycle Processes*. Geneva, Switzerland: International Organisation for Standardisation / International Electrotechnical Commissions / Institute of Electrical and Electronics Engineers. ISO/IEC/IEEE 15288:2015.

Leveson, N.G. 2011. *Engineering a Safer World: Systems Thinking Applied to Safety*. Cambridge, MA: Massachusetts Institute of Technology (MIT).

Presidential Council of Advisors on Science and Technology. 2010. *Report to the President: Realizing the Full Potential of Health Information Technology to Improve Healthcare for Americans: The Path Forward*. Washington, DC: Presidential Council of Advisors on Science and Technology, The White House. December 2010. Available at: https://www.whitehouse.gov/sites/default/files/microsites/ostp/pcast-health-it-report.pdf.

Reid, T.R. 2010. *The Healing of America: A Global Quest for Better, Cheaper, and Fairer Health Care*. New York, NY: Penguin Books.

Salinsky, E. and E. Gursky. 2006. "The Case for Transforming Governmental Public Health." *Health Affairs*. 25(4). 1017-2018. Available at: http://content.healthaffairs.org/content/25/4/1017.full

The Heritage Foundation. 2012. *One Year Later: Lessons from Recover After the Great Eastern Japan Earthquake*. Washington, DC: The Heritage Foundation. Special Report #108 on Asia and the Pacific. April 26, 2012.

The White House. 2006. *The Federal Response to Hurricane Katrina Lessons Learned*. Washington, DC: The White House. February 2006. Available at http://library.stmarytx.edu/acadlib/edocs/katrinawh.pdf

Primary References

None.

Additional References

None.

< Previous Article | Parent Article | Next Article >

SEBoK v. 2.1, released 31 October 2019

Overview of the Healthcare Sector

Lead Author: *Chris Unger*, ***Contributing Author:*** *Cyrus Hillsman*

This article describes some of the stakeholders of the healthcare sector and the factors which influence the application of systems engineering within it. For an overview of healthcare systems engineering and how it deals with these influences see the Healthcare Systems Engineering article.

The healthcare sector is a complex system made up of people, facilities, laws and regulations. It addresses current health, tries to ensure wellness, treats medical problems; creates new medication and medical devices; manages the health both individuals and populations; and helps determine regulations for safety, privacy, the environment, and healthcare delivery itself.

Stakeholders

There are many types of stakeholders in the healthcare sector. The space covers everyone from the general public – who have a stake in their own health and the health of those around them for issues like infectious disease – to the individual researchers who investigate current healthcare problems. The high-level groups of stakeholders include:

- The general public;
- Healthcare providers (such as doctors, nurses, clinics, and hospitals);
- Payers (such as insurance companies);
- Public health organizations;
- Researchers, scientists, and corporations in the pharmaceutical industry;
- Medical device manufacturers;
- Policy makers (particularly those with interest in public health, healthcare safety or privacy policies);
- Healthcare information technology technicians and organizations; and
- Professional organizations and societies relevant to the various aspects of the space.

The healthcare sector is an enormous area financially as well. For example, out of $2.87 trillion on healthcare spent in the US in 2010, the breakdown of components is:

US Healthcare Expenditures in 2010 (information from Emmanual 2014)

Hospital Care	$921B
Physician Services	$555B
Prescription Drugs	$280B
Nursing Home Care	$151B
Other Medical Products	$113B
Dental Services	$93B
Government Public Health	$84B
Other Professional Services	$79B
Home Health Care	$77B
Research	$48B

The sections below provide insight into the landscape for these the stakeholder groups where there is sufficient information currently available. More detail will be added as the healthcare aspects of the SEBoK mature and the team will take particular care to incorporate additional information from outside the US going forward.

Healthcare Delivery

The largest share of the money spent on healthcare in the US healthcare is in hospitals (almost a third). The number of hospitals has been relatively flat for the last 20 years. However, due to the growing cost pressures and increasing paperwork, there has been a general consolidation of hospitals into chains, and independent physician providers into hospitals or group practices. (Emmanuel 2014)

Overall Hospital Landscape (information from (AHA 2014))

Total Number of All U.S. Registered * Hospitals	5,627
Total Number of U.S. Community ** Hospitals	4,926
Total Number of Nongovernment Not-for-Profit Community Hospitals	2,870
Total Number of Investor-Owned (For-Profit) Community Hospitals	1,053
Total Number of State and Local Government Community Hospitals	1,003
Total Number of Federal Government Hospitals	213
Total Number of Nonfederal Psychiatric Hospitals	403

Hospitals range from small community hospitals to the New York-Presbyterian Hospital/Weill Cornell Medical Center with 2,259 beds (Becker 201)], or the University of Pittsburgh Medical Center Presbyterian with $12B in revenue in 2013. (Becker 2013).

Hospital chains tend to be less than 10 hospitals, with less than 10 chains having more than 10 hospitals (Becker 2015). The largest two have almost 200 hospitals (Community Health Systems with 188 and Hospital Corporation of America with 166). The largest systems by revenue are Kaiser Permanente and the Veterans Health Administration with revenue or budget of slightly over $50B each.

Medical Devices Manufacturers

The medical device development landscape is diverse, composed of many markets of intermediate size (many being above $10B in size, with high single digit to double digit growth rates). Some examples are, with projected market sizes in 2020, are:

Types of Medical Devices and Projected Market Share (Emmanuel 2014)

Medical Device Type	***Projected Market Share***
In-vitro diagnostics (IVD)	$75B
Endoscopy	$33B
Interventional Cardiology	$27B
Infection control	$17B
Minimally invasive surgery	$14B
Defibrillators	$13B
Dental Implants	$10B
Infusion pump	$ 7B
Magnetic Resonance	$ 7B
Digital Xray	$ 5B

As described in Healthcare Systems Engineering, this is the area of the healthcare sector that is most closely aligned with classic product-focused businesses.

Healthcare IT

There is a large uncertainty in what constitutes Healthcare IT. The most visible segment is the Electronic Medical Record (EMR) or Electronic Health Record (EHR), but there is also large markets in billing management, clinical decision support, image management, etc. But there is a divergence of market sizes with estimates around $60B [Bain, FierceIT] and some around $104B [Markets and Markets, MedGadget, and PRNewswire].

An EHR installation at a hospital is similar to an Oracle database installation at a company, where much of the cost is customizing the database and workflows to the institution's policies and workflows, and in training the users to the new system and standardized practices which come with IT and automation.

The top 10 Healthcare IT solution providers in 2015 (information from (Healthcare Informatic 2015))

Company	*2015 Revenue*
Optum	$5.2B
Cerner Corp.	$3.4B
McKesson	$3.1B
Dell	$2.9B
Cognizant	$2.7B
Philips	$2.7B
Xerox	$2.4B
Siemens	$2.0B
Epic Systems Corp.	$1.8B
GE Healthcare	$1.5B

Public Health Systems

The World Health Organization (WHO) defines public health as "all organized measures … to prevent disease, promote health, and prolong life among the population as a whole. Its activities aim to provide conditions in which people can be healthy and focus on entire populations, not on individual patients or diseases. Thus, public health is concerned with the total system and not only the eradication of a particular disease." (WHO 2016) Governments at each level define exactly what "public health" will encompass, but typically there are three areas: epidemiology, provision of health services, and workplace and environmental safety and policy. Epidemiology is the study and control health-related events, including disease. Various methods can be used to carry out epidemiological investigations: surveillance and descriptive studies can be used to study distribution; analytical studies are used to study determinants." (WHO 2016, "Health topics: Epidemiology"). Health services may include services such as preventive vaccinations, disease screening, or well-baby or well-child programs. Environmental safety can include developing policies for automobile or workplace safety, monitoring the quality of drinking water, or even conducting restaurant health inspections. In addition to these wide varieties of work, public health organizations are increasingly expected to be responsible for the health-related aspects of disaster and emergency response efforts.

In the US, the public health "system" is really a patchwork of independent healthcare departments. Each state or territory defines the scope and responsibilities of its own public health "department", requiring information and cooperation from hospitals, private physicians, emergency personnel, laboratory networks, and sometimes public health organizations from other states. (Gursky 2005)

Conclusion

In addition to each group of stakeholders being complex in itself, these stakeholders then interact and work together - or sometimes contradict one another. This makes the landscape of the healthcare systems engineering space itself complex and highlights the need for systems thinking and systems approaches when attempting to address any health-related issues or challenges.

References

Works Cited

AHA. 2014. "Fast Facts on US Hospitals." Chicago, IL: American Hospital Association (AHA). September 2014. Available at: http://www.aha.org/research/rc/stat-studies/fast-facts.shtml

Becker's Healthcare. 2015. "10 largest for-profit hospital systems | 2015". *Becker's Hospital Review*. June 30, 2015. Available at: http://www.beckershospitalreview.com/lists/10-largest-for-profit-hospital-systems-2015.html

Becker's Healthcare. 2014. "100 Largest Hospitals in the US". *Becker's Hospital Review*. August 7, 2014. Available at: http://www.beckershospitalreview.com/lists/8-7-14-100-largest-hospitals-in-america.html

Becker's Healthcare. 2013. "100 Top-Grossing Hospitals in the US". *Becker's Hospital Review*. June 24, 2013. Available at: http://www.beckershospitalreview.com/lists/100-top-grossing-hospitals-in-america-2013.html

Eliades, G., M. Retterath, N. Hueltenschmidt, and K. Singh. 2012. *Healthcare 2020*. Amsterdam, The Netherlands: Bain & Company. Available at: http://www.bain.com/Images/BAIN_BRIEF_Healthcare_2020.pdf.

Emmanuel, E.J. 2014 *Reinventing American Health Care: How the Affordable Care Act will Improve our Terribly Complex, Blatantly Unjust, Outrageously Expensive, Grossly Inefficient, Error Prone System.* New York City, NY: PublicAffairs.

Gold, A. 2014. "Global healthcare IT market projected to hit $66 billion by 2020." FierceHealthIT. April 1, 2014. Available at; http:/ / www. fiercehealthit. com/ story/ global-healthcare-it-market-projected-hit-66-billion-2020/ 2014-04-01.

Gursky, E. 2005. *Epidemic Proportions: Building National Public Health Capabilities to Meeting National Security Threats.* Arlington, VA: Analytic Services Inc. (ANSER). Healthcare Informatics. 2015."2015 HCI 100." Available at: http://www.healthcare-informatics.com/hci100/2015-hci-100-list

Markets and Markets. 2015. *North American Healthcare IT Market by Product (EHR, RIS, PACS, VNA, CPOE, mHealth, Telehealth, Healthcare analytics, Supply Chain Management, Revenue Cycle Management, CRM, Claims Management) by End User (Provider, Payer) - Forecast to 2020.* October 2015. Available at: http:/ / www. marketsandmarkets.com/Market-Reports/north-america-healthcare-it-market-1190.html

MedGadget. 2015. "Global Healthcare IT Market 2020 - Industry Survey, Market Size, Competitive Trends: Radiant Insights, Inc". November 2, 2015. Available at: http:/ / www. medgadget. com/ 2015/ 11/ global-healthcare-it-market-2020-industry-survey-market-size-competitive-trends-radiant-insights-inc.html

PRNewswire. 2015. "Healthcare IT Market Size to Reach $104.5 Billion by 2020: Grand View Research, Inc." October 15, 2015. Available at: http:/ / www. prnewswire. com/ news-releases/ healthcare-it-market-size-to-reach-1045-billion-by-2020-grand-view-research-inc-533012831.html

WHO. 2016. "Health Topics: Epidemiology." Geneva, Switzerland: World Health Organization. Available at: http:// www.who.int/topics/epidemiology/en/

WHO. 2016. "Trade, foreign policy, diplomacy, and health: Public Health." Geneva, Switzerland: World Health Organization. Available at: http://www.who.int/trade/glossary/story076/en/

Primary References

None.

Additional References

None.

< Previous Article | Parent Article | Next Article >

SEBoK v. 2.1, released 31 October 2019

Systems Engineering in Healthcare Delivery

Lead Author: *Cyrus Hillsman*, ***Contributing Authors:*** *Chris Unger, Nicole Hutchison*

The healthcare system is complex and adaptive and confronts significant challenges for which systems engineering tools are useful and necessary. The President's Council of Advisors on Science and Technology (PCAST) prepared a report concluding that healthcare improvement could be accelerated with the use of systems engineering. (PCAST 2014) They noted that they key incentives are wrong (fee for service vs. fee for outcomes), and key enablers are missing (access to useful data, lack accepted systems techniques and people trained in systems engineering)

This article provides an overview of healthcare delivery with some historical context, and describes some different approaches to systems engineering which have been found helpful in addressing healthcare delivery problems.

Human Centered Design

Healthcare delivery is not a product but a service and that makes it different than typical hardware or software design that may be seen in aerospace, defense, or even medical devices. There are three primary factors for these differences. First, quality in services can be difficult to measure objectively. Second, in this service system, care providers are continually making risk, cost, and quality of care decisions at the time of service. Each patient is unique and multiple pathologies and value streams are possible based upon any given patient's needs. Those needs are complemented by a care team that is unique and complex and includes the patients themselves, family support, medical professionals, hospitals, and even the industry in which it resides. Third, if returning to or maintaining wellness is considered to be the core value for a healthcare delivery system, then the patient's behaviors both within and outside any designed care plan has a significant role to play, because roughly half of all healthcare cost is derived from preventable disease. (Conover 2012)

Structure of the Healthcare Delivery Industry

The healthcare industry is large, diverse, and fragmented and this causes considerable complexity. This complexity is experienced both at the macro and micro levels.

At the macro level the healthcare industry is highly fragmented with over 50% of all healthcare workers employed in companies with less than 500 employees. (Griffith & White 2007) Nearly 1 million physicians practice medicine in the US; roughly half of these are in primary care and the rest are in over 30 specialties and many more subspecialties and clinics. In addition to physicians, some 5 million others in some 50 other specialties provide care to patients.

At the micro level, the complexity and pace of change make care difficult. Healthcare is a rapidly developing field with over 700,000 publications produced annually and the pace in fact is increasing. (Smith et al. 2013). That there are already 14,400 codes in the World Health Organization's *International Statistical Classification of Diseases and Related Health Problems* (ICD) complicates the issue. Add to this the regulatory and administrative burden of primary care providers interacting with approximately 200 specialists in any given year and the complexity that care providers face on a daily basis becomes clear.

In short, the healthcare delivery system is itself a complex adaptive system and represents a wicked_problem_, whereby any changes to the system intended to solve an issue will likely create other issues.

Improving Ongoing Operations

As mentioned, above caregivers are faced with many challenges and the goal of in systems engineering in healthcare delivery is to lessen that burden in a systematic way without significant disruption of current operations. To do this successfully requires several factors:

- First, as stated above, systems engineers have to acknowledge that they are dealing with a complex adaptive system that includes many wicked problems. An analogy is that what systems engineers experience in healthcare is like rewiring a house with the power turned on because whatever changes are made are to an existing system that must operate while the changes are being made.
- Second, “the system” in place is difficult to define. The "healthcare system" is actually a combination of many open systems and interdependencies with the system of interest may be unknown.
- Third, patient safety is always a concern and any actions that could affect patient safety must be very carefully considered. Often, "optimizing" a system may introduce a potential risk to patient safety. These system aspects are always in tension.
- Fourth, there is a bias towards the current (known) system versus a change leading to an unknown system. Any change will create a certain amount of disruption to an operational system that may be currently operating at or beyond capacity.
- Fifth, healthcare delivery systems are combinations of patients, providers, process, and products and therefore uncertainty is a daily reality. This level of uncertainty may not be amenable to typical agile approaches of 4-6 week sprints nor traditional waterfall methods.
- Sixth, local factors could play a significant role; therefore no two sites may perform an operation in exactly the same way.
- Seventh, the entire industry acts as a complex adaptive system with multiple intelligent agents working sometimes in partnership and sometimes in conflict with the goals of the system or patient.

Because of these factors and others the tradition of healthcare systems engineering has been to use adaptable human-centered methods. (Checkland 1999)

History of Healthcare Improvement Research

There have been many attempts to understand and improve healthcare both in the public and private domains. Examples include the National Healthcare Service Change Model, the efforts of the Agency for Healthcare Research and Quality, and the Institute for Healthcare Improvement. Here we outline some representative efforts.

Healthcare improvement has been shaped in part by four seminal works by the Institute of Medicine (IOM). *To Err is Human* reported that up to 98,000 patients were killed by healthcare each year. (Kohn, Corrigan, & Donaldson 2000) This put an emphasis on safety as a key quality of care metric. The following year the Institute of Medicine (IOM) broadened the concept of quality beyond safety to include six measures of quality. They determined that healthcare should be safe, effective, patient centered, timely, efficient, and equitable. (Institute of Medicine 2001) This report called Crossing the Quality Chasm included an appendix that documented poor quality and the severity of the issues of under use, over use, and potential for harm in medicine. A search for the underlying reasons for poor quality led to three primary reasons for poor quality. The three reasons were the growing complexity of science and technology; the increase in chronic conditions; and the failure to exploit information technology.

To address these concerns the IOM partnered with the National Academy of Engineering (NAE) to see what could be done from a systems engineering perspective to address the real challenges facing the industry in Building a Better Delivery System. (Compton et al. 2005). That was followed by the realization that standard systems engineering needed to be modified and healthcare was and would remain a human centered endeavor as stated in Best Care at Lower Cost (Smith et al. 2013)

Three Approaches

Although there are many accepted approaches to healthcare systems engineering and improvement, here we outline three that share common characteristics and are representative of most of the other methods.

The first approach is Lean Six Sigma which is a combination of two methods. Lean has its roots in the Toyota Production System (Ohno 1988) and the work of the International Motor Vehicle Program (Womack, Jones, & Roos 1990). Six-Sigma has its roots at Motorola and the work of Bill Smith. These two methods were combined by Michael L. George (see (George 2002) and (George 2003)). It includes techniques like value stream mapping, waste elimination, root cause analysis, and voice of customer. For additional information see Lean Engineering and Lean in Healthcare.

The second approach is based on industrial engineering, which has its roots in the work of Frederick Taylor and others. This approach includes tools such as discrete event simulation, ergonomics, production control, and operations research as shown in Figure 1. For additional information, see Systems Engineering and Industrial Engineering.

Insert Table ES-1 from Building a Better Delivery System here once we obtain the proper permissions.

The third approach is healthcare systems engineering. Traditional systems engineering uses a functional decomposition approach; see for example (Defense Systems Management College 2001). However, healthcare problems are often classified as wicked and complex and not amenable to traditional decomposition methods found in other areas of engineering. (Rouse & Serban 2014).

There are many tailored approaches to improving healthcare delivery, but almost all are based on one of these three approaches, or a combination of these.

Healthcare Systems Engineering

The basic systems engineering steps are similar to those for any industry specific applications, but the steps are tailored for healthcare. The traditional waterfall model of requirements, design, implementation, verification, and maintenance is interrupted in favor of almost continuous support. In many cases the closeout and transfer of the project to operational staff is more challenging in healthcare than in many industries.

Below is outlined a general methodology used by the US Department of Veteran's Affairs (VA) that may suit a wide variety of situations and programs, composed of 4 pillars: Define the Problem, Investigate Alternatives, Develop the Solution, and Launch and Assess the Solution. These 4 pillars are similar to classic mistake avoidance, development fundamentals, risk management, and schedule oriented approaches. (McConnell 1996); they are also similar to the Plan/Do/Check/Act methodology.

Define the Problem

As mentioned above, the patient is augmented by a care team consisting of family, friends, clinical staff, and many other support staff the patient will not directly encounter. This care team may not be familiar with the rigors of traditional engineering design. Because of this, a systems engineer may use a paired partnership model where engineers are embedded with clinical and administrative staff, family, and the patients themselves. In this concept, everyone is a designer and our goal is to provide them with the tools to contribute to the system design process. Even at this early stage, configuration management would be considered. Depending on the size of the rollout one alpha site and several beta sites may be used at any phase to avoid local optimal solutions that don't work globally.

Investigate Alternatives

During the proof-of-concept phase, visualizing the result is important for the reasons mentioned above. Therefore, one or more initial prototypes may be developed with the alpha site. The goal is to get to a minimally viable product as soon as possible to demonstrate the viability of the product or methodology. After the initial conversations and meetings, participants have a need to have a common understanding of how the system will work. The systems engineer would embrace the concept of operations with rich pictures, model based systems engineering, story boards, customer journey maps and other tools so that we all have a common understanding of the proposed system.

Develop the Solution

Using what has been learned from the minimally viable product feedback and incorporate that into the future state optimization, one would continue developing the prototype at the initial paired partner alpha site and then the trusted beta demonstration sites. In our case, stakeholders are a part of the development team and not an ancillary function. For this reason, demonstration is considered a key element of the communication plan when developing the solution.

Launch the Solution and Access the Performance

During evaluation and deployment phase, a systems engineer would have considered the future state optimization with corresponding alpha and beta sites. Live implementation would then be used for further testing and evaluation. At any phase feedback is encouraged and reflected in the next iteration of the solution. As mentioned previously abandonment and closeout even during the live phase may not be practical and in fact could be disadvantageous because not all possible needed configurations or situations would have been encountered.

Example Systems Engineering Tools

Below is a list of systems engineering tools which could be used at each of the four steps.

1. Define the Problem
    1. Establish the scope and context of the problem (define boundary conditions)
    2. Stakeholder identification and management
    3. Lifecycle mapping
    4. Value Stream Process Mapping
    5. SWOT analysis (Operational Deficiencies and Technological Opportunities)
    6. Workflow/Usability/Use Case analysis
    7. Observation Research
    8. Root Cause Analysis (Fishbone diagrams, 5 whys, …)
2. Investigate Alternatives
    1. Requirements management
    2. SE Evaluation Methods (Decision Trees, Quality Function Deployment (QFD))
    3. Trade-off Analysis
    4. Model-Based Systems Engineering (MBSE)
    5. Technical Risk Management
3. Develop the Solution
    1. Concept Development
    2. Architecting the solution (functional analysis, subsystem decomposition, interface definition and control, modeling)
    3. Define the implementation

    4. Process Redesign Techniques (including Lean Six Sigma)
    5. Active Integration
    6. Agile / Lean Development Principles (iterative development)
4. Launch and Assess the Solution
    1. Managing Change in Organizations
    2. Stakeholder Management, Change Management Techniques
    3. Spiral, Agile, and Lean Startup Delivery Practices (Minimal Viable Product delivery)
    4. Business Risk Management
    5. Metrics and benchmarking

During all phases, elements of cognitive & organizational psychology, industrial engineering, usability engineering, systems engineering, and other facets may be critical to implement a solution. Humans are *the* major part of the system and even the system of systems approach in healthcare.

Conclusion

Systems Engineering for Healthcare delivery shares many aspects with traditional SE, but differs significantly since healthcare delivery is a service (not a product) and due to the domain specific challenges. In particular, problem definition is a particularly 'wicked' problem, and measuring successful outcomes in a clear and objective fashion is challenging.

References

Works Cited

Checkland, P. 1999. *Systems Thinking, Systems Practice*. Hoboken NJ: John Wiley.

Compton, W.D., G. Fanjiang, J.H. Grossman, & P.P. Reid. 2005. *Building a better delivery system: a new engineering/health care partnership*. Washington DC: National Academies Press.

Conover, C.J. 2012. *American health economy illustrated*. Washington DC: American Enterprise Institute.

Defense Systems Management College. 2001. *Systems engineering fundamentals: Supplementary text*. Fort Belvoir, VA: The Press.

George, M.L. 2002. *Lean Six Sigma*. New York, NY: McGraw-Hill.

George, M.L. 2003. *Lean Six Sigma for Service*. New York, NY: McGraw-Hill.

Griffith, J.R., & K.R. White. 2007. *The Well-Managed Healthcare Organization*. Chicago, IL: Health Administration Press.

Institute of Medicine. 2001. *Crossing the quality chasm: A new health system for the 21st century*. Washington DC: National Academy Press.

Kohn, L.T., J. Corrigan, & M.S. Donaldson. 2000. *To err is human: Building a safer health system*. Washington DC: National Academy Press.

McConnell, S. 1996. *Rapid Development*. Redmond, WA: Microsoft Press.

Ohno, T. 1988. *Toyota Production System*. Portland OR: Productivity Press.

PCAST. 2014. *Better Health Care and Lower Costs: Accelerating Improvement through Systems Engineering*. Washington DC.: President's Council of Advisors on Science and Technology (PCAST).

Rouse, W.B., & N. Serban. 2014. *Understanding and managing the complexity of healthcare*. Cambridge, MA: The MIT Press.

Smith, M.D., R.S. Saunders, L. Stuckhardt, & J.M. McGinnis. 2013. *Best care at lower cost: The path to continuously learning health care in America*. Washington DC: National Academies Press.

Womack, J.P., D.T. Jones, and D. Roos.1990. *The machine that changed the world*. New York, NY: Harper Collins.

Primary References

PCAST. 2014. *Better Health Care and Lower Costs: Accelerating Improvement through Systems Engineering*. Washington DC.: President's Council of Advisors on Science and Technology (PCAST).

Additional References

None.

< Previous Article | Parent Article | Next Article >

SEBoK v. 2.1, released 31 October 2019

Systems Biology

Lead Authors: *Bridgette Daniel Allegro, Gary Smith*, ***Contributing Authors:*** *Chris Unger, Nicole Hutchison*

Systems biology is the computational and mathematical modelling of complex biological systems. Systems biology is a biology-based inter-disciplinary field of study that focuses on complex interactions within biological systems, using a holistic approach to biological research. From year 2000 onwards, the concept has been used in the biosciences in a variety of contexts. For example, the Human Genome Project is an example of applied systems thinking in biology which has led to new, collaborative ways of working on problems in the biological field of genetics. One of the outreaching aims of systems biology is to model and discover emergent properties of cells, tissues and organisms functioning as a system whose theoretical description is only possible using techniques which fall under the remit of systems biology. These typically involve metabolic networks or cell signalling networks. (Wikipedia Contributors 2016)

Systems Biology: A Vision for Engineering and Medicine

Organisms and Hosts Interact in Communities of Life

There is an increasing appreciation that microbes are an essential part of the ecologically-important traits of their host. Organisms do not live in isolation, but have evolved, and continue to evolve, in the context of complex communities and specific environmental conditions. Evolutionary biologists are increasingly able to integrate information across many organisms, from multiple levels of organization and about entire systems to gain a new integrated understanding that incorporates more and more of the complexity that characterizes interdependent species associations. Only when we begin to understand the molecular base for adaptation and interactions of communities of life, can we start to comprehend how ecosystems are functioning.

Addressing Different Levels of Organization of Organisms

Understanding the function of complex biological systems is one of the greatest challenges facing science. The rewards of success will range from better medicines to new engineering materials. The sequencing of the human genome, although of fundamental importance, does not even provide a complete parts list of the protein molecules that exist in a biological organism because of complexities of downstream processing and complex folding required to make a functioning receptor or enzyme from a long chain of amino acids. Furthermore, protein molecules do not function alone but exist in complex assemblies and pathways that form the building blocks of organelles, cells, tissues, organs, organ systems and organisms, including man. The functioning of brain or muscle, liver or kidney, let alone a whole person, is much greater than the sum of its parts.

Figure 1 - Levels of Structural Organization of the Human Body (source - https:/ / cnx. org/ contents/ Xh_25wmA@7/Structural-Organization-of-the#fig-ch01_02_01)'"

Internalizing the Complexity – Pushes the Boundary of Systems Thinking Capability

To tackle this problem (understanding biological systems) requires an iterative application of biomedical knowledge and experiment with mathematical, computational and engineering techniques to build and test complex mathematical models. Systems and control engineering concepts, a modular approach and vastly increased computing capacity are of critical importance. The models, once developed and validated, can be used to study a vastly greater range of situations and interventions than would be possible by applying classical reductionist experimental methods that usually involve changes in a small number of variables. This new approach is now termed "Systems Biology". It allows insight into the large amount of data from molecular biology and genomic research, integrated with an understanding of physiology, to model the complex function of cells, organs and whole organisms, bringing with it the potential to improve our knowledge of health and disease. Systems Biology will inevitably become an approach that pervades scientific research, in much the same way that molecular biology has come to underpin the biological sciences. It will transform the vast quantities of biological information currently available into applications for engineering and medicine.

Natural Patterns and Engineered Patterns Can Be a Source of Inspiration - in Both Directions

Biological organisms are much more complicated than any machine designed by man. However, there are similarities between the way in which organs and whole organisms are assembled from molecules and cells and the design methods used by engineers in the construction of complex systems. The application of such methods to biology will, however, require novel engineering tools to be developed since biological systems possess key features that artificial ones do not. Specifically, biological systems have an exceptional capacity for self-organization and assembly, using rules and mechanisms that have been shaped by natural selection. Biological systems also have significant capacity for continuing self-maintenance through turnover and renewal of component parts. Perhaps the property that distinguishes biological systems most is their ability to auto-adapt their organization to changing circumstances through altered gene expression, or more directly, through signal transduction and modification of proteins. This adaptation culminates at higher levels of organization as evidenced by phenomena such as the development of resistance to antibiotic therapy or tolerance to recreational drugs. The mechanisms by which component parts interact are often highly stochastic in nature; that is, susceptible to the play of chance, which becomes particularly important when only a few components are being considered. Nevertheless, biological systems are robust.

Advancements in Methods for Predicting "What If" in the Behavior of Complex Adaptive Systems

Advances in engineering design and techniques carry a significant potential in driving the progress of Systems Biology. Interventions to biological systems intended to improve health, whether environmental, pharmacological or clinical, need to be carefully thought through and carried out to maximize benefit and reduce harm. The refinement of techniques and tools enables devices and systems to achieve a defined performance within precise tolerance limits, potentially allowing better interventions to complex biological systems. They will be increasingly necessary to permit more reliable system-wide predictions of the effects of biomedical advances and to achieve desired clinical results to a predefined tolerance, or at least to have a quantitative bound on the biological uncertainty.

Transdisciplinary Approaches are Needed to Address the Complex Bio-system Problems

Research in the field of Systems Biology requires close interactions and collaborations between many disciplines that have traditionally operated separately such as medicine, biology, engineering, computer science, chemistry, physics and mathematics. Systems Biology demands a focus on the problem as a whole and therefore a combination of skills, knowledge and expertise that embraces multiple disciplines. The success of leaders in the field of Systems Biology will depend strongly on the extent to which they accomplish the creation of the environment that researchers need to develop an understanding of different working cultures, and manage also to implement strategies that integrate these cultures into shared working practices.

Systems Biology: Relevance to Healthcare

Complex Diseases Demand Systemic Approaches

Over the past few decades, pharmaceutical R&D has focused on creating potent drugs directed at single targets. This approach was very successful in the past when biomedical knowledge as well as cures and treatments could focus on relatively simple causality. Nowadays, the medical conditions that affect a significant proportion of the population in industrialized countries are more complex, not least because of their multifactorial nature. The sequencing of the human genome has led to a considerable increase in the number of potential targets that can be considered in drug discovery and promises to shed light on the etiology of such conditions. Yet, the knowledge of the physiological properties and the role that these targets play in disease development is still limited.

Diminishing Returns in the Single Target Approach to Disease

In terms of drug targets, there is a case that much of ‘the low hanging fruit’ was picked in the period between the late 1940s and the mid-1980s. The decline in output of new molecular entities and medicines recorded over the last 20 years, despite the steadily growing R&D expenditure and significant increase in sales, bears testimony to the fact that advances with new targets are more difficult and that R&D projects have become much more prone to failure. A basic problem is that the many factors that predispose to, and cause, complex diseases are poorly understood let alone the way in which they interact. The very fact that there are multiple drivers for these conditions suggests that a reductionist approach focusing on individual entities in isolation is no longer appropriate and may even be misleading. It is therefore necessary to consider ‘novel’ drugs designed to act upon multiple targets in the context of the functional networks that underlie the development of complex diseases. Many of the new developments are likely to turn into effective medicines when combinations of drugs are used to exert a moderate effect at each of several points in a biological control system. Indeed, many common diseases such as hypertension and diabetes are already treated with a combination of two or three medicines hitting different targets in the control network that underlies the condition. Investigating the possible combinations by trial and error in man is onerous but feasible with two components. However, it quickly becomes extremely complicated with three components and well-nigh impossible with four or more. Systems Biology, promises to assist in the development of more specific compounds and in the identification of optimal drug targets on the basis of their importance as key ‘nodes’ within an overall network, rather than on the basis of their properties as isolated components.

Individualized Medicine, Tuned to the Individual and Their Circumstances

Increasingly powerful drugs will be aimed at a decreasing percentage of people and eventually at single individuals. Modelling can be used to integrate in vivo information across species. Coupled with in vitro and in silico data, it can predict pharmacokinetic and pharmacodynamics behavior in humans and potentially link chemical structure and physicochemical properties of the compound with drug behavior in vivo. Large-scale integrated models of disease, such as diabetes and obesity, are being developed for the simulation of the clinical effects resulting from manipulations of one or more drug targets. These models will facilitate the selection of the most appropriate targets and help in planning clinical trials. Coupling this approach with pertinent genomic information holds the promise of identifying patients likely to benefit most from or to be harmed by, a particular therapy as well as helping in the stratification of patients in clinical trials. Symptoms that diagnose a disease do not necessarily equate to a common cause.

Systems Biology is arguably the only research approach that has the potential to disentangle the multiple factors that contribute to the pathogenesis of many common diseases. For example, hypertension, diabetes, obesity and rheumatoid arthritis are known to be polygenetic in origin although individual genes may not have been identified. Ultimately, the prevention of these conditions rests upon a comprehensive approach that engages with each of the more important predisposing factors, genetic and environmental, that operate upon individuals. A systems approach is already proving valuable in the study of complex scientific subjects and the research aimed at the prevention and management of medical conditions. Illustrative examples are neuroscience, cancer, ageing and infectious diseases.

A Healthcare Paradigm Reinforcing the Causes of Health and Not Just the Treatment of Disease

Notwithstanding the hugely important role that Systems Biology plays in understanding disease and designing drugs that treat them, the greatest opportunities may lie in health maintenance and disease prevention. Even modest measures that could retard the effect of ageing on brain, heart, bones, joints and skin would have a large impact on the quality of life and future healthcare demands of older people and consequently on the provision of health services. Young people are vulnerable too. Multifactorial diseases such as diabetes, obesity, allergies and autoimmune conditions are becoming prevalent in younger people and unless effective measures are taken to prevent an early and significant decline in their health, healthcare demand will increase exponentially. It is apparent that multiple and diverse factors interact in determining health, quality of life and ageing. These include genetic makeup, microbiota, diet, physical activity, stress, smoke and alcohol, therapeutic and social drugs, housing, pollution, education, and only a systems approach will permit the understanding of how best to prevent and delay health decline.

References

Works Cited

Wikipedia contributors. "Systems biology." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 20 Aug. 2016. Web. 12 Sep. 2016.

Primary References

None.

Additional References

Bosch, T.C.G. and M.J. Mc-Fall-Ngai. 2011. "Metaorganizsms as the new frontier." *Zoology*. 114(4): 185-190. September 2011. http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3992624/

Dollery, C. and R. Kitney. 2007. *Systems Biology: a vision for engineering and medicine*. London: The Academy of Medical Sciences and The Royal Academy of Engineering. Available at https://www.acmedsci.ac.uk/viewFile/publicationDownloads/1176712812.pdf

Endy, D. 2005. "Foundations for engineering biology." *Nature*. 438(7067): 449-453. 24 November 2005. Available at: http://www.nature.com/nature/journal/v438/n7067/abs/nature04342.html

Harvard Medical School. 2010. "Department of Systems Biology." Cambridge, MA: Harvard Medical School, Harvard University. Available at: https://sysbio.med.harvard.edu/

Kitano, H. 2002. "Systems Biology: A Brief Overview." *Science.* 295(5560): 1662-1664. 01 March 2002. Available at: http://science.sciencemag.org/content/295/5560/1662.

Sauser, B., J. Boardman, and D. Verma. 2010. "Toward a Biology of Systems of Systems." *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans.* 40(4): 803 - 814. Available at: http://ieeexplore.ieee.org/document/5467221/

Wikipedia contributors. "Systems biology." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 20 Aug. 2016. Web. 12 Sep. 2016.

< Previous Article | Parent Article | Next Article >

SEBoK v. 2.1, released 31 October 2019

Lean in Healthcare

Lead Author: *Bohdan Oppenheim*, ***Contributing Authors:*** *Chris Unger, Nicole Hutchison*

Lean Thinking, or Lean for short, originated in Toyota factories in the 1960s, was “transplanted” to the U.S. in 1992 with the publication of Womack and Jones' *Lean Thinking: Banish Waste and Create Wealth in Your Corporation* (2003), and evolved globally to practically all work domains: healthcare, engineering and systems engineering, science, administration, supply chain, government, banking, aviation, and many others (Oppenheim 2011). Lean has proven itself as the most effective methodology for improving operations identifying and eliminating waste from work processes. (E.g. Womack and Jones 2003; Oppenheim 2011; Graban 2012; Toussaint and Gerard 2010; and Oehmen 2012) Since 2003, Lean has established itself in healthcare operations.

Overview of Lean in Healthcare

Entire medical organizations (e.g., Theda Care, WI; Jefferson Healthcare, WA; Virginia Mason, WA; Geisinger Health (now called ProvenCare), PA; St. Elizabeth, Tilburg, The Netherlands, and numerous others (e.g. Graban 2012; Toussaint and Gerard 2010)) have been transformed with Lean. These sources contain rich data on specific improvements. Most leading healthcare institutions now have Lean centers of excellence or use Lean consultants, including Kaiser Permanente, Mayo Clinic, UCLA, Veterans Administration, and others. Lean has proven itself in reducing turnaround time of clinical tests, the time spent by patients in emergency departments, operating suites, pharmacies and clinics. Lean improvements in healthcare on the order of 30-50% are routine because traditional healthcare operations are burdened with this much waste, which remains “unseen” by the employees unless they are trained in Lean. Lean is now an established paradigm for improving healthcare delivery operations: increasing quality of healthcare, delivering care faster, shortening patient time in the system, increasing the time of medical professionals with the patient, reducing bureaucracy, increasing capacity of operations, and reducing healthcare costs and frustrations. (Graban 2012; Toussaint and Gerard 2010)

Lean does not mean that people have to work faster or "attach roller blades to move around faster". In Lean, systems employees work at their regular ergonomic and intellectual speeds. The time savings come from finding and eliminating idle states (e.g., waiting in numerous queues in the emergency departments), reduction of mistakes and rework, elimination of non-value adding tasks, and more streamlined movements of patients, staff, equipment, and supplies. And, most emphatically, Lean does not mean “mean layoffs”. Quite the opposite is true: Lean improves human relations at work and changes the culture from the traditional "blaming and shaming" to teamwork and cooperation focused on the good of the patient. (Graban 2012 (in particular see the endorsements from eight medical professionals on pages ii and iii) and (Toussaint and Gerard 2010)

With the endorsement of *Lean for Systems Engineering with Lean Enablers for Systems Engineering in the Wiley Series*, (Oppenheim 2011) the International Council on Systems Engineering (INCOSE) has effectively adopted Lean as one of its essential competencies. This book was followed with a major joint Project Management Institute (PMI)-INCOSE-MIT publication of (Oehmen 2012) integrating Lean with Systems Engineering and Program Management. Indeed, when applied with Systems Engineering and Systems Thinking, Lean becomes a powerful weapon in bending the healthcare cost curve and improving the quality of care.

Three concepts are critical to the understanding of Lean: value, waste, and the process of creating value without waste, which has been captured into the so-called Six Lean Principles, as follows.

- **Value**: M. Porter (2010) suggested that patients value three levels of care: (1) survival and the degree of recovery; (2) the time required to get back to normal activities, and (3) the sustainability (individual and social cost) of treatments.
- **Waste**: Table 1 lists the eight categories of waste used in healthcare. (Graban 2012; Toussaint 2010)

Table 1. Eight Waste Categories Used in Lean Helathcare (SEBoK Original).

Waste Type	Healthcare Examples
1. Waiting	Patients wait in numerous queues in clinics, test facilities, ERs, pharmacies, and for insurance approvals; MDs wait for next activity to occur (e.g. test results, information, approvals.)
2. Over-processing	Performing work that is not valued or needed, e.g. MDs and RNs spending time on computer filling out bureaucratic forms that nobody will review.
3. Over-production	Performing more work than needed for value. Transport of a patient in a wheelchair performed by expensive medical professionals because of the lack of transporters.
4. Inventory	Excess inventory costs. Expired supplies that must be thrown away.
5. Transportation of Patients	Transportation of patients over long distances to test offices in hospitals. Poor layout of hospitals, EDs, or test facilities.
6. Motion of Staff	Staff walking over long distances to fetch supplies, and between patients and central hospital stations.
7. Defects	Treatment of hospital infections. Failed and repeated tests, repeated paperwork. Surgical cart missing an item. Wrong medicine.
8. Waste of Human Potential	Burnout of medical staff. Frustrated employees quit making suggestions for improvements.

Table 2 lists the six Lean Principles (Graban 2012) and provides healthcare examples.

Table 2. Six Lean Principles (SEBoK Original)

Principle Name	Explanation
1. Value	Specify value from the perspective of the customer: the patient.
2. Value Stream	Identify all the value-added steps across the entire process, crossing all departmental boundaries, linking the steps into a seamless process, and eliminating all steps that do not create value.
3. Flow	Keep the processes flowing smoothly through all the steps, eliminating all causes of delay, such as batches of patients or items, and quality problems.
4. Pull	Avoid pushing work onto the next step or department; let work and supplied be pulled, as needed, when needed.
5. Perfection	Pursue perfection through continuous improvement, Kaizen events, implement best work standards, checklists, training, and promote improvement teams and employee suggestions.
6. Respect People	Create work environment based on synergy of cooperation, teamwork, great communication and coordination. Institute leadership. Abandon the culture of blaming and shaming.

Lean Practices

Lean healthcare strongly promotes engaging and leading employees. Lean places a big value on continuous education and training of employees at all levels. Lean management promotes standardization of best practices ("the best known way of doing it", but not necessarily "identical"), checklists, redundancies, patient safety and privacy rules, and patient data security and cybersecurity. Lean advocates visual management, with electronic or "black" boards updated in real time and displaying all information important for the local employees to manage their operation efficiently. Patient safety is still a significant problem in the U.S., in 1999 causing almost 250,000 deaths (Institute of Medicine, 1999) and medical errors occur in one of three admissions. Instead of "blaming and shaming" Lean promotes error and harm prevention and deep root-cause analysis, implementing processes and tools that make it impossible to create an error.

Systems Thinking and Lean

Healthcare is the most complex socio-technological system in our society, consuming nearly 20% of the U.S. GDP. Healthcare should be safe, effective and evidence based (Berwick 2011), as well as affordable and accessible, efficient, patient centered, timely, well integrated, and inclusive of latest science. (Oppenheim 2015) Healthcare has many stakeholders: the patients, medical professionals, medical facilities, hospitals, clinics, labs, medical equipment makers and users, pharmaceuticals, healthcare researchers and academia, insurances, employers, federal & state governments and international disease prevention centers, military and veteran's administration, fire departments and ambulances and others. The number of potential interactions (interfaces) in this hyper-system is extensive, and many interfaces are nonlinear, "wicked" (interacting with unpredictable humans), often creating unintended consequences and emergent behaviors. Because of these vast complexities, healthcare leaders (e.g. Kanter 2015) point out the need for intensive application of systems thinking and Lean when addressing these challenges. Attempting to solve the complex healthcare problems without systems thinking risks myopic and unsafe attempts which create more problems than they solve. Attempting to solve the challenges without Lean inevitably promotes excessive wastes, costs, and inefficiencies. Good healthcare needs both, Systems Thinking and Lean, to be applied simultaneously.

Lean and Agile in Six Healthcare Value Streams

The Healthcare Working Group of INCOSE identified six following value streams for HSE: A. Systems Engineering for medical devices B. Systems Engineering for healthcare informatics and medical records C. Healthcare delivery (operations) D. Biomedicine and big data analytics E. Pharmaceutical value streams F. Healthcare public policy

As described above, Lean is extraordinarily effective and well established in improving healthcare delivery operations (C). Agile is highly effective in (B) because this value stream works with software, the domain from which Agile originated. Since the stream (A) is the most similar to traditional systems engineering, Agile is expected to be effective therein, although Agile is not yet highly popular in healthcare outside of the software domain. Elements of Lean improvements which are localized and weakly convoluted (e.g., Kaizen events) have strong overlap with Agile/Scrum methodology. (Medinilla 2014)

MBSE and Lean

A highly powerful Model Based Systems Engineering (MBSE) is clearly the tool of choice for those applications where the benefit from multiple use of a standardized (reference) architecture and standard model compensates for the significant effort of creating such a model or architecture. (OMG 2016) In healthcare the value streams (A), (B), (E) and potentially F are the most conducive to the application of MBSE. Lean thinking is applicable to any healthcare operation without limitation. The Lean improvements always begin with the so-called Gemba waste walks, during which experts together with local process stakeholders walk along all the process steps, interviewing stakeholders and identifying and measuring the wastes wherever they occur. The rich menu of Lean thinking processes and tools is then applied to eliminate the wastes. Training and active participation of local stakeholders is always required.

Examples of Lean Improvements

1. In Jefferson Healthcare, WA: (Murman 2010)
   - In Acute Myocardial Infarction (a severe heart attack) time is critical as the greatest loss of heart muscle is in the first two hours. Recommended treatment is catheter insertion of balloon within 90 min of the contact with the patient (wherever the patient happens to be located). The Lean approach has reduced the treatment time from 165 min to 20-60 min at the patient site, vastly increasing patient survival rate.
   - The five Jefferson Healthcare clinics increased the cumulative available clinic hours from 1400 to 5600 in two years of Lean improvements which were focused on reorganizing medical staff schedules and eliminating wasted times, with no staff additions. The available clinic hours directly translate into billable visits: 1175 additional patients have been seen in 2009 compared to 2008 across the five clinics.
   - The Operating Room daily "on time start" of actual operations went from 14% to 96% using Lean tools for process planning and workplace organization.
   - Harder to measure is the culture change, although the staff participation at Lean improvement events was at 50%.
2. In Kaiser Permanente Southern California: (Oppenheim 2015)
   - In nine regional clinical laboratories Lean improvements cut the turnaround time for laboratory results by between 30 and 70%, with significant corresponding reductions of cost, rework, errors and work morale, and without hiring new staff or adding equipment.
   - In two Emergency Departments (ED) the average patient length of stay was reduced by 40% by the elimination of various idle states. The ED capacity increased accordingly.
   - The amount and cost of inventory of supplies on hand was reduced by nearly 30% by introducing the Just-in-Time tools of Lean.
3. In Alegent Health, NE (Graban 2012) the turnaround time for clinical laboratory results was reduced by 60% in 2004 without adding new staff or equipment; and by another 33% from 2008 to 2010.
4. In Kingston General Hospital, Ontario (Oehmen) the instrument decontamination and sterilization cycle time was reduced by 54% while improving productivity by 16%.
5. In Allegheny Hospital, PA the central-line associated bloodstream infections were reduced by 76%, reducing patient death from such infections by 95% and saving $1 million.
6. In UPMC St. Margaret Hospital, PA (Graban 2012) the readmission rates for chronic obstructive pulmonary disease (COPD) patients were reduced by 48%.
7. In ThedaCare, WI [3] the waiting time for orthopedic surgery was reduced from 14 weeks to 31 hours (from first call to surgery); improved inpatient satisfaction scores of "very satisfied" rose from 68% to 90%.
8. In Avera McKennan, SD [3] the patient length of stay was reduced by 29%, and $1.25 million in new ED construction was avoided.
9. In Denver Health, CO [3] the bottom-line Lean benefit was increased by $54 million through cost reduction and revenue growth, and layoffs were avoided.
10. In Seattle Children's Hospital, WA $180 million in capital spending was avoided through Lean improvements.

These examples demonstrate that Lean is successful in cost and throughput time reductions, and improvements in quality and patient and staff satisfaction. The improvements of this level are possible, even routine – because the amount of initially-invisible waste in traditional healthcare organizations is so high. The broad range of operations described in the examples manifest that Lean is applicable across the board to healthcare operations, without limitations.

Education in Lean Healthcare

Increasingly, Lean Healthcare becomes an inherent part of Healthcare Systems Engineering (HSE) Master's Programs, e.g. (Loyola Marymount University 2016) which has been developed in collaboration with Kaiser Permanente. The program includes two courses in Lean, basic and advanced, focused on improving operations in clinics, hospitals, emergency departments, clinical laboratories, radiology testing, operating rooms, pharmacies, supply chain, and healthcare administration. After the basic courses in systems engineering, project management, and systems thinking, the students also take courses on healthcare system architecting, modeling and simulations; medical data mining and analytics; systems engineering for medical devices, healthcare enterprise informatics; and healthcare delivery systems. All these advanced courses contain elements of Lean thinking because all these subdomains risk being burdened with waste and poor quality if Lean is ignored. Simply put, Lean is not really an optional extra if you want to achieve efficiency and effectiveness.

References

Works Cited

B.W. Oppenheim, B.W. 2015. "Lean Healthcare," INCOSE Healthcare Working Group webinar. San Diego, CA: International Council on Systems Engineering. April 30, 2015. Available at: https:/ / onedrive. live. com/ redir?resid=147E5C4249DA0EFB%21142

Berwick, D. 2009, "National Forum Keynote, Institute for Healthcare Improvement." Cambridge, MA: Institute for Healthcare Improvement. Available at: http:/ / www. ihi. org/ IHI/ Programs/ AudioAndWebPrograms/ BerwickForumKeynote2009.htm (accessed July 4, 2011)

Graban, M. 2012. Lean Hospitals; Improving Quality, Patient Safety, and Employee Engagement. Boca Raton, FL: CRC Press.

Kanter, M.K. 2015. "Strategic Partnership of Healthcare and Systems Engineering." San Diego: INCOSE Healthcare Working Group presentation, 2015

Loyola Marymount University. 2016. "MS Degree Program in Healthcare Systems Engineering." Available at: CSE.lmu.edu/graduateprograms/systemsengineering/healthcaresystemsengineeringms/

Medinilla, Á. 2014. ''Agile Kaizen: Managing Continuous Improvement Far Beyond Retrospectives.'' New York, NY: Springer, 2014

Murman, E. 2010 "The Lean Aerospace Initiative." Boston MA: Lean Advancement Initiative (LAI) Annual Conference.

Oehmen, J. 2012. The Guide to Lean Enablers for Managing Engineering Programs. PMI-INCOSE-LAI MIT. May 2012.

OMG. 2016. "MBSE Wiki." Available at: http://www.omgwiki.org/MBSE/doku.php (last accessed March 29, 2016)

Oppenheim, B.W. 2011. Lean for Systems Engineering with Lean Enablers for Systems Engineering. Hoboken, NJ: Wiley Series in Systems Engineering and Management.

Porter, M. 2010. "What is Value in Healthcare?" New England Journal of Medicine. 363: 2488-2481. 08 December 2010.

Toussaint, J. and R. Gerard. 2010. On the Mend: Revolutionizing Healthcare to Save Lives and Transform the Industry. Cambridge, MA: Lean Enterprise Institute. 06 June 2010.

Womack, J.P. and D. T. Jones. 2003. Lean Thinking: Banish Waste and Create Wealth in Your Corporation. Washington, DC: Free Press.

Primary References

None.

Additional References

None.

< Previous Article | Parent Article | Next Article (Part 5) >

SEBoK v. 2.1, released 31 October 2019

Article Sources and Contributors

Letter from the Editor *Source*: https://www.sebokwiki.org/d/index.php?oldid=57764 *Contributors*: Apyster, Bkcase, Cnielsen, Dholwell, Eleach, Kguillemette, Radcock, Rcloutier, Smenck2, Wikiexpert

BKCASE Governance and Editorial Board *Source*: https://www.sebokwiki.org/d/index.php?oldid=57626 *Contributors*: Apyster, Bkcase, Cnielsen, Dhenry, Kguillemette, Radcock, Rcloutier, Smenck2

Acknowledgements and Release History *Source*: https://www.sebokwiki.org/d/index.php?oldid=57682 *Contributors*: Apyster, Asquires, Bkcase, Cnielsen, Ddori, Dhenry, Dholwell, Eleach, Janthony, Jgercken, Kguillemette, Nicole.hutchison, Radcock, Rcloutier, Smenck2, Smurawski, Wikiexpert

Cite the SEBoK *Source*: https://www.sebokwiki.org/d/index.php?oldid=57684 *Contributors*: Apyster, Bkcase, Cnielsen, Dholwell, Kguillemette, Smenck2

Bkcase Wiki:Copyright *Source*: https://www.sebokwiki.org/d/index.php?oldid=57613 *Contributors*: Apyster, Bkcase, Cnielsen, Dhenry, Kguillemette, Radcock, Smenck2, Wikiexpert

Product Systems Engineering *Source*: https://www.sebokwiki.org/d/index.php?oldid=57177 *Contributors*: Bkcase, Blawson, Dcarey, Dhenry, Dholwell, Janthony, Jgercken, Kguillemette, Mhaas, Rlpineda, Skmackin, Smenck2, Smurawski, WikiWorks, WikiWorks753, Wikiexpert, Zamoses

Product Systems Engineering Background *Source*: https://www.sebokwiki.org/d/index.php?oldid=57178 *Contributors*: Bkcase, Dhenry, Dholwell, Janthony, Jgercken, Kguillemette, Mhaas, Rlpineda, Smenck2, Smurawski, WikiWorks, WikiWorks753, Wikiexpert

Product as a System Fundamentals *Source*: https://www.sebokwiki.org/d/index.php?oldid=57179 *Contributors*: Bkcase, Dhenry, Dholwell, Janthony, Jgercken, Kguillemette, Mhaas, Rlpineda, Smenck2, Smurawski, WikiWorks, WikiWorks753, Wikiexpert

Business Activities Related to Product Systems Engineering *Source*: https://www.sebokwiki.org/d/index.php?oldid=57180 *Contributors*: Bkcase, Cnielsen, Dhenry, Dholwell, Janthony, Jgercken, Kguillemette, Mhaas, Rlpineda, WikiWorks753, Wikiexpert

Product Systems Engineering Key Aspects *Source*: https://www.sebokwiki.org/d/index.php?oldid=57181 *Contributors*: Bkcase, Dhenry, Dholwell, Janthony, Jgercken, Kguillemette, Mhaas, Rlpineda, Smenck2, Smurawski, WikiWorks, WikiWorks753, Wikiexpert

Product Systems Engineering Special Activities *Source*: https://www.sebokwiki.org/d/index.php?oldid=57182 *Contributors*: Bkcase, Cnielsen, Dhenry, Dholwell, Janthony, Jgercken, Kguillemette, Mhaas, Rlpineda, Smenck2, Smurawski, WikiWorks, Wikiexpert

Service Systems Engineering *Source*: https://www.sebokwiki.org/d/index.php?oldid=57382 *Contributors*: Apyster, Bkcase, Blawson, Cnielsen, Dhenry, Dholwell, Janthony, Jgercken, Jmartin, Kguillemette, Mhaas, Rlpineda, Rturner, Smenck2, Smurawski, WikiWorks, WikiWorks753, Wikiexpert, Zamoses

Service Systems Background *Source*: https://www.sebokwiki.org/d/index.php?oldid=57445 *Contributors*: Bkcase, Blawson, Cnielsen, Dhenry, Dholwell, Janthony, Jgercken, Kguillemette, Rlpineda, Smenck2, Smurawski, WikiWorks, WikiWorks753, Wikiexpert, Zamoses

Fundamentals of Services *Source*: https://www.sebokwiki.org/d/index.php?oldid=57184 *Contributors*: Bkcase, Blawson, Dhenry, Dholwell, Janthony, Jgercken, Kguillemette, Mhaas, Rlpineda, Smenck2, Smurawski, WikiWorks753, Wikiexpert, Zamoses

Properties of Services *Source*: https://www.sebokwiki.org/d/index.php?oldid=57185 *Contributors*: Asquires, Bkcase, Blawson, Bwells, Dhenry, Dholwell, Janthony, Jgercken, Kguillemette, Mhaas, Rlpineda, Smenck2, WikiWorks, WikiWorks753, Wikiexpert, Zamoses

Scope of Service Systems Engineering *Source*: https://www.sebokwiki.org/d/index.php?oldid=57475 *Contributors*: Bkcase, Blawson, Dhenry, Dholwell, Janthony, Jgercken, Kguillemette, Mhaas, Rlpineda, Smenck2, Smurawski, WikiWorks, WikiWorks753, Wikiexpert, Zamoses

Value of Service Systems Engineering *Source*: https://www.sebokwiki.org/d/index.php?oldid=57458 *Contributors*: Bkcase, Blawson, Cnielsen, Dhenry, Dholwell, Janthony, Jgercken, Kguillemette, Mhaas, Rlpineda, Rturner, Smenck2, Smurawski, WikiWorks, WikiWorks753, Wikiexpert, Zamoses

Service Systems Engineering Stages *Source*: https://www.sebokwiki.org/d/index.php?oldid=57509 *Contributors*: Asquires, Bkcase, Blawson, Cnielsen, Dhenry, Dholwell, Janthony, Jgercken, Kguillemette, Mhaas, Rlpineda, Rturner, Smenck2, WikiWorks, WikiWorks753, Wikiexpert, Zamoses

Enterprise Systems Engineering *Source*: https://www.sebokwiki.org/d/index.php?oldid=57189 *Contributors*: Apyster, Bkcase, Blawson, Dfairley, Dhenry, Dholwell, Janthony, Jgercken, Jmartin, Kguillemette, Mhaas, Skmackin, Smenck2, Smurawski, WikiWorks753, Wikiexpert, Zamoses

Enterprise Systems Engineering Background *Source*: https://www.sebokwiki.org/d/index.php?oldid=57190 *Contributors*: Afaisandier, Bkcase, Blawson, Dfairley, Dhenry, Dholwell, Janthony, Jgercken, Jmartin, Kguillemette, Mhaas, Skmackin, Smenck2, Smurawski, WikiWorks753, Wikiexpert

The Enterprise as a System *Source*: https://www.sebokwiki.org/d/index.php?oldid=57391 *Contributors*: Bkcase, Blawson, Dhenry, Dholwell, Janthony, Jdahmann, Jgercken, Jmartin, Kguillemette, Mhaas, Skmackin, Smenck2, Smurawski, WikiWorks, WikiWorks753, Wikiexpert, Zamoses

Related Business Activities *Source*: https://www.sebokwiki.org/d/index.php?oldid=57192 *Contributors*: Apyster, Bkcase, Blawson, Dfairley, Dhenry, Dholwell, Janthony, Jgercken, Jmartin, Kguillemette, Mhaas, Skmackin, Smenck2, Smurawski, WikiWorks, WikiWorks753, Wikiexpert, Zamoses

Enterprise Systems Engineering Key Concepts *Source*: https://www.sebokwiki.org/d/index.php?oldid=57193 *Contributors*: Afaisandier, Bkcase, Blawson, Dhenry, Dholwell, Janthony, Jgercken, Jmartin, Kguillemette, Mhaas, Skmackin, Smenck2, Smurawski, WikiWorks753, Wikiexpert, Zamoses

Enterprise Systems Engineering Process Activities *Source*: https://www.sebokwiki.org/d/index.php?oldid=57194 *Contributors*: Afaisandier, Asquires, Bkcase, Blawson, Dhenry, Dholwell, Janthony, Jgercken, Jmartin, Kguillemette, Mhaas, Skmackin, Smenck2, Smurawski, WikiWorks753, Wikiexpert, Zamoses

Enterprise Capability Management *Source*: https://www.sebokwiki.org/d/index.php?oldid=57195 *Contributors*: Afaisandier, Asquires, Bkcase, Blawson, Dhenry, Janthony, Jgercken, Jmartin, Kguillemette, Mhaas, Skmackin, Smenck2, Smurawski, WikiWorks753, Wikiexpert, Zamoses

Systems of Systems (SoS) *Source*: https://www.sebokwiki.org/d/index.php?oldid=57422 *Contributors*: Apyster, Bkcase, Blawson, Dhenry, Dholwell, Eleach, Henshaw, Janthony, Jdahmann, Jgercken, Kguillemette, Mhaas, Radcock, Skmackin, Smenck2, Smurawski, WikiWorks, Wikiexpert, Zamoses

Architecting Approaches for Systems of Systems *Source*: https://www.sebokwiki.org/d/index.php?oldid=57197 *Contributors*: Apyster, Bkcase, Blawson, Dhenry, Dholwell, Eleach, Henshaw, Janthony, Jdahmann, Jgercken, Kguillemette, Mhaas, Skmackin, WikiWorks753, Wikiexpert, Zamoses

Socio-Technical Features of Systems of Systems *Source*: https://www.sebokwiki.org/d/index.php?oldid=57481 *Contributors*: Afaisandier, Bkcase, Blawson, Dhenry, Dholwell, Eleach, Hdavidz, Henshaw, Janthony, Jdahmann, Jgercken, Kguillemette, Mhaas, Radcock, Skmackin, Smurawski, WikiWorks, WikiWorks753, Wikiexpert, Zamoses

Capability Engineering *Source*: https://www.sebokwiki.org/d/index.php?oldid=57199 *Contributors*: Apyster, Bkcase, Blawson, Dhenry, Dholwell, Eleach, Henshaw, Janthony, Jdahmann, Jgercken, Kguillemette, Mhaas, Skmackin, WikiWorks753, Wikiexpert, Zamoses

Healthcare Systems Engineering *Source*: https://www.sebokwiki.org/d/index.php?oldid=56743 *Contributors*: Bkcase, Nicole.hutchison, Radcock, WikiWorks753

Overview of the Healthcare Sector *Source*: https://www.sebokwiki.org/d/index.php?oldid=57200 *Contributors*: Bkcase, Mhaas, Nicole.hutchison, Radcock, WikiWorks, WikiWorks753

Systems Engineering in Healthcare Delivery *Source*: https://www.sebokwiki.org/d/index.php?oldid=57489 *Contributors*: Bkcase, Mhaas, Nicole.hutchison, WikiWorks, WikiWorks753

Systems Biology *Source*: https://www.sebokwiki.org/d/index.php?oldid=57387 *Contributors*: Bkcase, Mhaas, Nicole.hutchison, WikiWorks, WikiWorks753

Lean in Healthcare *Source*: https://www.sebokwiki.org/d/index.php?oldid=57204 *Contributors*: Bkcase, Mhaas, Nicole.hutchison

Image Sources, Licenses and Contributors

File:Rob_Sm for Web.jpg *Source*: https://www.sebokwiki.org/d/index.php?title=File:Rob_Sm_for_Web.jpg *License*: unknown *Contributors*: Rcloutier

File:RobSignature2.jpeg *Source*: https://www.sebokwiki.org/d/index.php?title=File:RobSignature2.jpeg *License*: unknown *Contributors*: Bkcase

File:Hutchison profilephoto.png *Source*: https://www.sebokwiki.org/d/index.php?title=File:Hutchison_profilephoto.png *License*: unknown *Contributors*: -

File:PSE_Intro_Fig1.png *Source*: https://www.sebokwiki.org/d/index.php?title=File:PSE_Intro_Fig1.png *License*: unknown *Contributors*: Smenck2, Smurawski

File:PSE_Intro_Fig2.png *Source*: https://www.sebokwiki.org/d/index.php?title=File:PSE_Intro_Fig2.png *License*: unknown *Contributors*: Smenck2, Smurawski

File:PSE_Intro_Fig3.png *Source*: https://www.sebokwiki.org/d/index.php?title=File:PSE_Intro_Fig3.png *License*: unknown *Contributors*: Smenck2, Smurawski

File:PSE_Intro_Fig4.png *Source*: https://www.sebokwiki.org/d/index.php?title=File:PSE_Intro_Fig4.png *License*: unknown *Contributors*: Smenck2, Smurawski

File:IPDP PSE Background Figure 1.png *Source*: https://www.sebokwiki.org/d/index.php?title=File:IPDP_PSE_Background_Figure_1.png *License*: unknown *Contributors*: Smenck2, Smurawski

File:PSE_PSEB_Fig2.png *Source*: https://www.sebokwiki.org/d/index.php?title=File:PSE_PSEB_Fig2.png *License*: unknown *Contributors*: Smenck2, Smurawski

File:PSE_PAAS_Fig1.png *Source*: https://www.sebokwiki.org/d/index.php?title=File:PSE_PAAS_Fig1.png *License*: unknown *Contributors*: Smenck2, Smurawski

File:PSE_PAAS_Fig2.png *Source*: https://www.sebokwiki.org/d/index.php?title=File:PSE_PAAS_Fig2.png *License*: unknown *Contributors*: Smenck2, Smurawski

File:PSE_PSEKA_Fig1.png *Source*: https://www.sebokwiki.org/d/index.php?title=File:PSE_PSEKA_Fig1.png *License*: unknown *Contributors*: Smenck2, Smurawski

File:PSE_PSEKA_Fig2.png *Source*: https://www.sebokwiki.org/d/index.php?title=File:PSE_PSEKA_Fig2.png *License*: unknown *Contributors*: Smenck2, Smurawski

File:PSE_PSEKA_Fig3.png *Source*: https://www.sebokwiki.org/d/index.php?title=File:PSE_PSEKA_Fig3.png *License*: unknown *Contributors*: Smenck2, Smurawski

File:PSE_PSEKA_Fig4.png *Source*: https://www.sebokwiki.org/d/index.php?title=File:PSE_PSEKA_Fig4.png *License*: unknown *Contributors*: Smenck2, Smurawski

File:PSE_PSESA_Fig1.png *Source*: https://www.sebokwiki.org/d/index.php?title=File:PSE_PSESA_Fig1.png *License*: unknown *Contributors*: Smenck2, Smurawski

File:SSE_SSB_Fig1.png *Source*: https://www.sebokwiki.org/d/index.php?title=File:SSE_SSB_Fig1.png *License*: unknown *Contributors*: Smenck2, Smurawski

File:SSE_FOS_Fig1_no_white_space.png *Source*: https://www.sebokwiki.org/d/index.php?title=File:SSE_FOS_Fig1_no_white_space.png *License*: unknown *Contributors*: Smenck2

File:SSE_FOS_Fig3.png *Source*: https://www.sebokwiki.org/d/index.php?title=File:SSE_FOS_Fig3.png *License*: unknown *Contributors*: Smenck2, Smurawski

File:SSE_FOS_Fig4.PNG *Source*: https://www.sebokwiki.org/d/index.php?title=File:SSE_FOS_Fig4.PNG *License*: unknown *Contributors*: Smenck2, Smurawski

File:SSE_VoSSE_Fig1v2.png *Source*: https://www.sebokwiki.org/d/index.php?title=File:SSE_VoSSE_Fig1v2.png *License*: unknown *Contributors*: Smenck2, Smurawski

File:ESE-F01.png *Source*: https://www.sebokwiki.org/d/index.php?title=File:ESE-F01.png *License*: unknown *Contributors*: Smenck2, Smurawski

File:ESE-F02.png *Source*: https://www.sebokwiki.org/d/index.php?title=File:ESE-F02.png *License*: unknown *Contributors*: Smenck2, Smurawski

File:ESE-F03.png *Source*: https://www.sebokwiki.org/d/index.php?title=File:ESE-F03.png *License*: unknown *Contributors*: Smenck2, Smurawski

File:ESE-F04.png *Source*: https://www.sebokwiki.org/d/index.php?title=File:ESE-F04.png *License*: unknown *Contributors*: Smenck2, Smurawski

File:ESE-F05.png *Source*: https://www.sebokwiki.org/d/index.php?title=File:ESE-F05.png *License*: unknown *Contributors*: Smenck2, Smurawski

File:ESE-F07.png *Source*: https://www.sebokwiki.org/d/index.php?title=File:ESE-F07.png *License*: unknown *Contributors*: Smenck2, Smurawski

File:ESE-F08.png *Source*: https://www.sebokwiki.org/d/index.php?title=File:ESE-F08.png *License*: unknown *Contributors*: Smenck2, Smurawski

File:ESE-F09.png *Source*: https://www.sebokwiki.org/d/index.php?title=File:ESE-F09.png *License*: unknown *Contributors*: Smenck2, Smurawski

File:ESE-F10.png *Source*: https://www.sebokwiki.org/d/index.php?title=File:ESE-F10.png *License*: unknown *Contributors*: Smenck2, Smurawski

File:ESE-F11.png *Source*: https://www.sebokwiki.org/d/index.php?title=File:ESE-F11.png *License*: unknown *Contributors*: Smenck2, Smurawski

File:ESE-F12.png *Source*: https://www.sebokwiki.org/d/index.php?title=File:ESE-F12.png *License*: unknown *Contributors*: Smenck2, Smurawski

File:ESE-F13.png *Source*: https://www.sebokwiki.org/d/index.php?title=File:ESE-F13.png *License*: unknown *Contributors*: Smenck2, Smurawski

File:ESE-F15.png *Source*: https://www.sebokwiki.org/d/index.php?title=File:ESE-F15.png *License*: unknown *Contributors*: Smenck2, Smurawski

File:ESE-F06.png *Source*: https://www.sebokwiki.org/d/index.php?title=File:ESE-F06.png *License*: unknown *Contributors*: Smenck2, Smurawski

File:ESE-F30_Process_Breakdown.png *Source*: https://www.sebokwiki.org/d/index.php?title=File:ESE-F30_Process_Breakdown.png *License*: unknown *Contributors*: Smenck2, Smurawski

File:ESE-F21.png *Source*: https://www.sebokwiki.org/d/index.php?title=File:ESE-F21.png *License*: unknown *Contributors*: Smenck2, Smurawski

File:ESE-F22.png *Source*: https://www.sebokwiki.org/d/index.php?title=File:ESE-F22.png *License*: unknown *Contributors*: Smenck2, Smurawski

File:Use Cases ISO 14971 15288.png *Source*: https://www.sebokwiki.org/d/index.php?title=File:Use_Cases_ISO_14971_15288.png *License*: unknown *Contributors*: Bkcase