



# Guide to the Systems Engineering Body of Knowledge (SEBoK) version 0.75

## Part 4

Please note that this is a PDF extraction of the content from [www.sebokwiki.org](http://www.sebokwiki.org). The content contained here is extracted in the order listed under the [SEBoK Table of Contents \(TOC\)](#). **Note:** Some graphics may appear distorted due to the PDF processing. Please visit [www.sebokwiki.org](http://www.sebokwiki.org) for a clear version of the graphic.

*The SEBoK wiki and all articles included in it (including those within this PDF) are copyright 2012 by Stevens Institute of Technology. All rights reserved. Images used by permission remain the property of their respective copyright holders.*

You may use the contents of this wiki and all articles included in it on the conditions that (a) you acknowledge that your use is with permission of Stevens Institute of Technology, and (b) you indemnify and hold harmless Stevens Institute of Technology, the editors, and the authors, from any and all liability or damages to yourself or your hardware, software, systems, or third parties, including attorneys' fees, court costs, and other related costs and expenses, arising out of your use of this wiki and all articles included in it, irrespective of the cause of said liability.

Stevens Institute of Technology makes this wiki and all articles included in it available on an "as is" basis and makes no warranty, expressed or implied, as to their accuracy, capability, efficiency, merchantability, or functioning. In no event will Stevens Institute of Technology, the editors, or the authors be liable for any general, consequential, indirect, incidental, exemplary, or special damages, even if they have been advised of the possibility of such damages.

# Part 4

From .75 Sebokcase

## Part 4

- Part 4: Applications of Systems Engineering
  - Knowledge Area: Product Systems Engineering
    - Topic: Product Systems Engineering Background
    - Topic: Product as a System Fundamentals
    - Topic: Business Activities Related to Product Systems Engineering
    - Topic: Product Systems Engineering Key Aspects
    - Topic: Product Systems Engineering Special Activities
  - Knowledge Area: Service Systems Engineering
    - Topic: Service Systems Background
    - Topic: Fundamentals of Services
    - Topic: Properties of Services
    - Topic: Scope of Service Systems Engineering
    - Topic: Value of Service Systems Engineering
    - Topic: Service Systems Engineering Stages
  - Knowledge Area: Enterprise Systems Engineering
    - Topic: Enterprise Systems Engineering Background
    - Topic: The Enterprise as a System
    - Topic: Related Business Activities
    - Topic: Enterprise Systems Engineering Key Concepts
    - Topic: Enterprise Systems Engineering Process Activities
    - Topic: Enterprise Capability Management
  - Knowledge Area: Systems of Systems (SoS)
    - Topic: Architecting Approaches for Systems of Systems
    - Topic: Socio-Technical Features of Systems of Systems
    - Topic: Capability Engineering

## Part 4: Applications of Systems Engineering

This part of the SEBoK focuses on the application of systems engineering to the creation and life cycle management (glossary) of various types of systems. In particular, the engineering of product systems (glossary), service systems (glossary), enterprise systems (glossary) as well as System of Systems (SoS) (glossary).

### Knowledge Areas

Part 4 contains the following knowledge areas:

- Product Systems Engineering
- Service Systems Engineering
- Enterprise Systems Engineering
- Systems of Systems (SoS)

### Systems Engineering Application Paradigms

Part 3 Systems Engineering and Management introduces a paradigm that identifies the general goal of all systems engineering

efforts; that is the transformation of the specific stakeholder need into a system product or service that provides for the need. Given this framework, the various knowledge areas of Part 3 provide deeper insight into how to accomplish activities related to the systems engineering life cycle.

The following paradigms provide insight into the primary commonalities and differentiations of the various applications of systems engineering presented here in Part 4. The commonality is that all products and services are developed to provide for the capabilities needed in an operational environment in the form of activities. The supply of the products and services and the nature of the operational environment differentiate the applications.

## Product / Service Provisioning

The paradigm in Figure 1 illustrates the main result of Traditional Systems Engineering (TSE), that is, the supply of products and services that fulfill the capability needs as activities in an operational environment. In TSE, the operational environment is often described by some form of Concept of Operations (ConOps) (glossary) that can include use cases (glossary) of the activities of the system in operation.

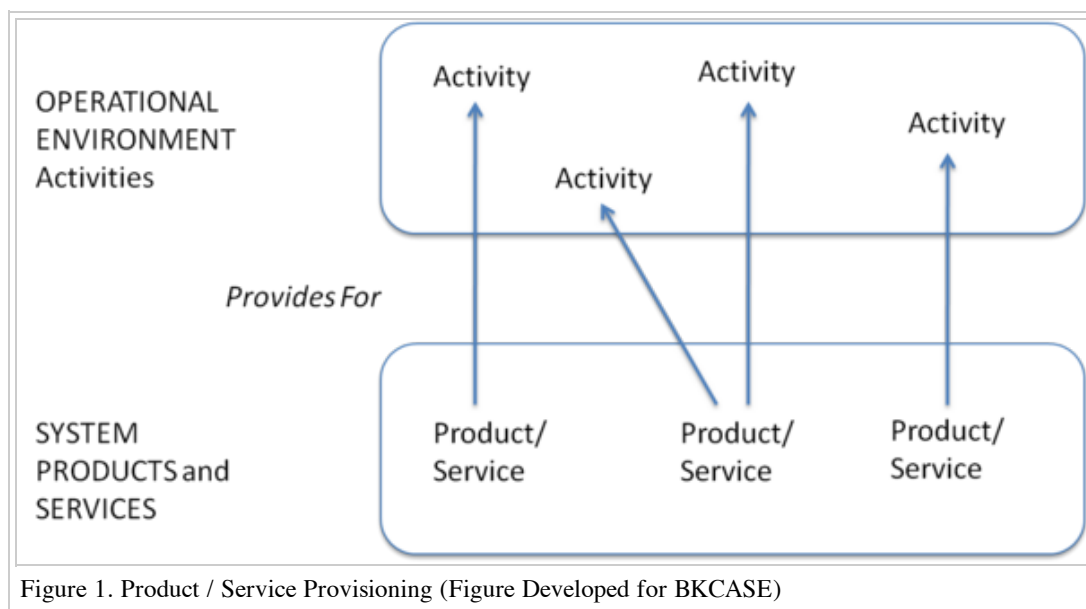


Figure 1. Product / Service Provisioning (Figure Developed for BKCASE)

It is important to take into account the binding (glossary) of the products and services to the activities. As implied by the figure, the operational environmental needs are met by directly binding products and/or services to the needs of one or more of the activities of the operational environment. In a manner consistent with ISO/IEC 15288, the products and services are composed of systems in which hardware, software, and human elements are supplied. Thus a mixture of physical products (often including software) and services such as logistics, help-desk functions, etc., may be provided. The supply of products and services can be a result of a direct acquisition or the result of acquiring products or services that have been provided for a marketplace. Note that the provisioning of products and services when humans are involved, like in a hospital, may even be classified as sociotechnical systems (glossary).

## Service System Provisioning

The paradigm in Figure 2 illustrates the provision of Service Systems. The growth of this model has been stimulated to a large extent due to the widespread availability of IT services such as Service Oriented Architectures (SOA), Software as a System (SaS), and Cloud Computing.



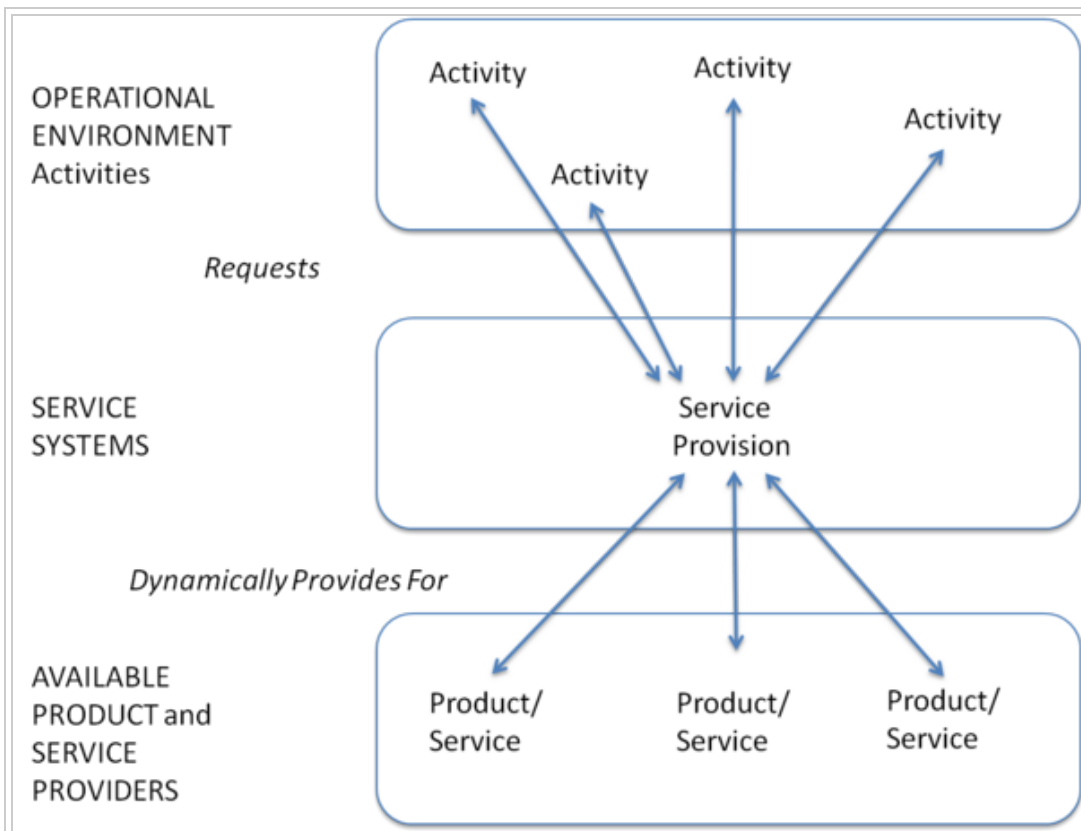


Figure 2. Service System Provisioning (Figure Developed for BKCASE)

A primary difference between Service Systems Engineering (SSE) and TSE is the dynamic binding (glossary) of products and services, potentially from multiple providers, in order to provide a desired service. Thus, the service is dynamically composed from a given service request. In the background there are a variety of hardware, software, and even human elements that provide the elements of a service. In contrast to an operational environment that is sustained, the service requested may be viewed as a temporary environment established to provide the service and it disappears after the service has been provided.

## Enterprise System Provisioning

The paradigm in Figure 3 illustrates the provision of Enterprise Systems Engineering (ESE) to meet the capability needs in the form of enterprise activities.

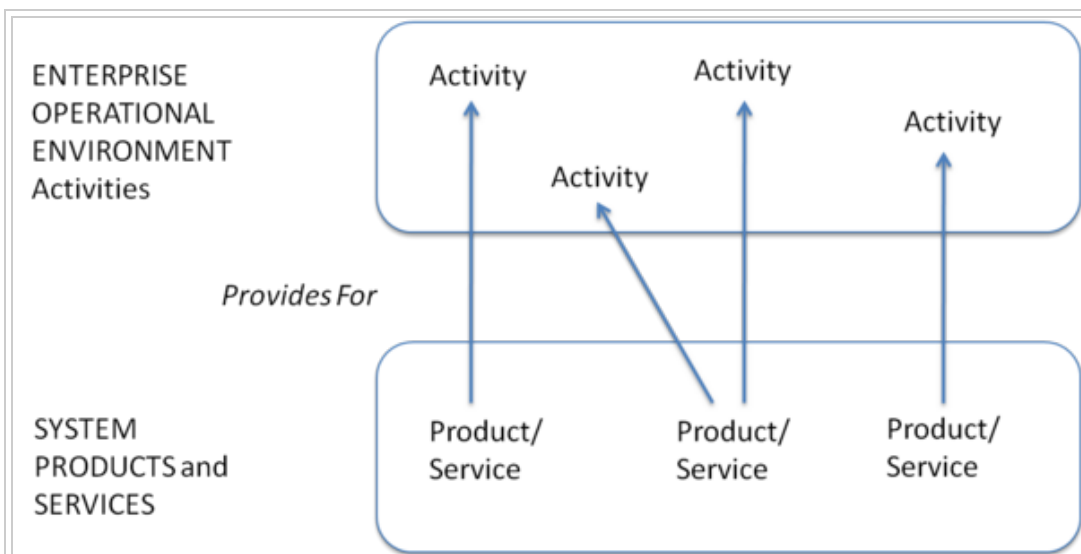


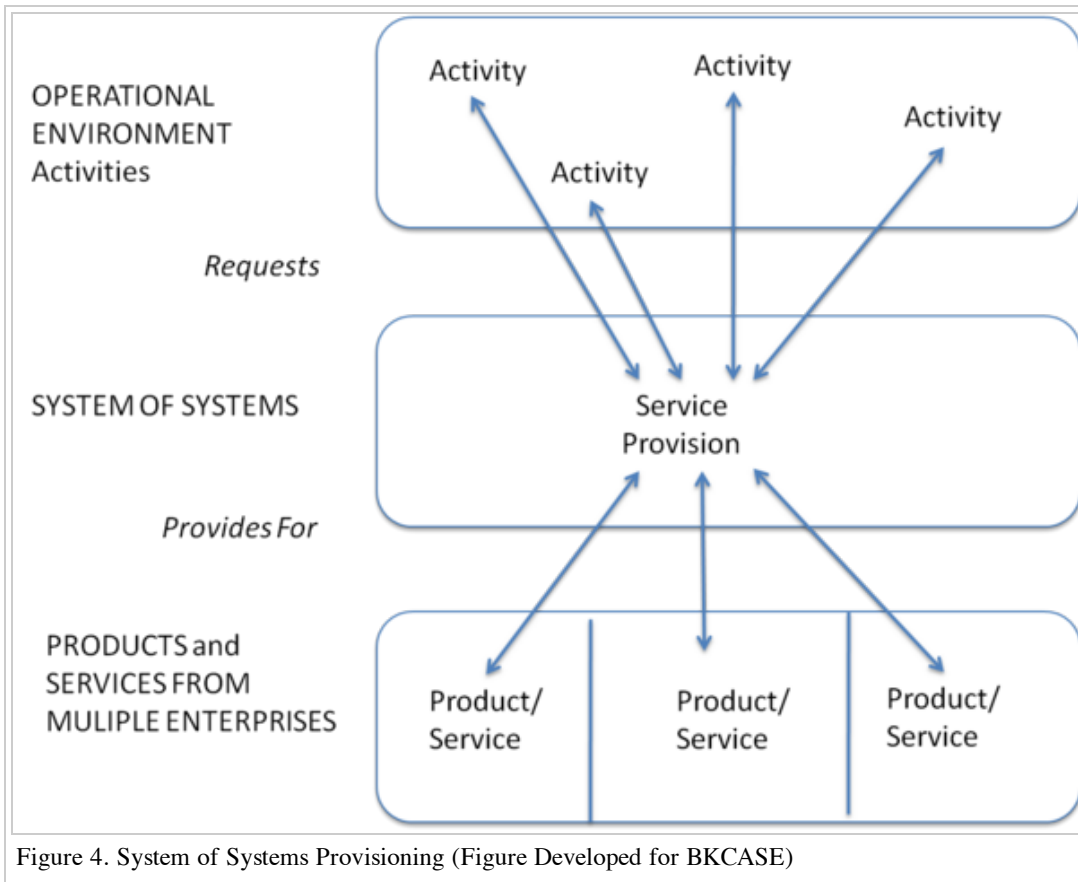
Figure 3. Enterprise System Provisioning (Figure Developed for BKCASE)

This form is similar to TSE except that the activities of the operational environment typically rely upon services provided by

Systems of Processes (both business processes (glossary) and system life cycle processes) that are essential in operating the enterprise. In addition, the products and services that are supplied to the enterprise include those that support their own value added production of products and services as well as their infrastructure systems.

## System of Systems Provisioning

Finally, the paradigm in Figure 4 illustrates the provisioning of products and services by multiple enterprises in forming a System of Systems (SoS) (glossary) that is created to meet the complex needs of an operational environment. Note that this could be meeting a crisis situation or meeting the needs of a new type of operation where multiple enterprises provide a new form of added value based upon their combined products and services.



All of the enterprises that contribute to an SoS are independent and provide value-added products or services that are meaningful individually.

System of Systems Engineering (SoSE), particularly when responding to some form of natural or manmade crisis, is initiated in the form of a request for service in order to meet a complex situation that has arose. In relationship to the system coupling diagram SoSE involves accelerated engineering of the respondent system that will interact with the situation system. Thus, instantiated system assets in the form of products and/or services are supplied from multiple enterprises and quickly integrated in providing the respondent service. In a manner similar to Service Systems, the operational environment disappears when the crises has been handled. SoSE is also applied in merging the operations of multiple enterprises in order to provide some new form of service. In contrast to unplanned crises, this is the result of planned development. The formation of the Department of Homeland Security in the USA is an example. In the private sector, multiple organizations may integrate their enterprise operations in order to meet an opportunity in a new marketplace.

## References

### Works Cited

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.

## Primary References

No primary references have been identified for version 0.75. Please provide any recommendations on additional references in your review.

## Additional References

No additional references have been identified for version 0.75. Please provide any recommendations on additional references in your review.

---

< Previous Article | Parent Article | Next Article >

## Comments from SEBoK 0.5 Wiki

No comments were logged for this article in the SEBoK 0.5 wiki. Because of this, it is especially important for reviewers to provide feedback on this article. Please see the discussion prompts below.

## SEBoK 0.75 Reviews



Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

Like

Add New Comment

Post as ...

Showing 0 comments

Sort by Popular now  Subscribe by email  Subscribe by RSS

blog comments powered by **DISQUS**

# Product Systems Engineering

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 markets segments. A market can be consumer based (e.g. private enterprises or general consumers) or it can be public (not-for-profit), that addresses the strategic needs of a country/region (military, healthcare, educational, transportation, energy, etc). NPDP has two significantly overlapping and integrated activities:

1. Systems Engineering: which includes concept generation, engineering design/development, and deployment
2. Market Development: which 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.

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, and 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 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, and include aspects such as: Acquired vs. Offered Products, Product Lifecycle & Adoption rates, Integrated Product Teams (IPTs) and Integrated Product Development Teams (IPDTs), product architectures, requirement 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 to product deployment.

## Topics

The Product Systems Engineering knowledge area 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

## Key Terms and Concepts

### Product

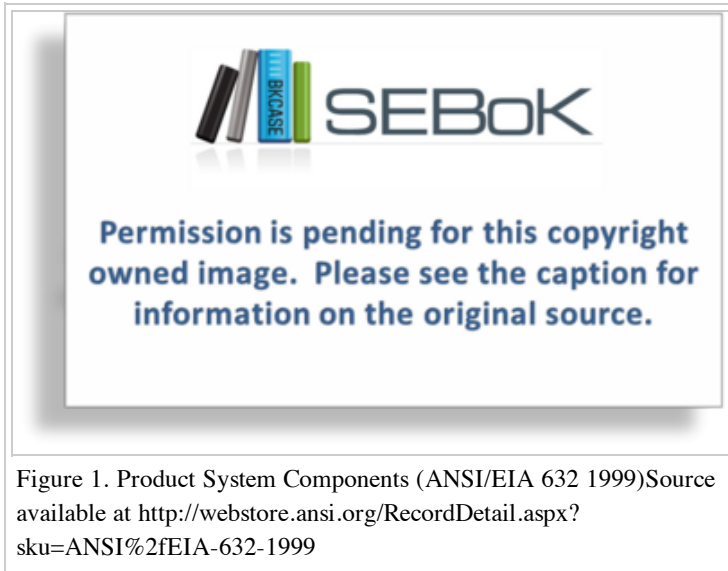
A product is an artifact that is created by some person or by some process such as a manufacturing process, software source code compilation and integration, building construction, creative writing process, or data processing.

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 has referred to *thing or things produced*.

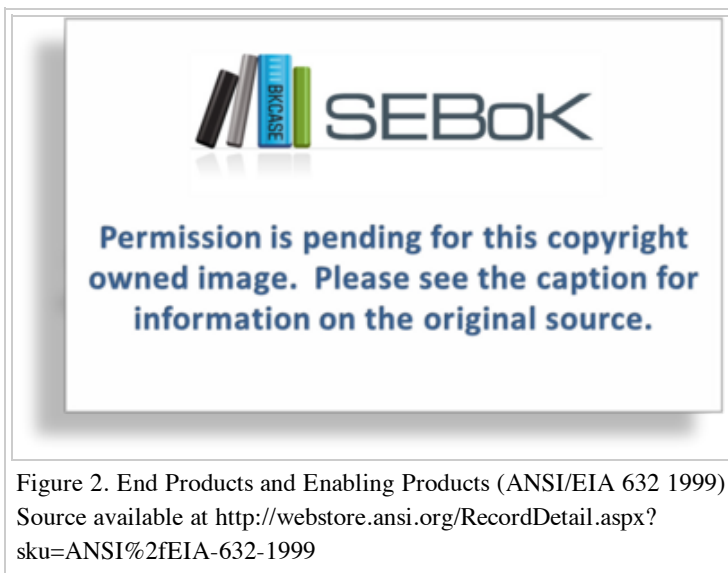
In economics and commerce, products belong to a broader category of *goods*. The economic meaning of product was first used by political economist Adam Smith. In marketing, a product is anything that can be offered to a market that might satisfy a want or a need. In retailing, products are called merchandise. 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 1998 standard, the just term *system* but the scope of the standard is clearly restricted to product systems.



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 warranted.



## 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 (glossary). 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.

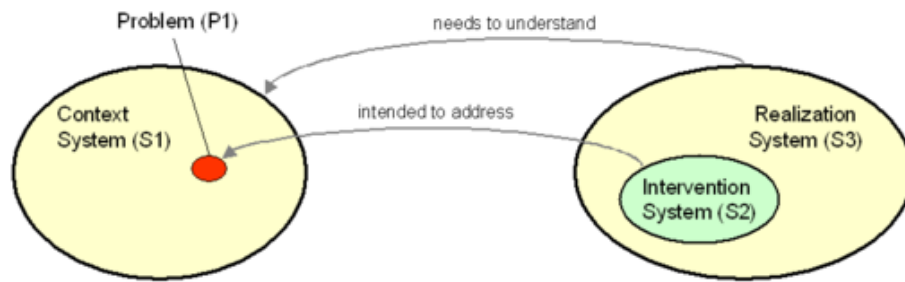


Figure 3. Realization system that creates the intervention to solve a problem (Martin 2004) Figure 3.  
Realization system that creates the intervention to solve a problem (Martin 2004) Reprinted with  
Permission of Aerospace

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 area 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.

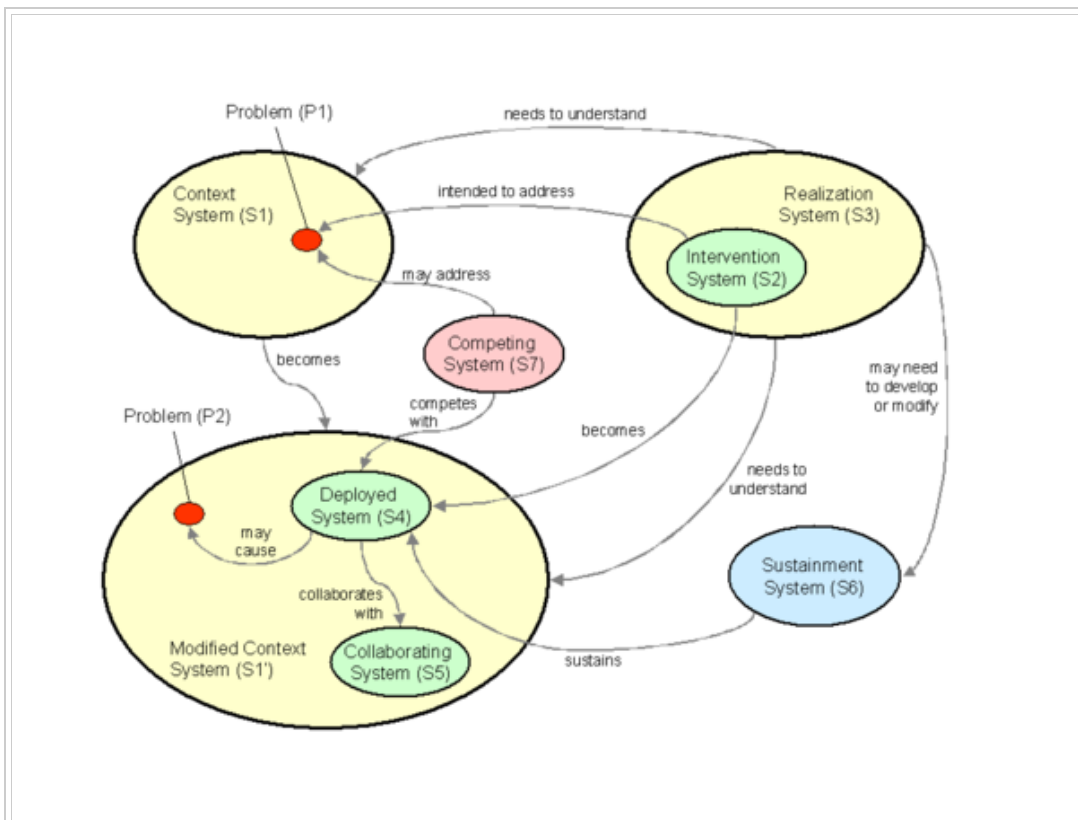


Figure 4. Product sustaintment system in support of the deployed product system (Martin 2004)

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 subassemblies 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. These three levels are:

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

Value will only be fully realized 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. 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) (glossary) 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 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 a method entitled '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 632-1998.
- 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, Klein, M., and Clements, P. 2000. ATAM; Method for Architecture Evaluation, Technical Report. CMU/SEI-2000-TR-004, ESC-TR-2000-004.
- Lawson, H. 2010. *A Journey Through the Systems Landscape*. London, UK: College Publications, Kings College, UK.
- Magrab, E., Gupta S., McCluskey, P., and Sandborn, P. 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, Toulouse, France, 2004.
- Phillips, F. 2001. *Market-oriented Technology Management - Innovating for Profit in Entrepreneurial Times*. Berlin, Germany: Springer-Verlag. ISBN 3-540-41258-1
- Pugh, S. 1990. *Total Design: Integrated Methods for Successful Product Engineering*. Upper Saddle River, NJ, USA: Prentice Hall. ISBN-10: 0201416395, ISBN-13: 978-0201416398.
- Ring, J., 1998. "A Value Seeking Approach to the Engineering of Systems." Proceedings of the IEEE Conference on Systems, Man, and Cybernetics. p. 2704-2708.
- Sage, A., and Rouse, W. (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 Clark, K. 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 632-1998.

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

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

## Additional References

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, Klein, M., and Clements, P. 2000. ATAM; Method for Architecture Evaluation, Technical Report. CMU/SEI-2000-TR-004, ESC-TR-2000-004.

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

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, Toulouse, France, 2004.

Phillips, F. 2001. *Market-oriented Technology Management - Innovating for Profit in Entrepreneurial Times*. Berlin, Germany: Springer-Verlag. ISBN 3-540-41258-1

Pugh, S. 1990. *Total Design: Integrated Methods for Successful Product Engineering*. Upper Saddle River, NJ, USA: Prentice Hall. ISBN-10: 0201416395, ISBN-13: 978-0201416398.

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

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

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

---

< Previous Article | Parent Article | Next Article >

## Comments from SEBoK 0.5 Wiki

No comments were logged for this article in the SEBoK 0.5 wiki. Because of this, it is especially important for reviewers to provide feedback on this article. Please see the discussion prompts below.

## SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

# Product Systems Engineering Background

## Product Types

A system is by definition composed of components. The system itself is a component of a larger system, and you can also view each component as a system on its own.

A single component consists of 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 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 components to include? 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 hypercompetitive 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.

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

## Relationship between Product SE 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 lifecycle perspective, “from cradle to grave” or “lust 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 (IOC) 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 of four scenarios (Phillips 2001):

- Use well-established technologies to help the enterprise improve the efficiency of current operations
- Use well-established technologies to help the enterprise into new kind of operations
- Use leading edge technologies to improve the efficiency of current operations or
- 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. Or, it may possess new or different characteristics that offer new or additional benefits to the customer. Finally, it may be entirely new and satisfy a newly-defined customer want or market niche. <http://www.businessdictionary.com/definition/product-development.html>

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 rapid response.

Products themselves are now multidisciplinary in nature, meaning that product development must have close participation not only from the different specialty engineering fields (mechanical, electrical, industrial, materials, and so on) but also from finance to analyze total cost of development (including production), marketing and sales (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, for example by deploying Integrated Product Teams (IPT) and Integrated Product Development Teams (IPDT).

## Product Systems Engineering

Product Systems Engineering strives for efficient use of company resources in order to achieve business objectives and deliver a quality product. PSE 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 analysis of the product and its related elements such as manufacturing, maintenance, support, logistics, phase-out, and disposal, all of 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 MPT 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:

- Determining the overall scope of needs of the product system
- Defining product and system requirements
- Considering all interactions between the different elements of the product system
- Organizing and integrating engineering disciplines
- 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 SE professionals on the need for the SE process to become more rapidly adaptable. Platform-based solutions to resolve some of these challenges (Infrastructure as a Service, Platform as a Service, Software as a Service) are being studied and proposed (MITRE 2010) (Boehm 2010).

## Product Development Process

The Integrated Product Development Process (IPDP) process starts with customer/market needs with the objective to:

- Deliver products that satisfy and exceed customer expectations
- Rapid response to customer demands through adaptive product offerings
- Respond to changing business environments
- Incorporate systems thinking, generate new ideas, and co-create value

IPDP methods are continuous in nature with a goal to produce products whose cost, performance, features, time-to-market help increase company profitability and market share. Figure 1 provides a snapshot of an Integrated Product Development Process (IPDP). The IPDP is divided into four stages (Magrab et al. 2010)

### Stage I: Product Identification

During this stage, the enterprise aims to identify a product idea that will be a good business investment for the company. Some of the outputs from a good Product Identification Stage include demonstration of strong customer need, determination of potential markets, business profitability, and sustainable product competitive advantage.

During this stage the SE process plays a key role working with Product Managers and the IPT to have back of the envelope input to assess needed technology innovation, viability of existing technologies, estimated time of development and cost of technology development, technology risk, propose technology development road map and functionality/features releases if any, etc. This initial assessment uses Rough Order of Magnitude (ROM) estimates within +/- 40% taken by the Product Managers to analyze the feasibility of a Business Case and come to a decision on entering the Concept Development stage, also known as Decision Gate -1

### Stage II: Concept Development

The main goal of the Concept Development stage is to generate and develop Methods, Processes, and Technologies (MPTs) concepts that will satisfy the product's performance goals. Some of the evaluation criteria that these concepts must fulfill are:

- Company's core competencies can satisfy the requirements to produce the products
- Low technical and business risk
- Minimal change in market conditions and competitors
- Manufacturing resource requirements are close to planned allocations
- Prototypes indicate product's economic viability and manufacturing feasibility

During this stage SE supports the IPDT in the analysis of alternatives (different concept solutions), identifying different operational scenarios and modes of operation, identifying the functional requirements of the products, defining Technical Performance Measures (TPM), identifying technology risks and performance risks, identifying the main components of the products and required interfaces among them, etc. This stage is highly interacting and iterative among several IPDT and in some instance because of the complexity of the products a Systems Engineering Integration Team is required to ensure analysis of all the possible solutions.

Techno-economical feasibility in terms of Time and Cost to develop the product are usually within +/-10%; in many instances it is possible at this stage to already have a Technology Roadmap developed to guide the product managers on possible phases for product releases. At this stage a decision is made to continue or not with the full development of the product according to a Product Roadmap jointly worked by the IPT through joint Preliminary Design Reviews.

### Stage III: Design and Manufacturing

This stage includes creation of engineering drawings for the product, product configuration items specs, design for manufacturability/producibility, Design for X (DFX), manufacturing design plans, production plans and schedules, completion of a test production run that ensures that product meets customer requirements and quality criteria, and a plan for full production.

PSE works closely with the Projects Managers and product managers to create a Systems Engineering Management Plan (SEMP) to manage the technical effort. Requirements traceability; product architecture requirements and views; Operational requirements; Integration, Verification and product Validation plans; Modeling, Simulation and test & evaluation of the Product system under different scenarios to evaluate TPM; launch readiness plans including end-user test plans, operational readiness, etc. are many of the activities carried out by SE.

### Stage IV: Launch

During this stage the product is delivered to its potential markets. A launch is considered successful when the product meets its quality goals, satisfies customer requirements, and realizes the business plan goals. PSE plays a consultant role with the product manager for the analysis and validation of TPM, test results and accuracy, and during the Continuous improvement process to monitor product and product system technical performance and product quality.



Figure 1. Integrated Product Development Process (Magrab, et. al. 2009) Source Available at <http://www.crcpress.com/product/isbn/9781420070606>

## Relationship between Product SE and Technology Development

As technological advancement accelerates, product life cycles become shorter, especially for high technology products. As a result,

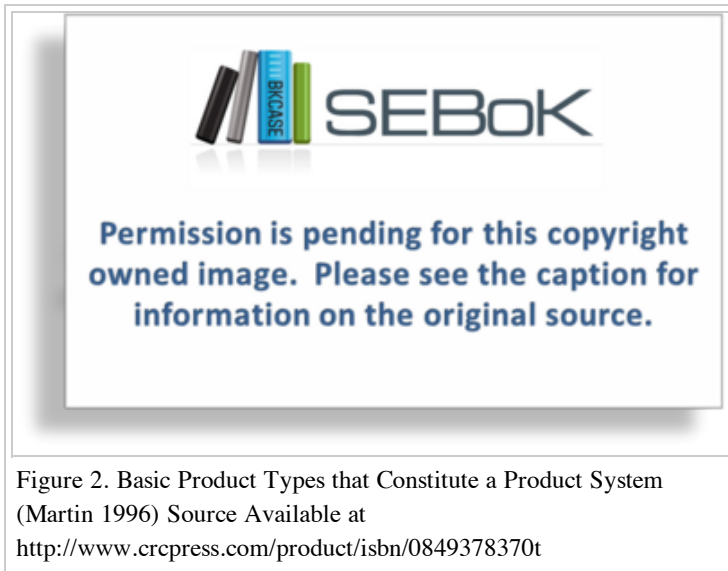
enterprises risk having outdated or obsolete products that have lost pace with markets trends, technology trends or customer expectations.

PSE should bring awareness of technology changes and trends to the analysis of new product ideas or innovations. This affects the Time & Cost inputs into the technical feasibility analysis of the product. The result should include a roadmap of required technology developments which is then used to create the overall roadmap for the new product offering.

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

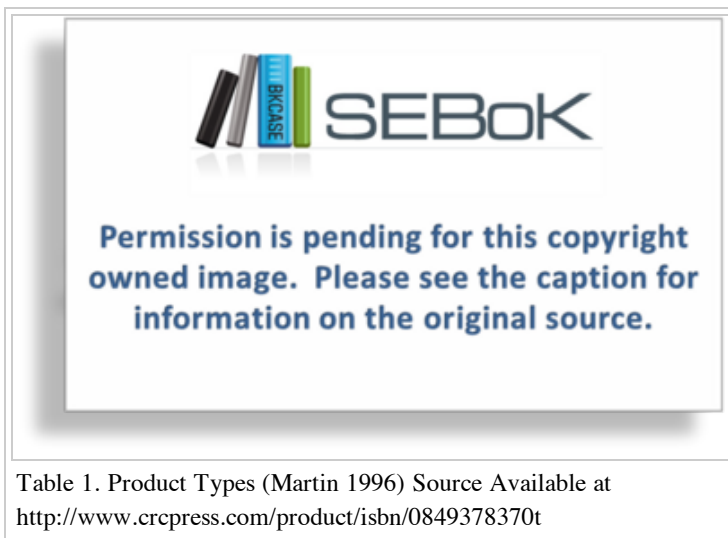
On the other hand, when technology developments or breakthroughs drive product innovation or the generation of new markets, we may say that technology developments generate requirements on product features and functionalities. Then, among the factors which dictate decisions about introducing products include Technology Readiness Levels (TRL), Integration Readiness Levels (IRL), Manufacturing Readiness Levels (MRL), System Readiness Levels (SRL), and Operational Readiness of the enterprise to launch the product system. See Product Readiness Article.

Understanding the entities that compose the product is a non-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 must be factored into the IPDP for the successful deployment and proper use of the product.



## Product Type Examples

Examples of each product type are shown below. (Martin 1996)



Materials could be thought of as basic raw materials, like steel, or as complex materials, like clad metals, graphite composites, or building aggregate material. Personnel are not normally thought of as a “product,” but that depends on which system you are working with. The NASA space program “system” certainly produces astronauts. It is not usually possible to just go out and hire the particular person you need 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. You could specify the interfaces for the person as well as assessing the maturity of that personnel “product.” These are a few examples of how Product SE can be applied to personnel products.

In Enterprise SE we may need education and training systems as a part of our personnel system in order to produce people with the right competencies and capabilities.

## References

### Works Cited

- 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. August 29, 2010.
- Grady, J. 2010. *Systems Synthesis- Product and Process Design*. Boca Raton, FL, USA: CRC Press.
- INCOSE. 2011. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, version 3.2.1. San Diego, CA, USA: International Council on Systems Engineering (INCOSE), INCOSE-TP-2003-002-03.2.1.
- Magrab, E., Gupta S., McCluskey, P., and Sandborn, P. 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 Babcock, D. 2007. *Managing Engineering and Technology*, fourth edition. Upper Saddle River, NJ, USA: Prentice Hall.
- Academy of Program/Project & Engineering Leadership (APPEL). 2009. NASA's Systems Engineering Competencies. Washington, D.C.: U.S. National Aeronautics and Space Association. Available at: [http://www.nasa.gov/offices/oce/appel/pm-development/pm\\_se\\_competency\\_framework.html](http://www.nasa.gov/offices/oce/appel/pm-development/pm_se_competency_framework.html)
- Phillips, F. 2001. *Market Oriented Technology Management: Innovating for Profit in Entrepreneurial Times*. Springer. 3-540-41258-1
- 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

No primary references have been identified for version 0.75. Please provide any recommendation on primary references in your review.

## Additional References

No additional references have been identified for version 0.75. Please provide any recommendation on additional references in your



review.

---

[< Previous Article](#) | [Parent Article](#) | [Next Article >](#)

## Comments from 0.5 Wiki

This article is new to the SEBoK for version 0.75. As such, there are no associated 0.5 comments.

### SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

## Product as a System Fundamentals

This article introduces fundamental concepts for studying product systems.

### Product Elements and Connections

Product systems consist of product elements and two kinds of connections: among elements, and 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 diagram, data flow diagram, free body diagram, interface control diagram, port specification, energy transfer diagram, 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 ten feet apart
- one element is inside another

Motion-related relationships:

- the relative velocity of two elements is X
- the relative acceleration between two elements is X

Temporal relationships:

- one element exists before another
- two elements must exist at the same time
- two elements must be separated in time by three years

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 roadmap, and project schedule chart.

### Social relationships

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 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.

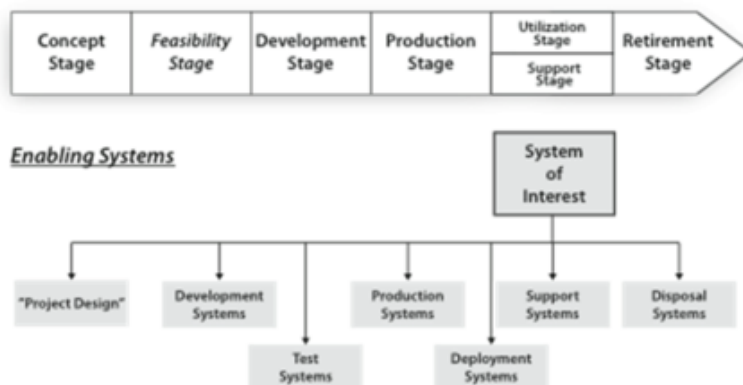


Figure 1. Example of Enabling Systems (Lawson 2010) Reprinted with permission of Harold "Bud" Lawson.

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 system of interest 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.
- Provided products are integrated into the system of systems.

## Product Architecture, Modeling & 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.” 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.”

A product's purpose (stakeholder's desire) is realized by a product system (the SOI). Because product systems are composed of different entities (components, assemblies, subsystems, information, facilities, processes, organizations, people) 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 & Validation plans created during the requirements phase are the basis of testing certify 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 purpose is articulated in terms of business objectives (market, cost, functionality, performance, 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 & sales personnel.

Different architectural Frameworks have been defined to guide product teams in defining the product architecture for commercial and for public enterprises. In general, an architectural framework describes a "view," meaning a "collection of models that represent the whole system from the perspective of a set of related stakeholder concerns."

Differences between acquired products and offered products play an important role in defining Product System requirements. Acquired products are lifecycle-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.

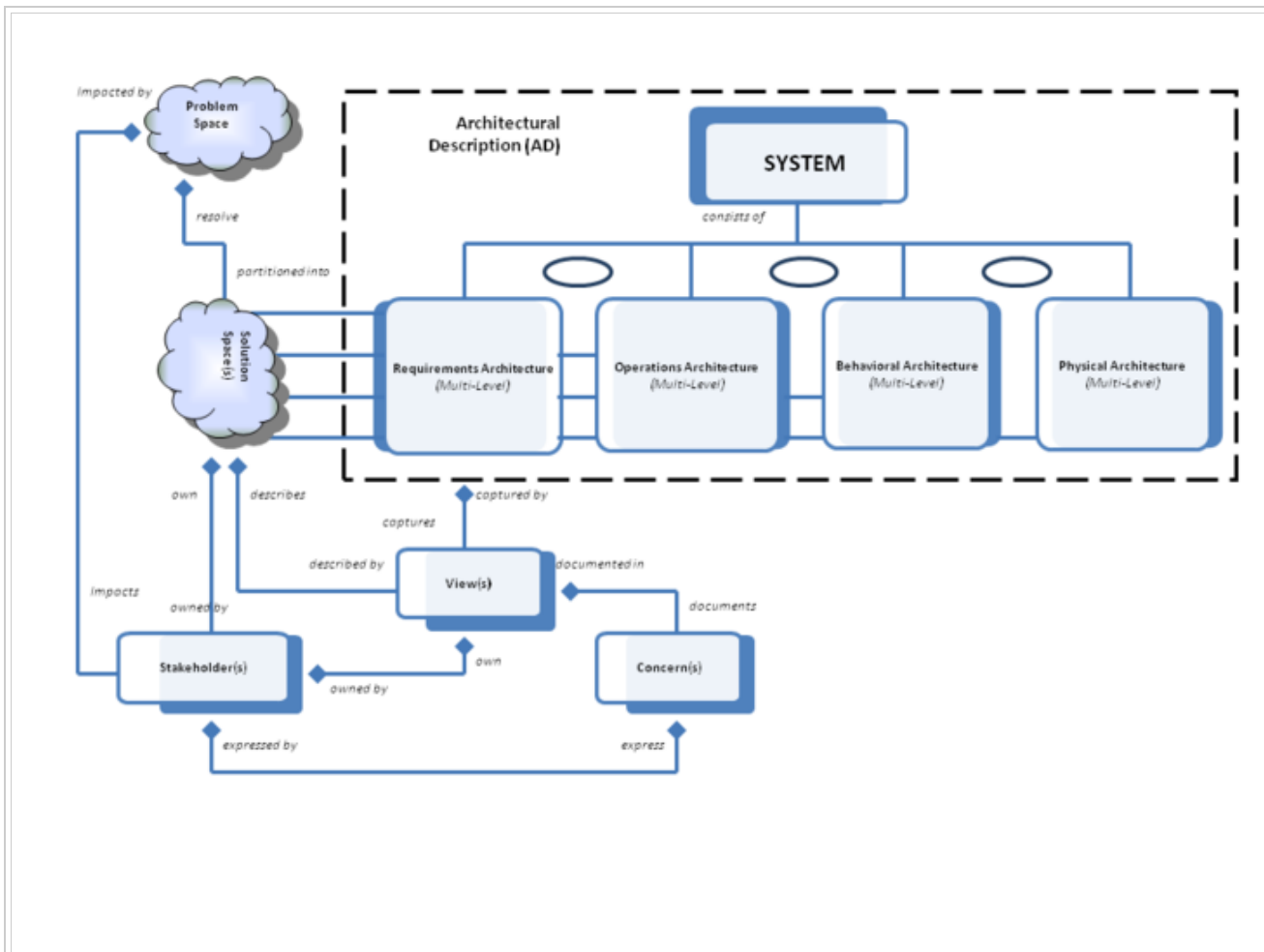


Figure 2. System Architectural Description Elements (Adapted from Wasson 2006) Reprinted with permission of John Wiley & Sons Inc.

## Specialty Engineering Integration

The INCOSE 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 & 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, usually 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 & Health
- 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 SE, 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 SE 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: 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," *The Thirty Elements of Systems Engineering*, third edition. New York, NY, USA: John Wiley & Sons.
- Grady, J. 2006. *System Requirements Analysis*. Elsevier. ISBN 978-0-12-088514-5
- Grady, J. 2006. "Chapter 3.7. System Requirements Analysis", *Specialty Engineering 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*. Auerbach Publications. ISBN 978-1-4200-7251-8
- INCOSE. 2011. *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, version 3.2.1. San Diego, CA, USA: International Council on Systems Engineering (INCOSE), INCOSE-TP-2003-002-03.2.1.
- 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 42010. 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, Kings College, UK.
- Maier, M. and Rechtin, E. 2009. *The Art of Systems Architecting*, third edition. Boca Raton, FL, USA: CRC Press.
- Wasson, C. S. 2006. *System Analysis, Design, and Development*. Hoboken, NJ, USA: John Wiley & Sons.

## Primary References

- ANSI/IEEE. 2000. *Recommended practice for architectural description for software-intensive systems*. New York, NY: 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. *System Requirements Analysis*. Elsevier. ISBN 978-0-12-088514-5
- Grady, J. 2010. *Systems Synthesis- Product and Process Design*. Boca Raton, FL, USA: CRC Press.
- INCOSE. 2011. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, version 3.2.1. San Diego, CA, USA: International Council on Systems Engineering (INCOSE), INCOSE-TP-2003-002-03.2.1.
- ISO/IEC/IEEE 42010. 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, Kings College, UK.
- Maier, M. and Rechtin, E. 2009. *The Art of Systems Architecting*, third edition. Boca Raton, FL, USA: CRC Press.
- Wasson, C. S. 2006. *System Analysis, Design, and Development*. Hoboken, NJ, USA: John Wiley & Sons.

## Additional References

- Grady, J. 2006. "Chapter 3.7. System Requirements Analysis", *Specialty Engineering Requirements Analysis*. New York, NY, USA: Academic Press.
- Eisner, H. 2008. "Chapter 7. Essentials of Project and Systems Engineering Management," *The Thirty Elements of Systems*

*Engineering*, third edition. New York, NY, USA: John Wiley & Sons.

Hybertson, D. 2009. *Model-oriented Systems Engineering Science: A Unifying Framework for Traditional and Complex Systems*. Auerbach Publications. ISBN 978-1-4200-7251-8

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.

---

[< Previous Article](#) | [Parent Article](#) | [Next Article >](#)

## Comments from 0.5 Wiki

This article is new to the SEBoK for version 0.75. As such, there are no associated 0.5 comments.

## SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

# Business Activities Related to Product Systems Engineering

This topic discusses the interfaces between product systems engineering and other 'back office' and management activities of the enterprise.

## Marketing, Product Life Cycle Management & Quality Management

PSE includes critical and robust interfaces with related business activities such as marketing, product lifecycle management (PLCM), and quality. Traditionally, PLCM has been a critical stage in the IPDP and continues to be an important tool for Lifecycle management after product deployment. PLCM provides an important component of the PSE end-to-end view with the other component being the “breadth” component which captures everything that is relevant to the system at each life cycle stage. More currently, focus has started to shift to the idea of managing not just the life of the product but to expand the view and to manage 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 zero every time.

PSE also includes interfaces with the marketing function; in particular, PSE works closely with the business and market development organizations to elicit products 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, and define products, 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 Systems Engineering and Procurement/Acquisition.

The links between PSE and Quality are just as critical. The relationships between Product SE 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 the tangible product have been extensively used historically. More recent approaches that acknowledge and match Product SE’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 cooperative 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. Product SE should play an important role in the design and implementation of any quality management system. See 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 (SEI, 2008), CMMI helps "integrate traditionally separate organizational functions, set process improvement goals and priorities, provide guidance for quality processes, and provide a point of reference for appraising current processes."

## Project Management & Business Development

The end-to-end view mandated by PSE requires strong relationships with project management and to 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) (glossary) is the key document covering the activities, milestones, organization, and resources 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. (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. Product Systems Engineering’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, Product Systems Engineering 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 [http://en.wikipedia.org/wiki/Business\\_development](http://en.wikipedia.org/wiki/Business_development).

## Supply Chain Management & Distribution Channel Management

Product SE provides the following information to the supply chain management function in the 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)
- 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.

Product SE 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
- 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 (glossary) is defined in various ways, but always 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 social – e.g. a car may improve the ability to satisfy the transport needs of a family. Users acquire products (e.g. military equipment, cars, “productised” service offerings from airlines and taxi companies, etc) to contribute to satisfying their capability needs.

Capability Management (glossary) involves identifying and quantifying capabilities (existing, new or modified) that will be needed to meet enterprise goals, and selecting a coherent set of product and services across all components of Capability (glossary) 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 best use of existing product 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, 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; then 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. Product systems engineering 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 analysed by (Elliott, B, et al ) and is best viewed as the same basic SE process conducted at a much higher tempo (Kemp & 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 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 system 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 early in the product development that the system will be testable. 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 test capabilities necessary.

## 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 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

Kossiakoff, A. and N. Sweet. 2003. "Chapter 11," *Production, Systems Engineering Principles and Practice*. John Wiley & Sons.

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.

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

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

### Primary References

No primary references have been identified for version 0.75. Please provide any recommendations on primary references in your review.

### Additional References

No additional references have been identified for version 0.75. Please provide any recommendations on additional references in your review.

---

[< Previous Article](#) | [Parent Article](#) | [Next Article >](#)

## Comments from 0.5 Wiki

This article is new to the SEBoK for version 0.75. As such, there are no associated 0.5 comments.

## SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

# Product Systems Engineering Key Aspects

## Acquired Products vs. 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 upon opportunities to develop and offer products and services to potential users of the product based on business objectives measured usually 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 building a supply chain. The complexities of dealing with supply chains must be accounted for with respect to cost, risk, schedule and thus can have an impact upon product or service maturity. (See articles under the Systems Engineering Organizational Strategy knowledge area in Part 5)

### Acquired Products

Given specific needs for a product or service typically results in some form of an agreement between the acquirer and a supplier as specified in the agreement processes of ISO/IEC 15288 (2008). The acquirer specifies the need and requirements for the properties of expected product or service and may or may not place specific requirements upon how the supplier is to organize their life cycle treatment of the product or system.

The degree of agreement formality 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 the 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 disagreement 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 the 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 & 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 WSOI (Wider System-of-Interest) 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.

In Figure 1, it is important to observe with respect to Definition that in an agreement with a supplier the outsourcing can involve

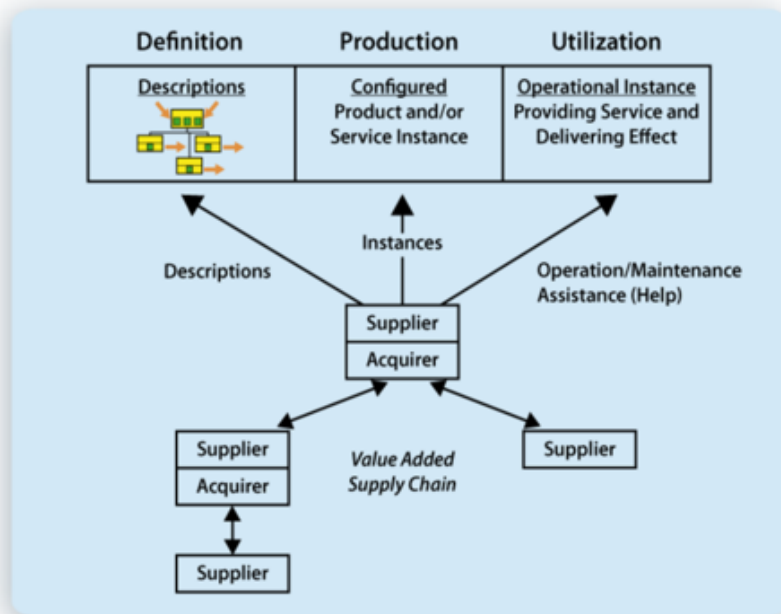


Figure 1. Supply Chain Structure (Lawson) Reprinted with permission of Harold "Bud" Lawson.

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.

With respect to Production, the outsourcing agreement with a supplier can vary from total production responsibility to the supply of 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 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 definitely is a structure tied together by agreement relationships. Further, the operation of the supply chain definitely 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.

**Distribution Channels.** 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, such as, 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.

Product SE may need to include special features in the product design to accommodate the needs of distribution channel elements, such as, 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, for interface fit checking and mass balance certification, for example.

## Product Lifecycle & Product Adoption Rates

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

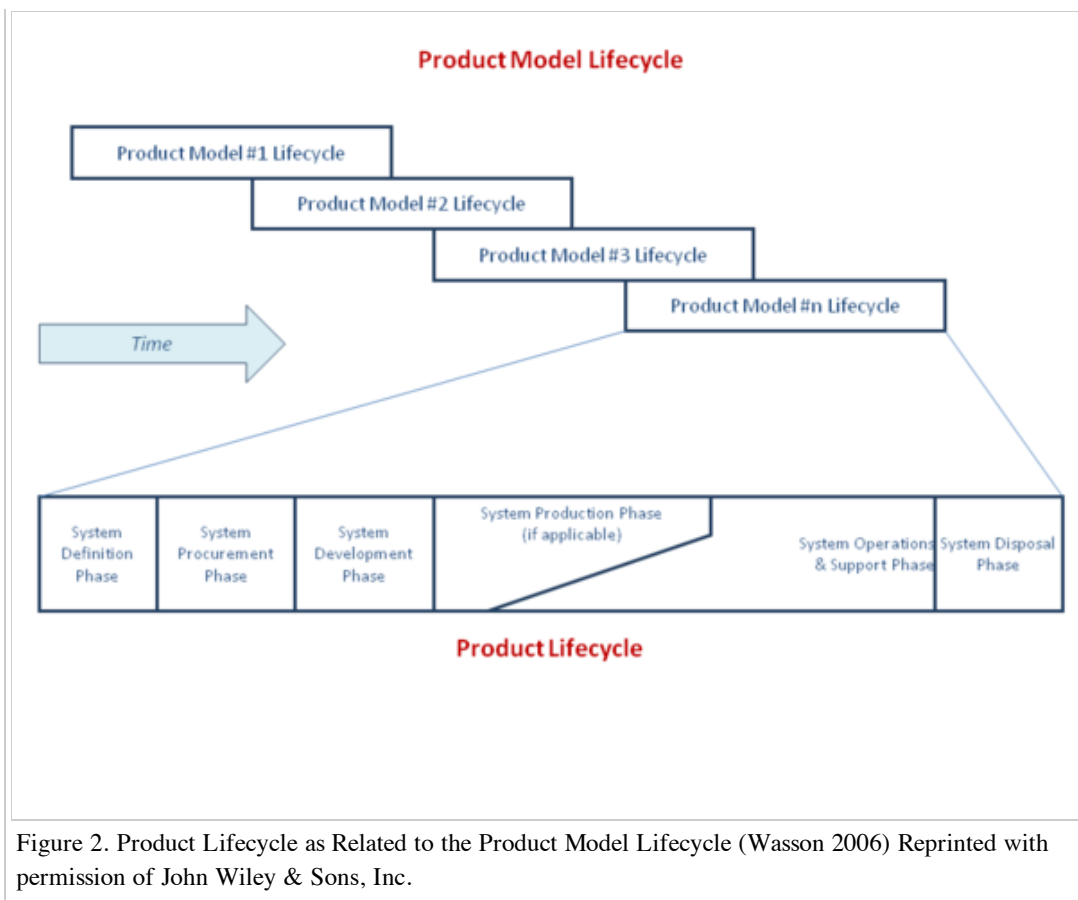


Figure 2. Product Lifecycle as Related to the Product Model Lifecycle (Wasson 2006) Reprinted with permission of John Wiley & Sons, Inc.

From an industry perspective, managing a product's lifecycle 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.  
([http://en.wikipedia.org/wiki/Product\\_lifecycle\\_management](http://en.wikipedia.org/wiki/Product_lifecycle_management))

There are many PLM tools and services available for facilitating the development and management of complicated product lifecycles 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](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 lifecycles 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. Product SE 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.

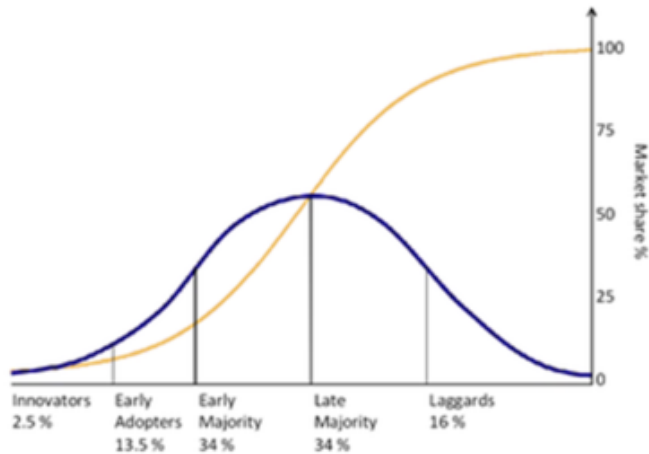


Figure 4. Rogers Innovation Adoption Curve (Source: <http://en.wikipedia.org/wiki/File:Diffusionofideas.PNG> Accessed February 6, 2012, Released by Tungsten, Public Domain)

## 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 1990's demanded development process (mostly input-output serial) to shorten their development time & 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 1990's 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 lifecycle. Later, as the importance of the process was recognized, the terminology was modified to Integrated Product and Process Development or IPPD. [INCOSE 2.a]

INCOSE SE Handbook, Version 3.1 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.

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 in addition to the disciplines required for acquired products. A good discussion can be found in (Phillips 2001), (Annachino 2003), (Morse, L., Babcock, D. 2007)

## Role of Architecture, Requirement & 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. The stakeholders have various views that they express based upon their particular viewpoints. 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 upon the life cycle thinking provided by the ISO/IEC 15288 standard; for example, 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 & Experimentation

Modelling, 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. For example:

- 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)
- Simulation is used to predict and optimise 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 and 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.
- 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, 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 – and system simulators of various levels of fidelity are used to familiarise 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 – “models” – so modelling is an enabler for simulation, prototyping and experimentation.

## Increasing Role of Software in Product Functionality

An important trend in marketed 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 have 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 the 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 chapter in SeBOK.



Recent advancements in IT and Software have assisted in their rapid utilization in Product Systems Engineering. Although, software development is 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 SW engineers to think of problem solving not only in software terms but also using the systems thinking approach. For this purpose, SW 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 & 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. that 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 into three fundamental integration components: functional organization, product and process integration.

PSE plays an important role to ensure well defined interfaces, interactions, relationships, information exchange, and processes requirements between / among 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 SE 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 (IPDT's) for detailed design and development. The IPDT's 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 product's architectural configuration and configuration tracking. As building blocks are put together interface requirements, information exchange, 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 of the Configuration Control Board (CCB).

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 any baseline changes to an interface for further recommendation to the CCB.

A configuration Item (CI) may be Hardware (HWCI), Software (CSCI), Firmware, subsystems, assemblies, non-Development items, commercial off the shelf (COTS) item, acquirer furnished equipment and/or processes. The reader is referred to (Wasson 2006), (Grady 2006) (INCISE v3.1) for detailed description of Configuration and Interface Control.

A product may experience hundreds of changes during its lifecycle 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 mechanism for bookkeeping and activity control need to be in place to identify, control, audit, account and trace interfaces, interactions, relationships among and between entities 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 (CM) & Risk Management (RM)

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 System 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 the Configuration Control Board (CCB) through Engineering Change Request (ECR) and/or Configuration Change Request (CCR). The CCB then analyzes the request to understand CI impacts and the feasibility (Time and Cost) for authorization, or rejection of change request(s). The lack of proper control and tracking of CI and product baselines may result in 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 baseline, documented, tested for backward compatibility and to ensure compliance with the integrated product functionality. Thus, successful implementation and lifecycle management of the product mandates a highly disciplined CM process that maintains proper control over the product and its components. The reader is referred

to INCOSE SE Handbook v3.1 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. ([http://en.wikipedia.org/wiki/Risk\\_management](http://en.wikipedia.org/wiki/Risk_management)). 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 to 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

- Anachino, M. 2003. *New Product Development: From Initial Idea to Product Management*. 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 edition. New York, NY, USA: John Wiley & Sons.
- Grady, J. 2006. *System Requirements Analysis*. Elsevier.
- Haimes, Y. 2008. *Risk Modeling, Assessment, and Management*, 3rd edition. New York, NY, USA: John Wiley & Sons.
- 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.
- Lawson, H. 2010. *A Journey Through the Systems Landscape*. London, UK: College Publications. Kings College, UK.
- 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.
- Wasson, C. S. 2006. *System Analysis, Design, and Development*. Hoboken, NJ, USA: John Wiley & Sons.
- Rogers, E. M. 2003. *Diffusion of innovations* (5th ed.). New York, NY: Free Press.
- Pugh, S. 1990. *Total Design: Integrated Methods for Successful Product Engineering*. Prentice Hall. ISBN-10: 0201416395, ISBN-13: 978-0201416398.
- Smith, P. and Reinertsen, D. 1997. *Developing products in half the time – new rules, new tools*. John Wiley and Sons. 2nd edition. ISBN-10: 0471292524, ISBN-13: 978-0471292524
- Reinertsen, D. 1997. *Managing the Design Factory: A Product Developers Tool Kit*. Simon & Shuster Ltd. ISBN-10: 0684839911 ISBN-13: 978-0684839912
- US Army, Chapter 2, section A.4 of Army Science and Technology Master Plan. downloaded from <http://www.fas.org/man/dod-101/army/docs/astmp/c2/P2A4.htm>, accessed on 12 Jan 2012

Kass, R. 2006. The logic of warfighting experiments. DOD CCRP – downloaded from [http://www.dodccrp.org/files/Kass\\_Logic.pdf](http://www.dodccrp.org/files/Kass_Logic.pdf), 12Jan 2012

## Primary References

No primary references have been identified for version 0.75. Please provide any recommendations on primary references in your review.

## Additional References

No additional references have been identified for version 0.75. Please provide any recommendations on additional references in your review.

[< Previous Article](#) | [Parent Article](#) | [Next Article >](#)

# Comments from 0.5 Wiki

This article is new to the SEBoK for version 0.75. As such, there are no associated 0.5 comments.

## SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

# Product Systems Engineering Special Activities

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 NASA (Mankins 1995) and later modified for use by the Department of Defense (DoD), Air Force Research Laboratory (AFRL), and 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**

<b>Technology Readiness Level</b>	<b>Description</b>
1. Basic principles observed and reported.	Lowest level of technology readiness. Examples might include paper studies of a technology's basic properties.
2. Technology concept and/or application formulated.	Invention begins. Examples are limited to analytic studies.
3. Analytical and experimental critical function and/or characteristic proof of concept.	Includes analytical and lab studies to physically validate predictions of separate elements of the technology. Examples include components not yet integrated.
4. Component validation in laboratory environment.	Basic technological components are integrated. This is relatively "low fidelity" compared to the eventual system.
5. Component validation in relevant environment.	Basic technological components are integrated with reasonably realistic supporting elements so it can be tested in a simulated environment.
6. Prototype demonstration in a relevant environment.	Representative prototype system tested in a relevant environment. Represents a major step up in a technology's demonstrated readiness. Examples include testing a prototype in a high-fidelity laboratory environment or in simulated operational environment.
7. Prototype demonstration in an operational environment.	Prototype near, or at, planned operational system. Represents a major step up from TRL 6, requiring demonstration of an actual system prototype in an operational environment.
8. System qualified through test and demonstration.	Technology proven to work in its final form and under expected conditions. Represents the end of true system development. Examples include developmental test and evaluation of the system.
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).

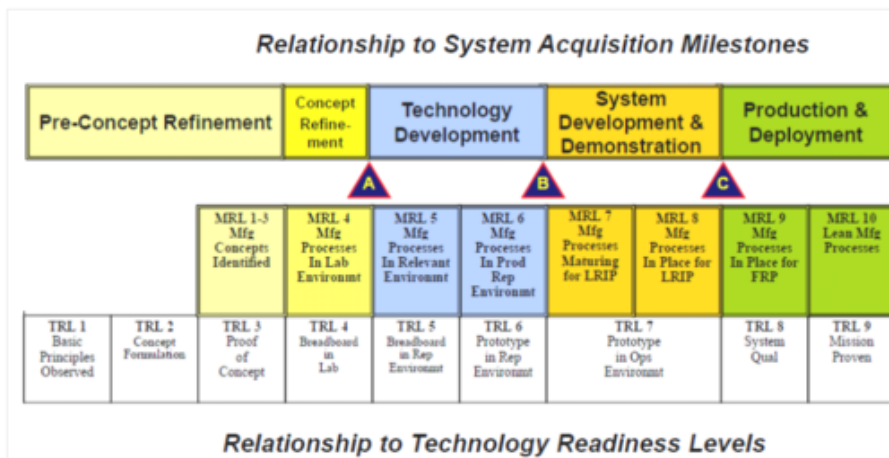


Figure 1. Technology Readiness Levels and Their Relationship to System Acquisition Milestones (Morgan 2008) Released by the Manufacturing Technology Division of the United States Air Force

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 (IRL)) 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 (Sausser 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 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 Handbook (INCOSE 2011) 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.*

The INCOSE Handbook also defines four methods for verification, inspection, analysis, demonstration and testing. 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 U.S. and Canada.

The best known certification is the airworthiness certification, which relates to the safety of flight for aircraft. In the U.S., 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)*

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, 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) 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* (Vivien 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 U.S., 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 U.S., 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 and ANSI/AIAA R-100, 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)/U.S. Department of Defense (DoD). Available at: [http://www.dau.mil/pubscats/PubsCats/13th\\_Edition\\_Glossary.pdf](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: U.S. 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" (online). Washington, DC, USA: Federal Aviation Administration (FAA). Available at: [http://www.faa.gov/aircraft/air\\_cert/airworthiness\\_certification/aw\\_overview/](http://www.faa.gov/aircraft/air_cert/airworthiness_certification/aw_overview/).
- INCOSE. 2011. *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, version 3.2.1. San Diego, CA, USA: International Council on Systems Engineering (INCOSE), INCOSE-TP-2003-002-03.2.1.
- Irish, V. 2005. *Intellectual Property Rights for Engineers*, 2nd edition. 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. "Systems Engineering Guide" (online). Available at [http://www.mitre.org/work/systems\\_engineering/guide/](http://www.mitre.org/work/systems_engineering/guide/)
- NERC. "North American Electric Reliability Corporation (NERC)" (online). Available at <http://www.nerc.com>.
- Sauser, B., Verma, D., Ramirez-Marquez, J. and Gove, R. 2006. *From TRL to SRL: The Concept of System Readiness Levels*. Proceedings of the Conference on Systems Engineering Research (CSER), Los Angeles, CA.
- 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.



## Primary References

FAA. 2011. "Airworthiness Certificates Overview" (online). Washington, DC, USA: Federal Aviation Administration (FAA). Available at: [http://www.faa.gov/aircraft/air\\_cert/airworthiness\\_certification/aw\\_overview/](http://www.faa.gov/aircraft/air_cert/airworthiness_certification/aw_overview/).

Irish, V. 2005. *Intellectual Property Rights for Engineers*, 2nd edition. 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. "Systems Engineering Guide" (online). Available at [http://www.mitre.org/work/systems\\_engineering/guide/](http://www.mitre.org/work/systems_engineering/guide/)

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

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.

## 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: U.S. 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.

---

[< Previous Article](#) | [Parent Article](#) | [Next Article >](#)

## Comments from 0.5 Wiki

This article is new to the SEBoK for version 0.75. As such, there are no associated 0.5 comments.

## SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

## Service Systems Engineering

The growth of services in the ever evolving global economy has brought much needed attention to Service Science (glossary) and Service Systems Engineering (glossary). Research focuses on developing formal methodologies to understand enterprise-end-user (Customer (glossary)) interactions from both socio-economic and technological perspectives, and to enable Value (glossary) 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

The topics included in this Knowledge Area include:

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

## Introduction

Service Systems Engineering (SSE) has usually been a proprietary Process (glossary) 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) (glossary) service providers. (AT&T SRP 2008; Eppinger 2001; Whitten and Bentley 2007; Freeman 2004) An example of a service realization process from American Telephone and Telegraph (AT&T) is shown in Figure 1 below.



Figure 1. AT&T Service Realization Process (AT&T 2008)Source available at

[http://www.corp.att.com/gov/contracts/fts\\_bridge/technical/04a\\_vol\\_I\\_](http://www.corp.att.com/gov/contracts/fts_bridge/technical/04a_vol_I_)

These early efforts were, and in some instances remain, very important for Product (glossary) 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 (glossary) (manufacturing, agriculture, etc.) and Service Business (glossary) distinctions are going away (Spohrer 2011). These services – or service innovations – must take into account social aspects, Governance (glossary) 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 (glossary) 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 entities, stakeholders, and in defining the information flows required for the governance, operations, administration, management and provisioning (OAM&P) of the service. (Maier 1998; Jamishidi 2009; Pineda 2010; Luzeaux and Ruault 2010). The unique characteristics and evolving methods of SSE are shown in Table 2 below (Tien and Berg 2003).



Permission is pending for this copyright owned image. Please see the caption for information on the original source.

Table 2. Service Systems Engineering: Unique Characteristics and Evolving Methods (Tien and Berg 2003, pg. 28, Table 11) Source available at <http://www.springerlink.com/content/6264k68125787u03/>

## Service SE Knowledge Area Topics

This Knowledge Area (KA) describes best practices in service systems engineering 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, 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 (NAE 2011).

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 (glossary) and Service Systems Engineering (SSE) (glossary).

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; particular attention is paid to the properties of service systems for the service sector such as transportation, consulting services, and healthcare.

The Scope of Service Systems Engineering and Value of Service Systems Engineering articles cover the value of SSE, defining (or using when available) service architectural frameworks, and the stages of the service development process from concept to Life Cycle Management (glossary).

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 & 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 (glossary) 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 (glossary), 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 reasonably be planned and predicted. Not so with innovation and breakthroughs. 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 SE

Mindful of the evolution taking place in the global economy and the world markets, it would be foolish 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. Available at: [http://www.corp.att.com/gov/contracts/fts\\_bridge/technical/07\\_vol\\_I\\_section\\_1.pdf](http://www.corp.att.com/gov/contracts/fts_bridge/technical/07_vol_I_section_1.pdf). accessed on June 1st 2011.
- Chang, C.M. 2010. *Service Systems Management and Engineering: Creating Strategic Differentiation and Operational Excellence*. New York, NY, USA: John Wiley & Sons, Inc. ISBN 978-0-470-42332-5.
- Eppinger, S. 2001. "Innovation at the Speed of Information." *Harvard Business Review* 79(1) (January 2001): 149-158.
- Freeman, R.L. 2004. *Telecommunication Systems Engineering*. 4th edition. New York, NY, USA: John Wiley & Sons. ISBN 0-471451-33-9; ISBN 978-0-47145-133-4.
- 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.
- Kano, N. 1996. *Guide to TQM in Service Industry*. Tokyo, Japan: Asian Productivity Organization. ISBN: 978-92833114300.
- Maier, M.W. 1998. "Architecting Principles for System of Systems." *Systems Engineering* 1(4): 267-284.

Maglio, P., C. Kieliszewski, and J. Spohrer. 2010. *Handbook of Service Science*. New York, NY, USA: Springer Science + Business Media. 1st Edition. ISBN-10: 9781441916273; ISBN-13: 978-1441916273.

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.

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

Spohrer, J.C. 2011. "Service Science: Progress & Directions." Presented at International Joint Conference on Service Science. 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: 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. ISBN 13:978-0-07-305233-5.

## 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., 2010, ISBN 978-0-470-42332-5.

Freeman, R.L. 2004. *Telecommunication Systems Engineering*. New York, NY, USA: John Wiley & Sons. 4th edition. New York, NY USA ISBN-10: 0471451339; ISBN-13: 978-0471451334.

Maglio, P., C. Kieliszewski, and J. Spohrer. 2010. *Handbook of Service Science*. New York, NY, USA: Springer Science + Business Media. 1st Edition. ISBN-10: 9781441916273; ISBN-13: 978-1441916273.

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

## Additional References

No additional references have been identified for version 0.75. Please provide any recommendations on additional references in your review.

---

[< Previous Article](#) | [Parent Article](#) | [Next Article >](#)

## Comments from SEBoK 0.5 Wiki

No comments were logged for this article in the SEBoK 0.5 wiki. Because of this, it is especially important for reviewers to provide feedback on this article. Please see the discussion prompts below.

## SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

# 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 a trend which has accelerated in the last 25-30 years and in 2006, services produced by private industry accounted for 67.8% of U.S. gross domestic product (GDP). The top sub-sectors included real estate and 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. PSS's 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 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." 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 (McAfee 2009).

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 (2009).

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 (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), Marketing Resource Management (MRM), Software Configuration Management (SCM), etc.), software as a service (SaaS), platform as a service (PaaS), cloud services, peer-to-peer 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 or (WS). Thus, 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 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, 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 eco-system.

## 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.

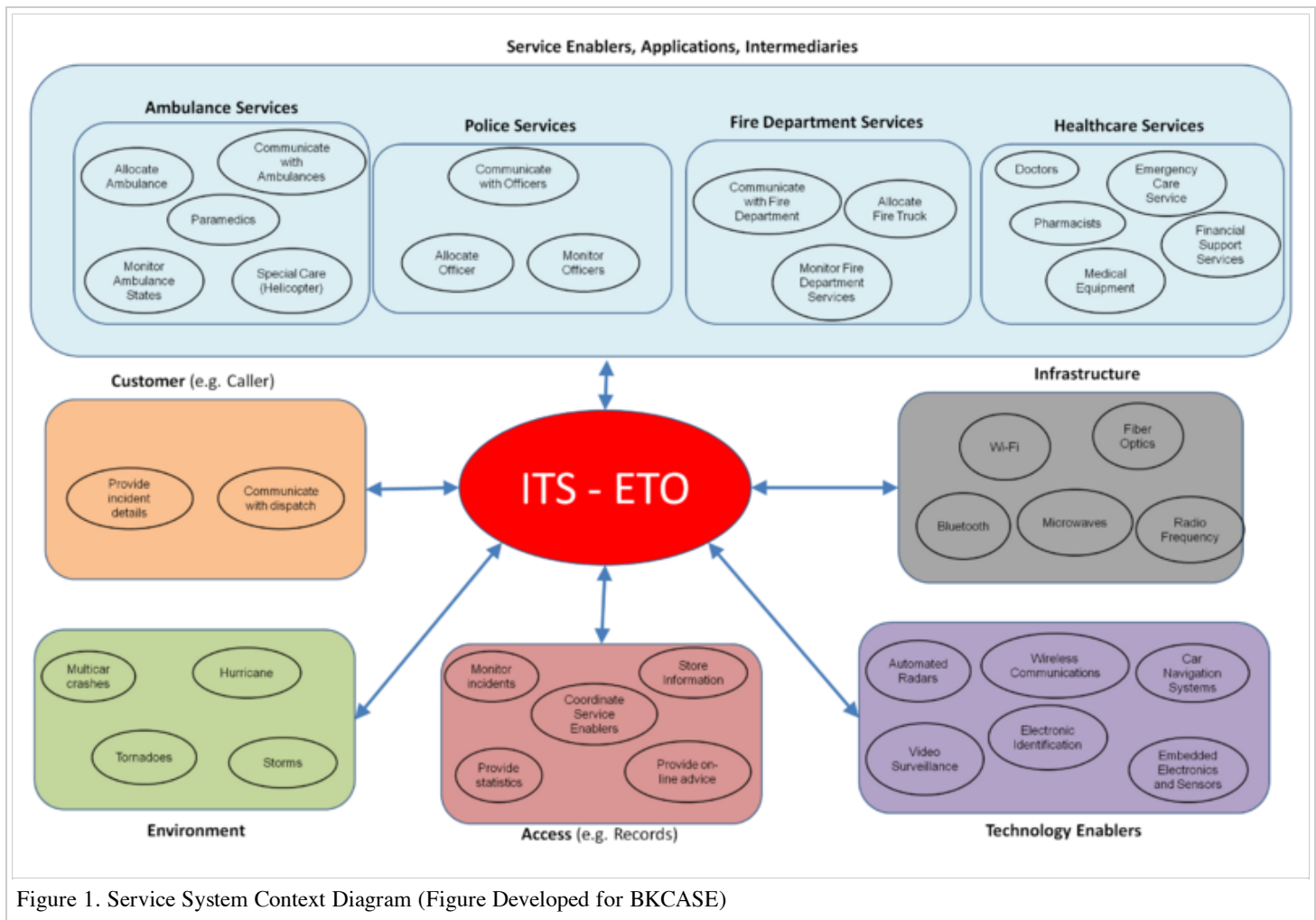


Figure 1. Service System Context Diagram (Figure Developed for BKCASE)

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 anytime, 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 lifecycles, 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. 271-337.

Cook, M. 2004. "Chapter : 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. Edmunds, Suffolk, UK: Professional Engineering Publishing Limited. 123-134.

Domingue, J., D. Fensel, J. Davies, R. González-Cabero, and C. Pedrinaci. 2009. "Chapter : 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 2007.

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.

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

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

Organization for Economic Co-operation and Development (OECD). 2000. *The Service Economy*, Science Technology Industry, Business and Industry Policy Forum Series, Paris, France: OECD. <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.C. 2011. "Service Science: Progress & Directions." Presented at the International Joint Conference on Service Science, May 25-27, Taipei, Taiwan.

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).

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

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, DOI:10.3233.

US DOT. 2011. "Emergency Transportation Operations." Research and Innovative Technology Administration, last modified June 16, accessed June 23, 2011. <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: 324-335.

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, Leicester, U.K.

## 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 2007.

Organization for Economic Co-operation and Development (OECD). 2000. *The Service Economy*, Science Technology Industry,

Business and Industry Policy Forum Series, Paris, France: OECD. <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: 324-335.

## Additional References

No additional references have been identified for version 0.75. Please provide any recommendations on additional references in your review.

---

[< Previous Article](#) | [Parent Article](#) | [Next Article >](#)

## Comments from SEBoK 0.5 Wiki

No comments were logged for this article in the SEBoK 0.5 wiki. Because of this, it is especially important for reviewers to provide feedback on this article. Please see the discussion prompts below.

## SEBoK 0.75 Reviews

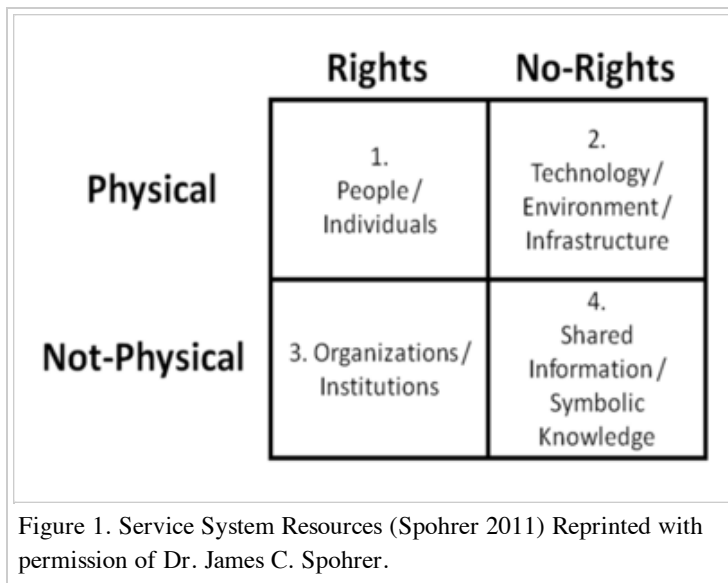
Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

# Fundamentals of Services

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).

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:



## 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; value could then be created and delivered in terms of E-services: 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 bricks-and-mortar service enterprises interact with their customers.

As discussed by Chesbrough (2011), Tien and Berg (2003), and Hipel, et al. (2007), 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.” Chang (2010) defines these features as:

- *Information Driven* - The creation, management, and sharing of information is crucial to the design, production and delivery of services.
- *Customer Centric* - Customers are generally 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 more 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 add value to the target clients. For profit-seeking service companies, the value so produced assures their profitability. For non-profit service entities, it reinforces the goodness of its 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 the 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 are the customer, provider, authority, and the competitors; Figure 2 below depicts value calculations from multiple stakeholder perspectives:

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 profitably to customers? Can we continue to improve?
3.Authority	Compliance (Taxes and Fines, Quality of Life)	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?

Figure 2. Value Calculation From Different Stakeholders' Perspectives (Spohrer 2011) Reprinted with permission of Dr. James C. Spohrer.

From an engineering design point of view, the service and business goals are an entry point 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, we then need to identify service system entities that could participate in the service delivery (people, organizations, technologies, processes).

## Service System Entities

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

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

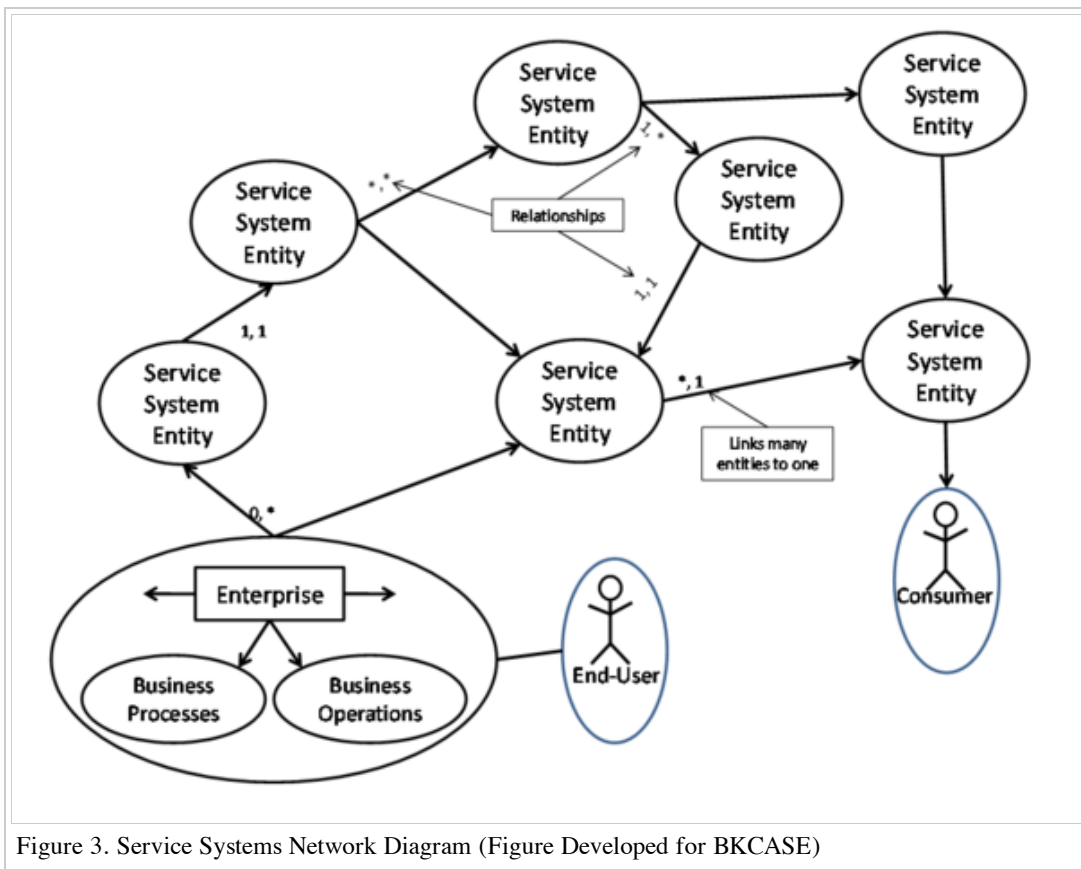


Figure 3. Service Systems Network Diagram (Figure Developed for BKCASE)

## Service System Hierarchy

But systems are part of other systems that 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 a business enterprise, a nation, 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. We extend this concept to create the service system hierarchy as described in Figure 4 below (Spohrer 2011; Maglio and Spohrer 2008; Maglio et al. 2010):

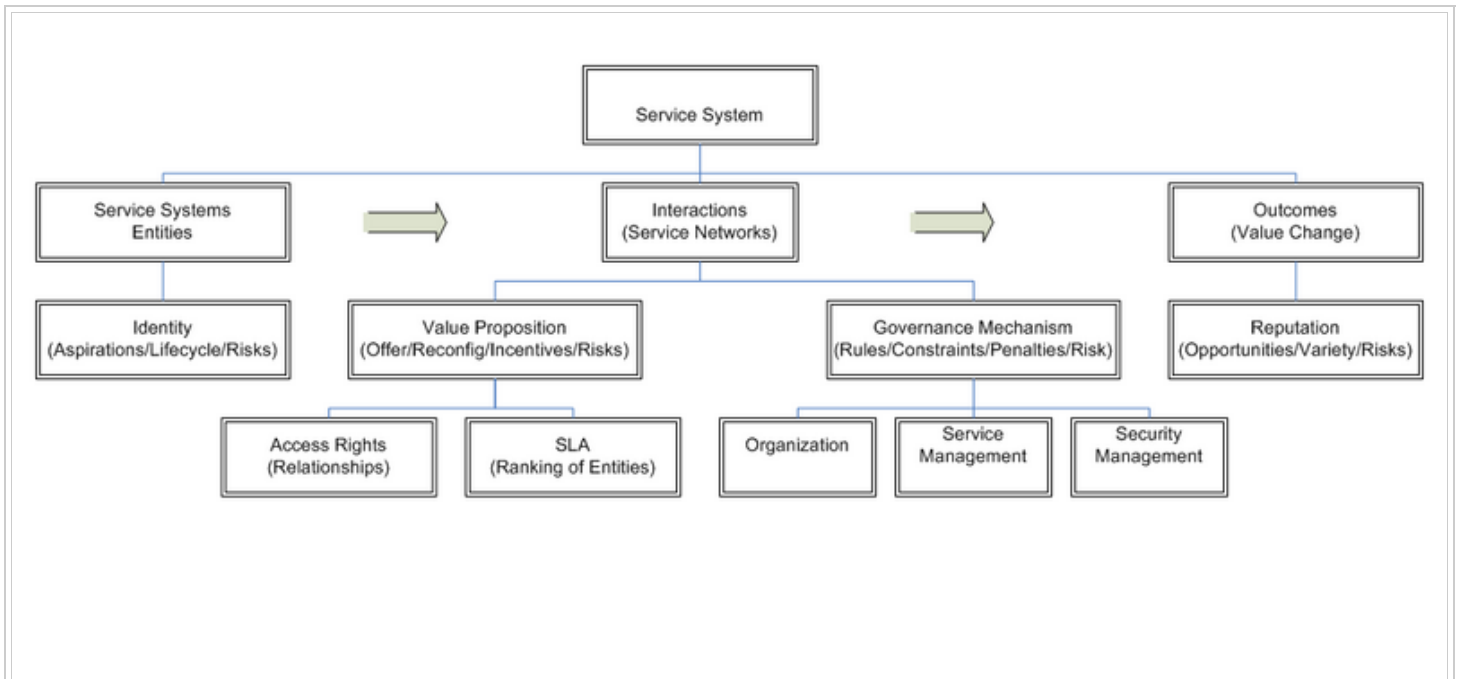


Figure 4. Service System Conceptual Framework (Sphorer 2011) Reprinted with permission of Dr. James C. Spohrer.

## 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) and 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 to ensure that the customized service behaves as expected. This adaptive behavior of service system implies that its 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*. 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.
- 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 at: <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. DOI: 10.1007.
- 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. Fährnich (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, 25-27 May 2011, Taipei, Taiwan.
- 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).
- 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, DOI: 10.1002. Hoboken, NJ: John Wiley & Sons.
- Theilmann, W. and L. Baresi. 2009. "Chapter: 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

- 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.
- 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 13-15, 2007, Cambridge, UK.
- Spath, D. and K.P. Fährnich (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, 25-27 May 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.

## Additional References

No additional references have been identified for version 0.75. Please provide any recommendations on additional references in your review.

---

[< Previous Article](#) | [Parent Article](#) | [Next Article >](#)

## Comments from SEBok 0.5 Wiki

No comments were logged for this article in the SEBok 0.5 wiki. Because of this, it is especially important for reviewers to provide

feedback on this article. Please see the discussion prompts below.

## SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

## Properties of Services

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 a single service system entity. 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 intangible quality of service (e.g., ambulance service availability and response time to an emergency request). The demand for service may have loads dependent on time of day, day of week, season, or unexpected needs (e.g., natural disasters, product promotion campaigns, etc.). For instance, travel services have peak demands during Christmas and Thanksgiving weekend; 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; while 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 (SLA)

A service level agreement (SLA) is a set of technical (functional) and non-technical (non-functional) parameters agreed between 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.

A 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 cost 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 things like availability, resilience, security, reliability, scalability, agreement duration, response times, repair times, usability, etc. A variety of service evaluation measures are shown in Table 1 below.





Permission is pending for this copyright owned image. Please see the caption for information on the original source.

Table 1. Service Systems Engineering: Service Evaluation Measures (Adapted from Tien and Berg 2003, pg. 27, Table 10) Source available at <http://www.springerlink.com/content/6264k68125787u03/>

## Service Key Performance Indicators (KPI)

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 the business processes and their components so as to ensure compliance with SLAs. The allocated measures generate "derived requirements" (Service Level Requirements (SLR)) for the system entities and their relationships, as well as 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 as 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; and 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+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), International Organization for Standardization (ISO) Standards 9001, Telecom Quality Management System Standards (TL) 9000, Information Technology Infrastructure Library (ITIL) V3 (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 (Bell System 1984). Telecommunication Systems Engineering (TCSE) played an important role in finding methodologies for correlation between perception and objective measures for the services of the 20th 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 21st 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 an expected service. This perception must be captured via surveys and interviews. The perceived service is then compared with the expected service by the customers, which leads to the perceived service quality as a result. 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 other 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 21st century will almost certainly see similar and probably accelerated technology development as seen in the prior three decades. We will also see mass collaboration become an established mode of operation. We are already seeing the beginnings of mass collaboration in developments like value co-creation where loosely entangled actors or entities come together to create value in unprecedented ways but which 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; that is, 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
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 chain, 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, banking and finance / business consulting, healthcare and family life, education and work life / jobs and entrepreneurship
3. **Systems that focus on Governing:** city, state, nation

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 to 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, Seoul, South Korea.

Bell System. 1984. *Engineering and Operations in Bell System*. Murray Hill, NJ, USA: AT&T Bell Labs.

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.

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.

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 Baresi, L. 2009. "Chapter: 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

Bell System. 1984. *Engineering and Operations in Bell System*. Murray Hill, NJ, USA: AT&T Bell Labs.

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

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

Theilmann, W. and Baresi, L. 2009. "Chapter : 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.

### Additional References

No additional references have been identified for version 0.75. Please provide any recommendations on additional references in your review.

[< Previous Article](#) | [Parent Article](#) | [Next Article >](#)

## Comments from SEBoK 0.5 Wiki

No comments were logged for this article in the SEBoK 0.5 wiki. Because of this, it is especially important for reviewers to provide feedback on this article. Please see the discussion prompts below.

## SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

# Scope of Service Systems Engineering

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 service systems engineering (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 different domains that range from 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.

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.

SSE	COLLABORATION DOMAINS
SSE Management	<ul style="list-style-type: none"> <li>• Management Science</li> <li>• Business Process Management</li> <li>• Cognitive Science</li> <li>• Decision Science</li> </ul>
Service Realization Process (SRP)	<ul style="list-style-type: none"> <li>• All engineering fields</li> <li>• Business Operations</li> <li>• Infrastructure Operations</li> <li>• Social Science</li> <li>• Computer Science</li> <li>• Management Science</li> <li>• Behavioral Science</li> <li>• Network Science</li> <li>• Computational Science</li> <li>• Systems Science</li> <li>• Decision Science</li> </ul>
Methodologies, Processes and Tools (MPT)	<ul style="list-style-type: none"> <li>• Natural Science</li> <li>• Business Science (BPMN)</li> <li>• Mathematical</li> <li>• All engineering fields</li> </ul>

Figure 1. Service Systems Engineering Domain Collaboration (Hipel et al. 2003, pg. 727) Reprinted with permission of © Copyright IEEE – All rights reserved.)

## 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).

For the sake of completeness of the description, 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 may be that the new request is 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, including considerations such as 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, Lifecycle 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 Figure 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.

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

Figure 2. Service Realization Process: Life Cycle Stages (Figure Developed for BKCASE)

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

## Service Design Management

Another important role of service systems engineering 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 both for 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 stakeholders 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:

- SSE Planning
- SSE Assessment and Control
- SSE Decision Management
- SSE Risk Management
- SSE Configuration Management
- SSE Information Management
- SSE Engineering Measurement

SSE plays a critical role in describing the needs of the intended service in terms of the service 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. "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" (Hsu 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, 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. *SeCSE Methodology*. SeCSE Service Centric Systems Engineering. SeCSE511680. Available at: [http://www.secse-project.eu/wp-content/uploads/2007/08/a5\\_d4-secse-methodology-v1\\_3.pdf](http://www.secse-project.eu/wp-content/uploads/2007/08/a5_d4-secse-methodology-v1_3.pdf).
- 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, Metz, France.
- Spark, D. 2009. "Real-Time Search and Discovery of the Social Web." Spark Media Solutions Report. Accessed September 2, 2011. Available at: <http://www.sparkminute.com/2009/12/07/free-report-real-time-search-and-discovery-of-the-social-web/>.

### Primary References

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

ITIL V3. 2007. *ITIL Lifecycle Publication Suite Books*. London, UK: The Stationery Office.

Lefever, B. 2005. *SeCSE Methodology*. SeCSE Service Centric Systems Engineering. SeCSE511680. Available at: [http://www.secse-project.eu/wp-content/uploads/2007/08/a5\\_d4-secse-methodology-v1\\_3.pdf](http://www.secse-project.eu/wp-content/uploads/2007/08/a5_d4-secse-methodology-v1_3.pdf).

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.

### Additional References

No additional references have been identified for version 0.75. Please provide any recommendations on additional references in your review.

[< Previous Article](#) | [Parent Article](#) | [Next Article >](#)

## Comments from SEBok 0.5 Wiki

No comments were logged for this article in the SEBoK 0.5 wiki. Because of this, it is especially important for reviewers to provide feedback on this article. Please see the discussion prompts below.

## SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by [DISQUS](#)

# Value of Service Systems Engineering

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

Service systems engineering uses disciplined approaches to minimize Risk (glossary) by coordinating/orchestrating social aspects, governance (glossary) (including security (glossary)), environmental, human behavior, business (glossary), 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 (glossary), 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 (glossary) and validation (glossary) in particular as they relate to the operations, administration, maintenance (glossary) 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 (glossary) management, service innovation, and social networks for the modeling of service system value chain. (Carroll, et al. 2010) The Systems Engineering (glossary) 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 (glossary) 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 (glossary) 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 Enabling Individuals to Perform Systems Engineering), the Service System Management and Engineering (SSME)-12 skills, as summarized by (Chang 2010) and listed below verbatim:

1. Management of Service Systems. These skills include scheduling, budgeting and management of information



systems/technologies, leadership

2. Operations of Service Systems. Engineers should be proficient in process evaluation and improvement, Quality (glossary) improvement, customer relationships, Uncertainty (glossary) management
3. Service Processes. These skills include performance measurements, flow charting, work task breakdown
4. Business Management. Business skills include project costing, business planning, 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 (glossary) 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)
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 (glossary) (IT), business reengineering, web services, etc.) through the creation of process-driven architectural frameworks.

Architectural 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 (glossary) systems of systems (see also Complexity). An architecture (glossary) is a Model (glossary) of the the system to describe the entities, the interactions and interoperability among entities, as well as the expected behavior (glossary), 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, 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 architectural 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 architectural 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 architectural 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 (glossary), 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) V3) 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:

Standards:

- Zachmann Framework (Zachman 2003)

- Business Process Modeling (BPM) (Hantry et al. 2010)
- The Open Group Architectural Framework (TOGAF)(TOGAF 2009)
- IT Service Management Information Technology Infrastructure Library (ITIL)(OGC 2009)
- Enhanced-Telecomm Operations Map (e-TOM) 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 Architectural Framework (DoDAF)
- Others.

#### Proprietary Enterprise Architectural 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
- Others.

As an example of architectural frameworks applications for service systems we present the “High Level Reference Model for the Smart Grid” developed by NIST in 2010 under the “Energy Independence and Security Act of 2007” ((EISA)):

*EISA designated the development of a Smart Grid as a national policy goal, specifying that an Interoperability (glossary) 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, information exchanges required among system entities) and technical needs of the smart grid (data structures, entities specifications, interoperability requirements, etc.).

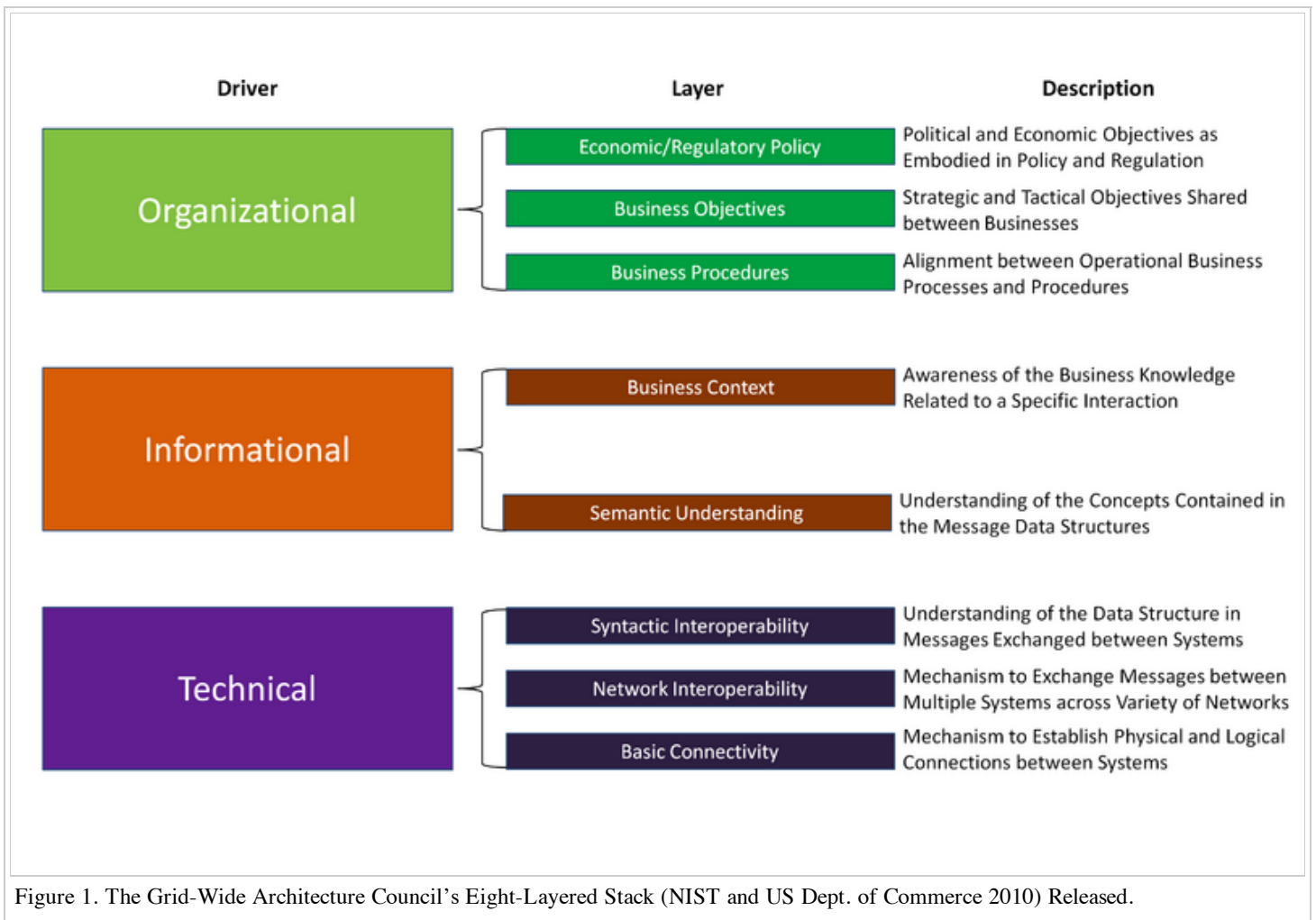
The NIST Reference Model uses this architectural framework to identify existing standards, identify new standards required for interoperability among interconnected networks and to enable innovations where smart grid components (energy sources, bulk generation, storage, distribution, transmission, metering, 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 architectural framework is being used today by different standards organizations, the Smart Grid Interoperability Panel (SGIP), and several smart grid working groups. For details on priorities, working programs and working group charters the user is referred to (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 (glossary) 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). Examples are:

- Web Services Development Life Cycle (SLDC)
- Rational Unified Process (RUP) for SOA
- Service Oriented Modeling and Architecture (SOMA)
- Service Oriented Analysis and Design/Decision Modeling (SOAD)



Further Research is required to understand the architectural implications of dynamic service configuration including the human behavior, social aspects, governance processes, business processes and implications on dynamic Service Level Agreement (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. "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. ISBN 978-0-470-42332-5.
- 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. ISBN 978-0-13234-482-1.
- e-TOM. 2009. "Business Process Framework." Morristown, NJ: TeleManagement Forum. Accessed on May 30, 2011. Available at: [www.tnforum.org/BusinessProcessFramework/1647/home.html](http://www.tnforum.org/BusinessProcessFramework/1647/home.html).
- Grasso, D. and D. Martinelli. 2007. "Holistic Engineering." *The Chronicle Review, The Chronicles of Higher Education*. "Section B."

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. "Chapter: Business Process Management." in Papazoglou, M.; Pohl, K.; Parkin, M.; Metzger, A. (Eds.) *Service Research Challenges and Solutions for the Future Internet S-Cube – Towards Engineering, Managing and Adapting Service-Based Systems* Berlin 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 on September 2, 2011. Available at: [ftp://ftp.hp.com/pub/services/itsm/info/itsm\\_rmwp.pdf](ftp://ftp.hp.com/pub/services/itsm/info/itsm_rmwp.pdf).

OGC (Office of Government Commerce). 2009. *ITIL Lifecycle Publication Suite Books*. London, UK: The Stationery Office. ISBN 978-0-11331-050-0.

Luzeaux, D. and J.R. Ruault (eds.). 2010. *Systems of Systems*. New York, NY, USA: John Wiley & Sons. ISBN 978-1-84821-164-3.

Maglio, P. and J. Spohrer. 2008. "Fundamentals of Service Science." *Journal of the Academy of Marketing Science* 36(1): 18-20. DOI: 10.1007 / ISBN 117-4-70070-058-9.

NIST. 2010. National Institute of Standard and Technology. "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 at: <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 on September 2, 2011. Available at: <http://docs.oasis-open.org/webcgm/v2.0/OS/webcgm-v2.0.pdf>.

Papazoglou, M., K. Pohl, M. Parkin, and A. Metzger. 1998. "Chapter: 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 Heidelberg, Germany: Springer-Verlag

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

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

## 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. ISBN 978-0-470-42332-5.

Erl, T. 2008. "SOA Principles of Service Design." Boston, MA, USA: Prentice Hall, Pearson Education. ISBN 978-0-13234-482-1.

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. "Chapter: Business Process Management." in Papazoglou, M.; Pohl, K.; Parkin, M.; Metzger, A. (Eds.) *Service Research Challenges and Solutions for the Future Internet S-Cube – Towards Engineering, Managing and Adapting Service-Based Systems* Berlin Heidelberg, Germany: Springer-Verlag: 27-54.

OGC (Office of Government Commerce). 2009. *ITIL Lifecycle Publication Suite Books*. London, UK: The Stationery Office. ISBN 978-0-11331-050-0.

NIST. 2010. National Institute of Standard and Technology. "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 at: <http://www.nist.gov/smartgrid/upload/FinalSGDoc2010019-corr010411-2.pdf>.

## Additional References

No additional references have been identified for version 0.75. Please provide any recommendations on additional references in your review.

< Previous Article | Parent Article | Next Article >

## Comments from SEBoK 0.5 Wiki

No comments were logged for this article in the SEBoK 0.5 wiki. Because of this, it is especially important for reviewers to provide feedback on this article. Please see the discussion prompts below.

## SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

# Service Systems Engineering Stages

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 (glossary) (TSE) process we lumped together the Concept and Feasibility Phase into a single Service Strategy/Concept, as discussed in the SEBoK Systems Engineering and Management Part. All of the stages of the SSDP take a similar iterative approach to fully understand the enterprise (glossary) capabilities, enterprise process impact, Information Technology (glossary) (IT) and technology impacts and customer (glossary) expectations. We have purposely added the Information Technology Infrastructure Library (ITIL) stage names 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.

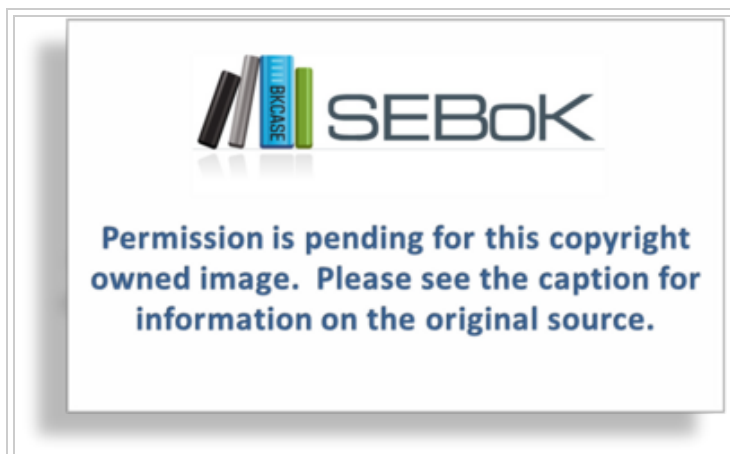


Figure 1. Converting business requirements into New Services  
(Adams et al., pg. 44) Source available at  
<http://www.tsoshop.co.uk/bookstore.asp?FO=1159966&Action=Book&ProductID=9780113311972&From=Sea>

## 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 (glossary) organization, new web service designers, new technology developments and/or IT, 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 needs/impacts on enterprise process capabilities, Operational (glossary) 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 (glossary), social, cultural, human and behaviors. The feasibility assessment also gives a +/- 30% estimate on time to develop and Cost (glossary) of development which are entry points into the Business Case to evaluate whether the service is viable with the constraints and estimates to develop and to market. At this time a decision (Decision Gate (glossary)) 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, customer within the markets to be targeted, and customer experience expected from the service (i.e., defining the Non-Functional Requirements (glossary) of the service such as Quality (glossary) Of Service (QoS), Availability (glossary), Reliability (glossary), Security (glossary) 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, 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 describing 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 (SLA) 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 (KPI)s, Technical Performance Measures (TPM)s, and Service Performance Measures (SPM)s according to intended SLA.

The SSE requirements analysis addresses the support systems for the governance (glossary), 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 (SOP) and the Operations Technical Plans (OTP).

## Systems Design/Development

The SRD, SOP and OTP have enough details about the service functions, operations, interfaces and information flows required among the different service system entities to analyze, identify and recommend end-to-end applicable architectural frameworks, to carry out trade-off analyses for the alternatives among service system entities, to describe and allocate relationships (interactions) among entities, and 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 V3 (OGC 2009) recommends inclusion of the following Service Design Processes:

- Service Catalogue Management
- Service Level Management
- Capacity Management
- Availability Management
- Service Continuity Management
- Security Management
- Supplier/Provider Management

## Service Integration, Verification & Validation

SSE defines Integration (glossary) 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 (IV&V) (glossary) (IV&V) plans need to include end-to-end Verification (glossary) and Validation (glossary) 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 the IV&V 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)
- 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 IV&V test plans and regression testing to ensure new developments work flawlessly with existing services.

ITIL V3 (OGC 2009) 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
- 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 V3 are:

- Event Management
- Incident Management
- Problem Management
- Request Fulfillment
- 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 modeling, definition, and Design (glossary) of the organization, processes and data structures of the service system (See also Representing Systems with Models). These tools and technologies include: Modeling, Simulation (glossary), development, test bed, and social environmental aspects of the intended or to be designed service. The tools fall into three main domains:

- **Business Process Management (BPM)**
- **Service Design Process**
- **Service Design Management**

**Business Process Management (BPM):** 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 in the alignment of 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 (glossary) 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 the reader is referred to (Hantry et al. 2010), (Carroll et al. 2010), (Andrikoupoulos et al. 2010), (Lin and Hsieh 2011), and (Ward-Dutton 2010).

**Service Design Process:** Architecture Frameworks (AF) and Enterprise Architectures (EA) are standards that help split Complex (glossary) systems (see also Complexity) into an interrelated, structured form. They describe the different characteristics of the Product (glossary) and services. System engineering modeling tools such as the Unified Modeling Language (UML) and System Modeling Language (SysML) help develop the AF and EA and greatly impact the continued evolution and successful implementation of complex projects. Service Oriented Architecture (SOA) and software-intensive systems architecture (IEEE P1471) are standards that apply architecture principles for specialised applications. Successful implementation of the architectural tools help identify critical interfaces and understand the allocations between components and functions.

Model-Based Systems Engineering (MBSE) (glossary) (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 (glossary). 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 the reader is referred to (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, 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 (glossary) improvements across the service. Lean Manufacturing, Six Sigma (glossary), swim lanes, balanced scoreboard, benchmarking, 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) 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) and product quality (ISO/IEC 9126), as well as information security management (ISO 27001) and evaluation (ISO 15408). The ITIL V3 describes best practices for IT Service Management which can be extended to include service systems.



## References

### Works Cited

- Adams, S., A. Carlidge, A. Hanna, S. Rance, J. Sowerby, J. Windebank. 2009. *ITIL V3 Foundation Handbook*. London, England: The Stationary Office: 44. ISBN 9780113311989
- Andrikopoulos, V., A. Bucchiarone, E. Di Nitto, R. Kazhamiakin, S. Lane, V. Mazza, and I. Richardson. 2010. "Chapter 8: Service Engineering." Papazoglou, M.; Pohl, K.; Parkin, M.; Metzger, A. (Eds.) . "Service Research Challenges and Solutions for the Future Internet S-Cube – Towards Engineering, Managing and Adapting Service-Based Systems" Berlin Heidelberg, Germany: Springer-Verlag: 271-337.
- Carroll, N., E. Whelan, and I. Richardson. "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. ISBN 978-0-470-52588-3.
- 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. Available at: [http://www.incose.org/ProductsPubs/pdf/techdata/MTTC/MBSE\\_Methodology\\_Survey\\_2008-0610\\_RevB-JAE2.pdf](http://www.incose.org/ProductsPubs/pdf/techdata/MTTC/MBSE_Methodology_Survey_2008-0610_RevB-JAE2.pdf)
- Friedenthal, S. 1998. "Object Oriented System Engineering: Process Integration for 2000 and beyond." Presented at System Engineering & Software Symposium. New Orleans, LA: Lockheed Martin Corporation.
- 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. "Chapter: Business Process Management." in Papazoglou, M.P., K. Pohl, M. Parkin, A. Metzger (eds.) . *Service Research Challenges and Solutions for the Future Internet S-Cube – Towards Engineering, Managing and Adapting Service-Based Systems*. Berlin Heidelberg, Germany: Springer-Verlag: 27-54.
- Hybertson, D.W. 2009. *Model-Oriented Systems Engineering Science: A Unifying Framework for Traditional and Complex Systems*. Boston, MA, USA: Auerbach Publications. ISBN # 978-1-4200-7251-8.
- OGC (Office of Government Commerce). 2009. "ITIL Lifecycle Publication Suite Books." London, UK: The Stationery Office. ISBN 978-0-11331-050-0.
- Lefever, B. 2005. SeSCE Methodology. SeCSE Service Centric Systems Engineering. SeCSE511680. Available at: [http://www.secse-project.eu/wp-content/uploads/2007/08/a5\\_d4-secse-methodology-v1\\_3.pdf](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.
- Pezuela, C., 2005. "Collection of Existing Service Centric Prototypes." Report A5.D1 Brussels, Belgium: European Union, Information Society Technology. <http://www.secse-project.eu/wp-content/uploads/2007/08/a5d1-collection-of-existing-service-centric-prototypes.pdf> Accessed on September 5, 2011.
- Ward-Dutton, N. 2010. "BPM Technology: Vendor Capability Comparison." MWD Premium Advisory Report. Horsham, West Sussex, UK: Macehiter Ward-Dutton (MWD) Limited. MWD Advisors. <http://www.mwdadvisors.com/library/detail.php?id=380> Accessed September 5, 2011.

### Primary References

- AT&T SRP. 2008. Technical Approach to Service Delivery. General Services Administration, AT&T Bridge Contract No. GS00Q09NSD0003, [http://www.corp.att.com/gov/contracts/fts\\_bridge/technical/07\\_vol\\_I\\_section\\_1.pdf](http://www.corp.att.com/gov/contracts/fts_bridge/technical/07_vol_I_section_1.pdf). accessed on June 1st 2011.
- Daskin, M.S. 2010. *Service Science*. New York, NY, USA: John Wiley & Sons. ISBN 978-0-470-52588-3.
- Erl, T. 2008. "SOA Principles of Service Design." Boston, MA, USA: Prentice Hall Pearson Education. ISBN 978-0-13234-482-1.

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. Available at:

[http://www.incose.org/ProductsPubs/pdf/techdata/MTTC/MBSE\\_Methodology\\_Survey\\_2008-0610\\_RevB-JAE2.pdf](http://www.incose.org/ProductsPubs/pdf/techdata/MTTC/MBSE_Methodology_Survey_2008-0610_RevB-JAE2.pdf)

Hybertson, D.W. 2009. *Model-Oriented Systems Engineering Science: A Unifying Framework for Traditional and Complex Systems*. Boston, MA, USA: Auerbach Publications. ISBN # 978-1-4200-7251-8.

OGC (Office of Government Commerce). 2009. *"ITIL Lifecycle Publication Suite Books."* London, UK: The Stationery Office. ISBN 978-0-11331-050-0.

Lefever, B. 2005. SeSCE Methodology. SeCSE Service Centric Systems Engineering. SeCSE511680. Available at: [http://www.secse-project.eu/wp-content/uploads/2007/08/a5\\_d4-secse-methodology-v1\\_3.pdf](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. ISBN 978-1-84821-164-3.

## Additional References

No additional references have been identified for version 0.75. Please provide any recommendations on additional references in your review.

---

[< Previous Article](#) | [Parent Article](#) | [Next Article >](#)

## Comments from SEBok 0.5 Wiki

No comments were logged for this article in the SEBoK 0.5 wiki. Because of this, it is especially important for reviewers to provide feedback on this article. Please see the discussion prompts below.

## SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

# Enterprise Systems Engineering

This knowledge area focuses on the application of systems engineering principles to enterprise systems (glossary). Enterprise systems engineering (glossary) is the application of Systems Engineering (glossary) principles, concepts, and methods to the planning, design, improvement, and operation of an enterprise.

## Topics

This series of articles will first provide (1) some background on the scope of ESE, imperatives for enterprise transformation, and potential SE enablers for the enterprise. It will then discuss (2) how ESE relates to system of systems (SoS) and a federation of systems (FoS). Next it will describe (3) related business activities and necessary extensions of TSE that enable (4) ESE activities. Each of the ESE process activities is discussed (5) in the overall context of the unique circumstances in the operation of a large and complex enterprise. Finally, it will show (6) how ESE can be used to establish and maintain enterprise operational capabilities. These six Topics are listed below.

The topics contained within this knowledge area include:

- 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

## Purpose

This article, and its sub-articles (listed as "Topics" below), 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).

This Knowledge Area includes 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

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. (ESE) is an emerging discipline that focuses on frameworks, tools, and problem-solving approaches for dealing with the inherent complexities (see also Complexity) 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 Rebovich and White (2011).

*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, [http://en.wikipedia.org/wiki/Dutch\\_East\\_India\\_Company](http://en.wikipedia.org/wiki/Dutch_East_India_Company))*

*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.*

## 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:

1. each other to coordinate functions, share information, allocate funding, create workflows, and make decisions, etc.; and
2. 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).

### 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 includes 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 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 throughout 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](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).

## 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 your 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 SE

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).*

## Services vs Products vs Enterprises

A Service System (glossary) 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 (glossary) 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 4), facilities, data, materials, techniques, and even services. Each of these product system elements can be "engineered."

*Note 4. 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.*

So, 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.

## Creating Value

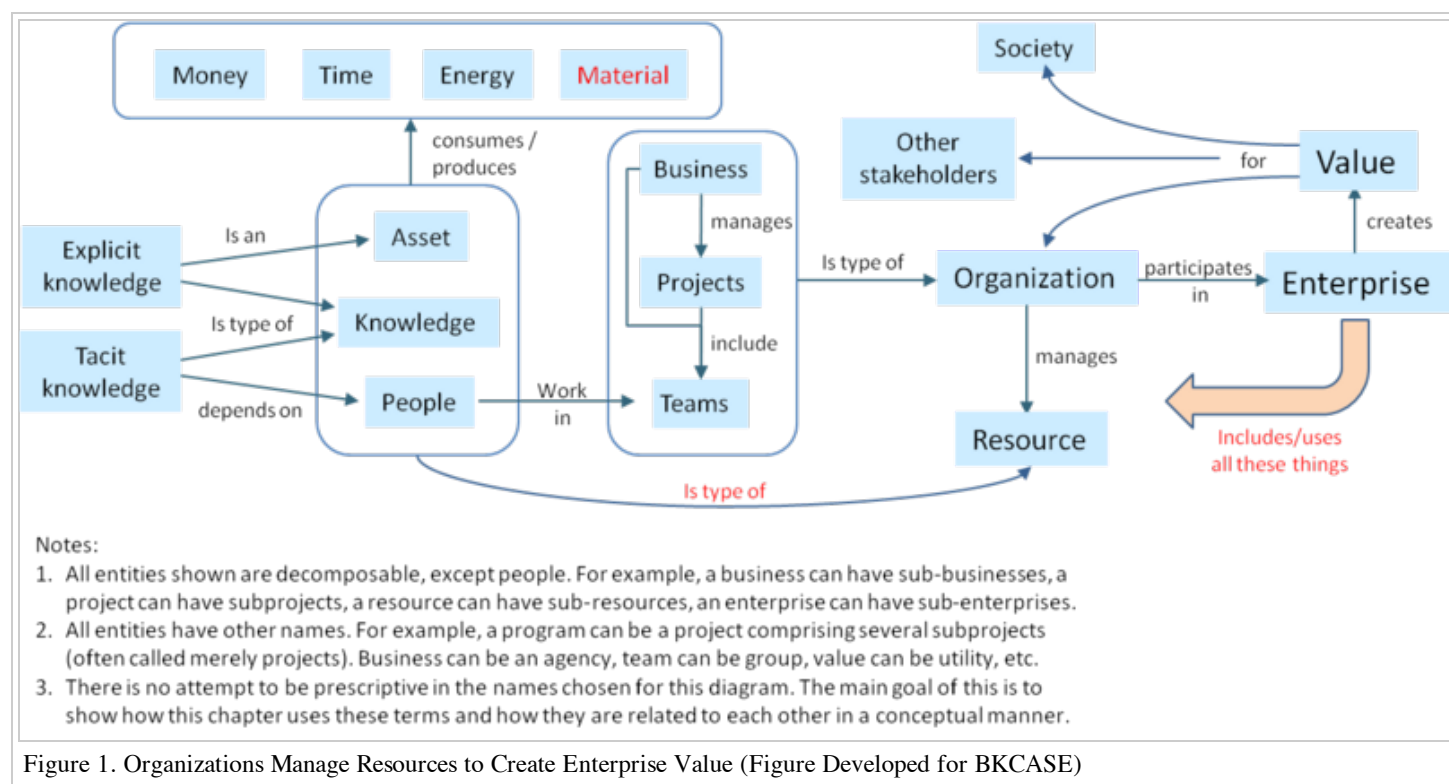
The basic concepts that drive the enterprise context of SE are shown in Figure 1. There are three types of organizations of interest – businesses, projects, and teams (see note 4). A typical business participates in multiple enterprises through its portfolio of projects. Large SE projects can be enterprises 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.*

## 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.

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.



*Note: Red text in this figure is to indicate changes from the version of this diagram in SEBOK draft v0.25.*

## Enabling SE 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 the project level. The article called Systems Engineering Organizational Strategy discusses enabling SE in the organization, while the article called Enabling Businesses and Enterprises to Perform Systems Engineering focuses on the cross-organizational functions at the business and enterprise levels. Competence of individuals is discussed in the article called Enabling Individuals to Perform Systems Engineering.

## Kinds of Knowledge Used by the Enterprise

Knowledge is a key resource for SE. 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.

## 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

- Dyer, L. and Ericksen, J. 2009. "Complexity-based Agile Enterprises: Putting Self-Organizing Emergence to Work." In A. Wilkinson et al (eds.). *"The Sage Handbook of Human Resource Management."* London, UK: Sage: 436–457.
- 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. 2009. "Engineering the Enterprise as a System." In *"Handbook of Systems Engineering and Management."*, eds. A. P. Sage, W. B. Rouse. 2nd ed. New York, NY, USA: Wiley and Sons, Inc.
- Stacey, R. 2006. "The Science of Complexity: An Alternative Perspective for Strategic Change Processes." In R. MacIntosh et al (eds.). *"Complexity and Organization: Readings and Conersations."* London, UK: Routledge: 74–100.

### Primary References

- 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*.", eds. A. P. Sage, W. B. Rouse. 2nd ed. New York, NY, USA: Wiley and Sons, Inc.
- Bernus, P., Nemes L., and Schmidt G., eds. 2003. "*Handbook on Enterprise Architecture*", Berlin & Heidelberg, Germany: Springer-Verlag.
- Valerdi, R. and Nightingale, D. J. 2011. "An Introduction to the Journal of Enterprise Transformation," *Journal of Enterprise Transformation*, 1(1), 1-6, 2011.

## Additional References

- Drucker, P. F. 1994. "The theory of business". *Harvard Business Review* (September/October 1994): 95-104.
- Fox, M., J. F. Chionglo, and F. G. Fadel. 1993. "A common sense model of the enterprise". Paper presented at A 3rd Industrial Engineering Research Conference, Norcross, GA, USA.
- Gøtze, J, ed. *Journal of Enterprise Architecture*. <https://www.aogea.org/journal>.
- Joannou, P. 2007. "Enterprise, systems, and software—the need for integration". *Computer*, IEEE, May 2007.
- 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. <http://www.tandf.co.uk/journals/UJET>.

---

[< Previous Article](#) | [Parent Article](#) | [Next Article >](#)

## Comments from SEBoK 0.5 Wiki

## SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

# Enterprise Systems Engineering Background

## 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 capability: 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).

The relationships among these elements are illustrated in the figure below, which connects to the previous figure at "People," "Organization," and "Enterprise." Organizational capabilities are addressed in the article on Systems Engineering Organizational Strategy, and individual competencies are addressed in the article on Enabling Individuals to Perform Systems Engineering as they relate to the principles, theories, and practices of organizational behavior.

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. See Figure 1 below.

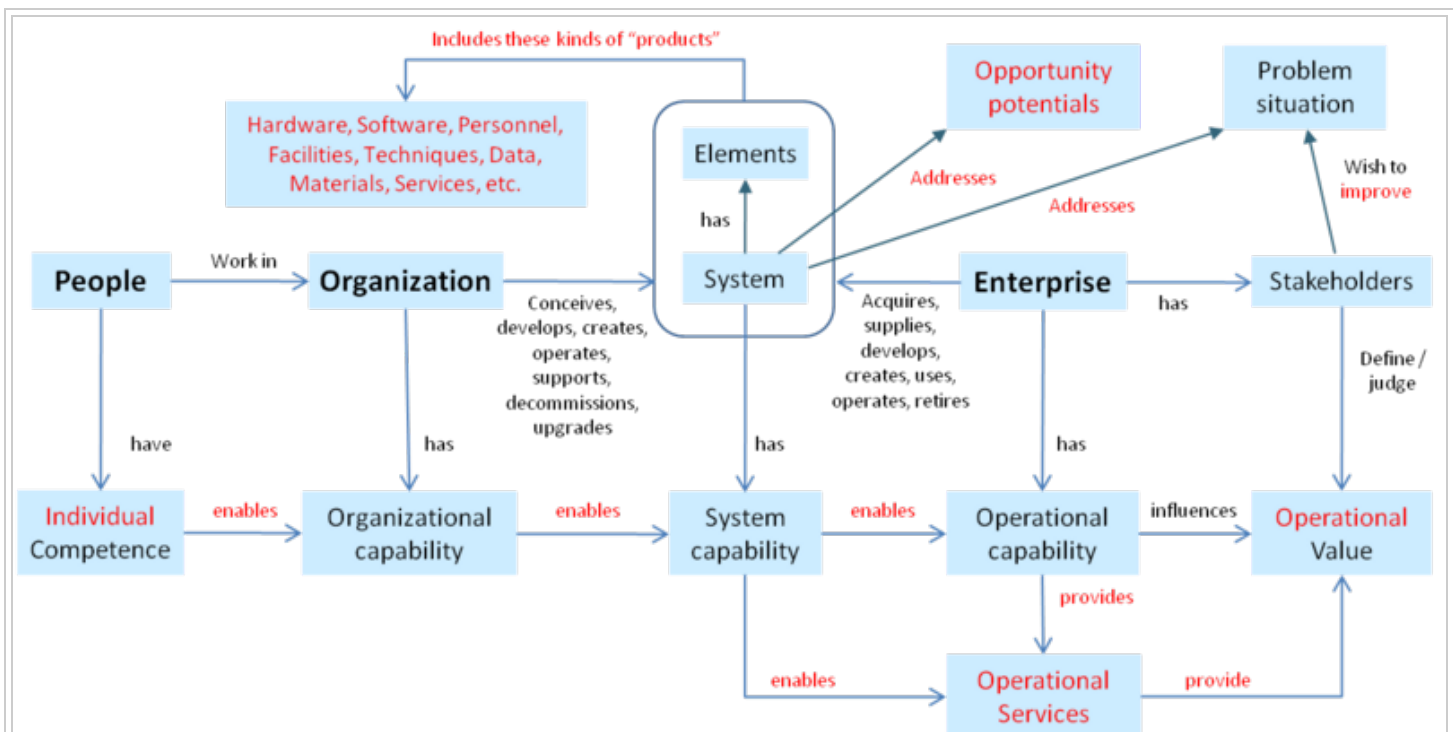


Figure 1. Individual Competence Leads to Organizational, System & Operational Capability (Figure Developed for BKCASE)

*Note: Red text in this figure is to indicate changes from the version of this diagram in SEBOK draft v0.25.*



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 (even a service can be one of the elements of the system).

## Organizational Design

The competencies of individuals are important to the overall organizational capability as discussed in the article on Enabling Individuals to Perform Systems Engineering. 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. This organizational design will be based on organizational design patterns and their tradeoffs, as discussed in the article on Systems Engineering Organizational Strategy.

## 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). Figure 2 below shows the typical kinds of components (shown here as “domains”) in an enterprise (Trous 2010) that may 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 Lawson (2010, chap. 8).

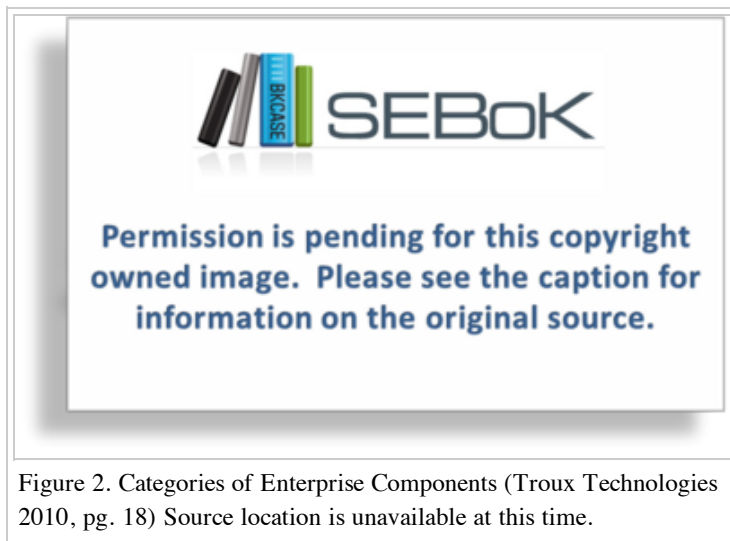


Figure 2. Categories of Enterprise Components (Trous Technologies 2010, pg. 18) Source location is unavailable at this time.

## 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 (Rouse 2009) that are essential to address in order to ensure that the enterprise will succeed:

- 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
- 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 it contributes 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 1).

*Note 1. “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.*

## 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.” The term “intraprise” is used here to denote the many systems internal to the enterprise. The intraprise includes more than just the information systems; it also includes the social and cultural “systems” of the enterprise (see the article called Enabling Businesses and Enterprises to Perform Systems Engineering).

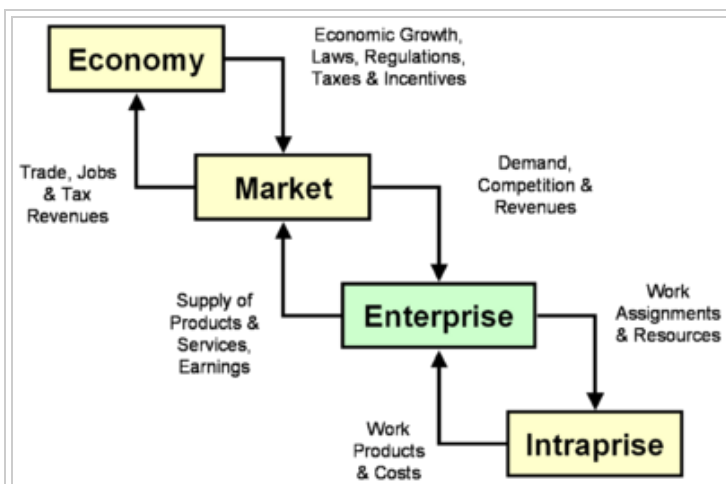


Figure 3. Context for Enterprise Transformation (Rouse 2009)

Reprinted with permission of John Wiley & Sons Inc.

## 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 don't) (Giachetti 2010) (See also Representing Systems with Models).

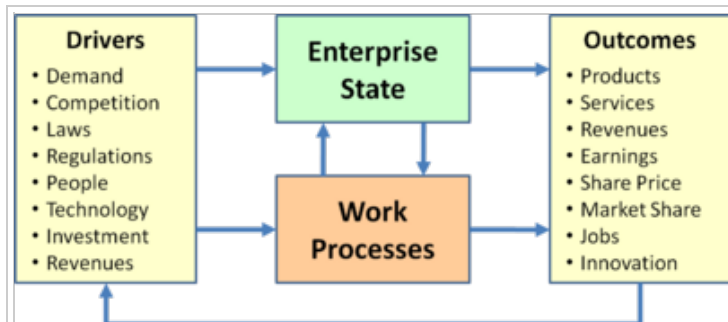


Figure 4. Drivers and Outcomes for the Enterprise (Rouse 2009)  
Reprinted with permission of John Wiley & Sons Inc.

## 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:

- Value Opportunities: The lure of greater success via market and/or technology opportunities prompts transformation initiatives.
- Value Threats: The danger of anticipated failure due to market and/or technology threats prompts transformation initiatives.
- Value Competition: Other players' transformation initiatives prompt recognition that transformation is necessary to continued success.
- 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).

## Enabling the Enterprise

ESE, by virtue of its inherent transdisciplinarity (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 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 system (cas) engineering (CASE), refer to White (2009, 1-16) and McCarter and White (2009, 71-105).

**Table 1. Executive Concerns and SE Enablers (Rouse 2009) Reprinted with permission of John Wiley & Sons Inc.**

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

## References

### Works Cited

- 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.
- Lawson, H. 2010. *"A Journey Through the Systems Landscape."* Kings College, UK: College Publications.
- McCarter, B.G. and B.E. White. 2009. "Emergence of SoS, sociocognitive aspects." In Jamshidi, M. ed. *"Systems of systems engineering: Principles and applications."* Boca Raton, FL, USA: CRC Press, Taylor & Francis Group: 71-105
- Miller, J., and S. Page. 2007. *"Complex Adaptive Systems: An Introduction to Computational Models of Social Life."* Princeton, NJ, USA: Princeton University Press.
- 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. 2008. "Health Care as a Complex Adaptive System: Implications for design and management." *The Bridge, National Academy of Engineering* 38 (1) (Spring 2008): 17-25.
- Rouse, W. B. 2009. "Engineering the Enterprise as a System." In eds. A. P. Sage, W. B. Rouse. 2nd ed. *"Handbook of systems engineering and management."* New York, NY: Wiley and Sons, Inc.
- Sage, A. P. 2000. "Transdisciplinarity Perspectives in Systems Engineering and Management." In eds. M. A. Somerville, D. Rappaport *"Transdisciplinarity: Recreating Integrated Knowledge."* 158-169. Oxford, UK: EOLSS Publishers.
- Srinivasan, J. 2010. "Towards a Theory Sensitive Approach to Planning Enterprise Transformation." Paper presented at 5th European Institute for Advanced Studies in Management (EIASM) Workshop on Organizational Change and Development, September 23-24, 2010, Vienna, Austria.
- Troux. 2010. "Metamodeling and modeling with Troux Semantics." Austin, TX, USA: Troux Technologies. Version 9, July 2010.
- von Bertalanffy, L. 1968. *"General System Theory: Foundations, Development, Applications."* Revised ed. New York, NY: 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." Paper presented at 1st Annual IEEE Systems Conference, 9-12 April, 2007, Honolulu, HI, USA.

White, B. E. 2009. "Complex Adaptive Systems Engineering (CASE)." Presented at IEEE Systems Conference 2009, Vancouver, Canada, 23-26 March 2009.

## 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 eds. A. P. Sage, W. B. Rouse. 2nd ed. *"Handbook of systems engineering and management."* New York, NY: Wiley and Sons, Inc.

Srinivasan, J. 2010. "Towards a Theory Sensitive Approach to Planning Enterprise Transformation." Paper 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)". IEEE Systems Conference 2009, Vancouver, Canada, 23-26 March 2009.

## Additional References

No additional references have been identified for version 0.75. Please provide any recommendations on additional references in your review.

---

[< Previous Article](#) | [Parent Article](#) | [Next Article >](#)

## Comments from SEBoK 0.5 Wiki

No comments were logged for this article in the SEBoK 0.5 wiki. Because of this, it is especially important for reviewers to provide feedback on this article. Please see the discussion prompts below.

## SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

# The Enterprise as a System

To enable more efficient and effective enterprise (glossary) transformation, the enterprise needs to be looked at “as a system (glossary),” 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 (glossary) of enterprise systems from product/service systems is the inclusion of people as a component (glossary) of the system, not merely as a user (glossary)/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)

## 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) (glossary) (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).*

## Value Stream Analysis

Value stream analysis 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.

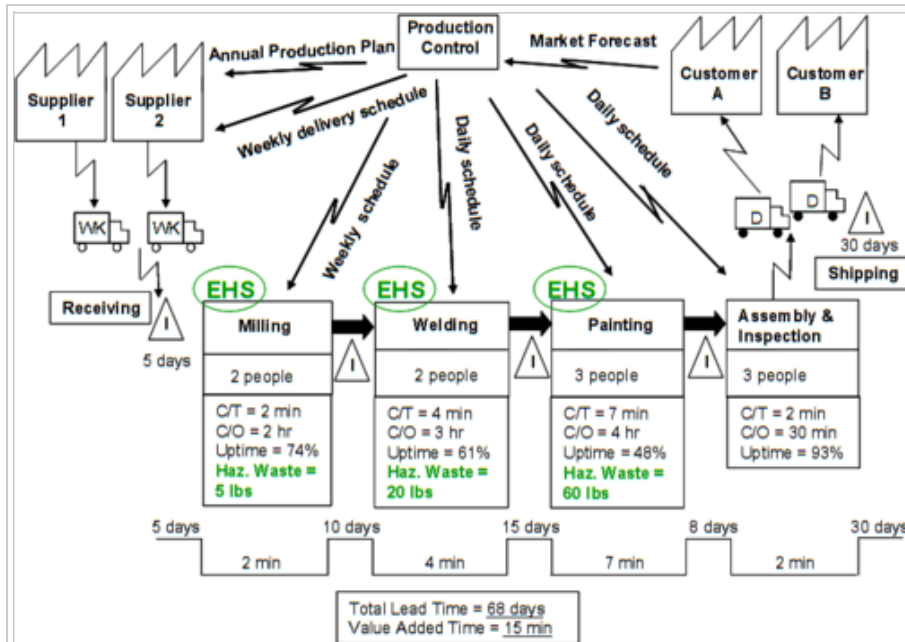


Figure 1. Value Stream Example (Source:

[http://en.wikipedia.org/wiki/Value\\_stream\\_mapping](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).

## 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 system-of-systems 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.
- **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 article called Systems of Systems (SoS).

## Federation of Systems

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). 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. An 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 a mistake 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 21st 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 SE and management by Sage and Cuppan (2001, 325-345):

- Subsidiarity
- Interdependence
- Uniform and standardized way of doing business
- Separation of powers
- Dual citizenship
- Scales of SE

## Scales of SE

According to Maier’s definition, an enterprise would not necessarily be called a SoS since the systems within the enterprise do not usually meet the criteria of operational and managerial independence. In fact, the whole purpose 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 and 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).

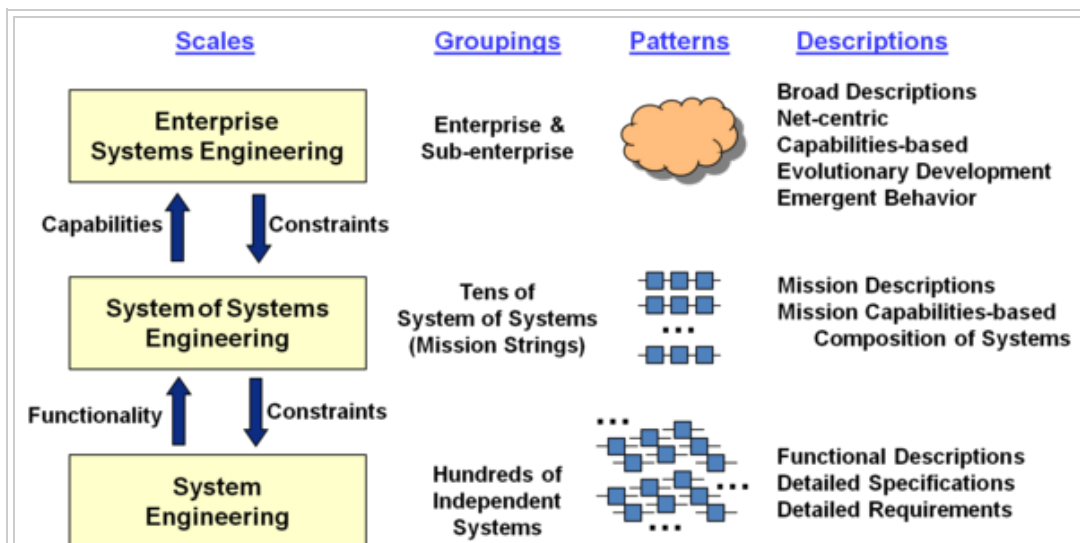


Figure. 2 Different Groupings and Patterns Revealed at Different Scales (DeRosa 2005) Reprinted with permission of © 2011. The MITRE Corporation. All Rights Reserved.

## Relationships between Enterprise and SoS

An enterprise may require a particular Operational Capability (glossary) that is brought into being by connecting together a chain of systems that together achieve that Capability (glossary). 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).

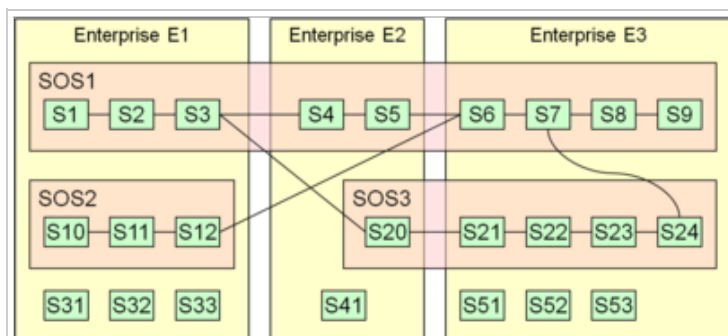


Figure 3. Relationships between an Enterprise and SoSs (Martin 2010) Reprinted with permission of The Aerospace Corporation.

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.

## References

### Works Cited

- AFSAB. 2005. "Report on System-of-Systems Engineering for Air Force Capability Development." Washington, DC: U.S. Air Force Scientific Advisory Board (AFSAB), U.S. 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*. November 2008.



- DeRosa, J.K. 2005. "Enterprise Systems Engineering," Presented at Air Force Association, Industry Day, Day 1, Danvers, MA, USA. 4 August 2005.
- 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]. Questa, NM, USA. Available from <http://www.parashift.com/docs/KmRaAeX.htm> Accessed September 6, 2011.
- DUS(AT). 2008. "Systems Engineering Guide for Systems of Systems, Version 1.0." Washington, D.C.: Deputy Under Secretary of Defense for Acquisition and Technology (DUS(AT))/U.S. Department of Defense (DoD) <http://www.acq.osd.mil/se/docs/SE-Guide-for-SoS.pdf>. Accessed September 6, 2011.
- 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* (May-June): 2-8.
- Handy, C. 1992. "Balancing Corporate Power: A New Federalist Paper." *Harvard Business Review* 70 (6) (November/December): 59-67.
- Hines, P. and N. Rich. 1997. "The Seven Value Stream Mapping Tools." *International Journal of Operations & Production Management* 1 (17).
- Joannou, P. 2007. "Enterprise, Systems, and Software — the Need for Integration." *Computer*, IEEE, May 2007.
- 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." Paper presented at 20th Anniversary International Council on Systems Engineering (INCOSE) International Symposium, 12-15 July, 2010, Chicago, IL, 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.
- Rouse, W. B. 2009. "Engineering the enterprise as a system." In A. P. Sage, W. B. Rouse. eds. *"Handbook of systems engineering and management."* 2nd ed. New York, NY, USA: Wiley and Sons, Inc.
- 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.
- 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) (December 2001): 325-45.
- Swarz, R. S. , J. K. DeRosa, and G. Rebovich 2006. "An Enterprise Systems Engineering Model," INCOSE Symposium Proceedings.

## 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*, IEEE, May 2007.

Rouse, W. B. 2009. "Engineering the Enterprise as a System." In A. P. Sage, W. B. Rouse. eds. *Handbook of systems engineering and management*. 2nd ed. New York, NY, USA: Wiley and Sons, Inc.

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.

## Additional References

Arnold, S., and H. Lawson. 2004. "Viewing Systems From a Business Management Perspective." *Systems Engineering, the Journal of the International Council on 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, the Journal of the International Council on Systems Engineering (INCOSE)* 4 (2): 118-33.

Nightingale, D., and D. Rhodes. 2004. "Enterprise systems architecting: Emerging art and science within engineering systems." Paper 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." Paper 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." *INSIGHT, A Publication of the International Council on Systems Engineering (INCOSE)* 6 (2) (January 2004).

Ring, J. 2004. "Seeing an Enterprise as a System." *INSIGHT, A Publication of the International Council on Systems Engineering (INCOSE)* 6 (2) (January 2004).

Valerdi, R., D. Nightingale, and C. Blackburn. 2009. "Enterprises as Systems: Context, Boundaries, and Practical Implications." *Information-Knowledge-Systems Management Journal* 7 (4) (December 2008): 377-99.

---

[< Previous Article](#) | [Parent Article](#) | [Next Article >](#)

## Comments from SEBoK 0.5 Wiki

## SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

# Related Business Activities

The following business (glossary) management activities can be supported by Enterprise Systems Engineering (ESE) (glossary) (ESE) activities:

- Mission & Strategic Planning
- Business Processes & Information Management
- Performance Management
- Portfolio Management
- Resource Allocation & Budgeting
- Program & Project Management

The figure below shows how these business activities relate to each other as well as the relative scope of ESE and product SE (Martin 2010).

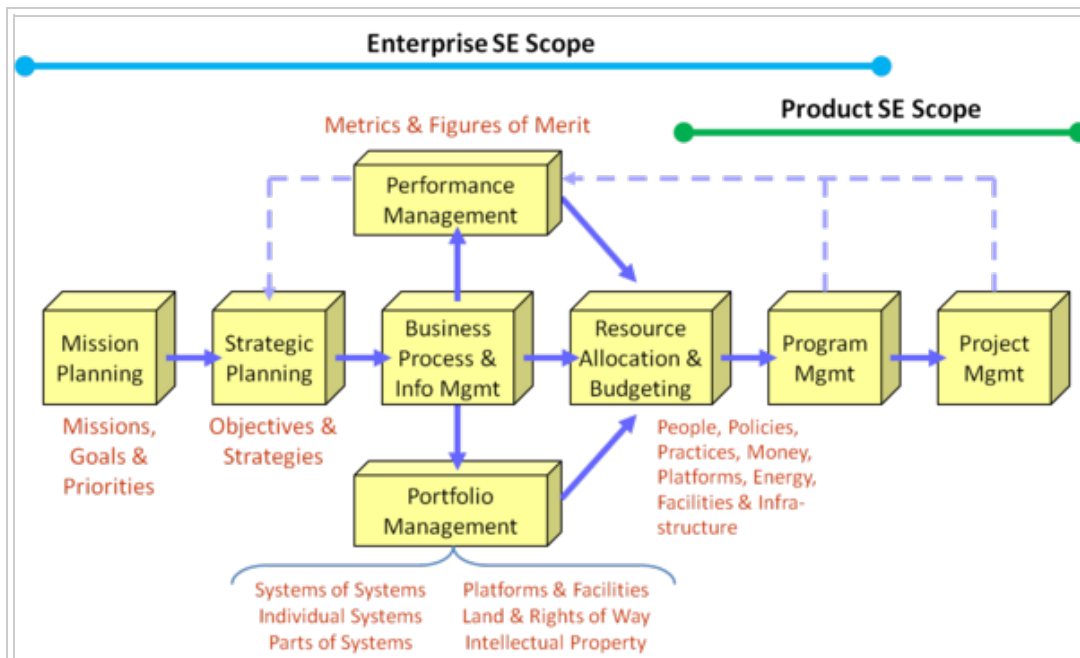


Figure 1. Related Business Activities (Martin 2010) Reprinted with permission of The Aerospace Corporation.

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. Traditional Systems Engineering (TSE) uses “requirements” to specify the essential features and functions of a system. An Enterprise (glossary), 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 (glossary) to discriminate between options and to select the appropriate balanced Portfolio (glossary) of systems and other enterprise resources.

The first three activities listed above are covered in Enabling Businesses and Enterprises to Perform Systems Engineering. The other business management activities are described in more detail below in 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 inherently uses the PDCA cycle as one of its fundamental tenets. After ESE develops the enterprise transformation plan, 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 replanning may be required. 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](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)

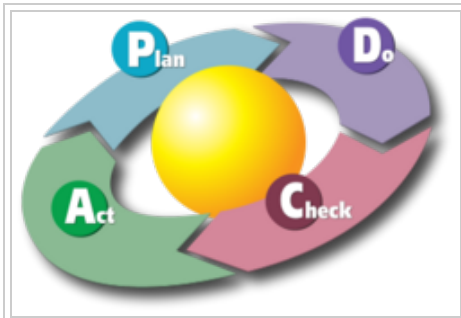


Figure 2. PDCA Cycle (Source: <http://en.wikipedia.org/wiki/PDCA>. Accessed July 2010 from the Public Domain. Diagram by Karn G. Bulsuk (<http://blog.bulsuk.com>))

## Portfolio Management

Program and Project Managers direct their activities as they relate to the systems under their control. Enterprise management, on the other hand, means directing the portfolio of items that are necessary to achieving the enterprise goals and objectives. 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) (glossary) 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 (glossary) 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 (glossary). 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 help it achieve its goals and objectives.

## Resource Allocation and Budgeting

The resource allocation activity is driven by the portfolio management definition of the optimal set of portfolio elements. Capability (glossary) 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.

## 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 (glossary) 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 (glossary) 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 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 (glossary), 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 15288 2008; Wasson 2006; INCOSE 2010; Blanchard and Fabrycky 2010).

## 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 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.

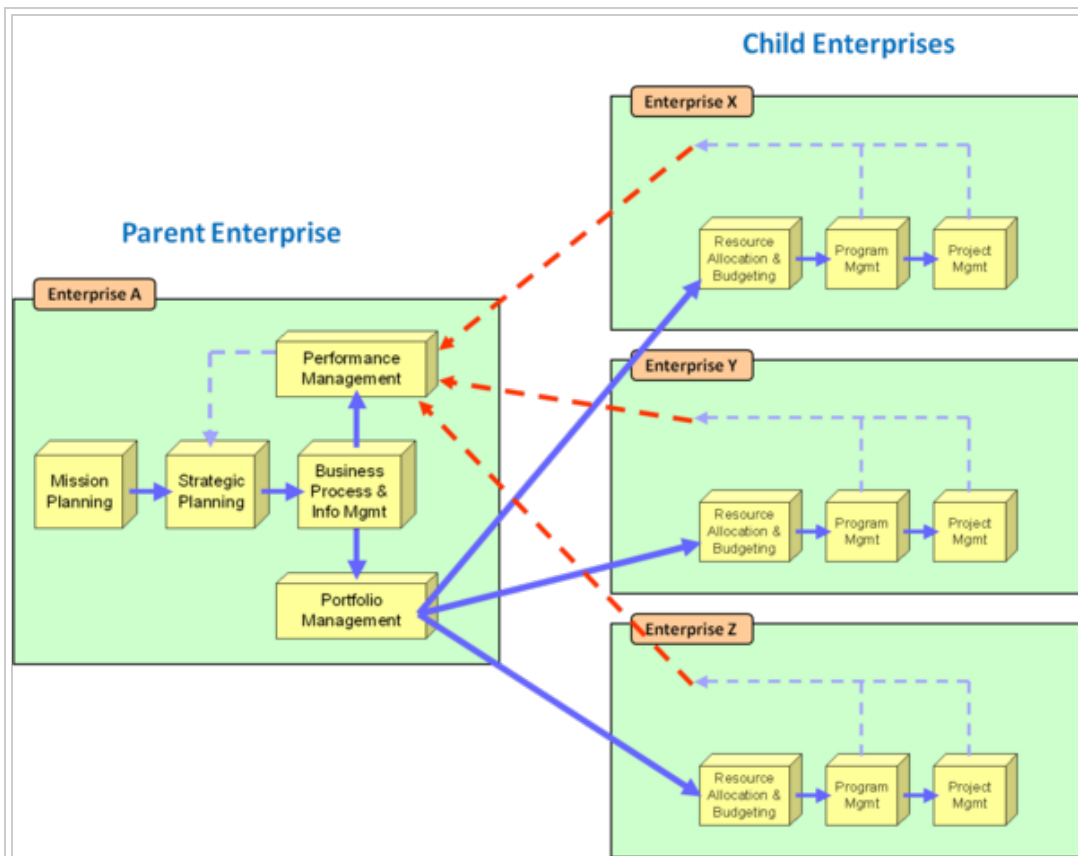


Figure 3. Parent and Child Enterprise Relationships (Martin 2010) Reprinted with permission of The Aerospace Corporation.

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.

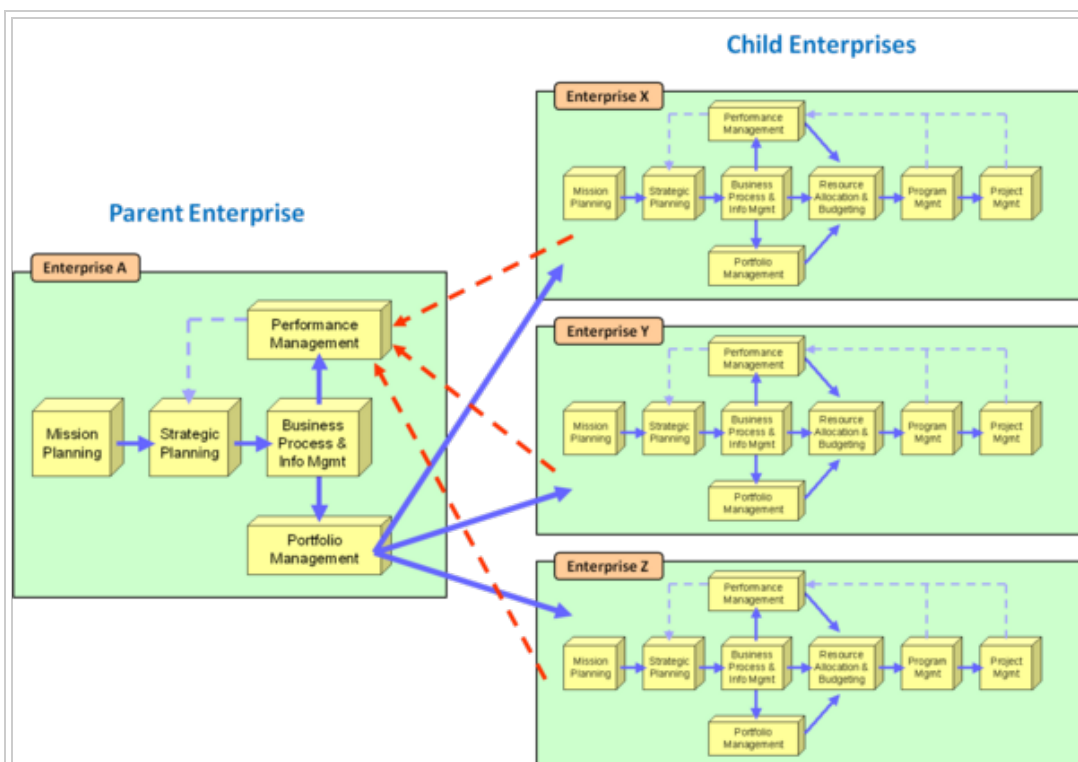


Figure 4. Mission and Strategic Planning at All Levels of Cooperating Enterprises (Martin 2010)

Reprinted with permission of The Aerospace Corporation.

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 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

- 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.
- 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]. 2010a Available from <http://www.ietf.org/overview.html> Accessed September 6, 2011.
- 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]. <http://www.ietf.org/tao.html#intro> Accessed September 6, 2011.
- INCOSE. 2010. *"INCOSE systems engineering handbook, version 3.2."* San Diego, CA, USA: International Council on Systems Engineering (INCOSE), INCOSE-TP-2003-002-03.2.
- ISO/IEC 15288. 2008. *"Systems and Software Engineering - System Life Cycle Processes."* Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC), ISO/IEC 15288:2008 (E).
- 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. 2010. "An Enterprise Systems Engineering Framework." Paper presented at 20th Anniversary International Council on Systems Engineering (INCOSE) International Symposium, 12-15 July, 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.
- 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. 2010. "An Enterprise Systems Engineering Framework." Paper presented at 20th Anniversary International Council on Systems Engineering (INCOSE) International Symposium, 12-15 July, 2010, Chicago, IL, USA.

### Additional References

- Arnold, S., and H. Lawson. 2004. "Viewing Systems from a Business Management Perspective." *Systems Engineering, the Journal of the International Council on 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, the Journal of the International Council on Systems Engineering (INCOSE)* 4 (2): 118-33.

Drucker, P. F. 1994. "The Theory of Business." *Harvard Business Review* (September/October 1994): 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 Management College.

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 >

## Comments from SEBoK 0.5 Wiki

## SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

## Enterprise Systems Engineering Key Concepts

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 (glossary) “single” system (glossary) (Blanchard and Fabrycky 2010; Hall 1989; Sage and Rouse 2009). Enterprise Systems Engineering (ESE) (glossary) (ESE) expands beyond this traditional basis to “consider the full range of SE services increasingly needed in a modern organization (glossary) 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 (glossary) of government acquisition (glossary) of very large, complex military and civil systems (e.g., F22 fighter jet and air traffic control system).

ESE encompasses this traditional role in system acquisition, but also incorporates Enterprise (glossary) strategic planning and enterprise investment analysis. 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



The MITRE Corporation has done significant development of ESE practices.

*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.” In addition to the TSE Process (glossary) areas, MITRE has included the following process areas in their ESE process (DeRosa 2005) to close the gap between ESE and product SE:

- Strategic Technical Planning
- Enterprise Architecture
- Capabilities-Based Planning Analysis
- Technology Planning
- 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.

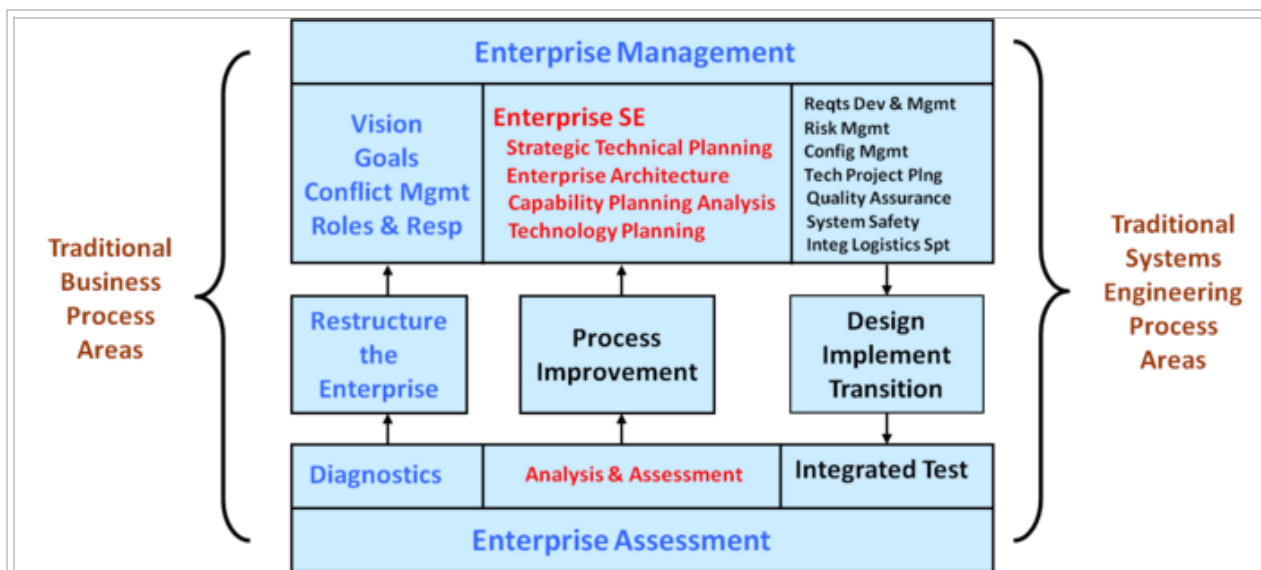


Figure 1. Enterprise SE Process Areas in the Context of the Entire Enterprise (DeRosa 2006) Reprinted with permission of © 2011. The MITRE Corporation. All Rights Reserved.

SE is viewed by many organizations and depicted in many process definitions as bounded by the beginning and end of a system development Project (glossary). 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 (glossary) purpose. Elliott and Deasley (Elliott and Deasley 2007) discuss the differences between development phase SE and in-service SE.

In contrast to TSE, ESE 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 (glossary). 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 (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 (glossary) 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 et al. 2006) (See also Complexity)*

## Extensions Beyond Traditional SE

These considerations were taken into account when extending the process areas for ESE. The acquisition-oriented SE concept areas in the Friedman-Sage framework (Friedman and Sage 2004, 84-96) were extended to cover the strategic and investment-oriented ESE activities described above. Four new process areas were added to the list as shown in the table below. Additions or changes are shown in *italics*.

To emphasize the fact that an enterprise is dealing with many systems within its portfolio of systems, the word “enterprise” is used in the first two process areas dealing with requirements and architecture (glossary). Portfolio and Program (glossary) views are emphasized, as well.

The word “program” is used to replace “system” since the programs are the system “elements” of the enterprise when you treat the Enterprise as a system. Projects are like “subsystems” of these programs. It is really the life cycle of the programs that is supported by enterprise management, while the Programs themselves are responsible for managing and supporting the life cycles of their particular systems. The word “portfolio” is used to emphasize the collections of Programs within the enterprise that must also be managed as a group.

**Table 1. Extension of TSE to ESE Process Areas (Martin 2010) Reprinted with permission of The Aerospace Corporation.**

Acquisition-Oriented Process Areas	Strategic-and Investment-Oriented Process Areas
<i>Requirements Definition and Management</i>	Enterprise Requirements Definition and Management
<i>System Architecture and Conceptual Design</i>	Enterprise Architecture and Conceptual Design
<i>System and Subsystem Detailed Design and Implementation</i>	Program and Project Detailed Design and Implementation
<i>Systems Integration and Interfaces</i>	Program Integration and Interfaces
<i>Validation and Verification</i>	Program Validation and Verification
<i>Deployment and Post Deployment</i>	Portfolio & Program Deployment and Post Deployment
<i>Life Cycle Support</i>	Portfolio & Program Life Cycle Support
<i>Risk Assessment and Management</i>	Opportunity and Risk Assessment and Management
<i>System and Program Management</i>	(expanded to include the 4 areas below)
	Strategic Technical Planning
	Capability-Based Planning Analysis
	Technology and Standards Planning
	Enterprise Evaluation and Assessment

## Enterprise Entities and Relationships

An enterprise “system” has different entities and relationships than you might find in a product/service system (see note 5). Products and services are usually treated as “assets” as shown in the figure below (Trous 2010). Other enterprise entities of interest are things like information, knowledge, skills, finances, policies, process, strategy, markets, and resources.

*Note 5. 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.

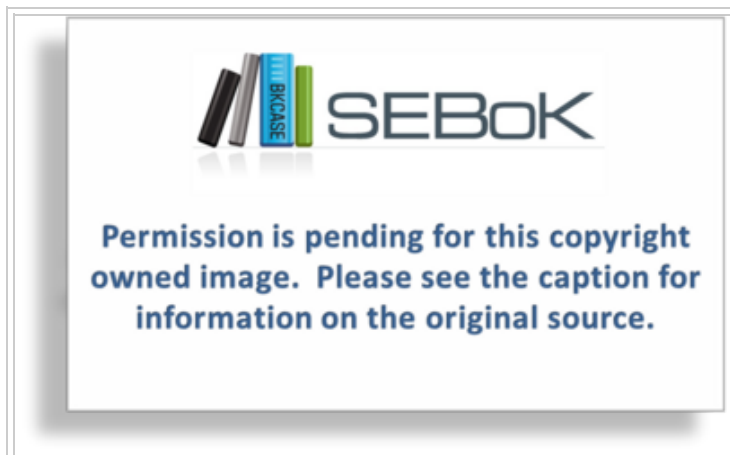


Figure 2. Asset Domain and Concept Domain Categories for Enterprise Entities (Troux Technologies 2010, pg. 17, Table 2)  
Source location is unavailable at this time.

The figure below shows examples of entities for applications, software, infrastructure and hardware in the asset domains of an enterprise along with typical entities for the analysis domain such as Risk (glossary), constraints, findings, recommendations, factors, and trends. Also shown here are the relevant relationships between such entities. This is from the Troux Semantics metamodel used in the Troux Architect Modeling Tool (glossary) for enterprise architecture activities. Other enterprise modeling tools have similar metamodels (or sometimes called “schemas”).

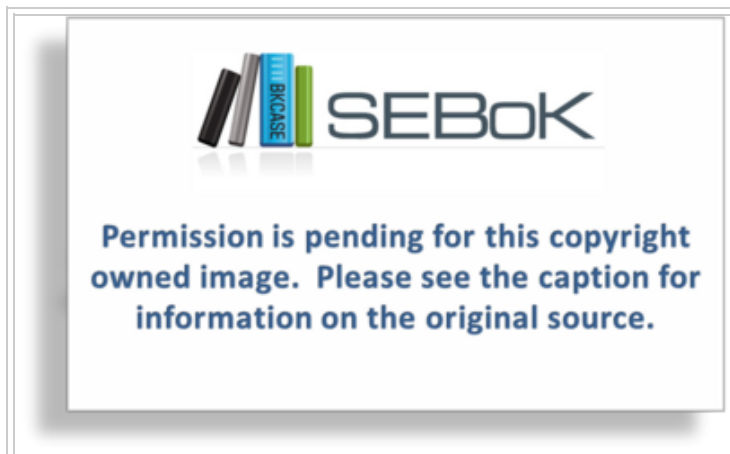


Figure 3. Examples of Enterprise Entities & Relationships (Troux Technologies 2010, pg. 22, 24, 42) Source location is unavailable at this time.

## Enterprise Architecture Frameworks & Methodologies

There are various frameworks and methodologies available that assist in the development of an enterprise architecture (glossary). Some of these are described below.

### TRAK Framework

The “standard” entities and relationships used in architecture modeling of an enterprise are specified in metamodels and viewpoint specifications in various domain-specific architecture frameworks. The figure below, as one example, shows the metamodel for the TRAK architecture framework (TRAK 2011).

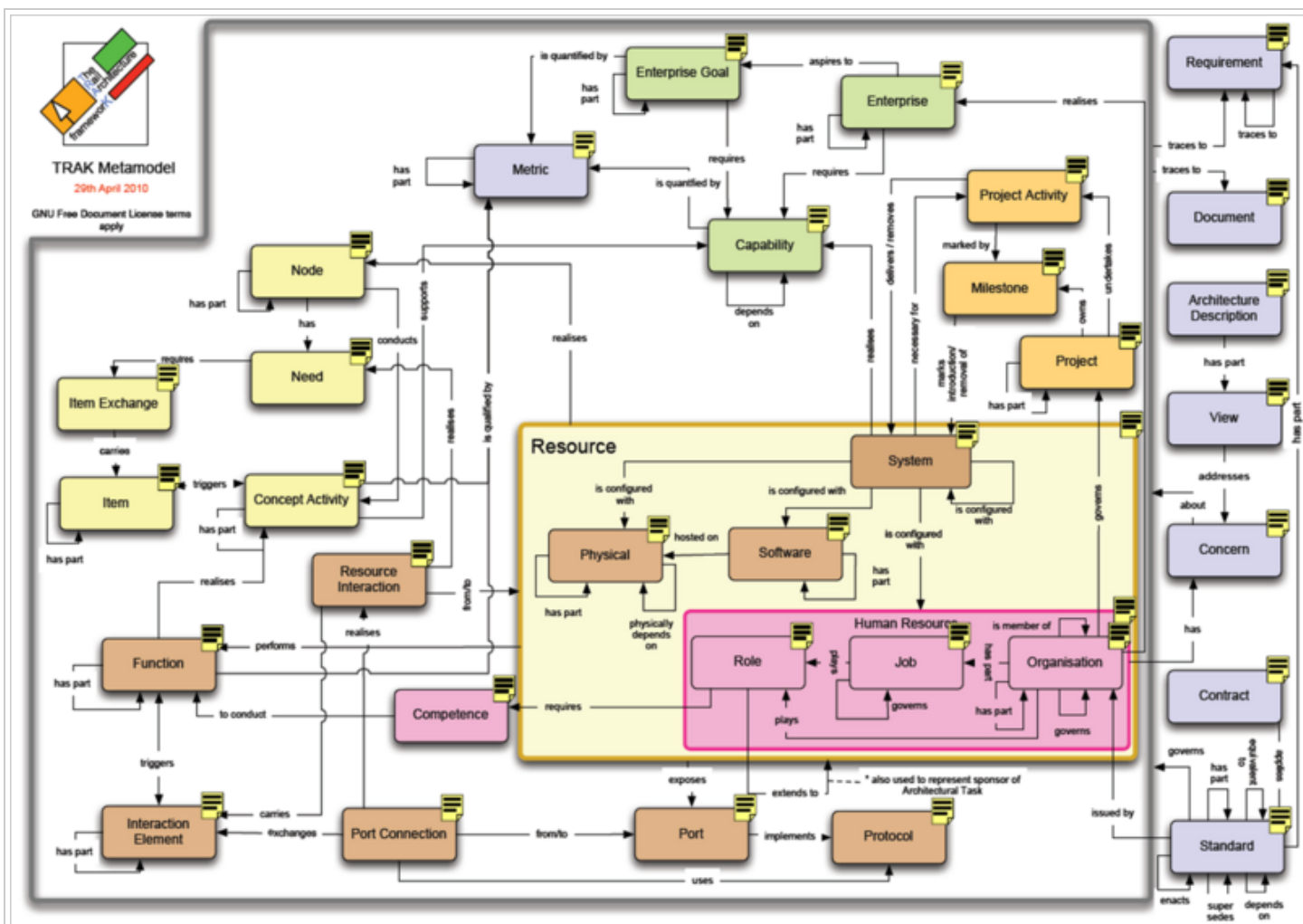


Figure 4. TRAK Metamodel (TRAK 2011) Released under the GNU Free Documentation License of Copyright (C) 2010 - 2011 UK Department for Transport. Source is available at <http://trakmetamodel.sourceforge.net>

## DODAF Framework

The figure below shows the metamodel for the United States Department of Defense (DoD) Architecture framework (DoD 2010).

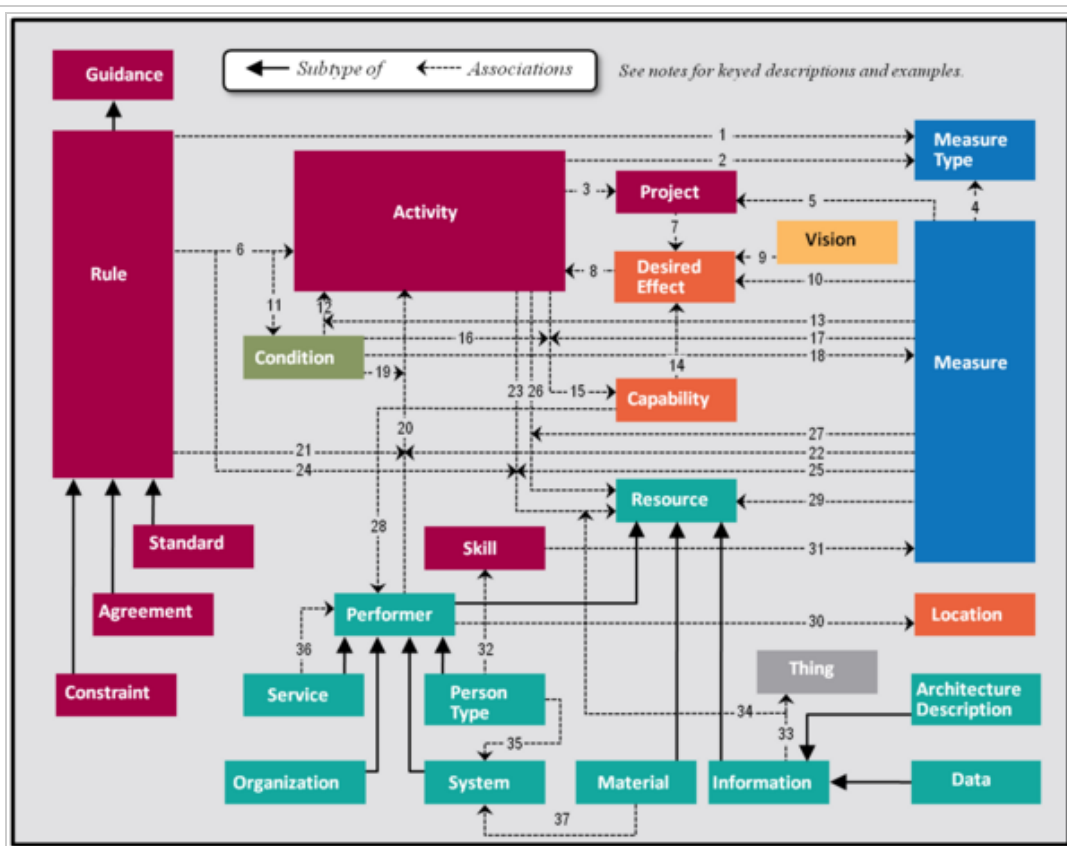


Figure 5. DoDAF Conceptual Data Model (DoD 2010) Released by the United States Department of Defense.

## TOGAF Framework

Some frameworks (like The Open Group Architecture Framework (TOGAF)) are more properly called methodologies since they focus on the process (see figure below) by which artifacts are created and how they are used. Other frameworks (like Zachman and CIO Council 1999) are more properly called taxonomies since they define and categorize the kinds of elements of interest to the enterprise analyst (Ref: A Comparison of the Top Four Enterprise-Architecture Methodologies, <http://msdn.microsoft.com/en-us/library/bb466232.aspx>).

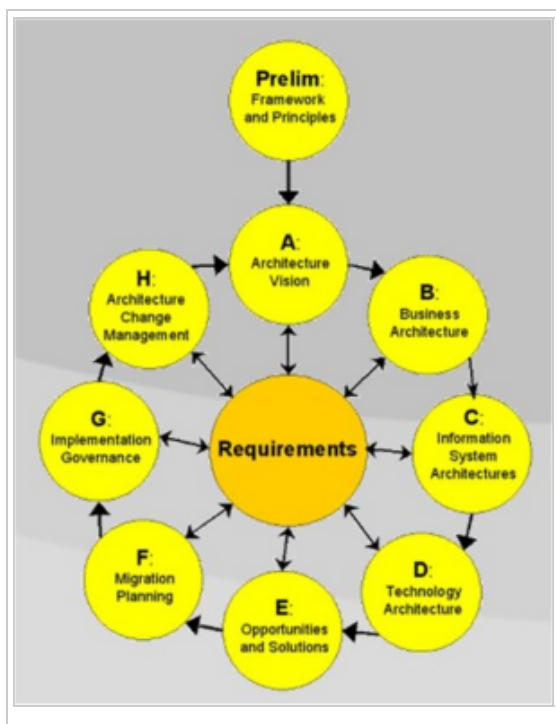


Figure 6. TOGAF Methodology (TOGAF 2009 - Image credited to Marley NASA /SCI 2003)  
(Source:  
[http://en.wikipedia.org/wiki/File:TOGAF\\_ADM.jpg](http://en.wikipedia.org/wiki/File:TOGAF_ADM.jpg))  
Accessed September 16, 2011. Released in the public domain from NASA.

## Zachman Framework

The figure below shows the Zachman architecture framework (taxonomy) (Zachman 1987 and 1992). The columns represent the six “interrogatives” of why, how, what, who, where, and when, and these can be considered to be “stakeholder concerns” of the enterprise stakeholders. These columns also represent data (i.e., the “what”), functions, networks, people, time, and motivation. The rows represent the different stakeholder “perspectives”: contextual (planners), conceptual (owners), logical (designers), physical (builders), and detailed (subcontractors or suppliers). These rows also represent the following “perspectives”: Scope (glossary) (i.e., contextual), business model, system model, technology model, and detailed representations.

	Why	How	What	Who	Where	When
Contextual	Goal List	Process List	Material List	Organizational Unit & Role List	Geographical Locations List	Event List
Conceptual	Goal Relationship	Process Model	Entity Relationship Model	Organizational Unit & Role Rel. Model	Locations Model	Event Model
Logical	Rules Diagram	Process Diagram	Data Model Diagram	Role relationship Diagram	Locations Diagram	Event Diagram
Physical	Rules Specification	Process Function Specification	Data Entity Specification	Role Specification	Location Specification	Event Specification
Detailed	Rules Details	Process Details	Data Details	Role Details	Location details	Event Details

Figure 7. Zachman Framework (Zachmann 1992 –Image credited to Dekker and SunSw0rd) (Source: [http://en.wikipedia.org/wiki/File:Zachman\\_Framework\\_Model.svg](http://en.wikipedia.org/wiki/File:Zachman_Framework_Model.svg) ) Accessed September 16, 2011. Released in the public domain by Wikimedia guidelines.

## References

### Works Cited

- 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.
- Carlock, P. and R. Fenton. 2001. “System of Systems (SoS) Enterprise Systems Engineering for Information-Intensive Organizations,” *Systems Engineering Journal* 4 (4): 242-261.
- CIO Council 1999. "Federal Enterprise Architecture Framework (FEAF), Version 1.1" Washington, DC, USA: Federal Chief Information Officers Council. [http://www.cio.gov/documents\\_details.cfm/uid/1F432311-2170-9AD7-F2053C10765E0E1C/structure/Enterprise%20Architecture/category/Enterprise%20Architecture](http://www.cio.gov/documents_details.cfm/uid/1F432311-2170-9AD7-F2053C10765E0E1C/structure/Enterprise%20Architecture/category/Enterprise%20Architecture) Accessed on September 7, 2011.
- DeRosa, J.K. 2005. “Enterprise Systems Engineering,” Presented at Air Force Association, Industry Day, Day 1, Danvers, MA, USA. 4 August 2005.
- 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." (17). London, England, UK: Royal Academy of Engineering.

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*." International series on systems science and engineering. 1st ed. Vol. 3. Oxford, UK: Pergamon Press.

Hitchins, D. 1993. "*Putting Systems to Work*." New York, NY, USA: John Wiley & Sons.

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, 10-15 July, 2005, Rochester, NY, USA.

Martin, J.N. 2010. "An Enterprise Systems Engineering Framework" Paper presented at 20th Anniversary International Council on Systems Engineering (INCOSE) International Symposium, 12-15 July, 2010, Chicago, IL, USA.

MITRE. 2004. "*MITRE 2004 Annual Report*." McLean, VA, USA: MITRE Corporation.

Rebovich, G. 2006. "Systems Thinking for the Enterprise: New & Emerging Perspectives." Paper 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.

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," INCOSE Symposium Proceedings.

TOGAF. 2009. "The Open Group Architecture Framework." Version 9. <http://www.opengroup.org/togaf/> Accessed September 7, 2011.

TRAK. 2011. "TRAK Enterprise Architecture Framework." <http://trak.sourceforge.net/index.html> Accessed September 7, 2011.

Troux. 2010. "Metamodeling and Modeling with Troux Semantics." Austin, TX, USA: Troux Technologies. Version 9, July 2010.

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, 10-15 July, 2005, Rochester, NY, USA.

Martin, J.N. 2010. "An Enterprise Systems Engineering Framework." Paper presented at 20th Anniversary International Council on Systems Engineering (INCOSE) International Symposium, 12-15 July, 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 and Francis Group.

Swarz, R.S., J.K. DeRosa, and G. Rebovich 2006. "An Enterprise Systems Engineering Model." INCOSE Symposium Proceedings.

## Additional References

Gøtze, J, ed. *Journal of Enterprise Architecture*. <https://www.aogea.org/journal>.

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.

Vernadat, F. B. 1996. *"Enterprise Modelling and Integration - Principles and Applications."* London, UK: Chapman and Hall.

---

< Previous Article | Parent Article | Next Article >

## Comments from SEBok 0.5 Wiki

No comments were logged for this article in the SEBoK 0.5 wiki. Because of this, it is especially important for reviewers to provide feedback on this article. Please see the discussion prompts below.

## SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

# Enterprise Systems Engineering Process Activities

## Systems Engineering Role in Transforming the Enterprise

### Enabling Systematic Enterprise Change

The Systems Engineering (glossary) (SE) Process (glossary) as applied to the Enterprise (glossary) 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) (glossary) (ESE) is that it “determines the balance between complexity (see also 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). They provide a reasonably good definition for an enterprise that captures well 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.*

### Effectiveness vs. 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)

### Enterprise Management Process Areas

The “System and Program (glossary) Management” process area is dropped from the matrix (see Table 1 in Enterprise Systems Engineering Key Concepts) since, in an enterprise, this process area is not directly applicable. Instead, this is the responsibility of



the Programs within the Enterprise. This process area is replaced with the following four process areas (which, altogether, are the key elements of the Enterprise Management process that oversees the “SE” of the enterprise as a whole):

- Strategic Technical Planning
- Capability-Based Planning Analysis
- Technology and Standards Planning
- Enterprise Evaluation and Assessment

## Strategic Technical Planning

The purpose of strategic technical planning 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 transdisciplinary technical principles and practices from psychology, sociology, organizational change management, etc. This process uses the roadmaps developed during technology and standards planning. 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 (glossary) to avoid or an Opportunity (glossary) to pursue in the technical strategy. The technical strategy is defined in terms of implementation guidance for the programs and projects.

## 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 (glossary) capability roadmaps (BCRMs) are related the Operational (glossary) 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 (glossary) 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 SE competency, see the Enabling Individuals to Perform Systems Engineering article.)

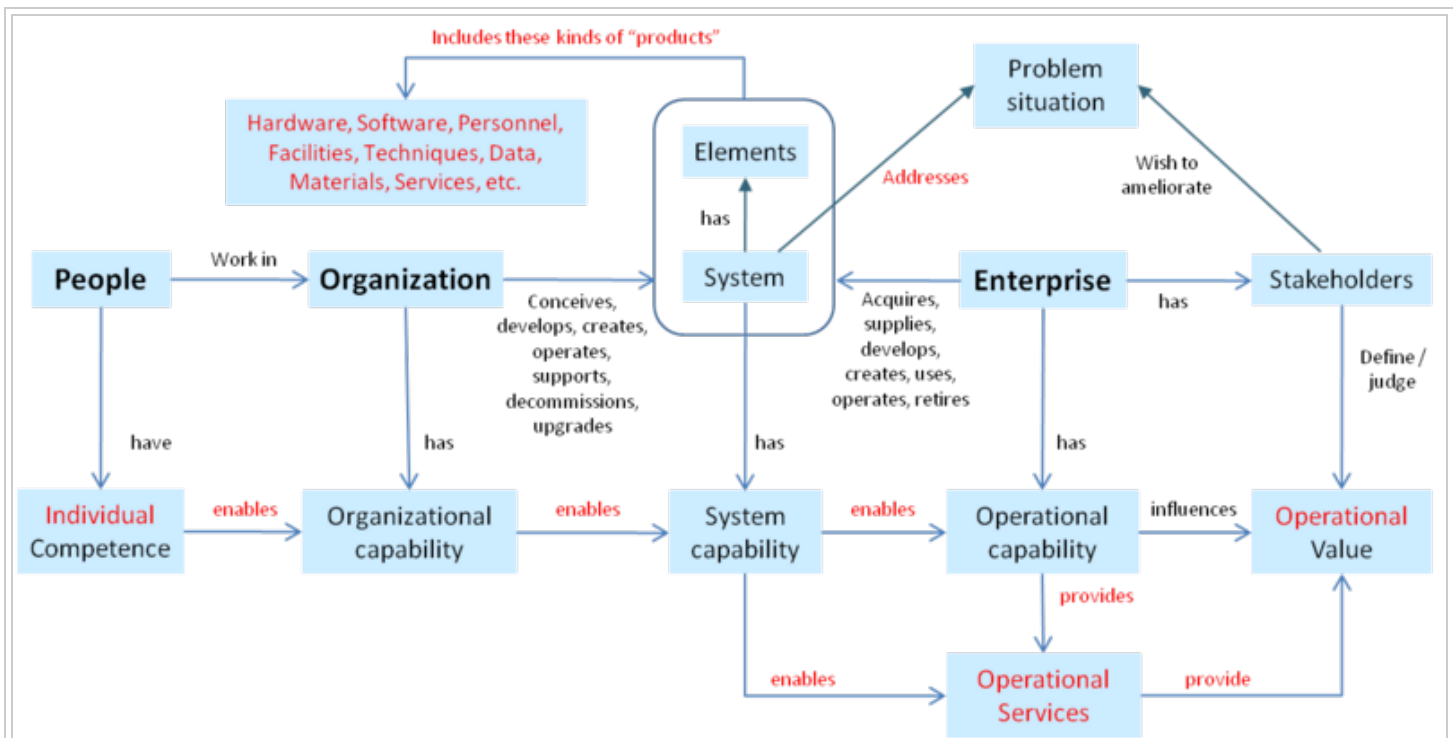


Figure 1. Organizational, System & Operational Capabilities (Figure Developed for BKCASE)

## 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 (glossary) 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 (glossary) and Agility (glossary) 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
- In-line, continuous performance monitoring and selective forensics (Roberts 2006).

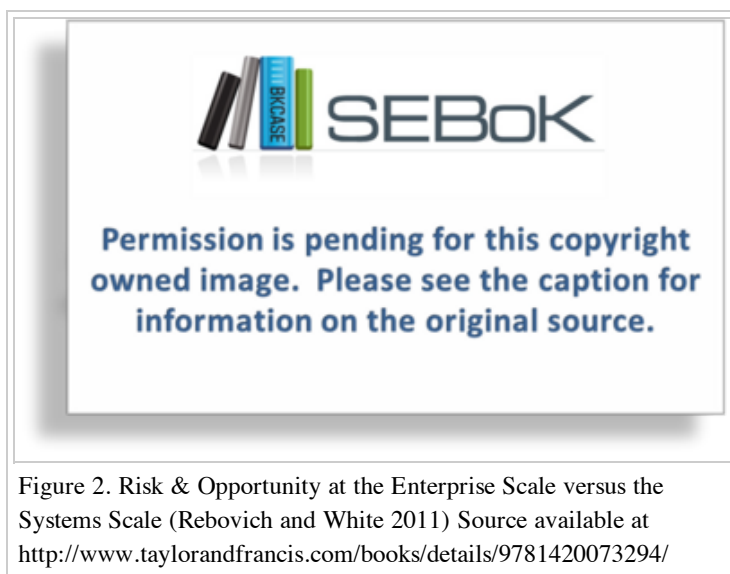
Enterprise Architecture (glossary) (EA) can be used as a primary tool in support of evaluation and assessment. EA can be used to provide a Model (glossary) to understand how the parts of the enterprise fit together (or don't) (Giachetti 2010). The Structure (glossary) and contents of the EA should be driven by the key business decisions (or, as shown in the six-step process in (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 in (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. In other words, EA can be viewed as either the "noun" (i.e., things) or the "verb" (i.e., actions). The literature uses these terms in both senses and it is not always clear in each case which sense is intended.

## 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.



# Enterprise Architecture and Requirements

EA goes above and beyond the technical components of Product (glossary) 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, Laszlo, 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 2007) is expanding its scope to include requirements on architecture frameworks (to be published in 2012).

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 15704 2000; ISO/IEC 10746 1998; NATO 2004; 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
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 : 3-9.

Bernus, P., L. Nemes, and G. Schmidt, eds. 2003. "*Handbook on Enterprise Architecture*", Berlin & Heidelberg, Germany: Springer-Verlag.

CIO Council. 1999. "Federal Enterprise Architecture Framework (FEAF), Version 1.1" Washington, DC, USA: Federal Chief Information Officers Council. [http://www.cio.gov/documents\\_details.cfm/uid/1F432311-2170-9AD7-F2053C10765E0E1C/structure/Enterprise%20Architecture/category/Enterprise%20Architecture](http://www.cio.gov/documents_details.cfm/uid/1F432311-2170-9AD7-F2053C10765E0E1C/structure/Enterprise%20Architecture/category/Enterprise%20Architecture) Accessed on September 7, 2011.

DoD. 2010. "DoD architecture framework (DoDAF)". version 2.0. Washington, DC: U.S. 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: Rick Doctor & Partners; Marcel Dekker, Inc.
- ISO 15704. 2000. *"Industrial Automation Systems — Requirements for Enterprise-Reference Architectures and Methodologies."* Geneva, Switzerland: International Organization for Standardization (ISO), ISO 15704:2000.
- ISO/IEC 42010. 2007. *"Systems and Software Engineering — Recommended Practice for Architectural Description of Software-Intensive Systems."* Geneva, Switzerland: International Organization for Standards (ISO)/International Electrotechnical Commission (IEC), ISO/IEC 42010:2007.
- ISO/IEC 10746. 1998. *"Information Technology — Open Distributed Processing — Reference Model: Architecture."* Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC), ISO/IEC 10746:1998.
- Martin, J.N., 2010. "An Enterprise Systems Engineering Framework." Paper presented at 20th Anniversary International Council on Systems Engineering (INCOSE) International Symposium, 12-15 July, 2010, Chicago, IL, USA.
- Martin, J.N., 2006. "An Enterprise Architecture Process Incorporating Knowledge Modeling Methods. PhD., George Mason University.
- Martin, J.N., 2005. "Using an Enterprise Architecture to Assess the Societal Benefits of Earth Science Research." Paper presented at 15th Annual International Council on Systems Engineering (INCOSE) International Symposium, Rochester, NY, USA.
- Martin, J.N., 2003a. "An Integrated Tool Suite for the NOAA Observing System Architecture." Paper presented at 13th Annual International Council on Systems Engineering (INCOSE) International Symposium, Arlington, VA, USA.
- Martin, J.N., 2003b. "On the Use of Knowledge Modeling Tools and Techniques to Characterize the NOAA Observing System Architecture." Paper presented at 13th Annual International Council on Systems Engineering (INCOSE) International Symposium, Arlington, VA, USA.
- McCaughin, K., and J.K. DeRosa. 2006. "Process in Enterprise Systems Engineering." Paper presented at 16th Annual International Council on Systems Engineering (INCOSE) International Symposium, 9-13 July, 2006, Orlando, FL, USA.
- MOD. 2010. "Ministry of Defence Architecture Framework (MODAF)." version 1.2.004 UK: U.K. Ministry of Defence. <http://www.mod.uk/NR/rdonlyres/04B5FB3F-8BBC-4A39-96D8-AFA05E500E4A/0/20100602MODAFDownload12004.pdf> Accessed September 8, 2011.
- 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." Paper presented at 16th Annual International Council on Systems Engineering (INCOSE) International Symposium, 9-13 July, 2006, Orlando, FL, USA.
- 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. <http://www.opengroup.org/togaf> Accessed on September 2, 2011.
- TRAK. 2011. "TRAK Enterprise Architecture Framework." <http://trak.sourceforge.net/index.html> Accessed September 7, 2011.
- White, B.E. 2006. "Enterprise Opportunity and Risk." Paper presented at 16th Annual International Council on Systems Engineering (INCOSE) International Symposium, 9-13 July, 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): 276-92.

## 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." Paper presented at 20th Anniversary International Council on Systems Engineering (INCOSE) International Symposium, 12-15 July, 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, Danvers, MA, USA. 4 August 2005.

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.

---

[< Previous Article](#) | [Parent Article](#) | [Next Article >](#)

## Comments from SEBoK 0.5 Wiki

No comments were logged for this article in the SEBoK 0.5 wiki. Because of this, it is especially important for reviewers to provide feedback on this article. Please see the discussion prompts below.

## SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

# Enterprise Capability Management

There are three different kinds of capability (glossary): Organizational Capability (glossary), System Capability (glossary), and operational capability (glossary). Management of organizational capability is addressed in the article called Enabling Businesses and Enterprises to Perform Systems Engineering. Management of system capability is addressed by the SE Management activities described in the articles called Systems Engineering Management and Product and Service Life Management. Management of operational capability is described herein.

The enterprise (glossary) has a current and planned (baseline (glossary)) operational capability, based on its past activities and on its current plans for change. The purpose of the enterprise capability management (glossary) 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 Information Technology Infrastructure Library (ITIL) best practices for operations management, starting with strategic operation planning (OGC 2009).

*The Information Technology Infrastructure Library (ITIL) is a set of concepts and practices for Information Technology Services Management (ITSM), Information Technology (IT) development and IT operations. ITIL gives detailed descriptions of a number of important IT practices and provides comprehensive checklists, tasks and procedures that any IT organization can tailor to its needs. ITIL is published in a series of books, each of which covers an IT management topic. ([http://en.wikipedia.org/wiki/Information\\_Technology\\_Infrastructure\\_Library](http://en.wikipedia.org/wiki/Information_Technology_Infrastructure_Library)).*

## Needs Identification & Assessment

The enterprise has key stakeholders that have Operational (glossary) 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 (glossary) activities. End user activities include such things as retail sales, entertainment, food services, and Business (glossary) travel. An example of an operational need is: "Provide transportation services to commuters in the metropolitan area of London."

In addition to operational needs, there are enterprise needs that relate to enabling assets the enterprise has in place that allow the Mission (glossary) 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 (glossary) 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 (glossary) 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 (glossary) EA can be used to Model (glossary) these states and the relative impact each enabling asset has on the desired state changes.

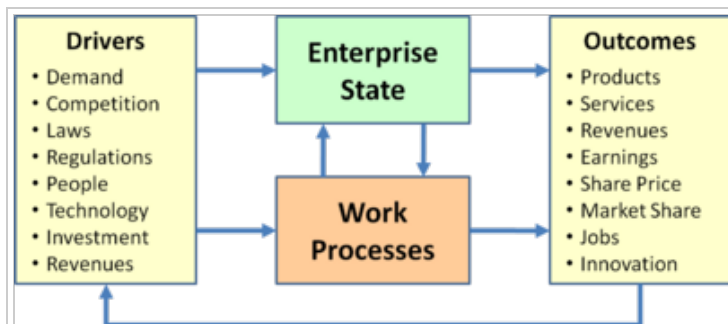


Figure 1. Enterprise State Changes Through Work Process Activities (Rouse 2009) Reprinted with permission of John Wiley & Sons Inc.

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. 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 as to 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 things 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 don’t) (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 15288 2008, architectural design process; ISO/IEC 42010 2007; ISO 15704 2000).

## **Opportunity Identification & Assessment**

The enterprise architecture is used to help identify opportunities for improvement. Usually these opportunities require investment of time, money, facilities, personnel, and so on. There might also be opportunities for “divestment,” which could involve selling of assets, reducing capacity (glossary), 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 (glossary) can be found in (Martin 2005; Martin et al 2004; Martin 2003). References: (Kaplan 2009) on portfolio management, (ISO/IEC 2008) 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 (glossary), 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 (glossary) (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 (glossary) 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 (glossary) 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. These 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. *Industrial Automation Systems — Requirements for Enterprise — Reference Architectures and Methodologies*. Geneva, Switzerland: International Organization for Standardization (ISO), ISO 15704:2000.
- ISO/IEC. 2008. *Systems and Software Engineering — System Life Cycle Processes*. Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC), ISO/IEC 15288:2008 (E).
- ISO/IEC. 2007. *Systems and Software Engineering — Recommended Practice for Architectural Description of Software-Intensive Systems*. Geneva, Switzerland: International Organization for Standards (ISO)/International Electrotechnical Commission (IEC), ISO/IEC 42010:2007.
- Kaplan, J. 2009. *Strategic IT Portfolio Management: Governing Enterprise Transformation*. Waltham, Massachusetts, USA: Pittiglio, Rabin, Todd & McGrath, Inc. (PRTM).
- Martin, J.N. 2005. "Using an Enterprise Architecture to Assess the Societal Benefits of Earth Science Research." Paper presented at 15th Annual International Council on Systems Engineering (INCOSE) International Symposium, Rochester, NY, USA.
- Martin, J.N. 2003. "On the Use of Knowledge Modeling Tools and Techniques to Characterize the NOAA Observing System Architecture." Paper presented at 13th Annual International Council on Systems Engineering (INCOSE) International Symposium, 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." Paper presented at 14th Annual International Council on Systems Engineering (INCOSE) International Symposium, 20-24 June, 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.

ISBN 978-0-11331-050-0.

Rouse, W. B. 2009. "Engineering the Enterprise as a System." In *Handbook of Systems Engineering and Management*., eds. A. P. Sage, W. B. Rouse. 2nd ed. New York, NY, USA: Wiley and Sons, Inc.

Saxena, M. S. 2009. *Capability Management: Monitoring & Improving Capabilities*. New Dehli: Global India Publications Pvt Ltd.

## Primary References

Kaplan, J. 2009. *Strategic IT Portfolio Management: Governing Enterprise Transformation*. Waltham, Massachusetts, 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*., eds. A. P. Sage, W. B. Rouse. 2nd ed. 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*. November 2008.

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." Paper presented at International Society for Pharmaceutical Engineering (ISPE) Conference on Concurrent Engineering (CE), 26-30 July, 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." Paper presented at 16th Annual International Council on Systems Engineering (INCOSE) International Symposium, 9-13 July, 2006, Orlando, FL, USA.

---

[< Previous Article](#) | [Parent Article](#) | [Next Article >](#)

## Comments from SEBoK 0.5 Wiki

No comments were logged for this article in the SEBoK 0.5 wiki. Because of this, it is especially important for reviewers to provide feedback on this article. Please see the discussion prompts below.

## SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

# Systems of Systems (SoS)

System of systems engineering (SoSE), while not a new discipline, is an opportunity for the systems engineering community to define the complex systems of the 21st Century (Jamshidi 2009a). While systems engineering is a fairly established field, SoSE represents a challenge for the present systems engineers at the global level. In general, SoSE requires considerations beyond those usually associated with engineering to include socio-technical and sometimes socio-economic phenomena.

## Topics

The topics contained within this knowledge area include:

- Architecting Approaches for Systems of Systems
- Socio-Technical Features of Systems of Systems
- Capability Engineering

## Definition and characteristics of Systems of Systems

There are many definitions of System(s) of Systems, some of which are dependent on the particularity of an application area. (Maier 1998) postulated five key characteristics of SoS: Operational independence of component systems, Managerial independence of component systems, Geographical distribution, Emergent behavior, and Evolutionary development processes. (Jamshidi 2009) has reviewed more than seven potential definitions of SoS and, although there is not all are universally accepted by the community, the following has received substantial attention:

*A SoS is an integration of a finite number of constituent systems which are independent and operatable, and which are networked together for a period of time to achieve a certain higher goal.*

It should be noted that according to this definition, formation of a SoS is not necessarily a permanent phenomenon, but rather a matter of necessity for integrating and networking them in a central way for specific goal such as robustness, cost, efficiency, etc.

DeLaurentis (DeLaurentis 2005) has added to the five SoS criteria above for SoS Engineering to include: inter-disciplinarity, heterogeneity of the systems involved, and networks of systems.

Not all SoS will exhibit all of the characteristics, but it is generally assumed that a SoS is characterised by exhibiting a majority of the Maier Criteria. Although the individual systems in a SoS are usually considered to have independent operational viability, it is sometimes the case that the SoS must contain some systems the only purpose of which is to enable the interoperation of the other component systems; i.e. the enabling systems cannot operate outside of the SoS.

## Types of SoS

SoS can take different forms. Based on a recognized taxonomy of SoS, there are four types of SoS (Maier 1998; Dahmann and Baldwin 2008). These are:

- **Directed.** Directed SoS are those in which the system-of-systems is created and managed to fulfil specific purposes and the constituent systems are subordinated to the SoS. The component systems maintain an ability to operate independently, but their normal operational mode is subordinated to the central managed purpose.
- **Acknowledged.** Acknowledged SoS have recognized objectives, a designated manager, and resources for the SoS; however, the constituent systems retain their independent ownership, objectives, funding, and development and sustainment approaches. Changes in the systems are based on cooperative agreements between the SoS and the system.
- **Collaborative.** In collaborative SoS the component systems interact more or less voluntarily to fulfil agreed upon central purposes. The central players collectively decide how to provide or deny service, thereby providing some means of enforcing and maintaining standards.
- **Virtual.** Virtual SoS lack a central management authority and a centrally agreed upon purpose for the system-of-systems. Large-scale behavior emerges—and may be desirable—but this type of SoS must rely on relatively invisible mechanisms to maintain it.

This characterization offers a framework for understanding SoS based on the origin of the SoS capability objectives and the relationships among the stakeholders for both the SoS and the systems.

## Emergence

The emergent behavior of a SoS is, put simply, the behaviors that are due either to the internal relationships among the parts of the SoS or as a response to its external environment. The behavior emerges because of interactions and it cannot arise from any of the constituent systems acting alone. Because of the strong element of humans and organizations in SoS, emergent behavior in a SoS can result from changes in the way systems are employed by users now that they are operating in a new and expanded context. Consequences of the emergent behavior may be viewed as negative/harmful, positive/beneficial, or neutral/unimportant by stakeholders of the SoS.

There is much concern about emergent behavior which is unexpected or cannot be predicted by knowledge of the system's constituent parts. For the purposes of a SoS, *unexpected* means unintentional, not purposely or consciously designed-in, not known in advance, or surprising to the developers and users of the SoS. In a SoS context, *not predictable by knowledge of its constituent parts* means the impossibility or impracticability (in time and resources) of subjecting all possible logical threads across the myriad functions, capabilities, and data of the systems to a comprehensive SE process.

## Application domains and the difference between System of System Engineering and Systems Engineering

Application of SoS is broad and expanding into almost all walks of life. Originally addressed in military applications, the defense sector has provided a base for some initial approaches to conceiving and engineering SoS which offer intellectual foundation, technical approaches and practical experience to this field. However, SoS is far from limited to defense. In fact, as you look at the world through a SoS lens it becomes clear that SoS concepts and principles apply across other governmental, civil and commercial domains. These include

- Transportation; e.g. integrated ground transportation, cargo transport, air traffic, highway management, space systems
- Energy; e.g. smart grid, smart houses, integrated production/consumption
- Health Care; e.g. regional facilities management, emergency services, personal health management
- Natural Resource Management, e.g. global environment, regional water resources, forestry, recreational resources
- Disaster Response; forest fires, floods, terrorist attacks
- Consumer Products; integrated entertainment, household product integration
- Media; film, radio, television

Understanding the environment in which a system or SoS will be developed and employed is central to understanding how best to apply SE principles within that environment. Observations regarding differences between individual or constituent systems and SoS are listed in Table 1 (Dahmann and Baldwin 2008; Neaga et. al. 2009). In each case, the degree of difference varies in practice and with complexity of current systems and system development environments - many of the SoS characterizations may apply to systems in certain circumstances.

**Table 1. Differences between Systems and Systems of Systems as They Apply to Systems Engineering (Figure Developed for BKCASE)**

	<b>Systems Engineering</b>	<b>Systems of Systems Engineering</b>
<b>Management and Oversight</b>		
System	Physical engineering	Socio-technical management and engineering
Stakeholder Involvement	Clear set of stakeholders	Multiple levels of stakeholders with mixed and possibly competing interests
Governance	Aligned management and funding	Added levels of complexity due to management and funding for both SoS and systems; SoS does not have control over all constituent systems
<b>Operational Environment</b>		
Operational focus (goals)	Designed and developed to meet common objectives	Called upon to meet new SoS objectives using systems whose objectives may or may not align with the SoS objectives

<b>Implementation</b>		
Acquisition/Development	Aligned to established acquisition and development processes	Cross multiple system lifecycles across asynchronous acquisition and development efforts, involving legacy systems, developmental systems, and technology insertion
Process	Well established	Learning and adaptation
Test and evaluation	Test and evaluation of the system is possible	Testing is more challenging due to systems' asynchronous life cycles and given the complexity of all the parts
<b>Engineering and design considerations</b>		
Boundaries and interfaces	Focuses on boundaries and interfaces	Focus on identifying systems contributing to SoS objectives and enabling flow of data, control and functionality across the SoS while balancing needs of the systems. OR Focus on interactions between systems' Difficult to define system of interest
Performance and Behavior	Performance of the system to meet performance objectives	Performance across the SoS that satisfies SoS use capability needs while balancing needs of the systems
Metrics	Well defined (e.g. INCOSE handbook)	Difficult to define, agree, and quantify

## References

### Works Cited

- Dahmann, J., and K. Baldwin. 2008. Understanding the Current State of US Defense Systems of Systems and the Implications for Systems Engineering. Paper presented at IEEE Systems Conference, 7-10 April, Montreal, Canada.
- DeLaurentis, D., and W. Crossley. "A Taxonomy-Based Perspective for System of Systems Design Methods," Paper 925, IEEE 2005 Conference on Systems, Man, and Cybernetics, Waikoba, HI, Oct. 10-12, 2005.
- Eaga, E.I., M. Hemshaw, 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*, 20th - 23rd April 2009, Loughborough University, 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." Paper presented at IEEE Systems Conference, 7-10 April, Montreal, Canada.
- Jamshidi, M. (ed.) 2009a. *Systems of Systems Engineering – Innovations for the 21st Century*. Hoboken, NJ, USA: Wiley. (ISBN 978-0-470-19590-1)
- Jamshidi, M. (ed.) 2009b. *Systems of Systems Engineering - Principles and Applications*. Boca Raton, FL, USA: CRC Press. (ISBN 978-1-4200-6588-6)
- Maier, M.W. 1998. "Architecting Principles for Systems-of-Systems." *Systems Engineering*. 1(4): 267-284.

### Additional References

Checkland, P.B. 1999. *Systems Thinking, Systems Practice*. Chichester, UK: John Wiley & Sons Ltd.

DeLaurentis, D., and W. Crossley. "A Taxonomy-Based Perspective for System of Systems Design Methods," Paper 925, IEEE

2005 Conference on Systems, Man, and Cybernetics, Waikoba, HI, Oct. 10-12, 2005.

Neaga, E.I., M. Henshaw, and Y. Yue. 2009. The Influence of the Concept of Capability-based Management on the Development of the Systems Engineering Discipline. Presented at 7th Ann. Conf. on Systems Eng. Research, 20th - 23rd April. Loughborough, UK.

Rebovich, Jr., G. 2009. "Chapter 6: Enterprise System of Systems", *Systems of Systems Engineering - Principles and Applications*. Boca Raton, FL, USA: CRC Press, pg. 169.

Ring J. 2002. "Toward an ontology of systems engineering." *INSIGHT*, 5(1): 19-22.

---

< Previous Article | Parent Article | Next Article >

## Comments from SEBoK 0.5 Wiki

## SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

# Architecting Approaches for Systems of Systems

A key part of the SoSE task is to establish a technical framework for composing systems to meet SoS needs, including possible changes in systems functionality, performance or interfaces, and evolving the SoS over time to meet changing SoS objectives. This technical framework is essentially an overlay to systems which comprise the SoS, and is often referred to as the *architecture* for the SoS. Notably, a SoS architecture does not address the details of the individual systems; rather, it defines the way the systems work together to meet user needs and addresses the implementation of individual systems only when their functionality is key to crosscutting issues of the SoS. Developing and evolving the SoS architecture is core to the engineering of a SoS.

## 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 Std 610.12). The architecture of a SoS is a persistent technical framework for governing the evolution of a SoS over time. An architecture for a SoS includes:

- Concept of operations (CONOPs); how the systems will be employed by the users in an operational setting
- Systems, functions, and relationships and dependencies, both internal and external
- End-to-end functionality and data flow as well as communications

The architecture of a SoS is constrained by the structure and content of the individual systems, particularly the extent to which

changes in those systems are affordable and feasible, since systems will typically need to continue to function in other settings in parallel with participation in the SoS. Architecture data on constituent systems can be an important input to SoS architecture development.

Developing a SoS architecture requires analysis and assessments of trades among different options. Architecture analysis may be supported by different assessment approaches. Available architecture frameworks (e.g. Zachman 1987; DoDAF; MoDAF) provide tools to collect and organize information to support SoS architecture development and represent architecture decisions.

The functional architecture is derived from the CONOPs. While for directed SoS, major changes in the systems may be possible, in other types of SoS legacy systems may constrain change.

The functional architecture provides a functional *picture* of the system. It details the complete set of functions to be performed within the SoS as well as the relationships among the functions. The output of the design process is the physical architecture that defines the physical components (constituent systems) of the SoS.

In most cases, due to legacy systems, the migration to an architecture of a SoS will be incremental. Ideally the architecture of a SoS will persist over multiple increments of SoS development, allowing for change in some areas while providing stability in others. This ability to persist and provide a useful framework in light of changes is a core characteristic of a good architecture. Over time, the SoS will face changes from a number of sources (e.g., capability objectives, actual user experience, changing concepts of operations and technology, and unanticipated changes in systems) which may all affect the viability of the architecture and may call for changes. Consequently the SoS systems engineer needs to regularly assess the architecture to ensure that it supports the SoS evolution.

## 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 systems engineer needs to consider the current state and plans of the individual systems as important 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.

First, as noted above, the managerial and operational independence of constituent systems can pose major challenges for the SoS architecture. Because systems are likely to continue to face new functional requirements and the need for technology upgrades independent of the SoS, there is an advantage to a SoS architecture which is loosely coupled — that is, it has limited impact on the individual systems, allowing for changes in functionality and technology in some systems without impact on others or on the SoS objectives.

Second, the independence of the constituent systems also means that these systems are typically not designed to optimize SoS objectives. In the area of trust for example, a system may severely constrain access to services to provide a level of security when a SoS may depend on free exchange of those services. (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.

Third, as introduced in the [Emergence] sub-section, there are risks associated with unexpected or unintended behavior resulting from combining systems with individual complex behavior. These become serious in cases where safety, for example, is threatened through unintended interactions among services provided by multiple constituent systems in a SoS.

Finally, 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 in many cases are very mature (e.g. in sustainment) or currently productively supporting other systems. 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. In parallel, there is 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 achieving synergy between these independent systems to achieve the desired overall system performance (Azarnoush, et. al. 2006).

Modelling and simulation is conducted to analyse architecture effectiveness and to verify architectural features. In the literature, researchers have addressed the issues of coordination and interoperability in a SoS (Abel & 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 tools (Zeigler, et. al. 2000a). In these research efforts, the major focus is given to DEVS architecture with JAVA. In (Mittall, 2000b), DEVS state machine approach is introduced. Finally, DEVS Modeling Language (DEVSML) is developed by using XML based JAVAL 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

The critical challenge of moving from SoS as a concept to the engineering of SoS is the significant technological, human, and organizational differences when considering system of systems engineering and management (Wells and Sage 2008). A potential approach to engineering of SoS can be the *open systems approach* to SoSE (Azani 2009). The open systems principles listed by Azani (Azani 2009) are:

- Open interface principle: open systems have permeable boundaries that allow them to exchange mass, energy, and information with other systems.
- Synergism principle: the co-operative interaction between constituent systems so that their combined effect is greater than the sum of the 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: energy and mass (material) are conserved within the SoS.
- Reconfiguration principle: the SoS reconfigures and adapts 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.
- Modularity principle: 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 (Azani 2009) has elaborated the open systems development strategies and principles utilized by biotic SoS, discussed their implications for engineering of man-made SoS, and introduced an integrated SoS development methodology for engineering and development of adaptable, sustainable, and interoperable SoS based on open systems principles and strategies.

(Hitchens, 2003, 107) has expressed the principles of open systems rather differently, in the context of systems life cycles, as the seven principles of 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 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 should be at a level of detail sufficient to be implementable by independent parties. Compliance with open standards is intended to ensure consistent results (Henshaw, et. al., 2011). These support 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 easy improvement and update of system capabilities by adding or changing components.

## Networks and network analysis

Because networks are such a common component of SoS, they warrant specific attention. In those SoS 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 as discussed across 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 has been explored extensively by the defense community, and the concepts of networks and network analysis extend beyond information sharing (Dickerson and Mavris 2009). For instance in U.S. Navy work on C4ISR as part of a SoS (Owens 1996) the term *network* included organizational aspects of C2 structure as well as communications.

Differences in architecting of enabling network SoS derive from the fact that these SoS are typically built on commercial technologies and architectures, and these 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 hence 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 up-front analyses to explore sensitivities through modeling, simulation, analysis, and/or lab experimentation to identify scalability issues or knees in the curve (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 the 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, 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 such as operating systems, communication hardware, and so on. For this purpose, a SoS needs a common language the SoS’s systems can speak. Integration also implies the control aspects of the SoS because systems need to understand each other in order to take commands or signals from other SoS systems (Cloutier et. al. 2009). 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 environment (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 nine more detailed levels:

- Network Transport
  - Physical interoperability
  - Connectivity and network interoperability
- Information Services
  - Data/object model interoperability
  - Sematic/information interoperability
  - Knowledge/awareness of actions interoperability
- People, Processes and Applications
  - Aligned procedures
  - Aligned operations
  - Harmonized strategy/doctrine
  - Political or business objectives

This spectrum of interoperability levels requires appropriate coherence at each level 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 context:

- Political context
  - Legal interoperability
  - Organizational interoperability

- Semantic interoperability
- Technical interoperability

The interoperability between the systems of a SoS is a fundamental design consideration for SoS that may be managed through the application of standards.

## Standards

There are currently no standards specifically written for SoS or SoSE products, processes, management, or other endeavours (Johnson 2009); however, there are many standards which address aspects of interoperability relevant to SoS. The term, *standard*, has a both general and specific meaning within the realm of contemporary standards. In general, a standard represents a document produced by a standards body, organization, or entity. A specific standard represents either a part of a document or the whole document that is required, recommended, or permitted to be used as practices, processes, formats, contents, etc.

Much of the current work in SoS is being done in engineering and acquisition; within engineering the concept of a universally agreed-upon set of guidelines for interoperability is important. Such guidelines provide four levels of standardization: compatibility, interchangeability, commonality, and reference (Johnson 2009), which are relevant to the SoS environment through the creation of compatibility, similarity, measurement, and symbol and ontological standardization. As the various disciplines relevant to SoS mature, standards will be required to ensure these four levels of the SoS Standardization are met (Jamshidi 2009b).

Future standards for SoS will most likely mimic current systems oriented standards by incorporating extensions for SoS perspectives and needs. An important need for systems is adaptability, and this will be even more important for SoS due to the uncertainties, technologies, systems, stakeholders, organizations, and other entities that may be a part of future evolution of the SoS. The other key area of standardization is interoperability and one effort to provide a semantic and syntactic interoperability standard is the development by US DoD C4ISR organization's *Levels of Information System Interoperability* (DoD 1998).

Standard approaches to modeling and simulating systems, processes, interactions, and other SoS activities are an important area in which standards for SoS may evolve. Model-based standards (ref) may also be a significant area of future development in this topic.

Overall, open standards are viewed as an effective way to reduce the risks associated with lack of interoperability in SoS. An open standard is standard that is consensus-based amongst a community of interest, and is published by and freely available within that community of interest (Henshaw et. al 2011). This has been emphasized in the software domain, for instance (Hall 2007) as a strategy for DoD to adopt open IT standards and to influence these appropriately through participation in relevant standards developing organizations and/or standards setting organizations in the area of information and communications technologies.

## References

### Works Cited

- Abel, A. and S. Sukkarieh. 2006. "The Coordination of Multiple Autonomous Systems using Information Theoretic Political Science Voting Models." In the Proceedings of the IEEE International Conference on System of Systems Engineering, 24-26 April, Los Angeles.
- 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.
- Azamouh, H., B. Horan, P. Sridhar, A.M. Madni, and M. Jamshidi. 2006. "Towards optimization of a real-world robotic-sensor system of systems". In the Proceedings of the World Automation Congress (WAC), July 24-26. Budapest, Hungary.
- Cloutier, R.M., J. DiMario, and H.W. Polzer. 2009. "Net-Centricity and System of Systems". Chapter in (Jamshidi 2009a)
- 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. Assistant Deputy Under Secretary of

Defense (LMR/LPS), and DoD Standards Executive, in *DSPO Journal* pg. 4-7, Mar/Jan.  
[http://www.dsp.dla.mil/app\\_util/content/newsletters/journal/DSPJ-01-07.pdf](http://www.dsp.dla.mil/app_util/content/newsletters/journal/DSPJ-01-07.pdf)

Henshaw, M.J.D. 2011. Assessment of Open Architectures within Defence Procurement. **(need complete ref)**

Hitchins, D.K. 2003. *Advanced Systems Thinking, Engineering and Management*. Norwood, MA, USA: ARTECH HOUSE, INC. (ISBN 1-58053-691-0).

IEEE. 1990. Standard Glossary of Software Engineering Terminology, Std 610.12-1990. Washington, DC: Institute of Electrical & Electronics Engineers (IEEE). (ISBN 1-55937-067-X).

Jamshidi, M. (ed.) 2009. *Systems of Systems Engineering - Principles and Applications*. Boca Raton, FL, USA: CRC Press. (ISBN 978-1-4200-6588-6)

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." In the Proceedings of the IEEE International Conference on System of Systems Engineering, 24-26 April, Los Angeles, CA, US.

Mittal, S. 2000. "DEVS Unified Process for Integrated Development and Testing of Service Oriented Architectures." Ph. D. Dissertation, Univ. Arizona.

NCOIC. 2008. "NCOIC Interoperability Framework (NIF(R))". Available at: <https://www.ncoic.org/technology/deliverables/nif/>

Owens, W.A. 1996. "The Emerging U.S. System-of-Systems." In The National Defense University, Institute of National Security Studies, Number 63, Washington D.C. February.

Rebovich, Jr., G. 2009. *Chapter 6: Enterprise System of Systems. Systems of Systems Engineering - Principles and Applications*. Boca Raton, FL, USA: CRC Press, pg. 169.

Sahin, F., M. Jamshidi, and P. Sridhar. 2007. "A Discrete Event XML based Simulation Framework for System of Systems Architectures." In the Proceedings of the IEEE International Conference on System of Systems, 16-18 April 2007, San Antonio, TX, US.

Wells, G.D. and A.P. Sage. 2008. *Engineering of a System of Systems. 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." In the Proceedings of the IEEE International Conference on System of Systems Engineering, 24-26 April 2006, Los Angeles, CA, US.

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.* 9;59(7):647-659.

- The paper defines and clarifies basic concepts of enterprise architectures. Then an overview on architectures for enterprise integration developed since the middle of the 1980s is presented. The main part of the paper focuses on the recent developments on architectures for enterprise interoperability. The main initiatives and existing works are presented. Future trends and some research issues are discussed and conclusions are given at the end of the paper.

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

- This is the paper in which the five characteristics of, or criteria for, Systems of Systems are defined. It is the most frequently cited paper for SoS definition.

## 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: Office of the Assistant Secretary of The Navy (Research Development And Acquisition). ADA427961. Available at: <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, *Towards interoperability for European public services*. Available at: [http://ec.europa.eu/isa/strategy/doc/annex\\_ii\\_eif\\_en.pdf](http://ec.europa.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,
- Mittal, S. 2000a. "Extending DoDAF to Allow DEVS-Based Modeling and Simulation." *Journal of Defense Modeling and Simulation (JDMS)*. 3(2).
- Zeigler, B.P, D. Fulton, P. Hammonds, and J. NutaroJ. 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 >](#)

## Comments from SEBoK 0.5 Wiki

## SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

# Socio-Technical Features of Systems of Systems

## Human and Organizational Nature of SoS

System of Systems (SoS) (glossary) (SoS) contain many types of systems among which are, what are often termed, Enterprise (glossary) Systems (Chen et. al. 2008). There are many different definitions of enterprise: within a SoS Environment (glossary) an enterprise system could be described as a Complex (glossary), socio-technical system that comprises interdependent resources of people, information, and technology that must interact with each other and their environment in support of a common Mission (glossary) (See also Enterprise Systems Engineering. Emerging ‘soft’ issues critical to the Design (glossary) and operation of Systems of Systems can be identified as follows (see Hubbard et. al. 2010),

- Decision making in SoS, including issues in autonomy, authority, responsibility and ethics
- Measures of Enterprise SoS performance

- Impact of culture and cultural attributes on multinational and multicultural team performance
- System of Systems ethics, governance, and regulation
- System of Systems experimentation
- Shared/distributed situational awareness
- Alternative approaches to training, e.g., virtual reality, gaming
- SoS lead and lag 'soft' metrics, e.g., improved mental and physical workload measurement techniques
- Enterprise System agility and resilience, e.g., dynamic allocation and reallocation of function, the human in the loop
- Enterprise SoS leadership and motivational issues

The holy grail of being able to look into the future by evaluating the Effectiveness (glossary), impact or added value of alternative enterprise system configurations, prior to deployment, is still a long way off. Such a Capability (glossary) would greatly enhance an enterprise's ability to dynamically (re-)configure appropriate systems (people, Process (glossary), and technology) to achieve the performance required to produce designated capability in different contexts and to avoid enterprise structures that are susceptible to undesirable emergent behaviour (See also Emergence) including adverse circumstances such as accidents, disasters. Enterprise System models not only provide the means to visualize, represent, and analyse the inner workings of an Enterprise SoS, but can also constitute the building blocks of an Enterprise SoS Architecture (EA).

An EA is an Architecture (glossary) of an organization that supports strategy, analysis, and planning by stakeholders to determine how the organization can most effectively achieve its current and future objectives. (1) An EA Framework provides a methodology to describe how an EA is organized, structured, and operates in terms of people, processes, Product (glossary), Information Technology (glossary) (IT) and resources in order to achieve its goal. (1A) 1 + 1A = tiger team document (replace with appropriate description)

Existing models and enterprise system architectures and Frameworks (e.g. Zachman, CIMOSA, GERAM, VERAM, ToVE , PERA DoDAF, MODAF) tend to deal with enterprise elements such as Resources, Information Flows and Functions well, but a) within a process framework and b) they do not show a sufficient capability to include *soft* enterprise characteristics such as policies, culture, competencies, decision making structures, etc. within dynamic models. Hence, changes in one or more of these characteristics are not shown in overall organizational system performance. The following points can be made with reference to EAs:

- Architecture is foundational for managing modern enterprises and planning enterprise integration.
- An EA framework is an organized collection of ingredients (tools, methodologies, modeling languages, models, etc.) necessary to architect or re-architect whole or part of an enterprise.
- For a given enterprise, the enterprise architecture describes the relationships among the mission assigned to the enterprise, the work the enterprise does, the information the enterprise uses, and the physical means, human labor, and IT that the enterprise needs.

The prime advantage of an EA is to provide a common view (in the form of models) of what is going on in the enterprise to relevant actors or stakeholders of the enterprise. The second decisive advantage of an EA is that it provides a sound basis for the management of change that occurs throughout the Life Cycle (glossary) of the enterprise. Vernadat (1996) combines the two methodologies of enterprise modeling and enterprise integration and advocates a systematic engineering approach called Enterprise Engineering, for modeling, analysing, designing and implementing integrated enterprise systems.

Enterprise modelling (EM) is concerned with the representation and specification of the various aspects of enterprise operations; namely, functional aspects to describe what are the things to be done and in which order; informational aspects to describe which objects are used or processed; resource aspects to describe what or who performs things and according to which policy; and organizational aspects to describe the organizational structure and the responsibility frame within which things are being done. These Enterprise System models can be combined within an EA framework to provide a dynamic overview of the enterprise system.

Although there are several models available to assess the structure and performance of organizations (e.g. Castka 2001; Curtis et. al. 2001; Tannenbaum et. al. 1996), few if any of these models provide quantitative and qualitative measures of performance and none are truly able to provide a direct multi-point, measurable cause and effect link between the various *soft* attributes of an enterprise system and its performance. It is clear, though, that success factors from a human perspective do center upon the structure of communication (stakeholder management) and decision making processes and systems within the overall System of Systems

## The SoS mindset

SoS problems often exhibit many of the characteristics of so-called *wicked problem* (Rittel and Webber 1973):

- Problems are extremely complex and not bounded or stable
- They do not have unique, right solutions, but rather solutions that are either better or worse than others, and they do not have a definitive formulation
- SoS requirements are often volatile with changing constraints and moving targets
- Stakeholders have different views, and
- Understanding the whole context is difficult and critical.

SoS problems relate to both hard (mechanical, electronic, and Software (glossary)) and soft (people, organizations, and regulatory) systems considerations. Research must nowadays include mixed methods and approaches (Conklin 2005) that include both quantitative and qualitative techniques, making this a very challenging area intellectually.

## References

### Works Cited

- Castka, P. B. 2001. "Factors Affecting the Successful Implementation of High Performance Teams." *Team Performance Management* 7 (7/8), 123-134.
- Curtis, B., W. E. Hefley, and S. A. Miller. 2009. "People Capability Maturity Model (P-CMM)" Version 2.0, 2nd Ed. Software Engineering Institute. Carnegie Mellon University. <http://repository.cmu.edu/cgi/viewcontent.cgi?article=1048&context=sei>
- Conklin, J. 2005. "Dialogue Mapping: Building Shared Understanding of Wicked Problems." Chichester, West Sussex, England, UK: John Wiley & Sons, Ltd. 1st ed. ISBN 978-0-47001-768-5
- Hubbard, E-M., C. E. Siemieniuch, and M. A. Sinclair. Hodgson A. 2010. "Working towards a Holisticorganisational Systems Model." Presented at 5th Int. Conf. Systems of Systems Engineering (SoSE), Loughborough, UK. 22-24 June.
- Rittel, H. W. J. and M. M. Webber 1973. "Dilemmas in a General Theory of Planning." *Policy Sciences* 4 Amsterdam, The Netherlands: Elsevier Scientific Publishing Company, Inc.: 155–169. In Cross, N. 1984. Ed. "*Developments in Design Methodology*." Chichester, West Sussex, England, UK: John Wiley & Sons, Ltd. pp. 135–144
- Tannenbaum, S. I., E. Salas, and J. A. Cannon-Bowers 1996. "Promoting Team Effectiveness." In West, M. A. "*Handbook of Work Group Psychology*." Chichester, West Sussex, England, UK: John Wiley & Sons, Ltd.
- Vernadat, F. B. 1996. "*Enterprise Modeling and Integration: Principles and Applications*." London, England, UK: Chapman and Hall Publishers.

### Primary References

- Checkland, P. B. 1981. *Systems Thinking, Systems Practice*. Chichester, West Sussex, England, UK: John Wiley & Sons, Ltd.
- Hubbard, E-M., C. E. Siemieniuch, and M. A. Sinclair. Hodgson A. 2010. "Working towards a Holistic Organisational Systems Model." Presented at 5th Int. Conf. Systems of Systems Engineering (SoSE), Loughborough, UK. 22-24 June.
- Rittel, H. W. J. and M. M. Webber 1973. "Dilemmas in a General Theory of Planning." *Policy Sciences* 4 Amsterdam, The Netherlands: Elsevier Scientific Publishing Company, Inc.: 155–169. In Cross, N. 1984. Ed. "*Developments in Design Methodology*." Chichester, West Sussex, England, UK: John Wiley & Sons, Ltd. pp. 135–144

### 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. <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. <http://www.cit.gu.edu.au/~bernus/taskforce/geram/versions/geram1-6-3/v1.6.3.html>.

ISO 14258:1998. "*Industrial automation systems — Concepts and rules for enterprise models*." Geneva, Switzerland: International Organization for Standardization;

ISO 19439:2006. "*Enterprise integration — Framework for enterprise modelling.*" Geneva, Switzerland: International Organization for Standardization.

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. ISBN 978-6-13-025337-0

---

[< Previous Article](#) | [Parent Article](#) | [Next Article >](#)

## Comments from SEBoK 0.5 Wiki

No comments were logged for this article in the SEBoK 0.5 wiki. Because of this, it is especially important for reviewers to provide feedback on this article. Please see the discussion prompts below.

## SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

# Capability Engineering

## Capability Engineering Perspectives

The term Capability (glossary) 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 (glossary) and management, Through 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 different communities. Nevertheless, there are some common statements that one can make about capability engineering and its relationship to Systems Engineering (glossary).

Capability is the ability to do something; it is not a synonym for a system Function (glossary) or system purpose. A corollary of the statement above is that capability engineering is significantly different from Product (glossary) systems engineering and broader than (though it incorporates) the Process (glossary) perspective of systems engineering.

Capability engineering is very similar in scope to views of systems engineering such as Ring's Value Cycle (Ring 2002) and layers 1-4 of Hitchins' Five Layer Model (Hitchins 1994). Indeed, Capability engineering is the overarching approach that links Value (glossary), purpose, and solution of a systems problem. As such, capability engineering comprises mindset (Holistic (glossary) thinking, assumptions), trade-offs, Design (glossary), processes, values and policy, and outcomes. The processes for capability engineering are similar to traditional systems engineering, but the mindset and system Boundary (glossary) are different.

Capability is realised through a combination of components that include hard (equipment) and soft (people and processes) systems together with supporting services and infrastructure.

The relationship of capability to System of Systems (SoS) (glossary) can be colloquially summarised as *Capability is an emergent property of system of systems*.

## Services view of SoSE

As has been discussed throughout the Systems of Systems (SoS) knowledge area, 'system of systems' is typically viewed 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 an SoS is information and the SoS brings together multiple information systems to support the information needs of a broader community. These 'Information Technology (glossary) (IT) based' SoSs have the same set of characteristics of other SoSs and face many of the same challenges. Current IT has adopted a 'services' view of this type of SoS, and increasingly apply a 'service oriented Architecture (glossary)' approach to design and management (Erl 2008). This approach decouples the architectural approach from the implementation, defining the functions to be provided across the SoS in terms of 'services'. This approach allows for evolution of the underlying technical approaches in the supporting systems while maintaining continuity in supporting the needs of users across the SoS. While most commonly applied to IT based SoS, this services conceptualization of SoS can be applied to SoS of all types.

## References

### Works Cited

Erl, T. 2008. "SOA Principles of Service Design." Boston, MA, USA: Prentice Hall Pearson Education. ISBN 978-0-13234-482-1.

### 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, 20-23 June 2011. Denver, US.

- Output of the INCOSE UK Capability Working Group includes an in-depth analysis of capability engineering and its relationship to systems engineering.

### Additional References

No additional references have been identified for version 0.75. Please provide any recommendations on additional references in your review.

---

[< Previous Article](#) | [Parent Article](#) | [Next Article \(Part 5\)>](#)

## Comments from SEBoK 0.5 Wiki

## SEBoK 0.75 Reviews

Please provide your comments and feedback on SEBoK 0.75 below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account (click "DISQUS" button). Please look for the posts by "BKCASE" for specific issues on which the author team would like feedback.

blog comments powered by **DISQUS**

Retrieved from "[http://www.sebokwiki.org/075/index.php?title=Part\\_4&oldid=33061](http://www.sebokwiki.org/075/index.php?title=Part_4&oldid=33061)"