

# Service Systems Engineering Stages

From SEBoK

Service Systems Engineering Stages

---

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

---

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

---

## Contents

- 1 Service Strategy/Concept
- 2 Requirements Analysis and Engineering
- 3 Systems Design/Development
- 4 Service Integration, Verification & Validation
- 5 Service Transition/Deployment
- 6 Service Operations/Continuous Service Improvement (CSI)
- 7 Service Systems Engineering Tools & Technologies
- 8 References
  - 8.1 Works Cited
  - 8.2 Primary References
  - 8.3 Additional References

## Service Strategy/Concept

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

A high-level feasibility assessment of the concept is then carried out by the integrated service development team (ISDT) to assess the needs/impacts on enterprise process capabilities, operational

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

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

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

## **Requirements Analysis and Engineering**

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

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

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

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

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

# Systems Design/Development

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

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

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

# Service Integration, Verification & Validation

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

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

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

# Service Transition/Deployment

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

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

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

## Service Operations/Continuous Service Improvement (CSI)

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

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

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

## Service Systems Engineering Tools & Technologies

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

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

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

**Service design process:** Architecture frameworks (AF) and enterprise architectures (EAs) are standards that help split complex systems (see also Complexity) into an interrelated, structured form. They describe the different characteristics of the products and services. Systems engineering modeling tools, such as the unified modeling language (UML) (OMG 2010a) and system modeling language (SysML) (OMG 2010b), help develop the AF and EA and greatly impact the continued

evolution and successful implementation of complex projects. Service oriented architecture (SOA) and systems and software engineering architecture (ISO/IEC/IEEE 2011) are standards that apply architecture principles for specialized applications. Successful implementation of the architecture tools helps identify critical interfaces and improves understanding of the allocations between components and functions.

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

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

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

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

## References

### Works Cited

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

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

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

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

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

- Friedenthal, S. 1998. "Object Oriented System Engineering: Process Integration for 2000 and beyond." Presented at System Engineering & Software Symposium, Lockheed Martin Corporation, 1998, New Orleans, LA.
- Hantry, F., M.P. Papazoglou, W. van den Heuvel, R. Haque, E. Whelan, N. Carroll, D. Karastoyanova, F. Leymann, C. Nikolaou, W. Lamersdorf, and M. Hacid. 2010. "Business Process Management," in *Service Research Challenges and Solutions for the Future Internet S-Cube - Towards Engineering, Managing and Adapting Service-Based Systems*, edited by M. Papazoglou, K. Pohl, M. Parkin, and A. Metzger. Berlin and Heidelberg, Germany: Springer-Verlag. p. 27-54.
- Hybertson, D.W. 2009. *Model-Oriented Systems Engineering Science: A Unifying Framework for Traditional and Complex Systems*. Boston, MA, USA: Auerbach Publications.
- IEEE. 1998. IEEE 1220-1998, *IEEE Standard for Application and Management of the Systems Engineering Process*. Washington, DC, USA: Institute of Electrical and Electronics Engineers.
- ISO/IEC. 1998. ISO/IEC 14598-5:1998, *Information technology — Part 5: Process for evaluators*. Geneva, Switzerland: International Organization for Standardization / International Electrotechnical Commission.
- ISO/IEC. 2003a. ISO/IEC TR 9126-2:2003, *Software engineering — Product quality — Part 2: External metrics*. Geneva, Switzerland: International Organization for Standardization / International Electrotechnical Commission.
- ISO/IEC. 2003b. ISO/IEC TR 9126-3:2003, *Software engineering — Product quality — Part 3: Internal metrics*. Geneva, Switzerland: International Organization for Standardization / International Electrotechnical Commission.
- ISO/IEC. 2004. ISO/IEC TR 9126-4:2004, *Software engineering — Product quality — Part 4: Quality in use metrics*. Geneva, Switzerland: International Organization for Standardization / International Electrotechnical Commission.
- ISO/IEC. 2005. ISO/IEC 27001:2005, *Information technology — Security techniques — Information security management systems — Requirements*. Geneva, Switzerland: International Organization for Standardization / International Electrotechnical Commission.
- ISO/IEC. 2008a. ISO/IEC 15408-2:2008, *Information technology — Security techniques — Evaluation criteria for IT security — Part 2: Security functional components*. Geneva, Switzerland: International Organization for Standardization / International Electrotechnical Commission.
- ISO/IEC. 2008b. ISO/IEC 15408-3:2008, *Information technology — Security techniques — Evaluation criteria for IT security — Part 3: Security assurance components*. Geneva, Switzerland: International Organization for Standardization / International Electrotechnical Commission.
- ISO/IEC. 2009. ISO/IEC 15408-1:2009, *Information technology — Security techniques — Evaluation criteria for IT security — Part 1: Introduction and general model*. Geneva, Switzerland: International Organization for Standardization / International Electrotechnical Commission.
- ISO/IEC/IEEE. 2011. ISO/IEC/IEEE 42010:2011, *Systems and software engineering — Architecture description*. Geneva, Switzerland: International Organization for Standardization / International Electrotechnical Commission.
- Lefever, B. 2005. "SeSCE Methodology." Rome, Italy: SeCSE Service Centric Systems Engineering. SeCSE511680. Available: [http://www.secse-project.eu/wp-content/uploads/2007/08/a5\\_d4-secse-methodology-v1\\_3.pdf](http://www.secse-project.eu/wp-content/uploads/2007/08/a5_d4-secse-methodology-v1_3.pdf).
- Lin, F., and P. Hsieh. 2011. "A SAT View on New Service Development." *Service Science*. 3 (2): 141-157.

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

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

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

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

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

## Primary References

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

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

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

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

## Additional References

None.

---

< Previous Article | Parent Article | Next Article >

**SEBoK v. 2.4, released 19 May 2021**

Retrieved from

"[https://www.sebokwiki.org/w/index.php?title=Service\\_Systems\\_Engineering\\_Stages&oldid=61470](https://www.sebokwiki.org/w/index.php?title=Service_Systems_Engineering_Stages&oldid=61470)"

- 
- This page was last edited on 18 May 2021, at 19:48.

