Volume 1, Issue 5 November 2007

Continuity of Operations Leadership Series for Government

Integrating Continuity of Operations (COOP) into the Enterprise Architecture

Systems Pillar

Table of Contents

Table of Figures

Systems

Systems are the mechanisms by which organizational operations occur. Typically, information technologies are transformed into information systems such that those systems are aligned with operational activities of the organization. From a Continuity of Operations (COOP) perspective, it is important to develop systems that support anticipated COOP environments or can be realigned with new, and often unexpected, COOP operational requirements. To do this effectively, organizations must develop operational COOP requirements and inject them into the systems development process. The process, if effective, will generate systems that can support a variety of COOP situations.

By definition, a system is a set of interrelated entities in which each entity is related to or interacts with at least one other entity of the system. Therefore, a system implies interaction between its components. Man-made systems are created with a particular purpose or function in mind. Typically, systems are composed of physical machines such as mechanical, electronic, electromechanical, chemical and software elements. However, systems can also be non-physical, such as processes, procedures, methods and algorithms. Regardless of what a system consists of, it is typically designed to work as a coherent entity.

System design often depends on the environment. Most, if not all, systems are open because they must interact with the environment in which they exist. Whatever is not part of the system may be part of the system's environment. Potential COOP situations represent a new environment in which an organization must be prepared to operate.

A system is a set of smaller elements referred to as subsystems. A subsystem can be considered a system, unto itself, that interacts with other subsystems to define the total system. This is an important aspect of systems because it implies that systems are often defined by arbitrary boundaries and that these "system" boundaries are dependent upon the observer or system designer.

Systems typically have several common characteristics, including:

- A structure with a defined interior, exterior and boundary
- Inputs and outputs
- A structure and function made up of its component parts, internal interfaces and processes

The Nature of Systems

The nature of systems is often confusing because they consist of more than just the physical or functional elements of which they are made. While this may be difficult to understand, it is conceptually true and can be demonstrated with very simple examples. For instance, an automobile sitting in a garage is a system that has a defined purpose and function, but if the automobile is disassembled into a thousand pieces, it is still an automobile, but it is no longer a system. These two automobiles, when compared, will have the same mass and the same components, but one will have functionality and the other one will not. The difference between the two is that the assembled automobile has active interfaces that provide functionality not available to the automobile that is in pieces. The implication is that a system, to be valid, must have all of its constituent components intact and operating. Figure 1 shows the interactions or interfaces between the subsystems of a system. A system is not merely the sum of its parts, but a function of the sum of its parts.

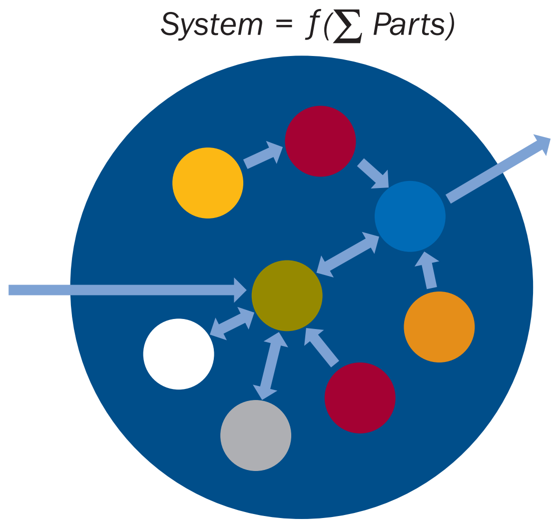

Figure 1: The Nature of Systems

All systems are subsystems and all subsystems are systems. Systems have an environment with which they interact and, therefore, must be considered subsystems of a larger system, whether it is intended or not. An automobile accepts air from the environment and releases other types of gases back to the environment. Therefore, it is part of the larger natural system. Similarly, an automobile requires fuel, roads and a maintenance system to make it useful. Therefore, an automobile is part of a larger transportation system. Along those same lines, an automobile has subsystems, such as an engine, that have inputs and outputs. An engine can be viewed as a system that fits into an environment such as an automobile.

A system that is considered from all aspects can be very complex. To properly contemplate systems from a creation and implementation perspective, they must be considered from the perspective of how they will exist and how they will be used. With this in mind, systems can be considered larger than their hardware and software components. Systems can be viewed from a high-order perspective to be made up of people, processes and products, where the products are the hardware and software. As shown in Figure 2, the elements of a system can be grouped into these three categories. Each of the system elements can then be further decomposed into the primary system functions. These functions are the actions that will make a system come into existence and provide value to the people or customers for whom it was developed.

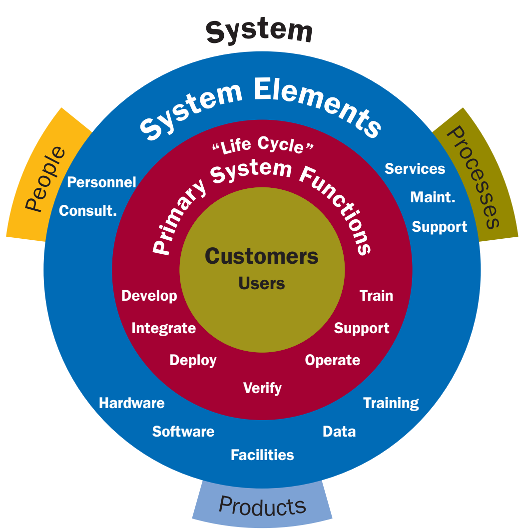

Figure 2: The System View

Why is this system model important to COOP and COOP planning? The model serves as a guide to ensure that COOP planners consider all aspects of the systems they must contemplate for successful COOP operations. A system that exists and functions well during normal operations may not function as well or may not function at all during COOP. Planners should carefully evaluate existing systems and upgrade existing systems or design new systems to operate in a COOP environment.

System of Systems

As stated earlier, systems can be decomposed into subsystems. These subsystems can be viewed as systems in their own right. Conversely, systems can be recognized as part of a larger supersystem. Using the previous example of automobiles, every automobile is a system, but each is also part of a supersystem made up of roads, traffic control, fuel, auto fabrication, auto repair and so on. As such, the automobile system becomes an integral part of this larger supersystem. In fact, automobiles are designed specifically to fit within this larger system. Otherwise, they could not be driven or maintained. From the larger supersystem's point of view, the transportation system is a "system of systems."

A system of systems usually implies a large-scale, interdisciplinary effort involving multiple distributed systems. An enterprise that has multiple core functions and communications domains will normally be concerned with a system of systems types of issues. As such, COOP activities must account for a system of systems when particular systems are unavailable.

Information Systems

There are many different types of information systems in an organization. From an enterprise point of view, these systems are often integrated to form a system of systems. Organizations, especially bureaucratic ones, are typically hierarchical in nature, and each level of the hierarchy may have specialized information systems that contribute to its level of functionality. As examples, the knowledge level of an organization may have Computer-Aided Design (CAD) applications, while the management level may have Management Information Systems (MIS) and Decision Support Systems (DSS). It is very common for different organizational divisions or functional units to have their own information systems, such as finance and accounting systems or human resource management systems

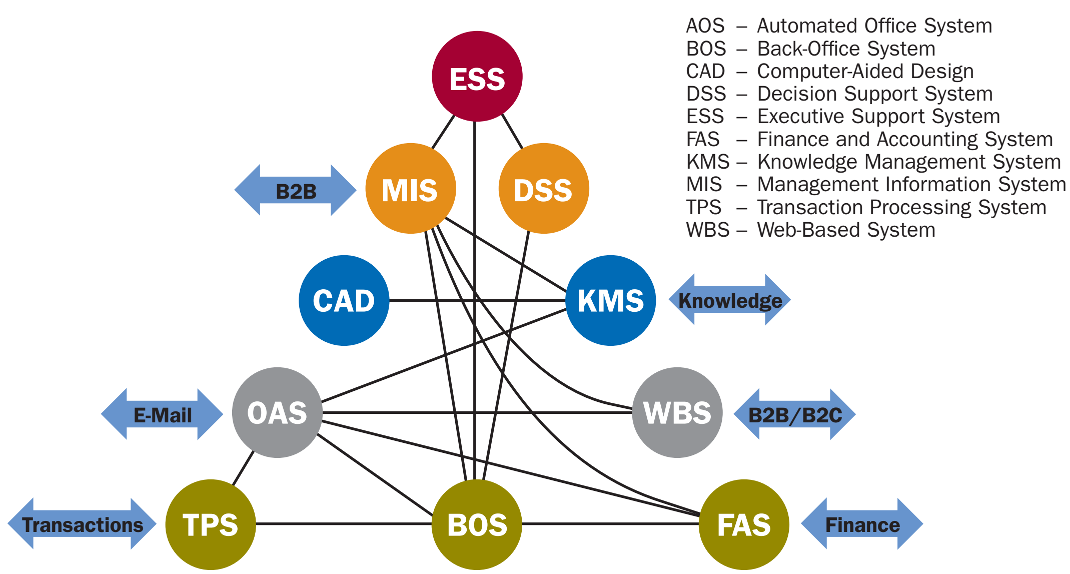

Figure 3: Information Systems

Regardless of the types and variability of information systems in an enterprise, there is usually a need to integrate many of the systems so that they can share data and information. Typically, lower-order information systems deal with data from transactions and pass it up to higher-level systems to be regressed, analyzed and converted into actionable information. The higher level the information is processed, the more aggregated it may become.

The COOP issue with information systems is that there needs to be a full understanding of required data and information and the relationships that must continue to exist once a COOP situation arises. It will be necessary to create data and information flow models for the organization's essential functions and their supporting processes. These flow models will contribute to an understanding of the data and information dependencies that will exist in various types of circumstances and will help to identify what types of data and information need to be stored, replicated, backed up and secured.

System Life Cycle

The system life cycle is an examination of a system or proposed system that addresses all phases of its existence to include conceptualization, design and development, production and/or construction, deployment, operation, maintenance and support, retirement, phase-out and disposal. In essence, the system life cycle is the path of the system from conception to termination.

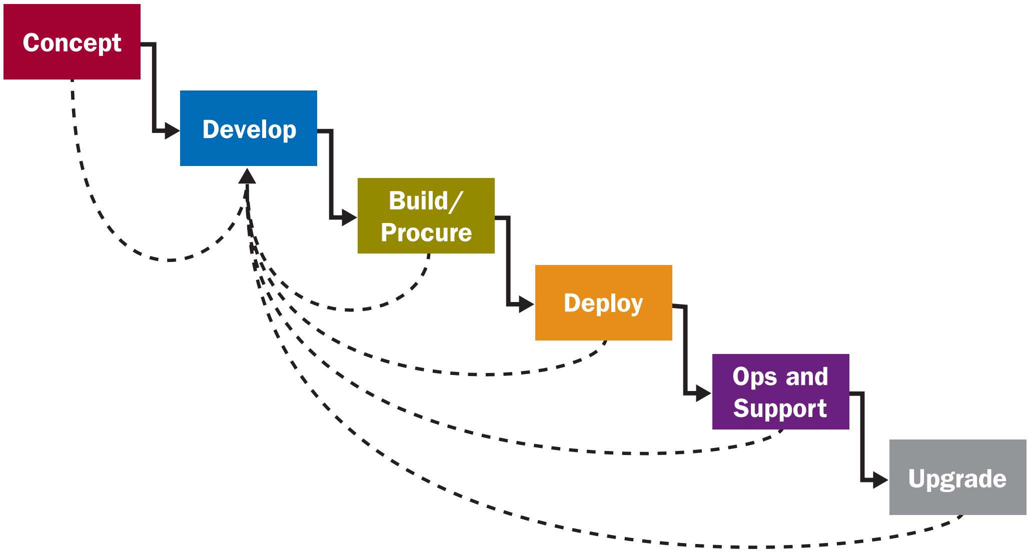

Figure 4: System Life Cycle

A critical aspect of the system life cycle is that the system development phase of the life cycle is the phase of the life cycle where all future life cycle phases are planned and designed. For instance, as an automobile is designed for a particular model year, it is designed and codified as a "build-to" specification. This build-to specification will define the form, fit and function of every component and subsystem of the automobile. Likewise, the designers will, if necessary, design new tools for fabricating components and will design the building and assembly process. Additionally, designers will need to identify the types and numbers of spare parts that will be needed and will calculate the automobile's reliability and maintainability parameters so that warrantees can be developed. All of these development issues will be initially designed and determined during the development stage.

Information system development requires the same rigorous approach as other complex systems. The development of information systems follows some form of conceptualization phase. The development process requires a complete understanding of how the system will be built, configured, deployed, operated, supported and eventually replaced.

Where does COOP fall within the system life cycle? Obviously, COOP needs to be an element of the operations and support phase. To ensure that COOP is fully integrated into the system life cycle and that it is accounted for from an operational perspective, COOP requirements need to be injected into the development process. For systems that already exist, COOP requirements need to become part of a systems engineering process that is geared to a system upgrade. If it is not possible to upgrade the system to account for COOP requirements, it is likely the system will need to be replaced or augmented in another way, perhaps with another system.

Conceptualization

Conceptualization or the conceptual design phase is a stage of the life cycle where needs are identified and examined. Based on what is understood at this early stage, there will be alternative, potential solutions that are evaluated and validated. The objective of this phase is to identify what potential solutions make sense and discard potential solutions that are obviously invalid. Additionally, conceptual designers will identify potential users and stakeholders and attempt to elicit requirements from them. These requirements may be related to the system's entire life cycle.

Once the requirements are understood and the potential candidate solutions identified, the conceptual phase will culminate with a full description of what needs to be developed and designed. Normally, a specification will be developed known as the "design-to" specification.

The major steps of the conceptual stage are:

- Identify and validate the need
- Identify the customers and stakeholders
- Determine operational scenarios
- Identify viable, alternative solutions
- Analyze the feasibility of alternatives
- Perform requirements analysis
- Develop design-to specifications
- Review the conceptual design
  - Conduct Conceptual Design Review
- Decide to progress to development stage

Development

The development phase of the system life cycle is often divided into two parts. The first part is known as preliminary design and the second part is known as detailed design and development. The development phase, as a whole, is actually much more than the design of a solution. Development is related to the definition of a complete system solution that includes the details of the remaining life cycle elements.

Preliminary Design

Preliminary design attempts to focus on a functional architecture of the system and often ignores post-development issues. It is normally a continuation and refinement of the conceptual phase.

In many cases, subsystems can be identified that are capable of performing system functions. These subsystems will be tested to ensure that they meet system specification. Interfaces between subsystems can also begin to be defined. The result of the preliminary design effort is a refined design and development specification with enough detail to reduce development risk. There are, however, many alternative technical solutions that need to be further analyzed and tested.

The major steps of the preliminary substage are:

- Analyze and decompose the system functions
- Allocate requirements to functions and subfunctions
- Perform trade-off studies to determine best alternatives
- Synthesize alternative into solutions
- Develop engineering models
- Refine design-to specifications
- Review the preliminary design
  – Conduct Preliminary Design Review (PDR)
- Decide to progress to detail design and development stage

Detailed Design and Development

Following preliminary design, the detailed design and development process focuses on transforming a functional design into its physical elements and defining all of the system elements that need to be present to bring the system into existence and to operate as a solution to the original need.

This subphase involves the work of transforming preliminary designs into detailed designs. Of interest will be all aspects of the system's form, fit and functions and a comprehensive definition of the entire system's life cycle requirements and solutions, including manufacture or procurement, deployment, and operations and maintenance. Once these elements are defined, they become baseline requirements and are typically placed under configuration control. This substage will result in a complete definition of the system in terms of products, processes, materials and interface specifications.

The major steps of the detailed design and development substage are:

- Design the system element details
- Align system functions with physical solutions
- Develop engineering and prototype models and simulations
- Refine, define and specify the product, process, material and interface specifications
- Transform system specification into "build-to" or procurement specifications
- Establish the system baseline definition
- Control the system configuration
- Review the critical design
  – Conduct Critical Design Review (CDR)
- Decide to progress to the build/procure stage

Build/Procure

The build/procure phase of the system life cycle can also be referred to as the production or construction phase. Information systems normally have a build or procure effort as opposed to a production effort. Additionally, software "production" occurs during design since there is no manufacturing process involved. Fundamentally, software that is developed in-house may skip many of the elements of this phase and software that is purchased as Commercial Off-The-Shelf (COTS) will be acquired during this phase.

During this phase, products and systems are assembled and configured in accordance with the build-to specifications, and systems are procured in accordance with the procurement specifications. Often, pilot systems and processes will be deployed and tested in an operational environment so that assessments can be performed. These assessments can be performed in the early stages of the build/procure phase or the later part of the development phase. In either case, pilots will help to identify and correct system deficiencies.

The major steps of the build/procure stage are:

- Produce, construct and/or configure system elements
- Procure system elements
  - – Procure hardware
  - – Secure software licenses
- Procure utilities and circuits
- Assemble and integrate subsystems
- Conduct system pilots in operational environments
- Test system elements for operational acceptance
  - – Perform Operational Test & Evaluation (OT&E)
- Build or procure spare parts and components
- Review the system's readiness for deployment
  - – Conduct deployment readiness review
- Decide to progress to the deployment stage

Deployment

The deployment phase is the process of transferring the system from the developer to the operator and establishing all operational processes. Deployment itself may involve conducting site surveys of facilities, transporting and receiving hardware and software, and performing onsite assembly and configuration. In other words, some of the deployment activities may be similar to some of the build/procure stage activities.

One of the most important aspects of deployment will be establishing and initiating the system processes. Once a system is deployed, all the processes associated with operations, support, maintenance and so on, must be deployed as well. This effort will usually require the distribution of training and documentation and will require the establishment of procedures and policies. Each of these system elements should have been created during the development phase and implemented during the deployment phase.

The major steps of the build/procure stage are:

- Conduct site surveys
- Train operational and support personnel
- Transfer system assets
- Initiate operational processes
- Initiate policies, procedures and guidance
- Review the system's operational readiness
  - – Conduct operational readiness review
- Decide to progress to the operations and support stage

Operations and Support (Maintenance)

Once the system has been deployed, the system will enter the operational phase. The activation of operations does not require complete deployment since an Initial Operating Capability (IOC) milestone may be a first step in activating the system. However, regardless of how much of the system is activated, the support and maintenance processes must be in place and activated as well. As a system is completely deployed and achieves Final Operating Capability (FOC), the scope and scale of the operational and support process must be in place.

Support activities will include any function that helps to enable the system or helps it to operate in its intended fashion. An example of a support function would be help desk operations for an information system or service. Maintenance functions will include procedures for scheduled and unscheduled corrective actions or adjustments that are necessary to keep the system operating. Maintenance processes usually involve procedures for fault detection, isolation and repair. They will include logistics issues such as spare parts and maintenance personnel.

The major steps of the build/procure stage are:

- Operate the system in the users' environment
- Provide support functions for system elements and personnel
- Maintain the system and provide logistics
- Improve system through modifications and adjustments
- Review the system's performance and availability (ongoing)
- Decide to progress to the upgrade/replacement/disposal stage

Upgrade/Replacement/Disposal

Once the system has become operational, its service, efficiency and availability must be continuously monitored and evaluated. If it is determined that a system is approaching the end of its useful life, then the system will require some form of upgrade or replacement. A system must be disposed of if it will no longer be needed.

There often needs to be considerable planning and preparation for this phase. If a system is to be replaced or upgraded, the approach to accomplishing this must be determined and planned. The decision to upgrade or replace must be done in a time frame that will permit the replacement or upgrade to be instantiated prior to the current system failure or degradation. The most difficult issue to be addressed will be the specific process for upgrade or replacement. Any new system or upgrade elements will need to progress through their life cycle and development processes.

Upgrade and replacement can be more difficult than new system development because there are legacy systems with which to deal. Replacement systems will need to displace legacy system elements and upgrades will need to be integrated into existing legacy system. These can be extremely difficult processes and add to the complexity of the life cycle process.

The major steps of the upgrade/replacement/disposal are:

- Identify the need for upgrade, replacement or disposal
- Plan for transition, evolution or secession of operations
- Perform the system life cycle and development processes
  - Account for the existing legacy system
- Decompose the existing system elements and stand-down operations (for disposal)

Systems Engineering

Systems engineering is a complex endeavor that involves interdisciplinary fields of engineering and systems management. It addresses three high-level functions, including:

1. Transform user needs into a system description
2. Integrate the various technical disciplines to perform the engineering processes
3. Integrate system elements including products, processes, people, facilities, and so on, into a system solution

Transforming a user need into a system description describes the overall purpose of systems engineering. The goal of system engineering is the design and description of all of the system elements required for the entire life cycle of the system. While the systems engineering process may continue to function after the development process is complete, it will be used for assessing system modification and not for building, deploying or operating the system.

Systems engineering is not always considered an "engineering" discipline in the normal sense. As a discipline, it is a technical and a managerial effort that is often oriented toward integration rather than design. A standard engineering discipline—such as mechanical, electrical and civil—is more concerned with the design of system elements. With this perspective in mind, systems engineering is responsible for facilitating the interactions between engineering disciplines.

Finally, systems engineering is responsible for integrating the various system elements and providing guidance with regards to the system's entire life cycle. In this sense, the systems engineer will help to direct the technical efforts of developing system architectures, evaluate alternative solutions, control the system configuration and conduct design reviews.

Systems Engineering Process

The systems engineering process has come into existence over several decades and dates back to the 1940s when the concept of integrated, complex systems began a significant issue. The roots of systems engineering can be traced back to the development of telephony systems at Bell Laboratories as well as the early U.S. missile programs. Today, systems engineering processes have been standardized and codified. Examples of these standards include:

- IEEE 1220-2005, IEEE Standard for Application and Management of the Systems Engineering Process (September 9, 2005)
- ANSI/GEIA EIA-632, Processes for Engineering a System (September 1, 2003)
- MIL-STD-499B, Systems Engineering (Draft) (August 24, 1993)

The systems engineering process can vary significantly between organizations and may be applied differently to different systems. The process is considered "tailorable" such that it can be adjusted for various circumstances. The critical issue that is addressed by the process of systems engineering is the need for a systematic approach to development so that there is traceability and accountability. Two key elements of systems engineering traceability are the concepts of validating that system requirements address the needs of the system's users and stakeholders and verifying that the final system description fulfills those requirements. This is the fundamental nature of why systems engineering is performed.

Figure 5 shows a high-level view of the systems engineering process. Fundamentally, this process is divided into two elements, the technical process and the analysis and control process.

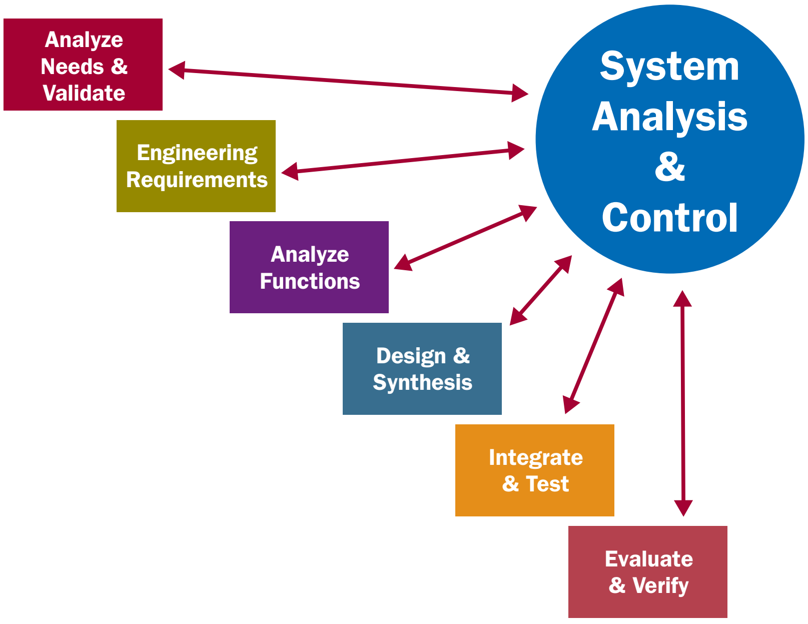

Figure 5: Systems Engineering Process

The technical process involves numerous technical disciplines that go far beyond the typical engineer fields. For information systems, the technologies may include computer engineering, computer science, network engineering, human factors engineering, psychology and so on. The analysis and control process will typically be aligned with project or program management functions. Some systems engineers may disagree with the view of systems engineering being both technical and managerial. The principle reason for this disagreement is that many systems engineers are or were design engineers and as systems engineers, they perform both the technical and managerial functions. This perspective does not discredit the view that the two functions are separate, even if the same people accomplish both.

Technical Process

The technical process of systems engineering is an iterative, waterfall process that has the objective of synthesizing a system solution and description. It does not matter what type of problem is to be solved or system is to be developed. The process is used to develop system processes as well as hardware and software component descriptions.

The technical process is initiated with a needs analysis and a needs validation effort. Once understood, the needs are translated into requirements and specifications through a rigorous requirements engineering effort. Requirements engineering ensures that all requirements are complete, unambiguous, testable, aligned, traceable and so on. These requirements are then used to define the system's functions and decompose those functions into subfunctions. The requirements are then allocated to the various functions, which will aid the eventual mapping of functions and requirements to solutions. In a similar fashion, the analysis and control process will typically allocate resources, such as schedule and funding, to the various functions so that there is an alignment between requirements and resources necessary to develop a solution.

Based on the functional decomposition, alternative solutions will be developed and analyzed in a design and synthesis effort. Trade-off studies will be performed to help determine the best alternatives for subsystems and systems. Once the best subsystem candidates are determined, they will be integrated together and tested to make sure that they work together and that the interfaces between them are well understood. Final solution evaluations are performed to verify that the overall system solution meets the requirements established for the system.

At any point in the technical process, iteration can take place and that iteration can go back any number of steps. It is not uncommon for a system to go through multiple iterations of each step in order to satisfy the users' needs.

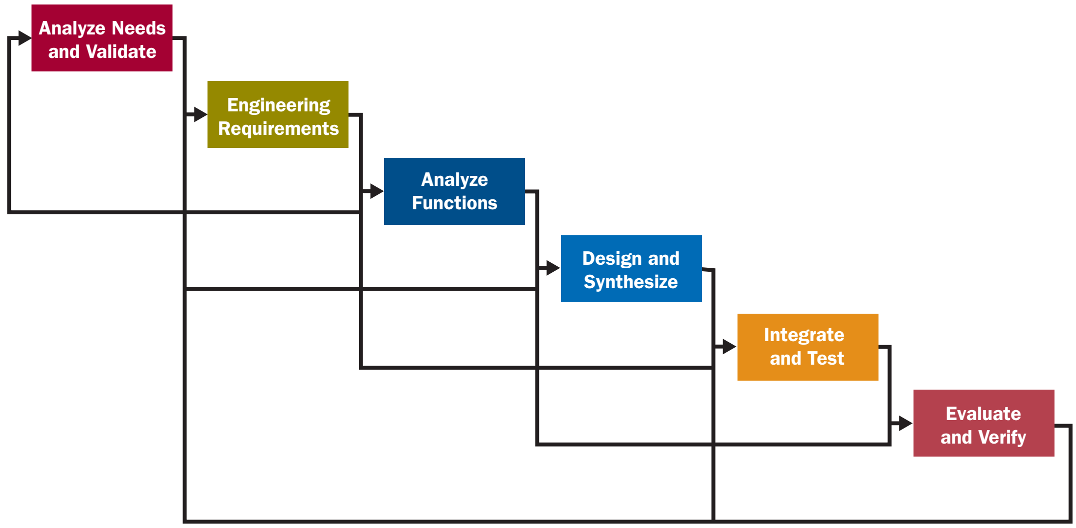

Figure 6: Iterative Technical Process

One fundamental issue with the technical process is that it will continue perpetually until an optimum solution is found, which may not exist. The reason for this is that over time, user requirements will change, a phenomenon known as "requirements creep," and available technological solutions will continue to advance and become available. Thus, without some form of control mechanism, the technical process of systems engineering may never be completed.

Analysis and Control Process

If the technical process of systems engineering were considered the machinery of system development, the analysis and control process would be the steering mechanism, fuel, oil and brakes for that machinery. It is this overarching managerial effort that manages the technical effort and provides it direction.

The analysis and control process will perform a number of ongoing tasks and will provide signals to the technical process regarding which actions to take and which actions to iterate. This management process will perform all planning functions and will develop the necessary guidance and controls to ensure that the system is managed to the plan. Risk planning and assessments will be accomplished to determine what courses of actions should be taken and what decisions are necessary to mitigate risks. The analysis and control effort is responsible for developing the function and physical architectures for the system and its life cycle. This process will direct the integration and testing as elements of the system are defined. From a control perspective, the process will determine when a solution should be solidified as a baseline so that formal configuration control can be implemented. Finally, the process will conduct periodic reviews that can be informal or formal in nature. Formal reviews will normally be aligned with key system milestones and will provide the necessary information for milestone decisions.

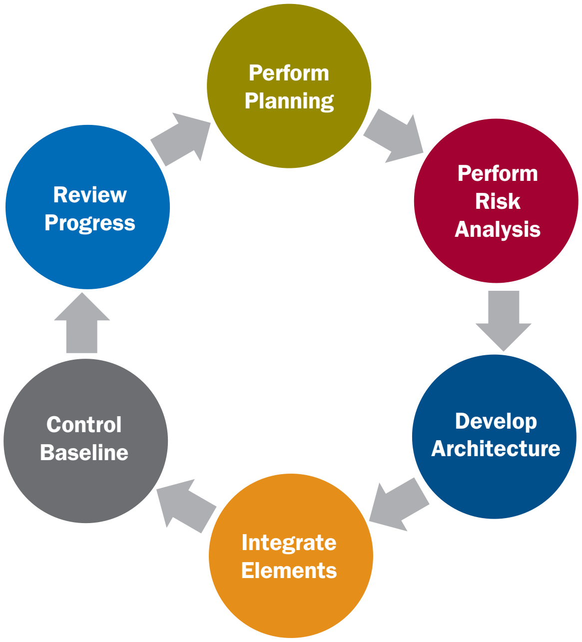

Figure 7: Analysis and Control Process

The analysis and control function is a cycle, such that each task of the process is being addressed regularly. Signals for action from the analysis and control process can be sent to the technical process at any time, but they are typically formalized in the documents and artifacts created by the management process, such as program plans, architectures, configurations and review directions.

The architectural development process of the analysis and control process is not related to enterprise architecture. The architectures developed in systems engineering are functional and physical. They provide a detailed, functional understanding of the system being developed and ultimately yield a physical architectural description of the system. The object is to first define the functions of the system and decompose them into subfunctions and then to develop a mapping of those functions to the products, processes and other system elements that will perform the functions.

Conclusion

A system is more than a product. It is a complex set of people, products and processes that work together to accomplish a stated function or objective. COOP will rely on many systems and successful COOP activities will likely rely on many systems or a system of systems. The essential systems that will be needed for COOP must be identified and understood with regards to their relation to the organization's essential functions. Once understood, an organization can begin to utilize existing systems processes to ensure that COOP is designed-in rather than bolted-on.

First and foremost, COOP requirements need to be fully understood and articulated from a life cycle perspective. What will be the operational activities for COOP and how will they be sustained for the duration of COOP operations? How will COOP assets, data, information, people and products be deployed? These issues must be addressed and folded into the system development process. For systems that already exist, COOP requirements may generate a system upgrade activity. When current systems cannot meet COOP needs and cannot be upgraded to meet COOP requirements, new systems may need to be developed. Existing, upgraded and replaced systems must fit into the larger organization system and must meet the needs of alternative potential COOP situations.

Systems engineering, as a process, is presented here as an example of a rigorous process for developing or upgrading systems. A rigorous process is necessary to ensure that needs are accounted for and to provide confidence that COOP is being integrated. The overarching need is to ensure that systems are created, upgraded and ultimately aligned with the organization's mission and operations to mitigate the risks associated with COOP.

About Juniper Networks

Juniper Networks, Inc. is the leader in high-performance networking. Juniper offers a high-performance network infrastructure that creates a responsive and trusted environment for accelerating the deployment of services and applications over a single network. This fuels high-performance businesses. Additional information can be found at www.juniper.net.

CORPORATE HEADQUARTERS
AND SALES HEADQUARTERS
FOR NORTH AND SOUTH AMERICA
Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, CA 94089 USA
Phone: 888.JUNIPER (888.586.4737)
or 408.745.2000
Fax: 408.745.2100
www.juniper.net

EAST COAST OFFICE
Juniper Networks, Inc.
10 Technology Park Drive
Westford, MA 01886-3146 USA
Phone: 978.589.5800
Fax: 978.589.0800

ASIA PACIFIC REGIONAL
SALES HEADQUARTERS
Juniper Networks (Hong Kong) Ltd.
26/F, Cityplaza One
1111 King's Road
Taikoo Shing, Hong Kong
Phone: 852.2332.3636
Fax: 852.2574.7803

EUROPE, MIDDLE EAST, AFRICA
REGIONAL SALES HEADQUARTERS
Juniper Networks (UK) Limited
Building 1
Aviator Park
Station Road
Addlestone
Surrey, KT15 2PG, U.K.
Phone: 44.(0).1372.385500
Fax: 44.(0).1372.385501

905007-001 January 2008