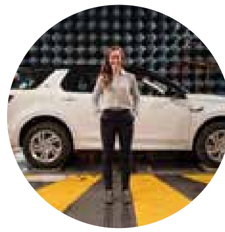
SYSTEMS ENGINEERING PRINCIPLES

CONTRIBUTORS

Michael Watson – Chair, Bryan Mesmer, Garry Roedler, David Rousseau, Javier Calvo-Amodio, Chuck Keating, William D. Miller, Scott Lucero, Rob Gold, Cheryl Jones, David Long, R. W. Russell, Aileen Sedmak

INCOSE
International Council on Systems Engineering

A better world through a systems approach

SYSTEMS ENGINEERING
PRINCIPLES

CONTRIBUTORS

Michael Watson – Chair, Bryan Mesmer, Garry Roedler,
David Rousseau, Javier Calvo-Amodio, Chuck Keating,
William D. Miller, Scott Lucero, Rob Gold, Cheryl Jones,
David Long, R. W. Russell, Aileen Sedmak

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

ISBN 978-1-937076-08-5

PREFACE

Systems engineering principles have been percolating in the systems engineering community for 30+ years. Based on the work done these past three decades, INCOSE has produced this first formal set of systems engineering principles peer reviewed by our sister organizations: AIAA, IEEE, and NDIA. These principles are not the final set but an initial set to help advance the discipline of systems engineering in application of the systems engineering processes, provide an indication of the basis of systems engineering, and spur further systems engineering research. INCOSE is excited to provide a further step in the advancement of the Systems Engineering discipline through the publishing of this first set of principles. It is hoped and expected that additional principles will be discovered/realized and that changes in the understanding of systems engineering practice or progress in the definition of the basis will lead to further updates of this set of principles. If the reader has any feedback or comments on this set of principles, they can post their input on incose.org/seprinciples. Many thanks go to the members of the INCOSE Systems Engineering Principles Action Team who were engaged, responsive, and constructively critical in the review of these principles collected from the previous works. Their insights and inputs were extremely valuable to the production of this publication. Special thanks also go to the peer review organizations from our sister societies, AIAA, IEEE, and NDIA. Their review and feedback made a tremendous improvement in the publication's formatting and clarity. We hope that this will provide you valuable guidance in your systems engineering endeavors whether practical application in the practice of systems engineering or research and development in the advancement of systems engineering.

INTRODUCTION

PURPOSE:

This publication captures the systems engineering principles for application by systems engineering practitioners.

SCOPE:

This publication addresses the systems engineering principles and hypotheses of future principles that are transcendent across all types of systems, system application contexts, and system life cycle phases.

DEFINITIONS:

Terms are defined in the Definition Section below. Definition of terms can be seen by hovering over the highlighted term in electronic formats.

Systems engineering, as does all engineering, exists to develop a solution to meet a need. This is the motivation of systems engineers to accomplish their work. But how do systems engineers accomplish this expansive challenge? Engineering disciplines have all developed from practice, sometimes over centuries or millennia, to more formal engineering basis for each discipline to answer this question. The engineering basis enables the accomplishment of more complicated and complex constructions, allowing engineers to achieve designs not previously possible. Following this progression, systems engineering practitioners have developed processes that guide the definition, development, operations, and retirement of systems stemming from an early set of Pragmatic Principles (Defoe 1993). As the processes have matured, experience has defined a set of principles and associated heuristics. Additionally, recent research into the fundamentals of systems engineering has identified new systems engineering principles and hypotheses. This publication captures these systems engineering principles and hypotheses for recognition and application by systems engineering practitioners. The codification of these principles and heuristics provides a consistent foundation for systems engineering processes and methodologies and a mechanism to evaluate and improve these processes and methodologies.

INTRODUCTION *continued*

The systems engineering principles and hypotheses in this publication represent the first edition. The current set are necessary but have not yet been determined to be a sufficient set. INCOSE formed a Systems Engineering Principles Action Team in early 2018. This team reviewed the current developments in systems engineering principles over a year, culminating in a set of systems engineering principles based on scientific research and practice heuristics. Peer review has provided validation of the current set. As systems engineering advances, further revisions or additions may occur through maturing of heuristics, scientific based research, mathematical developments, or sociological developments. As the discipline practice advances, a separate document in the future will also capture the systems engineering heuristics.

BACKGROUND

Systems engineering principles have slowly been emerging over the past 3 decades. Sometimes these have appeared as a mixture of rules of thumb or heuristics for practice and nascent principles. Systems engineering research literature contains several good articles on system principles. These principles provide a basis for the functioning of a system. They seek to group scientific axioms, laws, and scientific principles into a set of system principles. The main themes seen in the literature on system principles include system governance, system theory axioms, and system pathologies with a focus on complex systems and system of systems.

INCOSE compiled an early list of principles. These consisted of 8 principles and 61 sub-principles (Defoe 1993). These principles and subprinciples were important considerations in practice for the success of system developments and ultimately became the basis for the systems engineering processes. Other early work included a set of seven system science principles exhibited by systems (Hitchins 1992, 60–71). INCOSE also identified organizational principles in a set of 11 principles dealing with how to work successfully within an organization (Senge 1990). Project Performance International (PPI) (Halligan, 2002) has a set of SE principles that follow along the model set by Defoe providing considerations in the practice of SE, focusing on specific aspects within life cycle phases.

The Korean Council on Systems Engineering (KCOSE) provided a survey article of 8 works on systems engineering principles spanning the time from Defoe's principles through 2004 (Han, 2004), including an early version of the PPI principles. These 8 works showed evolution of systems engineering principles from practice focused to more transcendent focused principles. In 1997, the INCOSE Systems Engineering Principles Working Group (no longer active) generated a set of 8 principles building from the work of Defoe over the course of several years of discussions. These principles were a mixture of process basis, modeling guidelines, and an early worldview of the systems engineering focus. The Institute of Electrical Engineers (IEE, 2000), now part of the Institution of Engineering and Technology (IET), produced a set of 12 principles that also provided some basis for the systems engineering processes which are no longer extant. Lawrence Berkley National Laboratory (LNBL, 2001) produced a set of systems engineering principles that embody the concepts captured by the INCOSE systems engineering processes. In England, the Defence Engineering Group (DEG, 2002) produced a Systems Engineering Handbook with a brief set of principles guiding their processes and capturing some aspects of systems principles. Iowa State University reportedly produced a Systems Engineering Student Handbook containing a short list of systems engineering heuristic phrases stated as principles. The KCOSE paper also referenced a lecture on systems engineering principles from a course at the University of Southern California (USC) (Jackson, 2003). This lecture defined a principle as "a statement or generalization of a truth reflected in the systems engineering process", showing the focus on processes in the early systems engineering principle development.

Some early forms of systems engineering principles were also contained in textbooks on complex system development (Adamsen II, 2000). This set of principles assume a hierarchical system representation (complex systems have since shown to be more networks than hierarchies) and include statements on systems engineering processes. Finally, system architecting books also included some early systems engineering heuristics (Maier and Recthtin, 2002). These heuristics read as sayings about some aspect of systems engineering practice.

The KCOSE Technical Board reviewed these 8 sources and voted that 8 of the principles from these sources as a set of systems engineering principles, leading to an early form of transcendent principles consistent with the criteria defined above. These sources all show the early evolution stages of the systems engineering principles as people looked at both formal and informal (i.e., course notes and student handbooks) sources to try and understand systems engineering principles. The definition of the systems engineering processes in works such as the INCOSE Systems Engineering Handbook fulfilled some of the objectives of these early works on systems engineering principles and consolidated a lot of the work in this area. Recently, the need for more transcendent systems engineering principles has been recognized, as a guide for applying the processes.

Advances in system theory produced a set of unified propositions stated as seven axioms "from which all other propositions in systems theory may be induced." These seven axioms map to 30 scientific laws and principles (Adams et al. 2014). These axioms focus on the scientific basis of systems. Further work on these axioms provides an integration construct and a slightly different mapping to the underlying scientific laws and principles (Whitney et al. 2015). This work provides a strong integration and advancement in system theory, focusing on the principles behind the scientific basis of a system.

Complex system governance provides a set of nine metasystem functions "to provide control, communication, coordination, and integration of a complex system." These nine metasystem functions provide a basis for understanding complex systems and how to manage their acquisition or governance (Keating et al. 2016). These functions also extend to systems of systems engineering (Keating et al. 2017b).

System science approaches also incorporate systems theory leading to 10 concepts of systems theory and systems thinking (Sillitto 2014, 33–38). These 10 concepts focus on system principles providing a definition of system characteristics. A further development in system sciences produced a list of 12 systems sciences principles that also focus on the characteristics of systems (Mobus and Kalton 2015, 17–30). The statement and derivation of three principles of systems were formally derived (Rousseau 2018a, 665–681). In addition, an architecture of systemology and typology of system principles provides a good classification of scientific principles spanning from system philosophy through system practice (Rousseau 2018b). This work led to a framework for understanding system science principles (Rousseau 2018c).

System pathologies is another interesting approach to understand "circumstances that act to limit system performance or lessen system viability (continued existence) and as such they reduce the likelihood of a system meeting performance expectations." These pathologies define diagnostics for understanding systems derived from a set of 45 system laws and principles (Katina et al. 2016).

NASA undertook an independent development activity to develop the engineering and mathematical basis of systems engineering in 2011. NASA established the Systems Engineering Research Consortium consisting of 17 universities, Air Force Research Laboratories – Wright Patterson (AFRL-WP), and 5 small companies actively participating in different aspects of systems engineering through 2020. A set of more broadly applicable systems engineering postulates, principles, and hypotheses began to emerge in 2013. This consortium followed the approach of Ludwig Boltzmann in defining his postulates on gas distribution laws. Boltzmann's work is an early example of how to characterize the interactions of complex systems in seeking to understand the engineering and scientific basis of this system. According to Webster's Dictionary, a postulate is something assumed without proof to be true, real, or necessary. This led to the consortium articulating a set of postulates and hypotheses underlying systems engineering. They expanded the postulates in more detail as a set of systems engineering principles. This set of systems engineering postulates, principles, and hypotheses matured over the course of four years through peer reviews by the Consortium members, various papers, and conference panels. (Watson et al. 2014; Watson and Farrington 2016; Watson, Mesmer, and Farrington 2018; Watson 2018b; Watson 2018c) As this research progressed, a set of seven postulates, 14 systems engineering principles, and three hypotheses matured providing more specifics in the basis and application of systems engineering documented in a pair of NASA Technical Publications (Watson, Mesmer, and Farrington 2020a; Watson, Mesmer, and Farrington 2020b).

INCOSE established a Systems Engineering Principles Action Team to identify a set of principles and hypotheses to articulate the basic concepts that guide systems engineering in 2018. The team began with the multi-year work of the NASA Systems Engineering Research Consortium in the form of a whitepaper. The team mapped the systems engineering principles (i.e., axioms, laws, and principles) found in literature. This mapping provided a distilling of the earlier works into an initial set of INCOSE systems engineering principles and resulted in updates to the whitepaper. A review of this initial set by the team members occurred over a three-day meeting in Crystal City, Virginia in December 2018. An updated revision to the whitepaper resulted and led to discussion at the INCOSE International Workshop 2019 (IW 2019) open to the membership. Discussion at the IW 2019 resulted in updates captured in the article on the Systems Engineering Principles in the Insight Magazine in May 2019 (Watson, et al. 2019) and also presented at the INCOSE International Symposium 2019 (IS 2019). The INCOSE Training Working Group (TWG) sponsored a 3 session training series on the systems engineering principles in September and October 2019. Review and discussion continued as part of the INCOSE IW 2020. During 2019 and into the Spring of 2020, the team produced a set of articles to publish on the Systems Engineering Body of Knowledge (SEBoK). Working with the SEBoK editors, a survey article of systems engineering principles resulted from the work of the team. In the Spring and Summer of 2021, a series of peer reviews were held with the American Institute of Aeronautics and Astronautics (AIAA) Systems Engineer Technical Committee (SETC), Institute of Electrical and Electronics Engineers (IEEE) Systems Council and IEEE Systems, Man, and Cybernetics Society (SMC), and the National Defense Industrial Association (NDIA) that provided valuable input and feedback on the systems engineering principles. Through this process of mapping, review, and discussion a set of 15 principles and 3 hypotheses stated in this publication matured from the work of the INCOSE Systems Engineering Principles Action Team.

The principles define the system aspects and system influences that are of concern to the systems engineer. The hypotheses provide statements to advance in the understanding and prosecution of the systems engineering discipline.

DEFINITIONS

The definition of principle varies slightly across the literature. (Pratt and Cook 2017) A principle in this work is taken as a fundamental truth or proposition that serves as the foundation for a system of belief or behavior or for a chain of reasoning. (Oxford 2018) These principles guide the application of the processes and approaches for the discipline. The principles are transcendent in their scope. The Systems Engineering Principles Action Team defined a set of criteria based on understanding of the broad scope of the systems engineering discipline. The following set of criteria govern the identification of a principle:

- Transcend life cycle
- Transcend system type
- Transcend context
- Inform a world view on systems engineering
- Not be how-to statements
- Be supported by literature and/or widely accepted in the profession (it has been proven successful in practice across multiple organizations and multiple system types)
- Be focused, concise, and clear

Thus, a specific system type, a specific context in which the system is developed and operated, or a specific life cycle phase do not form the basis for a systems engineering principle. Yet the application of the principles does vary by these characteristics. Systems engineering heuristics, documented in a separate publication, capture the lessons learned from the more narrowly focused system contexts.

Heuristics are a form of guidance propositions emerging from practice in a given context or area. Heuristics, as abstractions of experience, are trusted, nonanalytic guidelines for treating complex, inherently unbounded, ill-structured problems. They are used as aids to decision-making, value judgments and assessments. Heuristics, in general, are not transcendent. Heuristics which are transcendent form the kernel for a systems engineering principle. (Rousseau, Pennotti, Brook 2022) Some of the current principles have a heuristic, or experience, basis. Heuristics are captured in the Heuristics database. **Hypotheses are potential principle statements that have varying support and that research can prove or disprove.** Systems engineering hypotheses provide statements to advance the understanding and application of the systems engineering discipline. The evidence for these varies, being either positive or negative evidence. In some cases, there are approaches or heuristics that operate on an individual hypothesis as stated. In other cases, there are approaches that assume these individual hypotheses are not true. Thus, these statements require a proof from the basis of systems engineering defined in Principle 15. Proofs provide the basis to promote a hypothesis to a principle or to remove a hypothesis that is disproven.

Elegant is a term used to describe a well-formed systems engineering solution. This concept appears in some of the principle and hypothesis discussions. Robert Frosch first introduced the idea that systems engineering is to produce an elegant design in a speech from 1967. (Frosch 1993) Mike Griffin expanded this into four characteristics of elegant systems as: System Efficacy, System Efficiency, System Robustness, and Minimizing Unintended Consequences. (Griffin 2010) System elegance then is 'a system that is robust in application, fully meeting specified and adumbrated intent, is well structured, and is graceful in operation.' (Watson, Mesmer, and Farrington 2020a)

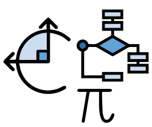
In reviewing the literature references, differences between system principles and systems engineering principles emerge (Watson 2018b). System principles address the functioning of a system, looking at the scientific basis for a system and characterizing this basis in a system context. Systems engineering principles address the engineering approach to developing and operating a system. These principles provide guidance in the application of systems engineering processes. The systems engineering approach must recognize and utilize the system principles, forming a relationship between systems engineering principles and system principles. This relationship exists in the scientific basis of systems engineering (see Principle 15 on page 34).

SYSTEMS ENGINEERING PRINCIPLES

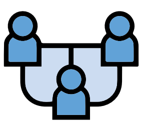
The systems engineering principles are stated in this section. Each principle is supported with a description, evidence that supports the principle (e.g., observable evidence of the application, proof from scientific evidence), and implications in systems engineering practice for application of the principle. Sub-principles are supported by a description and encompassed by the higher-level principle description, evidence, and implications.

ICONS

Systems Engineering Principles have a Physics/Logical focus or a Social focus or both. Icons represent each of these focuses and are used to denote the focus of each systems engineering principle. Where a systems engineering principle has both focuses, both Icons are used to denote the principle.



PHYSICAL/LOGICAL PRINCIPLES



SOCIOLOGICAL PRINCIPLES

PRINCIPLE 1:

Systems engineering in application is specific to stakeholder needs, solution space, resulting system solution(s), and context throughout the system life cycle.

DESCRIPTION: This is the first and foundational statement on systems engineering. The product (system) and its operational environment drive systems engineering and the system integrating physics, logic, and social and cognitive relationships (context) that are foundational to the specific product or system. Essential to this is the understanding of the mission or use of the product as formulated by the product goals. This includes the aspects of the system needed to operate in an elegant manner and thus considers the entire product life cycle.

Principle 1 represents all systems types.

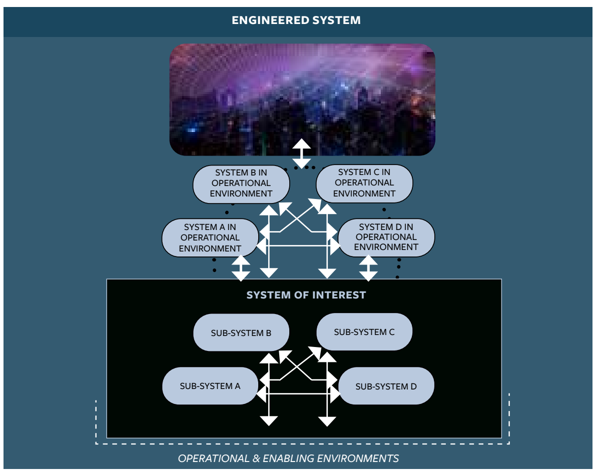
EVIDENCE: The ubiquitous tailoring of systems engineering approaches provides strong support for this postulate. Systems engineering approaches must be consistent with the system needs during development and operations. The NASA Systems Engineering Consortium research surveying the NASA 17 systems engineering processes provides support for this postulate indicating 72% of companies interviewed have systems engineering processes unique to their product. More than 7% of the respondents (Componation et al. 2013) do not follow a standard process.

IMPLICATIONS: This principle states that any application of systems engineering considers the specific system needs and organizational characteristics in development or operation. The systems engineering methods applied to a product will and should vary in emphasis and application based on the nature of that product, its physical application environment, and its context.

PRINCIPLE 2:

Systems engineering has a holistic system view that includes the system elements and the interactions amongst themselves, the enabling systems, and the system environment.

DESCRIPTION: From a physical, logical, and structural sense, a system is not a single mechanical, electrical, or chemical entity (e.g., it is not a single rod, wire, or chemical); it encompasses a set of interacting similar (e.g., mechanical) or dissimilar (e.g., mechanical, electrical, chemical) elements or subsystems providing capability(ies) not available independently. Systems engineering is concerned with combining multiple elements or subsystems, of various physical and logical types, into a best-balanced functional whole to accomplish the mission goals. This principle addresses the system integration aspects of systems engineering. Principle 3 addresses the discipline integration aspects below.



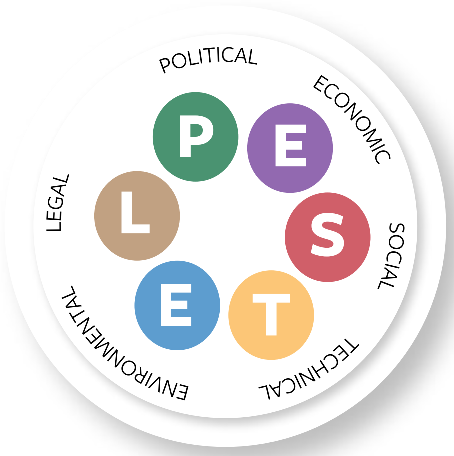
EVIDENCE: This principle provides the definition of a system that is consistent with mathematical category theory. This theory defines a category as a set of objects and the interaction among these objects. A system follows this definition where the system elements are the mathematical objects; the interactions between these elements, their enabling systems, and the environments are the interactions between the mathematical objects. Note, that category theory supports categories of categories. Thus, a system is the holistic category that may contain other categories such as subsystems and the enabling systems. In addition, the natural relation in category theory provides for the interaction with the system environment. Mac Lane's *Categories for the Working Mathematician* (1971) provides a good description of the mathematics in category theory.

In practice, individual engineering disciplines deal with the development of their specific functions extremely well. The integration of these functions with each other and with the environment define the interrelationships that drive the final system performance including emergent properties not evident from the individual system functions. Thus, the engineering addresses the individual functions well while the integration of the engineering functions makes these functions a system. The domain of systems engineering is the set of these integrated relationships. Category theory also supports this domain definition where the set of interrelationships fully defines the category (the system). The objects exist as the initiating and terminating points for these relations and so are still present, but the interrelationships can be seen to provide a definition of the system.

IMPLICATIONS: The systems engineer focuses on the interaction of these subsystems, not as a design engineer focused on the details, but as a well-versed integrator. These interactions, occurring simultaneously in system execution and operation, are the basis of the logical systems architecture. These system interactions, including interactions with the system environment, can drive the design as strongly as the subsystem functions themselves and, when coupled, can potentially create unexpected system responses. The systems engineer must understand and manage these responses in the full system context rather than a component context.

PRINCIPLE 3:

Systems engineering influences and is influenced by internal and external resource, political, economic, social, technological, environmental, and legal factors.



DESCRIPTION: The technical aspects of the system are not the only focus of systems engineering. Internal and external resource allocations influence the approach to systems engineering, as does the approach to systems engineering influence these resource allocations. In addition, political, economic, social, technological, environmental, and legal (PESTEL) factors (derived from business and marketing analysis [Oxford 2016]) are all factors in the context in which systems engineering executes. These factors are present in the organizational structure and culture. The system under development drives the development process that has a corresponding influence on the structure of the system's developmental and operational environment, including organizational structure and culture. Similarly, the structure and culture of the organization has an influence on the engineering of the system.

EVIDENCE: Organizational mirroring provides examples where the organization maps to system functions. Research in biased information sharing also shows that the organization maintains the system margin that is not always clearly identifiable in the system design. (Austin-Breneman, Yu, & Yang 2014; Austin-Breneman, Yu, & Yang 2015) Research in sociology has also indicated this principle. (Constant II 1993, 231) Thus, the organizational structure and culture has a big influence on the design and information visible within the design. Information on the design can vary by organization and can reside in the organization rather than be explicit in the system design, affecting both system development and operations.

IMPLICATIONS: The systems engineer must be cognizant of the resources, PESTEL factors, culture, the organizational interactions, and their potential impact on the design of the system. The systems engineer must understand how information flows through the organization, how the organization filters and interprets the flow of information, and how the system design or operational procedures capture the information flow. The systems engineer should work with project management and line management to address issues in organizational information flow and culture to improve the elegance of the system.

PRINCIPLE 4:

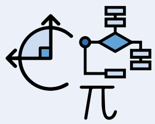
Both policy and law must be properly understood to not overly constrain or under constrain the system implementation.

DESCRIPTION: Every project has overarching constraints that extend beyond the physical and environmental. Specifically, most (if not all) projects have a limited budget and schedule. In addition, all systems must conform to established organizational and government policy and laws. Political processes vary by nation across the world yielding variations in the laws produced and the applicability to a system. These policies and laws put additional constraints on budgets, schedules, and technical solutions. They can incorporate standards from international and professional organizations. These factors provide a context in which the system is developed and operated. In addition, the system design choices also influence these factors. The understanding of legislators on what systems can actually achieve their intents drives government policy and law. Similarly, the types of systems the corporation or company chooses to develop influences corporate/company policy.

EVIDENCE: Every project has these constraints. Infinite budgets or schedules do not exist. Policy and law application pervade our systems. The legislators' understanding of solutions needed to accomplish their intents drive government policy and law. Similarly, corporate/company budgets and schedules are based on the executives' understanding of the budget and timeframe necessary to develop a system. This understanding drives the budget and schedule allocations that encompass both total funding and timeframe that the government or corporate /company executives will provide.

IMPLICATIONS: Policy and law act as important constraints on the system. Their application depends on the specific system developed and its context (Principle 1). Policy and law, often written in requirements-type language, are not requirements. The context for the policies and laws is much different, often being much looser than requirements and more likely reflecting high-level system expectations than specific system functional or operational choices. Often, most interpret policy as having more flexibility than law. The systems engineer should understand how much flexibility is acceptable by those who set the policy (whether government or organizational) and those who pass the laws (Watson, Mesmer, and Farrington 2020b, 140-149).

Social choices drive the establishment of these policy and law constraints. People make choices to define budget limits, schedule limits, policies, and laws, whether at the national or organizational level. Thus, physical and logical solutions through these constraints link social choice theory. These choices are based on an understanding of systems' abilities to achieve the government and corporate/company executives' intents. This understanding drives the budget and schedule allocations and the policies put in place. Similarly, the available budget, available expected duration, and existing policy and law can influence choices in the development of a system.



PRINCIPLE 5:

The real system is the perfect representation of the system.

DESCRIPTION: This principle provides a statement of the idea long espoused among statistical modelers. Models represent various aspects of a system, but the only complete, full, or perfect representation of the system is the system itself. This is particularly true for cyber systems and other non-deterministic systems where non-deterministic system response modeling approaches are not well defined. In modeling a system, the system model is composed of multiple models of aspects of that system, and each is an approximation to reality.

PROOF: Kullback-Liebler information provides a definition for "ideal" information (Burnham and Anderson 2002). This information measure indicating how close a particular model matches the real physical system is:

$$\mathrm{I}\,(f,g) = \int f\,(x)\log\,(f(x))\;dx - \int f(x)\log\,(g(x|\theta))\;dx \tag{1}$$

where,

f(x) is the real physical system,

g(x|θ) is the model of the system, and

θ are the model parameters.

Setting this relationship to zero provides a relationship to define the differences in a given model to the real system. This provides a proof that the perfect model of the system is the system itself.

$$\int f(x)\log\,(f(x))\;\;dx - \int\;f(x)\;\;\log\,(g(x|\theta))\;\;\;dx = 0 \tag{2}$$

$$\int f(x)\log\,(f(x))\;dx = \int f(x)\;\;\log\,(g(x|\theta))\;dx \tag{3}$$

Note, also that copies of systems are not physically identical.

$$\mathrm{f}_1\,(x) \neq \mathrm{f}_2\,(x) \neq \cdots \neq \mathrm{f}n\,(x) \tag{4}$$

Thus, a physical system only represents itself identically and not other physical copies of the same system (such as copies of the same model from a production line).

IMPLICATIONS: A perfect model, being the system itself, means all other models have limitations. Various system models can show various aspects of the system, but no system model can show the complete system. In addition, one copy of the physical system is not identical with another copy of the system. Even for software copies across multiple production units, corruption can occur in transfer or memory errors. Thus, variation in copies of the same physical system exist at various tolerance levels depending on the design and production approaches.

PRINCIPLE 6:

A focus of systems engineering is a progressively deeper understanding of the interactions, sensitivities, and behaviors of the system, stakeholder needs, and its operational environment.

DESCRIPTION: Progress of the system through development and operations produces a deeper understanding of the system as a whole. Progress through system development requires decisions that are more detailed. These detailed decisions result from this deepening understanding. The knowledge captured, maintained, and improved by systems engineering deepens as the discipline organizations complete their development work and the system functions are integrated. This deepening of understanding enables the systems engineering decisions necessary to produce an elegant system. The focus of systems engineering is on understanding the interactions of the system, many of which are not apparent until system integration (physical integration, logical integration, functional integration). This leads to a continuous reduction in system uncertainties and identification of system sensitivities. The systems engineer should understand the behavior of the system, including the emergent behaviors, prior to the operational phase. This necessitates systems engineering modeling tools to allow a sufficiently deep understanding of the system. As the system progresses through the life cycle, the systems engineer seeks the best balance of performance, cost, schedule, and risk. Understanding of the system could also regress, if organizational changes occur due to inactivity of an organizational element (loss of experience), retirement of key experienced individuals, or closure of suppliers. This loss of system understanding is more likely to occur in the later phases of the life cycle, such as operations, due to the lack of information flow between life cycle phases.

EVIDENCE: In practice, this deepening of understanding is in any system development or operation. The technical assessment process shows this as systems progress from concept review to requirements review to design review to acceptance review. This continues in the operational phases with lessons learned abundant for any system. This deepening of understanding of the system and its application drives commercial product upgrades or new models. Regression of system understanding can also occur in some life cycle extension activities. When system understanding is not maintained, the basis of systems specification becomes unclear, and some systems have been found not to perform (either underperform or over perform) to their system specifications. In addition, operational procedures can lose their basis that makes it difficult to determine when they should be retired or maintained as the system ages.

IMPLICATIONS: Systems engineers derive requirements as the system design progresses. Thus, while systems engineers define the mission requirements (part of understanding the mission context) at the beginning of development, they define the system requirements progressively. The requirements are a function of the design choices made and understood progressively throughout the development phase. This also applies to cost and schedules, particularly for new systems where the development or operations result in unexpected changes. Similarly, systems engineers develop models to predict system capabilities and then refine these models as they obtain testing and operational experience. System models gain fidelity as the design progresses and the systems engineer must manage the interaction between subsystem design maturity and system model maturity. These system models become the basis of system operations (Watson, Mesmer, and Farrington 2020a, 167-174). If systems engineers do not maintain the system basis, then the understanding of why certain procedures or specifications were defined can be lost. This becomes problematic for aging systems, particularly as they reach the generational gap for the workforce after 20 years of service.

There are several sub-principles that define aspects of this progressively deeper understanding of the system interactions, sensitivities, and behaviors.

SUB-PRINCIPLE 6(a):

Mission context is defined based on the understanding of the stakeholder needs and constraints.

The understanding and definition of the mission context (the system application use cases and constraints) is essential to a well-developed and operated system. An understanding of the stakeholder needs and constraints on the system defines the mission context. This requires an understanding of the stakeholder's relationship to the system in operation. Different stakeholders have different perspectives on what is important (operator versus maintainer versus general community). For example, the manufacturer, the driver, mechanic, and general public are all stakeholders for an automobile. The perspectives that each of these provide is different and can be either enforcing or conflicting. The manufacturer is concerned with production costs and appeal to customers. The driver is concerned with the general appearance, amenities, and ease of operation. The mechanic is concerned with accessibility to the vehicle's engine and components. The general public is concerned with safety and environmental impacts. The definition of the system application must bring together all of these perspectives.

SUB-PRINCIPLE 6(b):

Requirements and models reflect the understanding of the system.

The accuracy and completeness of system requirements and system models reflect the understanding of the system. A system that is not well-understood leads to poorly stated requirements, requirement gaps, and inaccurate system models and representations. The objective of systems engineering is to understand the system in its mission context (Principle 11(a)), which then produces the proper specification of requirements and proper representation of the system in the system models.

SUB-PRINCIPLE 6(c):

Requirements are specific, agreed-to preferences within the developing organization.

Preferences are attributes of an individual. The organization as a whole, however, must at some point consolidate these individual preferences and agree on specific values, such as performance, cost, and schedule that the system will achieve. These agreed-to preferences along with some agreement on the uncertainty in their measure are the system requirements. These are specific to the system in development. Systems engineering carefully defines the requirements (agreements) that are necessary for the successful completion and operation of the system. Integration of the disciplines is dependent on these requirements (agreements) between the different disciplines developing or operating the system. Configuration management is an important systems engineering function in maintaining these requirements (agreements) and managing their change in a consistent and coherent manner.

SUB-PRINCIPLE 6(d):

Requirements and system design are progressively elaborated as the development progresses.

The definition of mission requirements occurs early in the understanding of the system as a part of mission context. System requirements progress through a system requirements review based on configuration decisions. The remaining technical requirements derive from system design decisions that progress throughout the development phase. The definition of subsystem requirements occurs early in the design phase and the definition of component requirements occurs during detailed design activities.

SUB-PRINCIPLE 6(e):

Modeling of systems must account for system interactions and couplings.

System interactions and couplings vary, involving serial, parallel, nested, and looping relationships. These interactions may be mechanical, electrical, logical, chemical, etc. or any combination of these. Often, multiple peer relationships provide connections among system functions and the environment. Multidisciplinary Design Optimization (MDO) is a system modeling type that accounts for these. Looping, nested, and peer relationships support interactions and couplings not usually seen in hierarchical structures that generally only indicate parent/child relationships. In addition, hierarchical structures do not distinguish subtle interaction effects from strong interaction effects. Thus, system models must account for these interactions and couplings, clearly representing them.

SUB-PRINCIPLE 6(f):

Systems engineering achieves an understanding of all the system functions and interactions in the operational environment.

System functions and interactions in the operational environment provide the fabric for understanding the system. Requirements are essential agreements (Sub-Principle 6(c)) to allow the organization to work cooperatively but do not necessarily represent the function and interactions completely or evenly. Ideally, requirements are level (at the same level of detail in the design) and balanced in their representation of system functions and interactions. In practice, requirements are not level and balanced in their representation of system functions and interactions. Systems engineering ensures the system will perform as the designers expect based not only on their requirements but also on their models and designs. This leads to the principle that the proper performance of the system functions (outputs are within required ranges for a given input state) is a focus of system engineering. If the requirements are not truly level and balanced, a focus on the system functions will assist the systems engineer in ensuring a successful application of the system.

SUB-PRINCIPLE 6(g):

Systems engineering achieves an understanding of the system's value to the system stakeholders.

System success is contingent on the stakeholders' expectations, not on the quality of the system requirements, models, and design information. System value models that capture the stakeholder's preferences for the system provide the basis to perform system validation. A system value model represents the preferences of the system stakeholders in a holistic fashion. This enables finding the system with a set of attributes that best produces the system value correlated to the system most preferred by the system stakeholders. System success melds the system as designed and as built with the system as expected by the stakeholders. Often systems engineers assume the requirements reflect the stakeholder expectations. This is difficult to accomplish in practice due to the melding of external stakeholder expectations with developer expectations. Thus, requirements do not clearly reflect the stakeholder (internal or external) expectations in many system developments. These expectations are more clearly and directly represented in system value models. These system values models, constructed at the beginning of the system development, appear to provide a mathematical basis to define and guide the system development with the stakeholder's expectations throughout the system life cycle.

SUB-PRINCIPLE 6(h):

Understanding of the system degrades during operations if system understanding is not maintained.

The operational phase of a system is generally much longer than the development phase. This occurs for automobiles, consumer products, military systems, medical systems, aerospace, and petroleum industry, to name a few. The understanding gained in the development phase forms the basis for the operations and maintenance understanding of these systems. If systems engineers do not maintain this understanding, then future maintenance activities, such as obsolescence upgrades, new application upgrades, and corrective maintenance, require a relearning of the system and design decision basis that adds cost and time to the future system upgrades and repairs. Configuration management supports the maintenance of the system knowledge throughout the system life cycle. System models provide a mechanism to capture and maintain the understanding of the system (e.g., functions, interactions, production, operations) if the models are transitioned, maintained, and used in the system's operational and maintenance life cycle phases.

PRINCIPLE 7:

Systems Engineering addresses changing stakeholder needs over the system life cycle.

DESCRIPTION: Over time, the degree of consistency in stakeholder and user preferences tends to diminish due to environmental changes, emerging technologies, or changes in the makeup of stakeholder and user communities. For systems with long life cycle phases, these communities and their preferences can change significantly. We see this primarily in the operations phase and it also occurs in the development phase of long developments. This variation becomes more pronounced as the system lifetime increases. With more variation in stakeholders and stakeholder preferences, changes can be introduced to the system that can impact the system's ability to adapt to these preferences or stretch out system development duration. A key to managing these socially driven changes is to recognize when these shifts indicate the need for system modification, development of a different system, and the time for the current system to move into decommissioning.

EVIDENCE: This is a normal occurrence in the practice of systems engineering. The systems engineering processes deal with the change in stakeholders' expectations. These changes are a major source of change in mission context and system requirements.

IMPLICATIONS: This leads to instability in expectations for the system and in the system requirements. The systems engineers must be aware of these changes and account for them as early as possible. Early identification can provide for lower impacts to system development cost and schedule and to system operational change timeframes. Systems where stakeholders' expectations have the potential to change should employ more flexible systems engineering process application such as agile systems engineering to accommodate changes as the system moves through the life cycle.

PRINCIPLE 8:

Systems engineering addresses stakeholder needs, taking into consideration budget, schedule, and technical needs, along with other expectations and constraints.

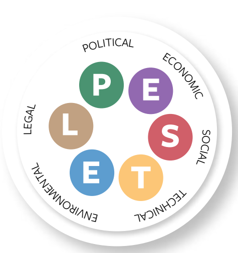
DESCRIPTION: Systems engineering solutions must address the stakeholders' needs and their constraints. Budget and schedule constrain the development and integration of the system, the operation and maintenance of the system, and the integration of the disciplines developing or operating the system. Note that budget is the amount allocated to execute the system development or operation and is not the actual cost. A focus of systems engineering is to keep the cost within the budget or recommend when the solution space defined by budget and schedule does not meet the intended system application. In addition, other expectations and constraints, such as environmental impacts, economic impacts, or social impacts, may also affect the system solution options. The systems engineer must account for each of these to ensure a system is developed and operated to satisfy the stakeholders' needs and constraints as captured by the mission context.

EVIDENCE: Solutions defined in response to the stakeholders' needs drive system cost, schedule, and other expectations and constraints. System budget and schedule problems result from a lack of understanding of the best balance of the system within the resource allocations provided and the technical needs of the stakeholders. Unexpected consequences can be realized by systems where environmental impacts, economic impacts, and social impacts are not recognized or understood.

IMPLICATIONS: System solutions account for not only the technical performance but also must fit the allocated budget, schedule for development and operation, and other expectations and constraints, such as environmental impact, and social impact. The systems engineer must understand the cost, schedule, and other impacts as well as they understand the technical performance of the system. The systems engineer develops this understanding from the initial concept definition and maintains it through the system lifecycle.

SUB-PRINCIPLE 8(a):

Systems engineering seeks a best balance of functions and interactions within the system budget, schedule, technical, and other expectations and constraints.



In accounting for all of the system needs and constraints and defining the system's mission context (system's application), the systems engineer seeks to obtain a best balance of all of the stakeholders' differing preferences. This best balance provides a system that most fully meets the system context (resource allocations, PESTEL factors, and the differing stakeholders' preferences). This balance requires a thorough understanding of the system and its mission context in order to achieve a best balance within the full system context.

PRINCIPLE 9:

Systems engineering decisions are made under uncertainty accounting for risk.

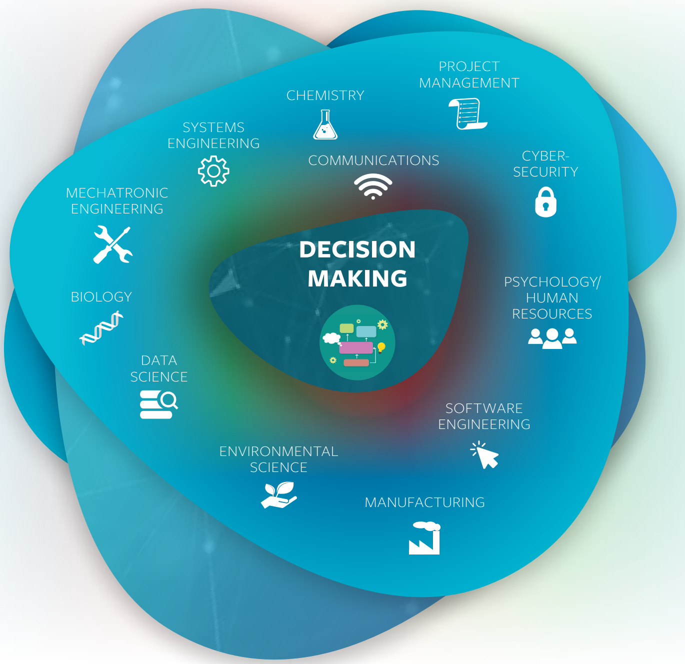
DESCRIPTION: Systems engineers progressively understand information about the system through the development process and through the operations process. There are several sources of uncertainty in the development and operations. Some of this is natural based on the progressive understanding of the system. Uncertainty exists due to the inability to predict the future with certainty and decision-making. This requires an understanding of a future system state that naturally entails a risk of the state not actually realized. Uncertainty arises from many aspects of systems engineering, including limited knowledge of system environments and social aspects of the organization that affects information maintenance, creation, and flow. Organizational cultural differences can lead to variation in defining risks and in risk tolerances that the systems engineer must incorporate. Risk tolerance between individuals in the development organization, operational organization, and the stakeholders varies dependent on the system context. In addition, system risk is an aggregation of uncertainties in the underlying system components and associated modeling. System risk tends to be at a higher level of aggregation and lower resolution than risk in the components of a system. This creates a mismatch that the systems engineer needs to address appropriately. Systems engineering must also understand sensitivities to ensure the proper focus on the different uncertainties. Systems engineering models the uncertainty and sensitivities throughout the process. Risk in decision-making comes from the need for a sufficient understanding of the system context and the knowledge that uncertainty does exist even as understanding improves.

EVIDENCE: Systems engineering risk processes exist to address this reality. The inability to predict future decisions and their impacts leads to risk in the decisions about the system. Selected system solutions have assumptions on what factors may or may not manifest themselves. In addition, the unknown unknown factors can drive risk unexpectedly. Systems engineers will recognize many of these factors as the system proceeds through development and operations, but they may not recognize them at the time needed for the decision.

IMPLICATIONS: Systems engineers are responsible for understanding the system and the system solution implications. Systems engineering must properly identify, track, and mitigate risk factors through the development. Systems engineers may realize new risks at any point in the development or operations life cycle phases. They should recommend risk mitigations understanding the impacts to system functionality, interactions, and stakeholder expectations. As the system decisions are made, the risks associated with the decision become apparent.

PRINCIPLE 10:

Decision quality depends on knowledge of the system, enabling system(s), and interoperating system(s) present in the decision-making process.



DESCRIPTION: Engineering decisions are made formally (i.e., control boards) or informally (i.e., study teams). Engineering organizations often create trade study or task teams to investigate and resolve specific problems that is a process of organizational flattening. Decision effectiveness depends on these decision-making bodies and organizationally flat teams involving the right decision makers with a sufficiently complete understanding of the decision context and the decision need. Having a full knowledge of the system and its context is contained in the decision-making body or team, protects against individual biases and reduces uncertainty (in particular mitigates the absence of information necessary for the body or team to make a decision or recommendation). Decisions are process dependent and information needed by the decision makers directly drives the decision methods.

EVIDENCE: Decisions made without a full understanding of the impacts on all phases of the system are known to be flawed in practice. These decisions lead to impacts to subsystems, enabling systems, and interoperating systems when the knowledge of these systems is not present among the decision makers.

IMPLICATIONS: Good decision quality requires the right knowledge be present in the decision-making process. Having the right knowledge mitigates individual biases and reduces uncertainties in system knowledge. This drives the membership of boards in the decision-making process, membership on trade study teams, integrated product team structures, and the approach for external coordination. Systems engineers should avoid decision-making processes where the system knowledge needed for the system decision is fragmented. Fragmented decision bodies lead to system decisions that do not properly balance all aspects of the system and the impacts to the enabling systems and interoperating systems. These fragmented systems often operation with unmitigated biases and uncertainties in the absence of knowledge about the system.

PRINCIPLE 11:

Systems engineering spans the entire system life cycle.

DESCRIPTION: Systems engineering is not just a development phase activity. It continues throughout system operation, decommissioning, and disposal. The organizational relationships and goals change as the system progresses through these phases, but systems engineering continues to integrate the system functions and the system disciplines throughout all phases of the system life cycle. Operations engineering is responsible for the operation of the system. Systems engineering is responsible for the various changes/upgrades to the system capabilities.

EVIDENCE: Systems engineering is well understood during the development phases. During the operational phases, systems engineering is still essential as the system goes through maintenance upgrades, new application adaptations, and obsolescence driven re-designs, among other stages. In addition, during decommissioning and disposal, systems engineering is essential to deal with the proper decoupling of the system and ensuring conformance with policy and laws affecting the system disposal.

IMPLICATIONS: As the system progresses through its life cycle, the need for systems engineering changes. A shift takes place from development to operations in terms of the scope of changes and organizational responsibility. Operations engineering is responsible for operating the system while systems engineering is responsible for the system changes/upgrades. The baseline operational system, then, becomes the medium in which operational phase system changes take place. The organization changes significantly as the system transitions from development to operations. Organizational relationships and needs are different. Culture can be very different between development and operational organizations. All of this affects the system and the systems engineering must deal with these organizational changes. Another organizational change and culture shift occurs during decommissioning and disposal.

A set of sub-principles defines the specific aspects of systems engineering throughout all of the system life cycle phases.

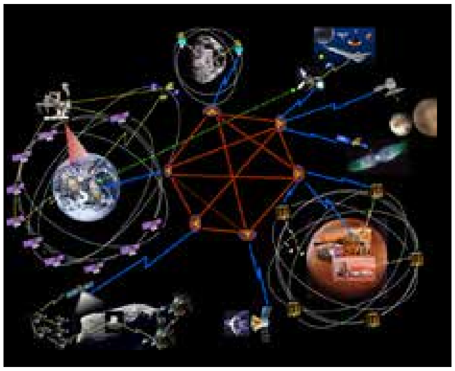
SUB-PRINCIPLE 11(a):

Systems engineering obtains an understanding of the system.

Understanding the system as a system rather than a collection of components is essential to the successful development of any system. The level of understanding of the system that the system engineer possesses underpins everything they do in terms of engineering the system.

SUB-PRINCIPLE 11(b):

Systems engineering defines the mission context (system application).



The systems engineer integrates all of the different stakeholder group's preferences, PESTEL factors, and resource allocations (budge and schedule) to produce a well-founded understanding of the mission context (system application). The mission context evolves from this integration and understanding activity and is the essential starting point for system development and operations activities.

SUB-PRINCIPLE 11(c):

Systems engineering models the system.

Systems engineering develops and maintains system-level models to aid in the design and analysis of the system as well as provide the necessary system basis for the operational and maintenance plans and procedures.

SUB-PRINCIPLE 11(d):

Systems engineering designs and analyzes the system.

Systems engineering performs design and analysis at the system level. Ideally, this is not merely a cognitive integration of the results of various discipline models, but rather uses system-level models to perform design at the system level. This then informs the system-level guidance to the discipline design to ensure the design closes at the system level during the design analysis cycles. Systems engineering performs system analysis of the integrated results from the discipline analysis in a coherent manner based on the system-level physics/logic.

SUB-PRINCIPLE 11(e):

Systems engineering tests the system.

System testing comes in many forms including developmental system testing, system qualification testing, verification testing, validation testing, certification testing, and acceptance testing. In all of these, systems engineering is a critical aspect of system testing. The systems engineer should define test objectives at the system level to ensure testing not only accomplishes specific discipline test objectives but also objectives at the system level. This can involve separate system tests, modification of discipline tests for system level objectives, or system-level analysis of test data to obtain a system-level understanding.

SUB-PRINCIPLE 11(f):

Systems engineering supports the production of the system.

The production (manufacture, fabrication, coding) of the system is an integrated activity between the system components and the tooling. In addition, changes during manufacturing often have system-level implications and can unexpectedly change system interactions. While this sub-phase is the purview of the manufacturing engineer, the systems engineer must stay involved to understand changes, update models, and perform analysis to ensure appropriate production changes at the system level.

SUB-PRINCIPLE 11(g):

Systems engineering supports operations, maintenance, and retirement.

Systems engineering has a key role in system operations defined by system interactions. Systems engineers obtain further understanding of the system interactions as the system operational experiences mature. This enhanced understanding leads to updates of system models used for operations and potential system maintenance upgrades or fixes. Similarly, systems engineering provides the understanding during decommissioning in how to de-integrate the system and dispose or repurpose system and organizational assets.

PRINCIPLE 12:

Complex systems are engineered by complex organizations.

DESCRIPTION: This principle is fundamental to the execution of systems engineering. The systems engineer must deal with both the complex system (the organization) that develops the system and the complex system itself. This dual focus forms the basis of systems engineering. The systems engineer is responsible for both integration of the system functions and the integration of the disciplines developing these functions. The social interaction within organizations working on complex systems is itself complex and is a strong driver in budget and schedule efficiency or inefficiency.

EVIDENCE: Major system failures have occurred due to the lack of information flow through the organization. Organizational structures, particularly for large system developments, are highly socially diverse with diversity in people, the engineering disciplines, and the organizational culture. Projects with more than one company involved see this organizational complexity increase tremendously. It is difficult in some organizational structures to understand how to share the information and what information to share.

IMPLICATIONS: Complexity resides not only in the system but also in the organization(s) developing and operating complex systems. Thus, systems engineers must deal with both the complexity of the system and the complexity of the development and operation organization(s).

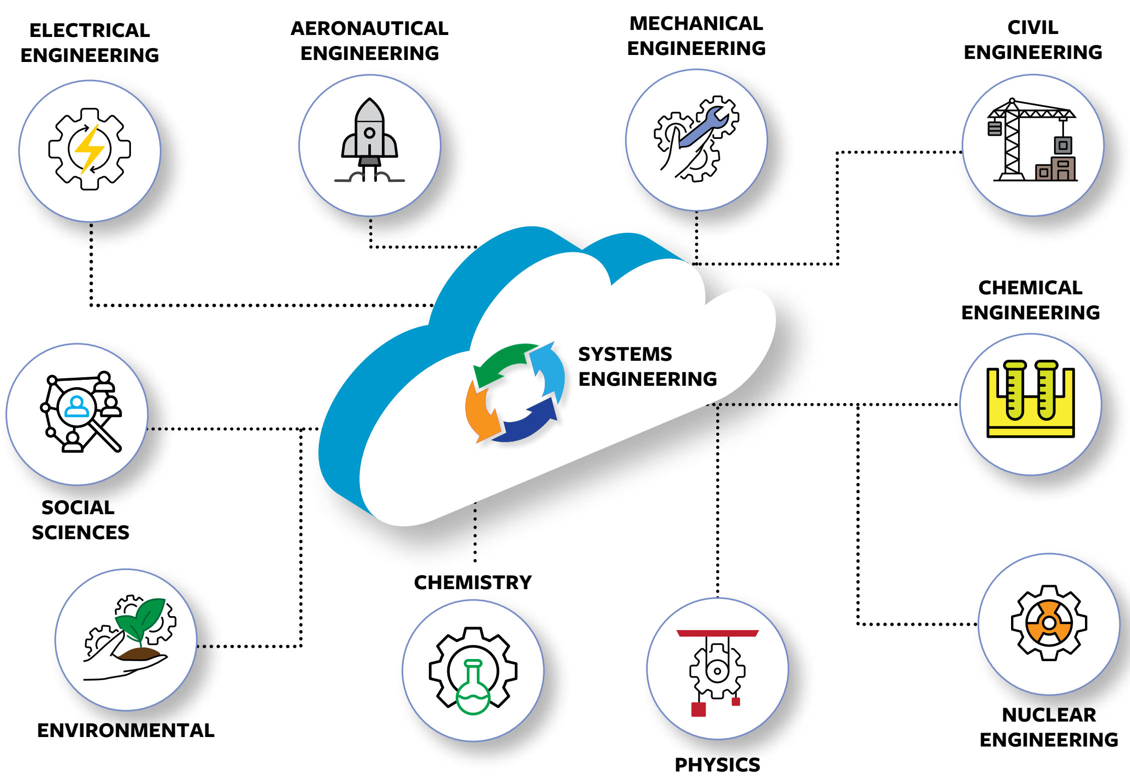
PRINCIPLE 13:

Systems engineering integrates engineering and scientific disciplines in an effective manner.

DESCRIPTION: The systems engineering discipline is its own engineering discipline, but is also dependent on other engineering, scientific, and social disciplines. Systems engineering seeks to integrate and incorporate these other disciplines in an elegant manner to produce an elegant system throughout the system life cycle.

EVIDENCE: Multiple engineering and scientific disciplines develop any complicated or complex system with many social aspects influencing the integration. These engineering and scientific disciplines with social influences work in an integrated fashion, formally and informally, to produce these systems.

IMPLICATIONS: The interaction of the disciplines is a focus of systems engineering. The objective is a basic understanding of each discipline with a detailed understanding of their interactions. This incorporates various organizational integration aspects. The systems engineer must be cognizant of the organizational and sociological influences on the system development and operations. The systems engineer must also "engineer" these relationships.



Systems Engineering Integration of Discipline Interactions

PRINCIPLE 14:

Systems engineering is responsible for managing the discipline interactions within the organization.

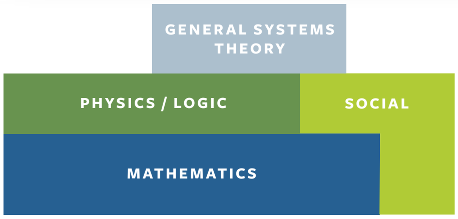
DESCRIPTION: The correspondence of the organization to the system (whether the organizational structure mirrors the system structure or not) is an essential mapping activity in managing the information flow and engineering of the system. The maturity of the engineering organization establishes the need for organizational structure formality. Successful development of a system by organizations inexperienced in that specific system will require structure that is more formal. Seasoned organizations with a specific system can operate successfully with little formal organization. Note that project management and organizational line management are concerned with organizational unit responsibilities and personnel matters. A concern of the systems engineer is how these units interact as part of system knowledge and understanding (system information) flows through the organization. The systems engineer works with project management and line management to resolve identified system information gaps or barriers in the organizational structure, as these gaps and barriers will lead to flaws in system design, manufacturing, and operation. System dynamics models provide an approach to this principle as discussed in "Engineering Elegant Systems: Theory of Systems Engineering". (Watson, Mesmer, and Farrington 2020a)

EVIDENCE: The engineering disciplines each create their building blocks of the system in coordination with other engineering disciplines. For example, system dynamics drive the structural loads. System efficiency increases at the expense of subsystem efficiency. Integrated performance of the system drives the best balance of system performance. Independent subsystem optimization leads to poorer system performance and system efficiency goes down. Appropriate information interchange among the engineering disciplines aids in the recognition of impacts to the overall system balance.

IMPLICATIONS: Systems engineers are responsible for understanding how the organizational structure and culture affect the flow of information about the system. The systems engineer ensures proper interaction between the engineering disciplines as they produce their aspect of the system. Similarly, in operations, the disciplines must work together to ensure consistent and intended system operation and maintenance. Creating a map of this information flow aides in understanding how this flow occurs within the organization. Where difficulties are identified, the systems engineer should discuss potential changes for improvement with project management and organizational (line) management. Adjusting systems engineering process flows may handle some difficult situations. Some may require organizational changes by the project manager or line management. These changes may solve one issue and make another information flow path more difficult in complex organizations. The systems engineer should evaluate each change and strive for the best balance of systems engineering process application with project and line organization structures.

PRINCIPLE 15:

Systems engineering is based on a middle range set of theories.



DESCRIPTION: There are many types of systems simply categorized as physical systems, logical systems, social systems, or some combination. Since there is not a unified theory of physics, nor a unified theory of logic, nor a unified theory of sociology, there is not a unified theory of systems engineering. Instead, systems engineering derives from a set of middle-range theories that form the basis of the system and the engineering of the system. The idea of middle range theories was applied in the development of sociology as a discipline. (Merton 1996) Middle range theories are a collection of theories that define the basis for the discipline. In the absence of unified theories, the middle range set of theories define major discipline concepts and workings. They may also point in the direction of a unified theory, if one exists. Systems engineering has four categories of middle range theories: systems theory, system physics/logic, mathematics, and sociology. None of these categories has been found to have a unified theory. If there were unified theories for these, systems engineering would still be dependent on more than one theory. Thus, middle range theories are essential to the definition of the systems engineering basis. Systems theory exists in various forms, such as general systems theory and system dynamics, and seeks to define the unique system aspects of the system. System theory does not replace the physical, logical, or social basis of a system but seeks to look at the interactions among the different system functions within the system constraints. All of these system theoretical bases have a mathematical underpinning. A mathematical structure that integrates the system, physical, logical, and social aspects of the system provide the mathematical framework of the system. Systems engineering then has four theoretical bases represented in the sub-principles below. These categories are broad systems theoretical basis, system specific physics/logic systems engineering theoretical basis, mathematical basis, and sociological theoretical basis.

EVIDENCE: Systems exist as either physical systems, logical systems, social systems, or some combination of these. These systems incorporate all of the sciences that define their physical, logical, and social nature. Systems theory provides further illumination on the nature of the integrated aspects of the system. Mathematical category theory provides the mathematical definition of a system. Category theory provides the mathematical structure to identify the system theoretical aspects from the physical, logical, and social functions and interrelationships of the system.

IMPLICATIONS: This middle-range set of theory provides a complete basis for the systems engineer to understand a system. The application will be specific to each system (the theories needed for a cyber-system are very different from those needed to build a ship). This structure provides for these differences and allows the systems engineer to incorporate the theories needed to understand both the system and the organization developing or operating the system. The systems engineer does not need expertise to design each component of the system; the system engineer is the expert in how to integrate these components into the intended system. This requires a broad understanding of several disciplines rather than a deep understanding in only one. The systems engineer must communicate clearly among the engineering disciplines, including understanding terminology differences and the use of similar terms to mean something different to a particular discipline. Φ to an optical engineer is the angular frequency of light while to the mechanical engineer working on the same system it means the angular rotational velocity of a component. Systems engineers should translate terminology and not try to enforce commonality among the engineering disciplines.

SUB-PRINCIPLE 15(a):

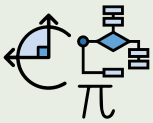
Systems engineering has a systems theory basis.

Systems theory currently has several different forms based on General Systems Theory. (Bertalanffy 1968) (Boulding 1956) (Hammond 2010) There have been various applications such as system dynamics (Forrester 1968), soft system methodology (Checkland 1981), interactive management (Warfield 1976) (Warfield 1994), etc. that seek to define the system aspects of a system. Work has been done to integrate these various theoretical approaches leading to a set of seven axioms of systems theory (Adams et al. 2015; Whitney et al. 2015). These theories form an important part of system engineering and allow for the identification of concepts, such as emergence, as system properties.

SUB-PRINCIPLE 15(b):

Systems engineering has a physical/logical basis specific to the system.

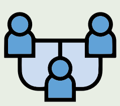
Systems engineering incorporates the fundamental physical and logical concepts specific to the system. Thus, the physical/logical basis of systems engineering incorporates the physical/logical basis of the system. The systems engineer must fully understand that this is different for different types of systems.



SUB-PRINCIPLE 15(c):

Systems engineering has a mathematical basis.

Mathematical category theory (Mac Lane 1971) provides a mathematical structure for systems engineering. A mathematical category provides a definition of a system that provides a structure to incorporate various physical, logical, and mathematical theories into a system representation. Category theory integrates several theories that are important to systems engineering. Systems engineers, in engineering the system, manage information about the system and its interactions, using this information to make development and operational decisions. The laws and relationships defined in information theory govern the information on the system. This also applies to the management of system information through the organization as contained. Note, that information theory has a set theory basis and naturally extends to the construction of a mathematical category. Systems engineers use information to control the system design or system operations that bring in control theory in a broad scope of controlling the information flow about the system and in defining the control methods to control system states within relevant acceptable ranges over time. Category theory provides for the interaction structure to show these control relationships for the system. Statistical engineering is also a significant mathematical tool that allows for systems understanding and accounts for uncertainties and sensitivities. Category theory allows for the absence of details within an element and allows for variations of relationships that support the application of statistics in defining system relationships.



SUB-PRINCIPLE 15(d):

Systems engineering has a sociological basis specific to the organization.

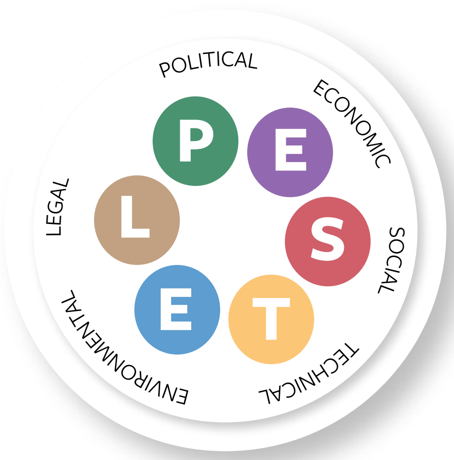
Systems engineering incorporates the fundamental sociological concepts specific to the development and operations organization. Understanding the social structure, culture, and interactions are essential to good information flow within and between organizations. Concepts such as specification of ignorance, common terminology, opportunity structures, role-sets, and the reclama (reconsideration) process are all important sociological approaches that are part of discipline integration. In bringing the disciplines together, the concepts of social ambivalence, social anomie, social dysfunction, insider-outsider behavior, unintended consequences, and the self-fulfilling prophecy are also present within organizations. These sociological principles provide key approaches to understand and manage the information flow through the organization as the disciplines integrate and share their information. This also provides a definition of the key sociological barriers to information flow through the organization (Watson, Andrews, and Larsen 2017; Watson et al. 2017).

SYSTEMS ENGINEERING HYPOTHESES

The hypotheses are statements that research can prove (or perhaps disprove). These statements challenge some of the heuristic notions found in complexity theory and are set in a practical application context (with real boundaries and constraints) rather than in a theoretical infinite context.

HYPOTHESIS 1:

If a solution exists for a specific context, then there exists at least one ideal systems engineering solution for that specific context.



DESCRIPTION: For a given system context that has a system solution, there exists an ideal (optimal or best-balanced) design for the system to accomplish the mission. Budget, schedule, decision timeframe, policy, law, and organizational culture define the context.

EVIDENCE: This hypothesis drives objective research into the question of an optimal system configuration (a best-balanced system). To achieve an optimal design requires having measures of goodness for designs considered as viable solutions to the problem. If a viable solution for the engineering problem exists then it can be defined in terms of a real function f that is variously called an objective function, loss function, cost function, utility function, fitness function or (in certain fields), an energy function or energy functional.

The problem of finding an optimal solution is a mathematically constituted problem of maximizing or minimizing a real function ranging over an allowed set. Optimization systematically chooses input values from within an allowed set and computes the value of the function. A viable solution that minimizes (or maximizes) this function represents an optimal solution (depending on the context).

The Weierstrass Optimization Theorem (also known as the Extreme Value Theorem) (Willis and Finney 2004, pg. 59) states that given a continuous real-valued objective function f_0 (x) where x ∈ X (where X is the variable space or decision space), subject to x ∈ D (were D is the constraint region) then f_0 will attain a minimum and a maximum at least once, that is, there are real numbers c and d in the closed interval [a, b] such that

$$f(c) \geq f(x) \geq f(d) \text{ for all } x \in [a, b] \qquad (5)$$

This implies that if viable solutions exist and there is a principled way of comparing the 'goodness' of designs, then there is a best (and worst) design given the context.

Hamilton's principle (Ginsberg 1998, pp. 282 - 285) directly shows this for a physical system through the relation:

$$\int_{t_1}^{t_2} (\delta T - \delta V + \delta W)dt = 0. \tag{6}$$

Where

δT = infinitesimal change in kinetic energy,

δV = infinitesimal change in potential energy,

δW = infinitesimal change in work, and

t = time defined over the range of the system performing work.

This principle shows there is a minimal path of work that balances the system work parameters in accomplishing its function. Exergy is an expansion of this principle and research on exergy efficiency of a rocket indicates that an optimal system, with an objective of thermodynamic efficiency, defines system efficiency across multiple configurations (Watson 2018a). This result has not previously been achievable in a quantifiable manner. In addition, the system value model offers the ability to define an objective function to optimize the system in each context.

IMPLICATIONS: This hypothesis makes no statement about a global optimum. Rather, this hypothesis states there is a local optimum within the confines of the specific developmental and operational context, such as PESTEL. Note, this means that if this context changes, the local optimum may also change. In the absence of the knowledge of a best balance, the system's development appears as a sociological balance of organizational preferences rather than a true best balance of the system.

HYPOTHESIS 2:

System complexity is greater than or equal to the ideal system complexity necessary to fulfill all system outputs.

DESCRIPTION: In each operational context and decision timeframe, the minimum system complexity required to fulfill all the system outputs is the optimal system complexity, and the complexity of alternative system designs are equal to or greater than the ideal (optimal). Note that this is not a simpler-is-better hypothesis. Minimal complexity involves all aspects of the system as defined by context in the Hypothesis 1 description. Being simple in only one context is not necessarily the system with the minimal complexity. The minimal complexity solution involves a best balance of the system and may lead to some aspects being more complex than alternatives and other aspects being less complex. Systems engineers define the minimal complexity holistically and not based on a subset of system aspects. The definition of system complexity is a much-debated topic.

EVIDENCE: This is similar to the statement of Occam's razor, "plurality should not be posited without necessity" (Duignan 2018). As Albert Einstein is reputed to have said, "everything should be made as simple as possible, but not simpler," that underlines a powerful truth of system modeling and systems engineering.

IMPLICATIONS: This hypothesis asserts that less complexity is preferable for a given context. This also states that a more complex system solution than the optimum can fulfill the system application, but not as elegantly. One must realize that the system complexity necessary to complete all intended outcomes of the system satisfies all its operational needs.

HYPOTHESIS 3:

Key stakeholders' preferences can be represented mathematically.

DESCRIPTION: A system results from a large set of decisions made by decision makers throughout an organization. To analyze a decision, three key elements are necessary: preference, beliefs, and alternatives. Hence, for a systems engineer to understand how an organization arrives at a particular system, an understanding of the set of decisions, each with their elements, is necessary. Each decision maker may have different preferences, beliefs, and alternatives. While each of these elements are challenging to understand, preferences are of particular interest to systems engineering as they relate to desired system goals. If different preferences are being used to make decisions on a system, then those decisions would be inconsistent with each other, meaning it is possible that given the same beliefs and alternatives decision makers may decide on different solutions. To enable consistent decision-making throughout the organization, systems engineers must elicit, represent, and communicate preferences of key stakeholders to drive to outcomes that the key stakeholder prefers. A mathematical representation supports the modeling of the preferences and enables analysis of the differences and commonalities in the preferences of different stakeholders.

EVIDENCE: Many systems engineering approaches use a representation of preference to guide decision-making. Goals in goal function trees, objective functions in multidisciplinary design optimization, payoffs in game theory (von Neuman and Morgenstern 1953), and utility functions in value-based engineering are just a few examples of mathematical representations of preferences used in systems engineering approaches. The premise of these approaches is that preferences are mathematically representable and enable a rank ordering of alternatives. Based on these examples, system engineers can create a mathematical function that rank orders alternatives in the same way that a preference does. Decision Theory also uses mathematical functions to rank order alternatives as an individual with their preference would and is widely advocated as a rigorous approach to design and systems engineering.

IMPLICATIONS: The accurate representation of stakeholder preferences enables the systems engineer to assess how well the system fulfills these preferences as the system progresses through its lifecycle. While system value modeling assumes a mathematical representation of preference exists, accurately representing preferences mathematically is still a significant challenge. The elicitation and formation of mathematical representations must become a significant task undertaken by systems engineers to adopt these approaches. Beyond enablement of approaches that strive to find the best system, mathematical representation of preferences also enables meaningful validation of the system. Mathematical representations of preferences allow comparison of the system characteristics with the stakeholder's preferences, answering the validation question: "does the system meet the stakeholder's intent."

ACKNOWLEDGMENTS

CONTRIBUTORS: Michael Watson – Chair, Bryan Mesmer, Garry Roedler, David Rousseau, Javier Calvo-Amodio, Chuck Keating, William D. Miller, Scott Lucero, Rob Gold, Cheryl Jones, David Long, R. W. Russell, and Aileen Sedmak.

REFERENCES

Adams, K. M., P. T. Hester, J. M. Bradley, T. J. Meyers, & C. B. Keating. 2014. "Systems Theory as the Foundation for Understanding Systems." Systems Engineering, 17(1): 112–123.

Adamsen II, Paul B., 2000. "A Framework for Complex System Development", chapter 7, CRC press.

Austin-Breneman, J.; Yu, B.Y.; and Yang, M.C., 2014. "Biased Information Passing Between Sub-Systems Over Time in Complex System Design," Paper Presented at the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Biennial International Conference on Dynamics for Design, and 26th

Austin-Breneman, J.; Yu, B.; and Yang, M.C., 2015. "Biased Information Passing Between Subsystems Over Time in Complex System Design," ASME Journal of Mechanical Design, Vol. 138, No. 1, November 2015.

Bertalanffy, L. 1968. General System Theory: Foundations, Development, Applications. Braziller, NY.Burnham, K. P., and D. R. Anderson. 2002. Model Selection and Multimodal Inference. 2nd ed. New York, US-NY: Springer.

Boulding, K. E. 1956. General systems theory—the skeleton of science. Management science, 2(3), 197-208.

Calvo-Amodio, J., & Rousseau, D. (2019). "The Human Activity System: Emergence from Purpose, Boundaries, Relationships, and Context". Procedia Computer Science, 153, 91-99.

Checkland, P. B. 1981. Systems thinking, systems practice John Wiley & Sons. New York.

Constant II, E. W., 1993. in The Social Construction of Technological Systems, ed. by Bijker, W. E., Hughes, T. P., & Pinch, T., MIT Press, 4th Printing, 1993, pp 221-242,

Componation, P. J., M. Dorneich, H. Guiping, P. A. Farrington, and J. L. Hanson. 2013. "Systems Engineering and Project Success in Government and Commercial Organizations." Paper presented at the American Society for Engineering Management International Annual Conference, Minneapolis, US-MN, 3-5 October.

Cutler, William. 1997. Presented at INCOSE Principles WG Session, Tuesday, August 5th, 1997, Los Angeles, CA

Defoe, J.C., ed. 1993. "National Council on Systems Engineering: An Identification of Pragmatic Principles, Final Report." Report, Systems Engineering Practice Working Group, Subgroup on Pragmatic Principles, NCOSE WMA Chapter.

DEG (Defence Engineering Group). (2002). The Defence Systems Engineering Handbook, University College London, p.12.

Duignan, B. 2018. "Occam's Razor." https://www.britannica.com/topic/Occams-razor..

Forrester, J. W. 1968. Principles of Systems. Productivity Press. Portland, OR.

Frosch, R.A. 1993. "A Classic Look at Systems Engineering," in Readings in Systems Engineering, F.T. Hoban and W.M. Lawbaugh (eds.), NASA SP–6102, Washington, DC, 1993.

Ginsberg, J. H. 1998. Advanced Engineering Dynamics. Cambridge University Press, Cambridge, UK.

Griffin, M.D. 2010 "How Do We Fix System Engineering?" 61st International Astronautical Congress, Prague, Czech Republic, September 21–October 1.

Halligan, Robert. 2002. Project Performance International Systems Engineering, Systems Engineering Principles.

Hammond, D. 2010. The science of synthesis: Exploring the social implications of general systems theory. University Press of Colorado.

Hitchins, D. 1992. Putting Systems to Work. John Wiley & Sons: Chichester, UK.

Katina, P. F. 2016. “Systems Theory as a Foundation for Discovery of Pathologies for Complex System Problem Formulation.” In Applications of Systems Thinking and Soft Operations Research in Managing Complexity, 227–267. Cham, CH: Springer.

Keating, C. B., P. F. Katina, R. Jaradat, J. M. Bradley, and A. V. Gheorghe. 2017. “Acquisition System Development: A Complex System Governance Perspective.” Paper presented at the 27th Annual INCOSE International Symposium, Adelaide, AU, 15-20 July. doi:10.1002/j.2334-5837.

Keating, C. B., P. F. Katina, A. V. Gheorghe, R. and Jaradat. 2016. “Complex System Governance: Advancing Prospects for System of Systems Engineering Applications.” Paper presented at the 13th Annual Acquisition Research Symposium, Monterey, US-CA, 5 May.

LBNL /Pub-3164SE, 2001. A Systems Engineering Primer for every engineer and scientist, pp. 6-7.

Mac Lane, S. 1971. Categories for the Working Mathematician. New York, US-NY: Springer-Verlag.

Maier, Mark W. and Rechtin, Eberhardt, (2002). The Art of Systems Architecting 2nd ed. CRC , Appendix A “Heuristics for systems-level architecting”

Mobus, G. E., and M. C. Kalton. 2015. Principles of Systems Science. New York, US-NY: Springer.

Merton, R. K. 1996. “On Sociological Theories of the Middle Ranges” in On Social Structure and Sci¬ence, P. Sztompka (ed.), Univ. of Chicago Press, Chicago, IL, pp. 41–50.

von Neumann, J., and O. Morgenstern. 2004. Theory of Games and Economic Behavior. 6th ed. Princeton, US-NJ: Princeton University Press.

Oxford Dictionary. 2018. Definition of Principle. www.oxforddictionaries.com.

Oxford College of Marketing. 2016. “What is a PESTEL analysis.” https://blog.oxfordcollegeofmarketing.com/2016/06/30/pestel-analysis/.

Pratt, J., Cook, S. C. 2017. “A Principles Framework to inform Defence SoSE Methodologies.” In 15th Annual Conference on Systems Engineering Research Disciplinary Convergence: Implications for Systems Engineering Research. eds., Madni, A. M., Boehm, B., Erwin, D., A., Ghanem, R., Wheaton, M. J., Redondo Beach, US-CA, 23-25 March.

Rousseau, D. 2018a. “Three General Systems Principles and Their Derivation: Insights from the Philosophy of Science Applied to Systems Concepts.” In Disciplinary Convergence in Systems Engineering Research. Cham, CH: Springer.

–––. 2018b. “On the Architecture of Systemology and the Typology of Its Principles.” Systems, 6 (1): 7.

–––. 2018c. “A Framework for Understanding Systems Principles and Methods.” Paper presented at 28th Annual INCOSE International Symposium, Washington, US-DC, 7–12 July.

Rousseau, D., Pennotti, M., Brook, P. 2022. Systems Engineering’s Evolving Guidelines. Report of the INCOSE Bridge Team, presented to the INCOSE Systems Science Working Group on January 31, 2022. Published on the SSWG’s Google Drive. Retrieved March 30, 2022, from https://drive.google.com/file/d/1JibL44sUh0ztefZQ5Rfy4kGiodXIy63n/view

Senge, P. M. 1990. The Fifth Discipline: The Art and Practice of the Learning Organization. London, GB: Random House.

Sillitto, H. 2014 Architecting Systems. Concepts, Principles and Practice. London, GB: College Publications.

Warfield, J. N. 1976. Societal Systems, Planning, Policy and Complexity.New York: Wiley Interscience

Warfield, J. N., & Cárdenas, A. R. 1994. A handbook of interactive management (p. 338). Ames: Iowa State University Press.

Watson, M. D., M. Griffin, P. A. Farrington, L. Burns, W. Colley, P. Collopy, J. Doty, S. B. Johnson, R. Malak, J. Shelton, D. R. Utley, M. C. Yang, and Z. Szajnfarber. 2014. “Building a Path to Elegant Design,” Paper presented at the American Society for Engineering Management 2014 International Annual Conference, Virginia Beach, US-VA, 15-18 October 15-18.

Watson, M. D., and P. Farrington. 2016. "NASA Systems Engineering Research Consortium: Defining the Path to Elegance in Systems." Paper presented at the 16th Annual Conference on Systems Engineering Research, Huntsville, US-AL, 22-24 March.

Watson, M. D., J. G. Andrews, and J. A. Larsen. 2017. "Principles of Sociology in Systems Engineering," Paper presented at the 2017 International Annual Conference of the American Society for Engineering Management, Huntsville, US-AL, 18-21 October.

Watson, M. D., J. G. Andrews, J. C. Eckley, and M. L. Culver. 2017. "Practical Application of Sociology in Systems Engineering," Paper presented at the 2017 International Annual Conference of the American Society for Engineering Management, Huntsville, US-AL, 18-21 October.

Watson, M. D. 2018a. "System Exergy: System Integrating Physics of Launch Vehicles and Spacecraft." AIAA Journal of Spacecraft and Rockets 55 (2): 451-461.

Watson, M. D. 2018b. "Engineering Elegant Systems: Systems Engineering Postulates, Principles, and Hypotheses Related to Systems Principles". Paper presented at the International Society for the System Sciences (ISSS) Annual Meeting, Corvallis, US-OR, 22-27 July.

Watson, M. D. 2018c. "Engineering Elegant Systems: Postulates, Principles, and Hypotheses of Systems Engineering." Paper presented at the AIAA Complex Aerospace Systems Exchange 2018, Future of Systems Engineering Panel, Orlando, US-FL,17-19 September.Willis, R., Finney, B. A. 2004. Environmental Systems Engineering and Economics. Springer Science+Business Media. US-NY.

Watson, M. D., B. Mesmer, and P. Farrington. 2018. "Engineering Elegant Systems: Postulates, Principles, and Hypotheses of Systems Engineering." Paper presented at the 16th Annual Conference on Systems Engineering, Charlottesville, US-VA, 8-9 May.

Watson, M. D., Mesmer, B., Roedler, G., Rousseau, D., Gold, R., Calvo-Amodio, J., Jones, C., Miller, William D., Long, D., Lucero, S., Russell, R. W., Sedmak, A., Verma, D., 2019. "Systems Engineering Principles and Hypotheses", INSIGHT Magazine, 21:1, Hoboken: Wiley (pp. 18-28).

Watson, M. D., B. Mesmer, and P. Farrington. 2020a. "Engineering Elegant Systems: Theory of Systems Engineering." NASA_TP_20205003644, NASA, Washington, D.C., August 2020.

Watson, M. D., B. Mesmer, and P. Farrington. 2020b. "Engineering Elegant Systems: The Practice of Systems Engineering." NASA_TP_20205003646, NASA, Washington, D.C., August 2020.

Webster's New World Dictionary, Third College Edition, ed. Neufeldt, V., Guralnik, D. B., Simon & Schuster, NY, 1988, pg. 1055.

Whitney, K., Bradley, J. M., Baugh, D. E., & Chesterman Jr., C. W. (2015). Systems theory as a foundation for governance of complex systems. International Journal of System of Systems Engineering, 6(1–2), 15–32.

Systems engineering principles have been percolating in the systems engineering community for 30+ years.

Based on the work done these past three decades, INCOSE has produced this first formal set of systems engineering principles peer reviewed by our sister organizations: AIAA, IEEE, and NDIA. These principles are not the final set but an initial set to help advance the discipline of systems engineering in application of the systems engineering processes, provide an indication of the basis of systems engineering, and spur further systems engineering research. INCOSE is excited to provide a further step in the advancement of the Systems Engineering discipline through the publishing of this first set of principles.

DR. JAVIER CALVO-AMODIO (Associate Professor, Industrial Engineering, Oregon State University) contributed to the development and applicability of the principles. Dr. Calvo-Amodio significantly contributed conceptual and theoretical foundations that support the validity of the systems principles.

ROB GOLD contributed to the development and review of the principles.

CHERYL JONES (Systems Engineer, US Army RDECOM) contributed to the development and review of the principles.

DR. CHUCK KEATING (Professor, Engineering Management and Systems Engineering, Old Dominion University) contributed to the development and critique of the principles. Dr. Keating significantly contributed to the underlying Systems Theory foundations embedded in the principles.

DAVID LONG (INCOSE Past President, CEO ViTech) was instrumental in orchestrating the initial INCOSE discussions on the Systems Engineering Principles emerging from literature in 2018.

D. SCOTT LUCERO (Research Faculty, Virginia Tech National Security Institute) provided early guidance that influenced development and applicability of the principles.

DR. BRYAN MESMER (Associate Professor, The University of Alabama in Huntsville) contributed to the overall consistency of the principles. Dr. Mesmer significantly influenced the decision-making aspects of the principles.

WILLIAM D. MILLER – MR. MILLER (Adjunct Professor, Stevens Institute of Technology; Editor-in-Chief, INSIGHT magazine; and 2013-2014 INCOSE Technical Director) ensures the principles are and remain fit for purpose as the keystone of the hard and soft sciences foundations (SF4SE) for the systems community's future of systems engineering (FuSE) initiative.

GARRY ROEDLER (INCOSE Past President, INCOSE Fellow & Retired Senior Fellow, Lockheed Martin) contributed to the development and review of the principles, and promoted the project across INCOSE and collaborating organizations.

DR. DAVID ROUSSEAU (Director, Centre for Systems Philosophy, INCOSE Fellow) contributed to the refinement and consolidation of the principles. Dr Rousseau significantly contributed to the conceptual clarity of the principles and the mapping of the principles to align inputs from across the published literature.

R. W. RUSSELL contributed to the development and review of the principles.

AILEEN SEDMAK contributed to the development and review of the principles.

DR. MICHAEL D. WATSON (NASA MSFC Advanced Concepts Office Technical Advisor) led the development, review, and maturation of these systems engineering principles at both NASA and as chair of the INCOSE Systems Engineering Principles Action Team.

INCOSE
International Council on Systems Engineering

A better world through a systems approach