SE Management is Not SE Core Competency: Time to Shift this Outdated, 60-Year-Old Paradigm

Charles Wasson
INCOSE Fellow and ESEP
Wasson Strategics, LLC
(330) 685-0850
office@wassonstrategics.com

Abstract. In the early 1950's the development of large, complex systems encountered two major challenges: (1) traditional Engineering methods were inadequate for coordinating and communicating designs and changes across multiple disciplines; and (2) projects were incurring unmanageable technical failures, cost overruns, and schedule slips. Exacerbating these challenges were growing conflicts between management and the engineers and scientists performing the engineering. These two challenges manifested themselves in the form of a "management gap," which emerged due to management frustrations with engineers and scientists' inability to articulate how the engineering process was performed, and (2) a "technology gap," which emerged due to engineers and scientists' frustrations with management's inability to understand how engineering was performed and the new technologies being implemented.

Central to these issues was the threat to the prevalent 1950's management paradigm of exercising authoritative control over subordinates by planning, organizing, staffing, directing, and controlling the tasks engineers and scientists performed. Corrective action solutions were urgently needed. Rather than solving the challenges, government as the acquirer of large, complex systems, decided to regain authoritative control over its contractors. As a result, the concept of Systems Management was introduced and mandated via a series of Systems, SE, and Engineering Management process standards.

Over the past 60+ years, the emerging field of Systems Engineering (SE), which originally focused on answering a key engineering question "Will the system work – i.e., 'be fit for purpose' when realized*?* (Ring, 2017) shifted to *"*did we follow our processes?" Projects corrected a "management" problem while neglecting the "engineering" question. As a result, projects continue to exhibit systemic performance issues. It is time to shift this outdated Systems Management paradigm and reestablish SE core competency as the "engine" for correcting SE contributions to project performance issues that seem so *intractable*.

Introduction

Despite significant investments over the past 30 years by industry, government, academic research, professional societies, and standards organizations, projects continue to have *limiting* degrees of success correcting project performance technical, cost, and schedule issues. To deal with the issues, these organizations pursued initiatives such as documenting Organizational Standard Processes (OSPs), Capability Maturity Model Integration (CMMI) assessments, ISO 15288 compliance, INCOSE System Engineering Professional (SEP) certifications (CSEP), SE handbooks, and so forth. These initiatives are a *necessary* but *insufficient* condition for project success. They address management oversight solutions to a technical problem, the lack of SE core competency. Referring to a quote attributed to Dr. Albert Einstein, "Insanity is continuing to do the same thing expecting a different result."

Methodology

The methodology for this paper consisted of a literature review of the topic. Based on the review, a Problem Statement was formulated and served as a basis for collecting actual case studies from industry and government based on the experiences of the author and others. Since SE Core Competency serves as the benchmark frame of reference for improvements, the author developed an architectural framework depicting the structural elements of a SE competency. To complement the architectural framework, a systemigram was developed to understand the factors that contribute to project performance issues and the role of SE core competency within the chain of performance effecters. Next, empirical data observations made over the past 25+ years by the author provided inputs for a Pareto diagram illustrating the low SE competency maturity level in typical projects. Analysis of the observations led to a key question: *Why are the SE competency maturity levels low despite claims to the contrary by enterprises?*

To answer this question, the author conducted literature research and personal interviews to understand how the Systems Management's process focus diverted attention from a key 1950s-1960s era engineering question: "*Will the system work – i.e., be fit for purpose – when realized?*" Since project-performance issues are multi-faceted, industry, government, academic, and other "influencers" were identified and analyzed to understand their contributions. This led to a final question: "*How do we correct the overwhelming imbalance between SE Management and SE Core Competency?* As a multi-faceted problem, corrective actions are provided for orchestrating change-management actions.

Problem Statement

Projects continue to exhibit systemic performance issues despite the introduction of Systems Engineering and Systems Management into large, complex system development in the early 1950's to correct engineered system technical failures and manage new technology risk.

Framing the Problem Space

To understand the context of the Problem Statement, let us begin our discussion with a few mini-case studies.

Mini-Case Study #1 – Doing All the Right Things Expecting a Different Result

Assume a customer schedules a site visit to a potential vendor's facility as shown in Figure 1. A "meet and greet" meeting is scheduled for the vendor's Executive Conference Room. An executive makes a presentation promoting their outstanding SE organizational capabilities such as documented processes, a CMMI Appraisal Rating of (1 – 5), ISO 9001 certification, ISO 15288 compliance, XX INCOSE CSEPs, YY personnel with MS degrees in SE, and ZZ with PhDs degrees in SE. Sounds impressive to some customers, but not to others.

From the back of the conference room a customer's voice breaks the rhythm of the presenter's presentation and says, "That's fine, but why do your projects continue to exhibit technical compliance, cost, schedule, and risk-performance issues project after project?"

Figure 1. Familiar scenario in industry.

After "doing the right things" – e.g., documented OSPs, ISO 15288 compliance, CMMI assessments, SEP certifications, etc. – that are intended to provide some level of confidence of the enterprise's SE capability to perform, *what is going on within these projects that continue to plague projects that lead to performance issues?"* Mini-Case Study #2 provides some insights concerning what occurs within a project.

Mini-Case Study #2 – Collaborative Engineering Team Development Environments

Referring to Figure 2, enterprises and projects perform to standards that include ISO 15288 Systems and Software Engineering – System Lifecycle Processes and the INCOSE SE Handbook SEHv4 (2015) - *A Guide for Lifecycle Management Processes and Activities*. As their titles indicate, these are management "process" standards. Enterprise command media typically requires development of Systems Engineering Management Plans (SEMPs) and others that tailor this guidance applicable to each project for application by multi-discipline Integrated Product Teams (IPTs), Product Development Teams (PDTs), and others.

Typically, each team consists of multiple engineering disciplines – e.g., hardware engineers such as electrical engineers (EEs) and mechanical engineers (MEs); software engineers (SwEs); specialty engineers such as human factors safety, reliability, maintainability, and others; quality assurance (QA) and software QA (SQA); manufacturing; procurement; and others. These engineers were educated to perform their respective engineering practices and they generally do so competently. The problem is they typically lack a *common* problem-solving and solution-development methodology, semantics, or decision-making methods required for today's collaborative, interdisciplinary team environments.

Engineers "engineer" their portion of a system or product based on (1) what they were educated and trained to accomplish, (2) tasks they are assigned, and (3) their Knowledge, Skills, and Abilities (KSAs). As a result, team decision-making is often a conglomeration of engineers with varying interdisciplinary KSAs (Figure 5) working together to perform an *ad hoc, endless loop,* Specify-Design-Build-Test-Fix (SDBTF) Engineering Process of activities that are often *chaotic, ineffective and inefficient*. Two adages characterize these activities: (1) "Every system (team) is perfectly designed to produce the results you are observing" (Figure 5) and (2) "Insanity it continuing to do the same thing over and over expecting a different result" (attributed to Einstein).

Process compliance to contract specification requirements is a *necessary* but *insufficient* condition for ensuring that a system, product, or service will satisfy stakeholder operational needs such as Ring's (2017) critical engineering question. However, as Wasson (2016, p. 24) observes, attempting to structure engineering development into a series of production line tasks to satisfy a managerial technical competency issue leads to "Paint-by-Number" engineering. That is, if you "paint" within the constraints of the process lines, the resulting system and its work product(s) will be engineering masterpieces that satisfy the customer's operational needs.

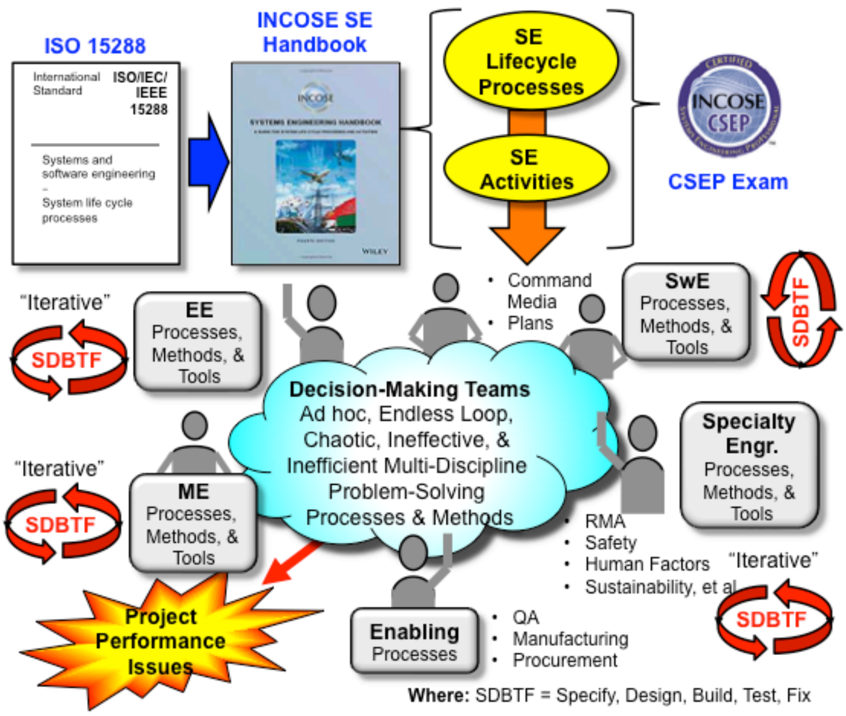


Figure 2. Understanding the SE challenges in today's interdisciplinary team environments.

Now, observe the SDBTF iterations, which refer to a common Specify-Design-Build-Test-Fix (SDBTF) engineering paradigm (Wasson, 2010, 2012, and 2016). SDBTF is a traditional *ad hoc, activity-based*, *endless-loop,* engineering paradigm derived from the Scientific Method (Wasson, 2016, pp. 295 – 296). Observe the terms *endless loop* and *activity-based* in the preceding sentence. Wasson (2016, p. 250) notes that if you reward engineers for activities, you get (endless) activities – e.g. Verify Subsystem #1. If you reward engineers for outcomes, you get outcomes – e.g., Subsystem #1 Verified." These are two different mindsets that illustrate how the focus on Systems Management processes and activities has contributed to project overruns and schedule slips.

Observe the SE Activities flow-down arrow in the upper center of Figure 2. Students learn the Scientific Method in public school, graduate, and move into higher education engineering programs. Higher education learning environments are often based on a traditional classroom-laboratory Plug and Chug – Design-Build-Test-Fix (DBTF) educational model. Since higher education includes science-based research, a student's Scientific Method methodology fits well into educational environments.

After graduation, graduates migrate into industry and government where a more complex Plug and Chug SDBTF Paradigm has thrived for decades. There, engineers naturally apply their educational Plug and Chug DBTF methodology to system development projects. System acquirers typically award contracts to "engineer" systems, not conduct research-based science projects. They expect you to deliver systems, products, or services based on your convincing proposal that lead to your enterprise's selection. Therefore, *why would you apply a research-based methodology* to a *project requiring a problem-solving and solution development methodology to engineer* systems? The problem is uninformed enterprises, executives, and managers often hear that the SE Process is "iterative and recursive." Since their defacto SDBTF paradigm is also "iterative and recursive," they must be performing SE. When the project experiences performance issues, SE is blamed (for their erroneous misperception).

Mini-Case Study #2 provides insights into how many enterprises perform SE (SDBTF engineering). The question is: *How do Project Managers (PMs) and engineers, who work in these environments, feel about (SDBTF) SE*? That brings us to Mini-Case Study #3.

Mini-Case Study #3 – Project Customer and Personnel Observations

Employees and consultants with industry and government note that customers, PMs, and engineers often privately voice their frustrations about project performance:

1. PMs observe: "Engineers can never finish a design on time or within budget! They are always 'tinkering' with the design! (SDBTF Engineering.) SEs are just coordinators! Why does my project have to pay for their paperwork?"
2. Executives assure customers: "We will work (nights, weekends, and holidays … whatever it takes) to get this system back on schedule using our SE (SDBTF Engineering) Process."
3. Engineers ask: "Why is everyone designated as a 'Systems Engineer'" by their manager *irrespective* of their requisite qualifications?

Technically, engineers embrace the concept of SE; however, they are not "buying" the administrative "selling" points promoted by professional societies. (Emes et al., 2005, p. 178) reinforce this point and cite Cowper and Smith (2003) who "identify the key barriers to promoting and 'selling' systems engineering as: the lack of SE awareness and understanding, the lack of a clear message about what SE is or is not, the confusion over the Systems Engineer's skill set, the need for a business case for SE, and the management of implementation risks."

Let us be clear, compliance to SE Management processes are a *necessary* condition for technical planning, assigning tasks and accountability, and tracking status, progress, and risk. However, processes are nothing more than navigational roadmaps based on best practices and lessons learned with outcome-based guideposts. Recognize that competent SEs and engineers performing to the processes produce and deliver systems that answer Ring's (2017) question, not processes. To illustrate the point, automobile roadmaps provide the flexibility to travel long distances between points A and B; however, driver competency determines how safely, efficiently, and effectively they can navigate the pathway options, hazards, and constraints to arrive on time. So, *what are the attributes of an SE core competency*?

Attributes of a SE Core Competency

Numerous authors have studied and authored papers on SE competency over many years. The topic has evolved into a series of SE Competency Frameworks and classifications such as Whitcomb, et al (2017); Gelosh et al (2017); and Grady et al (2011). Although these *evolving* frameworks continue to *converge*, the authors' (Wasson) intent is not to create another framework. What is important is to establish a framework to serve frame of reference for discussions in the remainder of this paper.

SE core competency, which is based on Knowledge, Skills, and Abilities (KSAs), consists of two primary elements as shown in Figure 3: SE core knowledge and SE application knowledge.

- SE Core Knowledge requires a strong educational course in SE concepts, principles, and practices. The course addressed here is not your typical abstract, high-level, SE management and acquisition course mislabeled as "Systems Engineering."
- SE Application Experience requires SE Application Experience *tempered* over a *minimum* of 25+ years of experience performing SE on many small to large, end-to-end projects in advancing leadership positions. Since SE Application Experience is abstract and vague, we need to further refine its attributes.

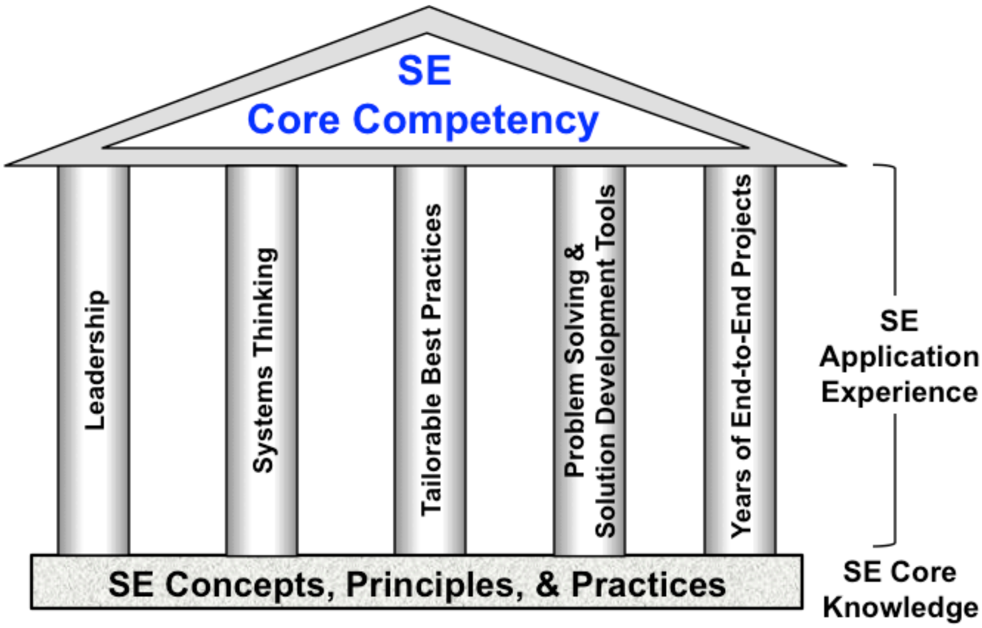


Figure 3. SE core competency attributes.

Observe the subtleties and relationships of the terms, SE Core Knowledge versus SE Application Experience, in terms of achieving SE Competency.

SE Core Competency = *f* [SE Core Knowledge (KSAs) + SE Application Experience]

As noted earlier in Mini-Case Study #3, SE is perhaps one of the most overused titles in engineering; "everyone is an SE." In many cases SEs are typically traditional discipline engineers – i.e., EEs, MEs, et al - and non-engineering professionals who exhibit *systems thinking* skills. If you investigate their KSAs, many have misperceptions of SE and exhibit discontinuities and voids in their knowledge that have been derived experientially over time. We will address this point later. This is not the fault of those engineers; the inconsistency reflects the lack of a bona fide SE course in undergraduate engineering education and industry and government's *misperception* of SE.

The preceding discussion provides an architectural framework depicting the elements of SE core competency. But *how does SE core competency relate to project performance issues*?

Understanding the SE Core Competency – Project Performance Links

Solving the SE core competency issue requires more than simply scoping and defining the term. Since understanding its relationship to project performance issues is key to our discussion, Figure 4 illustrates the linkages of SE Core Competency as the repository of an enterprise's SE KSAs to SE education and training to the project performance issues. To better understand how an enterprise's SE Core Competency KSAs impact project performance issues, we need to understand the state of SE in a typical workplace as discussed in Mini-Case Study #3.

The State of SE "Competency" in the Workplace

From an Organizational Development (OD) perspective, Wasson assessed the SE core knowledge of engineers, managers, and executives across a variety of business domains over a period of 30 years. Several key indicators emerged concerning SE KSAs related to job performance. The indicators consist of: SE awareness, SE concepts, SE principles, SE practices, SE definitions, SE application knowledge, and SE knowledge assimilation. Regarding the last two indicators:

- SE Application Knowledge encompasses the ability to effectively apply SE concepts, principles, and practices.
- SE Knowledge Assimilation encompasses the ability to visualize an entire system and its *dynamic interactions* (Griffin, 2010, p. 2) as the basis for performing technical trade-offs and workarounds, identifying sources of system defects such as: design flaws, deficiencies, and issues; faults and anomalies; and eliminating/reducing defects.

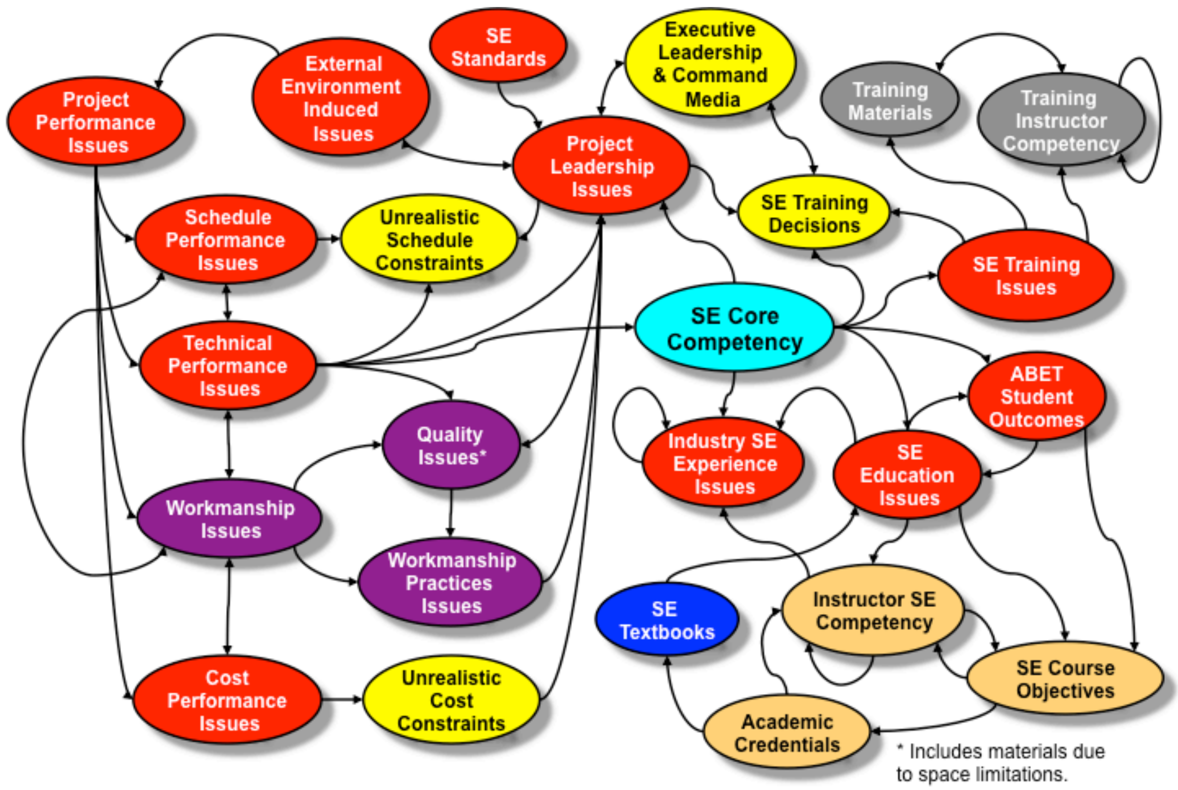


Figure 4. Systemigram tracing project performance issues to SE Core Competency, Enterprise and Project Leadership, and SE Education and Training.

Figure 5 provides a Pareto chart illustrating maturity level results across the SE Competency indicators; your enterprise experiences may be different. Observe the left to right ranking of the maturity level magnitudes beginning with SE Awareness, which serves as a gating filter for assessing the remaining indicator levels.

Two key observations:

1. If everyone is a SE as indicated in Min-Case Study #3, you would expect the maturity level to be a 10 (1 = Low; 10 = High) across all indicators. Obviously, that is not the case. Observe that Figure 5 illustrates SE competency maturity level magnitudes but does not reflect the breadth – e.g., average quantity or percentage of SEs – who could competently provide examples for a specific indicator. These results will provide the basis for our Figure 9 discussion later.

2. Lower indicator scores to the right of SE Concepts in Figure 5 illustrate KSA weaknesses that contribute to technical, cost, and schedule performance issues (Types I and II later in Figure 6.)

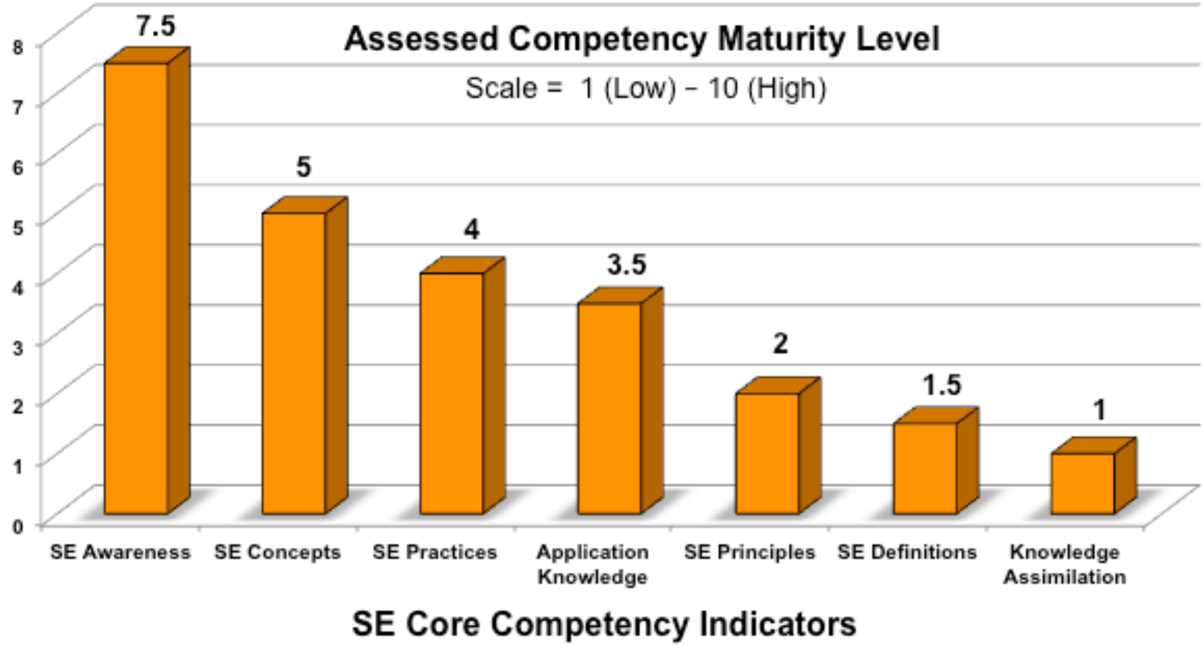


Figure 5. SE Core Competency Indicators - Author's Personal Assessment of Engineering Teams over 30 years on Small to Large, Complex System Development Projects.

The empirical data observations of Figure 5 suggest the general lack of SE core competency in many enterprises. Yet, many of these enterprises have "done all the 'right things'" – i.e., documented OSPs, CMMI, ISO 15288, INCOSE CSEPs, and so forth. In other words, an organization claiming to perform SE that exhibits low SE competency maturity levels such as Figure 5 lacks the SE core competency to answer Ring's (2017) critical engineering question. This is not to say that the project is incapable of developing a successful system based on *ad hoc, endless loop*, SDBTF engineering that is *inefficient* and *ineffective* and working nights, weekends, or holidays to deliver the system. In contrast, true SE core competency is based on integrated, interdisciplinary team decision-making applying common SE concepts, principles, and practices with minimal rework resulting in a higher probability of delivering a technically compliant system on-time and within budget.

You may ask: *Where is the SE core competency "bench strength" in Figure 5 required to respond to Ring's (2017) critical Engineering question:* "Will the system work – e.g., fit for purpose – when realized*?"* Based on evaluating SE organizations and teams over the past 30 years, enterprises often unwittingly presume that if they levy process standards compliance on the decision-making teams (Figure 2), who often lack a bona fide SE fundamentals course and exhibit the SE core competency indicator results shown Figure 5, magic will happen and projects will be successful.

So, how do the engineers and SEs in the Decision-Making Teams (Figure 2) acquire their SE KSAs? In general, SE core knowledge comes from two primary sources (Wasson, 2016, pp. 38 - 39).

- Formal SE Education – Theoretically, in a structured learning environment, Engineering students learn the *whats, whys, how tos, when tos,* and *where tos* for a given topic or situation.
- Experiential Learning – Occurs via personal initiatives such as mentoring and studying handbooks, textbooks, technical papers, and "Go Do" managerial tasking – e.g., "Go Do" a plan or "Go Do" a specification. In those situations, Engineers learn the *whats* and *how tos*; however, what is missing are the *whys*, *when tos,* and *where tos*. Many times critical technical decision-making depends on understanding the *whys*, *when tos,* and *where tos* to avoid negative outcomes and consequences.

The SE competency indicator results shown in Figure 5 reflect a convolution of team member KSAs that evolved from formal SE education and experiential learning.

Distilling the preceding discussions of SE core competency-- *what is the root cause that drives project technical, cost, and schedule performance issues*?

Summarizing Why Projects Continue to Exhibit Performance Issues

Based on the preceding SE core competency discussions, let us summarize the state of SE's contributions to project performance issues. Figure 6 illustrates our discussion. The matrix at the bottom of the graphic provides a foundational starting point.

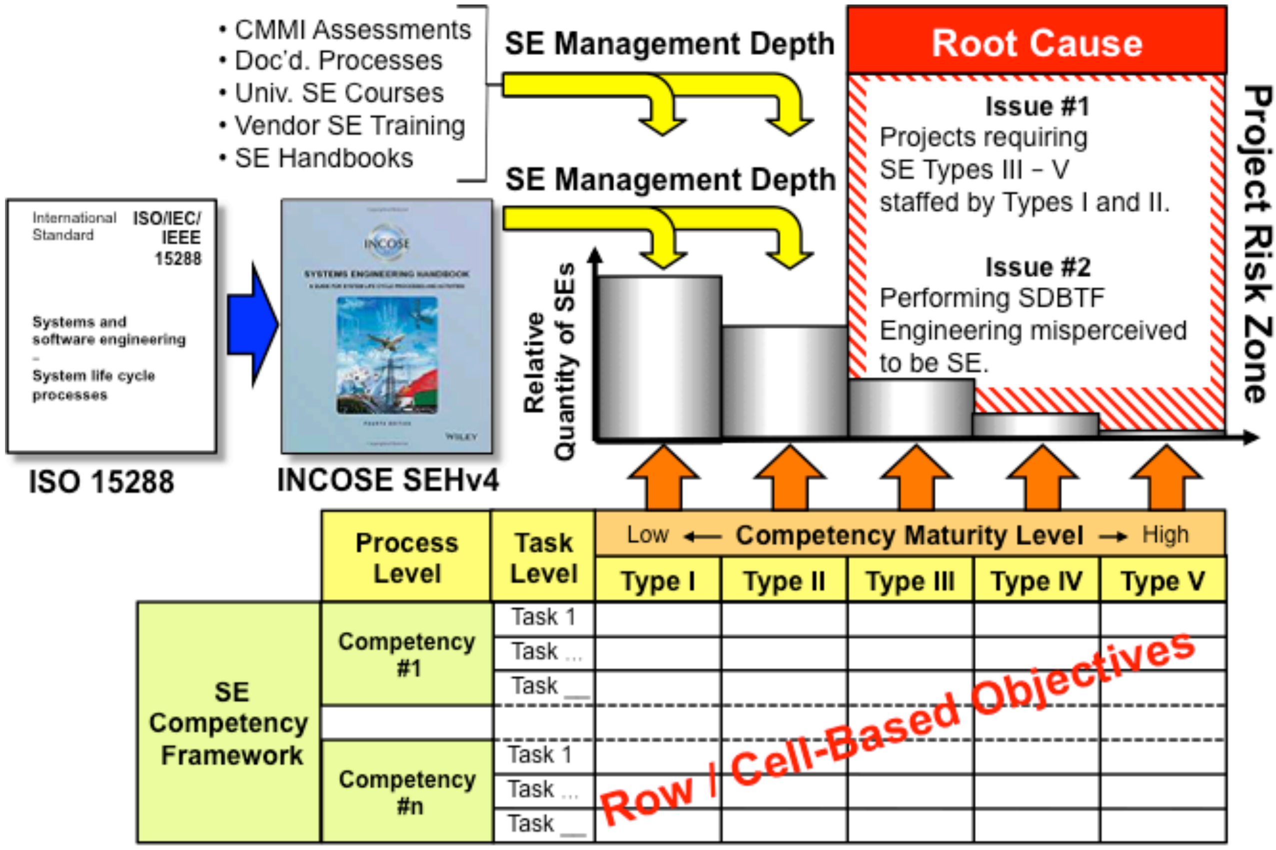


Figure 6. Relationships among SE Competency Frameworks, Kasser, et al (2009) five types of SEs, ISO 15288, the INCOSE SE Handbook, SE education, and project performance issues.

Whitcomb (2017); Gelosh et al. (2017); and others have developed SE Competency Frameworks that include a list of SE *competencies*. For example, the DoD SE Competency Framework consists of 39 competencies such as Stakeholder Requirements Definition, Requirements Analysis, etc. (Whitcomb, 2017). However, here is the challenge: unless the competency frameworks expand the competency topics to lower levels, we have not solved the issue. In fact, each of the *competencies,* which represent *activities* that SEs are expected to be capable of performing repeatedly and predictably, is characterized by a *process* consisting of *tasks* and *subtasks*.

One of the concerns about these frameworks is ensuring that they do not become a solution justifying the current state of SE. For example, the current, outdated view of SE is: (1) write (versus develop) specification requirements, (2) trace requirements (to what?), (3) conduct specification review (grammar and outline checks versus engineering), and (4) verify the system requirements (versus stress the system). These are high-level "procedural" checklist actions that do not ensure "engineering" is being performed. SE core competency requires a mental focus on "*system* engineering of the system" using processes based on best practices and lessons learned, not robotic check-off performance of a SE Management process. Consider the following example.

Griffin (2010, p.2) expressing a personal opinion states "While at its core systems engineering is concerned with interfaces between separable system elements, it should be realized that the more important concerns the dynamic behavior between the elements, not the numbers in the Interface Control Document (ICD)." This example illustrates how the focus on SE Management processes is misperceived by SEs, engineers, managers, and executives to represent SE core competency. Unfortunately, the process advocates, who may have had limited or no, actual "hands on" core SE experience, have turned SE Management processes into a profession.

Recognizing that: (1) SE core competency KSAs require many years to mature and (2) enterprise SEs have varying levels of experience, *how do we establish objective, measurable criteria that delineate various SE competency maturity levels*? Here is an example.

Kasser (2009, p. 6) suggest five types of SE maturity levels:

- **Type I.** This type is an "apprentice" who can be told how to implement the solution and can then implement it.
- **Type II.** This type is the most common type of systems engineer. Type II's have the ability to use the systems engineering process to figure out how to implement a physical solution once told what conceptual solution to implement.
- **Type III.** Once given a statement of the problem, this type has the necessary know-how to conceptualize the solution and to plan the implementation of the solution.
- **Type IV.** This type has the ability to examine the situation and define the problem (Wymore, 1993, p. 2).
- **Type V.** This type combines the abilities of the Types III and IV, and namely has the ability to examine the situation, define the problem, conceptualize the solution, and plan the implementation of the physical solution." (Please note that Wasson, 2016, pp. 293 – 312, reminds us that every system or entity has four domain solutions – requirements, operations, behavioral, and physical in that sequence. Premature development of the physical solution is outdated).

Interestingly, every enterprise has SDBTF discipline-based engineers who fit into each of these categories. However, that is different from having the requisite SE KSAs at those levels. We will address this point later in our discussion of Industry Contributions.

Each of these descriptions provides a basis for scoping for job labor categories. However, the problem is any "brand" of SE such as SDBTF Engineering characterized by an *endless loop* of *ad hoc activities* fits these five categories. For discussion purposes, let us accept Kasser, et al (2009, p. 6) Types I through V as a conceptual basis for establishing SE competency maturity levels that link to the competencies identified in the SE Competency Frameworks (Whitcomb, 2017); (Gelosh et al, 2017). Once that is established, all that remains is to complete the matrix by defining *objectives* for each task and subtask row and the intersecting cells for SE Types I - V.

The X-Y plot above the matrix plots relative quantities of competent SEs at various SE core competency maturity levels (Figure 5) available to work on technical projects in a typical enterprise. Referring to the upper left corner of Figure 6, ISO 15288 and the INCOSE SE Handbook specify *processes* and *activities* and earlier in Figure 2. In terms of the Kasser, et al (2009, p. 6) SE types, the SE competency *depth* of these processes and activities only go as far as Types I and II.

Likewise, the depth of documented OSPs, CMMI assessments, courses and vendor training mislabeled as "System Engineering," SE handbooks, and others typically only address Types I and II SE maturity levels. Figure 6 illustrates why moderate to large, complex projects continue to have SE related performance issues in spite of CMMI assessments intended to assess an enterprise's SE capabilities to perform prior to contract award. As a result, the Project Risk Zone shown in the upper right of Figure 6 illustrates the *root cause* of SE contributions to project performance issues, namely the lack of SE Types III – V core competency maturity levels. Although SE application experience is a key factor in SE core competency (Figure 3), years of experience in an engineering discipline such as EE, ME, SwE, and others does not equate SE Types III – V core competency.

The preceding discussions lead us to a key question: *How did SE management processes become the dominating focal point of SE at the expense of SE core competency?*

The Evolution of Systems Management– An Historical Review

During and following World War II, the need for increasingly complex systems and technologies strained traditional Engineering methods and Engineers' abilities to mentally deal with the technical complexities and interactions. Johnson (2002, 2013) provides an authoritative, historical perspective about the evolution of Systems Management and System Engineering in the 20th Century. He notes that "World War II was a crucible in which scientifically sophisticated technologies were rushed from research to development to production to operations" (Johnson, 2013, p. 671). His research provides an in-depth look at a variety of large systems development projects such as ballistic missiles, strategic defense, aircraft, missiles, rocketry, and so forth. Ring (2017) adds that nuclear power plants, nuclear-powered submarines, jet engines, and others presented challenging physics and Engineering problems in those days.

Traditional engineering disciplines such as EE, ME, et al, which were document-based, attempted to cope with the new challenges. The physical complexities of the challenges drove the need to seek better methods for coordinating and communicating among all the engineering disciplines working on a project. As the complexities of the technological problems grew, *tension* between engineers and management grew in the form of a "management gap" (Johnson, 2002, p. xi).

Managers planned, organized, staffed, directed, and controlled the activities and tasks their *subordinates* – e.g., engineers and scientists – performed. Drucker (1974, pp. 176-177) referred to the engineers and scientists of that timeframe as "knowledge workers." Management considered the engineers and scientists to be "lively and unruly" (Johnson, 2002, p. 2). Engineers were always late, over budget, and their systems failed to perform or had limited successes. For example, "roughly 50% of launches failed" (Johnson, 2013, p. 674). A clash between the management and engineer/scientist cultures was inevitable.

Similarly, a "technology gap" emerged between managers and their subordinates. Management's ability to understand and comprehend how engineers and scientists performed engineering and the technologies they applied became an issue. Engineers were unable to articulate the sequences of innovative and creative processes and tasks they used to perform engineering to management. Whereas traditional management focused on managing production lines, the engineering "knowledge workers" required innovation and creativity. Management created *standardized rules and procedures* referred to as "knowledge codification" (Johnson, 2002, p. 2). As a result, the management-subordinates hierarchy, which was based on managers planning, organizing, staffing, directing, and controlling tasks performed by their subordinates, was threatened. Managers needed better ways of regaining and exercising authoritative control over their subordinates.

Since these large projects were government acquisitions of aircraft, missiles, and other military items, the USAF recognized that it needed to gain control over the contractors' performance – i.e., late deliveries, technical failures, and cost overruns. Likewise, contractor management needed to regain control over the engineers and scientists. To solve the issue, two concepts emerged:

- The search for better engineering methods lead to consideration of the new field of SE, which emerged in the 1930s at Bell Laboratories (Kelly, 1950, p. 422).
- Systems Management consisting of Project Management (PM) and Configuration Management (CM) emerged on large, complex projects for the USAF. Over time, PM

enabled corporate management to organize and couple the traditional functions of management (planning, organizing, staffing, directing, controlling) with CM functions (configuration identification, configuration control, configuration status accounting, configuration reviews and audits) to manage and track engineering status, progress, performance, and risk.

Systems Engineering provided new methods of dealing with technological complexities as a means of managing programmatic and technical risk. Contrary to what we think SE was intended to accomplish, the key management purposes of SEs were: (1) creating and maintaining Engineering documents reflecting the current design and (2) communicating document changes and updates among the discipline engineers (Johnson, 2002, p. 2). The *subtlety* of the SE coordination and communication functions was based on *understanding* engineering rather than *performing* engineering. Based on what we know about the leadership role of SE today, the unintended consequence of the 1950's case for SE required more than simply coordinating and communicating. Unfortunately, the mindset established a stereotype that continues to plague SE today as exemplified in Mini-Case Study #3. As a result, the unintended consequence is engineering disciplines, which are rooted in physics and math, are slow to recognize SE as a discipline.

The Shift from SE to Systems Management Standards

To further ensure that contractors complied with requirements, the US Department of Defense (DoD) shifted to an emphasis on Systems Management as illustrated in Figure 7.

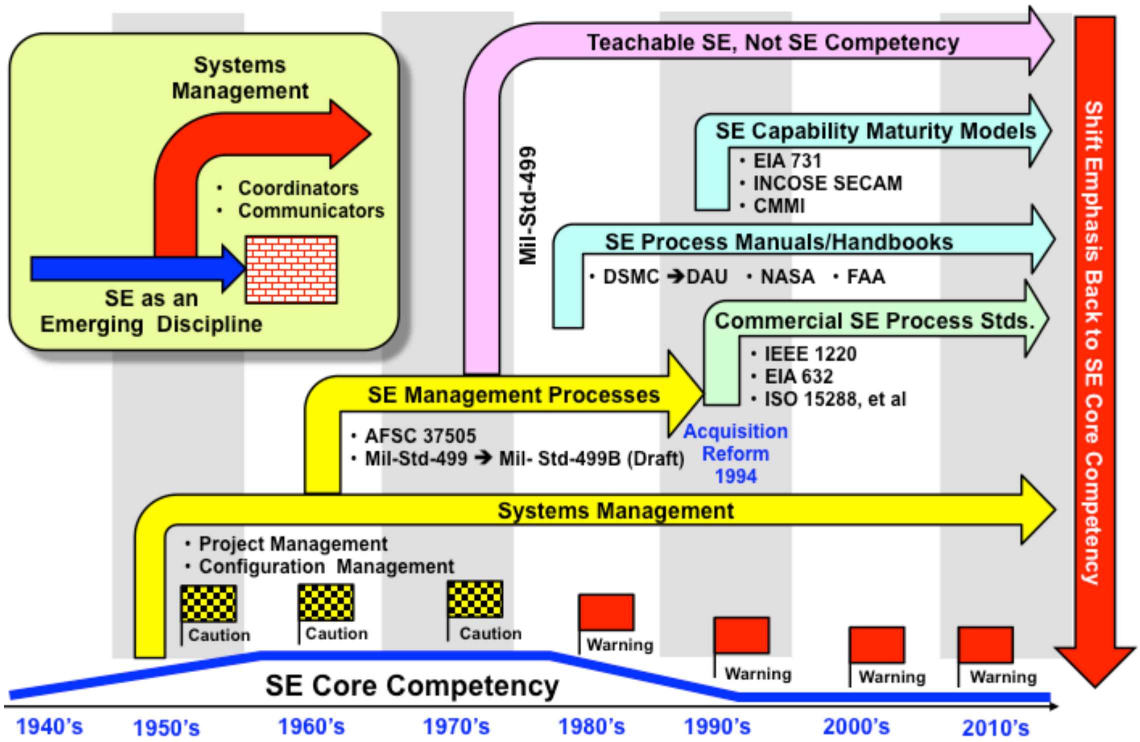


Figure 7. The evolution and propagation of Systems Management concepts via SE process standards.

SE Management Standards. Beginning in the 1960s, the USAF initiated a series of management standards for SE. Ring (2017) refers to these as "management protocols." These standards included:

- USAF AFSC 375-5 (1966) *Systems Management*
- Mil-Std-499 (1969) *Systems Engineering Management*
- Mil-Std-499A (1974) *Engineering Management*
- Mil-Std-499B (Draft - 1994) *Systems Engineering*

Observe the operative term, “management,” in most of these titles. Remembering those years, Blanchard (2017) notes that these titles were more than just subtle name changes.

- Systems Management related to the overall management of the system in question and *all* of its activities
- Engineering Management related to all of the engineering activities from a functional perspective
- System Engineering Management related to the management of all systems engineering activities; and so on.

Teachable Systems Engineering. In the 1960s and 1970s, as the engineering management standards evolved, SE courses took an SE Management approach influenced by Mil-Std-499. Ring (2017) refers to the early SE courses as “Teachable SE,” which accommodated the instructors’ knowledge and limited or no experience but not necessarily what Engineers needed to know to establish an SE knowledge competency (Figure 3) based on its concepts, principles, and practices.

DoD Acquisition Reform. The MIL-Std-499 series continued for 25 years until U.S. Secretary of Defense William J. Perry (1994) signed an Acquisition Reform Letter to transition the development of all standards to the commercial sector. Mil-Std-499B, which was in DRAFT form when the letter was signed, was never approved.

Commercial SE Process Standards. In 1994, the US DoD Acquisition Reform lead to the development of several SE Process standards:

- IEEE Std 1220-1994 (1995) Trial-Use *Standard for the Application and Management of the Systems Engineering Process*
- EIA/IS 632–1994 (1994), Interim Standard: *Processes for Engineering a System*
- ISO 15288:2002 - *System Engineering - System Life Cycle Processes*

SE Handbooks. U.S. Government organizations such as the Defense Systems Management College (DSMC), which is now the Defense Acquisition University (DAU), NASA, Federal Aviation Administration (FAA), the Department of Energy (DOE), and others, developed SE handbooks to serve as guides for their personnel. These documents were well-suited for acquisition engineers and PMs overseeing the development of systems.

Although early SE textbooks, such as Goode and Machol, and Hall, were written in the 1950s, the Defense Systems Management College (DSMC), NASA, US Army (FM-770-78, 1979), et al developed handbooks, which became “surrogate” SE textbooks for EEs, MEs, and other disciplines. Handbooks filled a void for engineers, who had limited access to SE courses and were learning to design and integrate their work products into “systems.” After all, *if the customer* was *using an SE handbook to assess the company’s work, the company’s engineers* needed *to read it*.

In 1990, the National Council on Systems Engineering (NCOSE) was established. Interestingly, the American Society for Engineering Management’s (ASEM) hosted a joint conference for the newly formed NCOSE and its 1st Annual NCOSE Conference in Chattanooga, TN (Brill, 1999, p. 260). NCOSE transitioned to the International Council on Systems Engineering (INCOSE) in 1995 (Honour, 1998, p. 9) and introduced its first SE Handbook in 1998. Did INCOSE’s SE management focus (Figure 6) evolve from ASEM engineering management roots?

SE Organizational Capability Maturity Models. During the 1980s as software-intensive systems experienced problems due to immature software development methods, industry and government shifted their focus on assessing the maturity of the Software Engineering capabilities. This lead to the formation of the Software Engineering Institute (SEI) at Carnegie-Mellon University.

After significant investment in software development improvements, researchers discovered that software developers were producing better software to poorly defined system and software requirements often written by traditional discipline engineers such as EEs designated as SEs. Since SE standards had evolved and matured over 30+ years, apparently there was little perceived need to "improve" SE. A more significant issue was improving SE contractor capabilities to perform. That is, *predictably* and *repeatedly* deliver systems, products, and services on-time, within budget, and compliant with specification requirements with minimal defects.

In the 1980s, automobile manufacturing made significant improvements in quality and performance. Since "documented processes" were one of the key contributors to the auto manufacturing improvements, Aerospace and Defense and other industries rationalized they could build aircraft, ships, satellites, etc. just like automobile manufacturers built cars. Subsequently, SE Management became more "process centric." Systems Management initiatives shifted to assessing an enterprise's SE capabilities to perform. As a result, several standards emerged:

- EIA/IS 632-1994 (1994) Interim Standard: Processes for Engineering a System.
- INCOSE Systems Engineering Capability Assessment Model (SECAM)
- CMMI (2000) - CMMI for Systems/Software Engineering

SE Core Competency (1950s – Today). Referring to the lower portion of Figure 7, observe that SE Competency *evolved* and *matured* from the 1940s through the early 1960s until emphasis shifted the focus to Systems Management. During the 1960s and 1970s, *caution flags* should have alerted industry and government to the potential long-term problem, especially from a System Safety perspective. SE Competency essentially plateaued from the 1970s until the 1980s when Teachable SE began to emerge in Engineering education. Again, *warning flags* should have alerted industry and government.

Kasser, et al (2009, p. 7), citing others, observes that "Research seems to show that early SEs tended to focus on the problem (Wymore, 1993), and finding the optimal solution" (Goode and Machol, 1959; Hall 1962). The authors add that most of the early SEs were Types III-V who performed tasks on a project and then moved on to other projects leaving Type IIs to "continue the development." Eventually, the number of new projects diminished resulting in layoffs of the Types III-Vs leaving a void in enterprise SE competency capabilities. As a result, Types I and IIs remained and took over Systems Engineering.

Based on the preceding discussion, a key question emerges: *How does SE Management influence and dominate the state of SE practice today*?

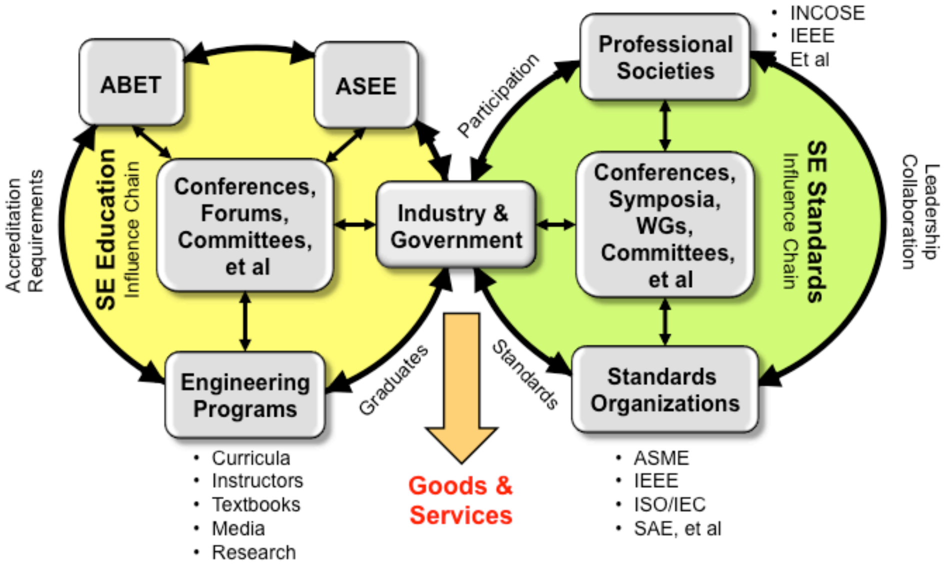
Understanding the Current State of SE Practice

When analyzing the state of SE practice today, two separate but interrelated influence chains emerge as shown in Figure 8: SE Education and SE Standards. Both of these influence chains share a common tangential intersection point, industry and government, which serve as the "proving ground" for Engineering and SE KSAs.

On inspection, Figure 8 makes sense – two interacting "systems" in harmony and balance. If these two influence chains get out of sync with each other, the potential for *confusion* and *conflict* abound in industry, government, and academia. Therefore, *stability* in the influence chain "thought processes" is important; however, there is a subtlety that is not readily apparent in Figure 8 that contributes to the state of SE today.

Referring to Figure 5, the author presented personal assessments of SE Competency Indicators across several business domains. If we combine the key "influencers" from the Engineering Education and SE Standards Influence Chains (Figure 8) with the SE core competency indicators

(Figure 5), Figure 9 emerges. For example, those who indicated an awareness of SE - i.e., familiarity, their remaining SE core competency indicators dropped off significantly to lower levels of maturity due to their inability to competently identify specific examples for each indicator. This is illustrated via decreasing maturity levels represented by the darker shades of gray.



- Figure 8. Industry and government are the "proving ground" for two influence chains: SE Education and SE Standards.

Observe the 3-D like vortex appearance in Figure 9 and its resemblance to a hurricane. Metaphorically, the hurricane symbolism reflects what has occurred over the past 60+ years of System Management. SE Management has become so deeply rooted and in-grained in media such as SE standards, handbooks, textbooks, and capability assessments that mindsets have evolved into a highly energized, rotational mass of "groupthink." Characteristic of "groupthink" paradigms, the "influencers" become so enamored by *what they perceive* SE to be, they tend to reject the fact that emphasis on SE Management processes and assessments *causes* rather than *resolves* project performance issues. Obviously, this is not *Systems Thinking* in action as evidenced by the project performance outcomes.

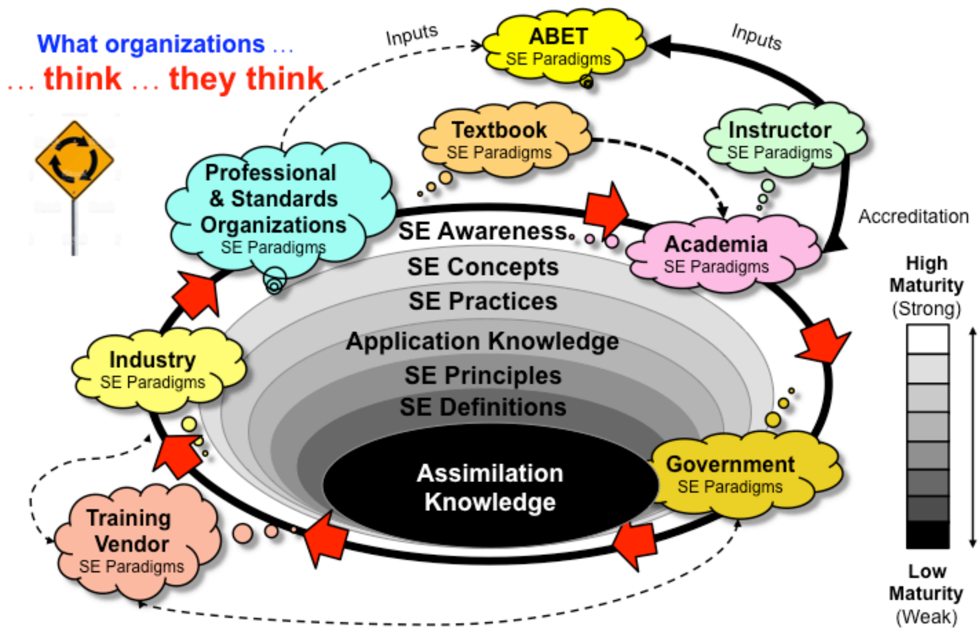


Figure 9. The "perfect storm" of SE Management "group think" and its impact on SE KSAs in a typical workplace based on SE core competency indicators (Figure 5).

To illustrate the preceding point, Ryschkewitsch et al. (2009, p. 4) observe: "Since the late 1980s, many aerospace related government and industry organizations have moved from a hardcore, technical leadership culture (the art) to one of systems management (the science). History has shown that many projects dominated by only one of these cultures suffer significant ill consequences."

So, *how do these "influencers" contribute to the 'groupthink' energy level*? Let us explore each one. Please note that each discussion below is dual-purposed: (1) it highlights problem areas that must be addressed and (2) it serves as opportunities for the respective organization to correct.

ABET Contributions. The US Accreditation Board of Engineering and Technology (ABET) establishes Criterion 3 Student Outcomes for Engineering Programs. For example, the current Student Outcomes related to SE include: "an ability to: …

> (c) "**design a system, component, or process** to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability.
>
> (d) **function on multidisciplinary teams.**
>
> (e) **identify, formulate, and solve engineering problems** …" (ABET, 2016, pp. 3-4).

One of the key principles of SE is *specificity* in defining specification requirements. There is an old SE adage that says "If you do not tell people what you want, you cannot complain about what comes back." In that context, ABET's criteria are written so *broadly* and *abstractly* that any SE course taught by a novice instructor could qualify as meeting these Student Outcomes criteria, at least by SE principles and best practices for specifying system requirements. From an SE perspective, ABET establishes criteria for Engineering Programs as performing entities functioning as "educational systems" to produce uniform and repeatable outcomes, i.e., degreed Engineers who can competently perform Engineering in industry and government.

Constructively, with student outcomes stated as abstract, high-level criteria, two questions emerge:

1. Is there any question why engineering graduates enter the industry and government workforces and exhibit the SE core competency indicator results shown in Figure 5?
2. Given industry and government project performance issues, why are ABET criteria not responding to the crisis? Are the ABET-industry-government feedback loops inoperable?

Academia (Engineering Program) Contributions. In general, the quality of engineering education is dependent on the strength and quality of its curriculum, instruction, and classroom materials. Some Engineering Programs and instructors have the foresight to recognize that in today's world, Engineering disciplines perform in collaborative, multi-discipline workplace environments. Courses attempt to integrate some of those concepts into their discipline instruction.

Over several decades, academia has made attempts to establish "capstone" courses and other methods to "expose" Engineering students to a collaborative and multi-discipline learning environment. Although these initiatives are admirable and provide opportunities for students, time restrictions, instructor industrial experience, and other factors often limit their effectiveness. Since ABET's Student Outcome Criterion 3 (d) requires the candidate to "… perform on multidisciplinary teams," structured team development and decision-making training is required.

Instructor Contributions. The strength of any Engineering Program or course resides in each instructor as a motivator, communicator, purveyor of industry knowledge and experience, and so forth. Wasson (2016, p. 42) refers to an educator's observation that academic instructors teaching an SE fundamentals course need at least 30 years of in-depth industry experience. They retire from industry, come to academia to teach, and leave after a few short years due to a lack of tenure and control over the course(s) they teach. In contrast, new Ph.D. instructors may have tenure but little or no industry experience. As a result, the number of *competent*, *qualified* instructors to teach SE courses is very limited. To solve the deficiency, instructors with qualifying credentials are recruited from industry; however, if the industry instructor's knowledge and experience is restricted to SE Management conceptual knowledge, the problem is *unresolved*.

In fairness to academic instructors, their experience is often instructional and research-based with little or no valid SE industry experience where SE core competency is *tempered* over time (Figure 3). As a result, they lack the requisite industry knowledge from working on many small to large, complex systems projects, end-to-end, over many years. Armstrong and Wade (2015) make the case that *SE can be learned by teaching it.* Although the author (Wasson) agrees "in principle" with their premise, the rhetorical question is: What is the frame of reference - i.e., SE *core competency* knowledge resource - from which they propose to teach?

To illustrate the point, Armstrong and Wade (2015, p. 5) quote Charles Babbage (1864): "On two occasions I have been asked, — 'Pray, Mr. Babbage, if you put into the machine wrong figures, will the right answers come out?'" Their paper's summary suggests "it would be helpful to increase their [instructors'] awareness of common learning roadblocks and why the student gets the answer wrong" as well as "types of systems engineering mistakes that are made and the thinking that causes them." These are *valid* concerns; however, the instructional learning process is *flawed* if the instructor fails to understand (1) the higher level SE Management is not the SE Core Competency issue and (2) the educational Plug and Chug SDBTF Engineering paradigm and replace it with a true multi-discipline SE problem-solving and solution-development methodology. The Mil-Std-499B (Draft) SE process, which was not approved, has been outdated for years. There is only one current SE textbook methodology that solves this problem.

Armstrong and Wade (2015) suggest instructors can learn SE by teaching it. Conceptually, SE can be learned by teaching, assuming a textbook based on SE core competency concepts, principles, and practices is selected. Additionally, the textbook should be selected based on *what* students need to enter industry and government and be productive on Day #1, not the instructor's personal, limited or no industry experience, comfort zone.

In terms of academia's preparedness to teach SE, Dixit and Valerdi (2007) observe that Engineering disciplines emerge within industry, accumulate a body of practitioner knowledge, and subsequently transition to the academic "theorists" for instruction. As evidenced by the lack of agreement by industry, government, professional societies, and standards organizations (Figures 8 and 9) concerning what a system and SE are, SE is still emerging as a body of knowledge and is not *sufficiently mature* in this author's opinion to transition to academic theorists. An exception being usage of a SE concepts, principles, and practices textbook; industry SE core competency KSAs (Figure 3); and the academic qualifications.

Author and Publisher Contributions. As a result of the confusion that exists in: (1) defining SE by professional societies and standards organizations and (2) the state of SE Management "group think," textbook authors get pulled into the undertow of the "group think" paradigm. Two problems emerge in traditional SE textbooks:

1. Problem #1 – Marketing textbooks and handbooks with "Systems Engineering" titles when, in fact, the scope and depth of the contents are more appropriately about System Acquisition and Management.
2. Problem #2 - Marketing "Principles" as a textbook subtitle when, in fact, the text contains *no explicitly stated principles*.

SE textbook and handbook titles should reflect the contents and vice versa.

Publishers are often dependent on authors as "experts" submitting textbook or handbook titles. In the case of SE, if an author or series editor's mental frame of reference for SE is defined by SE Management, "System Engineering" seems to qualify as a title. That paradigm creates *misperceptions* in readers' minds that SE Management contents represent what is required for SE core competency.

Government Contributions. Government, as an acquirer of systems, products, and services, is postured to assess System Developers and Services Providers who "claim" to understand and perform SE. Yet, project cost, schedule, and technical performance, which seems to evade many of its projects in the form of technical compliance issues, rework, defects, and cost and schedule, continues as noted by the GAO 17-77 (2016). Although SE Management "oversight" practices are appropriate for the type of tasks some government personnel perform, recognize that SE core competency is important in evaluating system developer performance. Here is an example.

Wasson (2016, p. 36) observes that when talking with traditional SDBTF Engineering paradigm enterprises and inquiring about their SE Process, they quickly point out that "they have a different 'brand' of SE." Why? Because they have heard that the SE Process is "iterative and recursive." Since their Plug and Chug SDBTF Engineering paradigm is iterative and recursive, they are, by definition, performing SE. Common semantics; entirely different concepts.

Additionally, the GAO reports such as GAO 17-77 (2016) highlight the correlation between project technical, cost, and schedule performance and the introduction of SE early into the acquisition process. For example, their findings note that the lack of SE or late involvement by SE correlates with low project performance in the form of technical risks, cost and schedule overruns, and so forth. Unfortunately, the GAO refers to SE in a generic context without some qualification of the form of SE performed such as true SE or *ad hoc, endless loop*, SDBTF Engineering. Most enterprises exist between these two extremes; typically toward SDBTF Engineering.

Industry Contributions. One of the clichés in industry was noted earlier in Mini-Case Study #3: *why does everyone have an SE job title*? Industry and government have been SE Management "proving grounds" for decades. Despite claims by executives and functional managers of having XX (quantity) SEs, many are discipline engineers designated with SE job titles. Wasson (2016, p. 39) observes that based on his career experience, less than 3% of so-called SEs in a typical Systems Engineering organization exhibit SE Types III - V competency shown in Figure 6. Most are specialty engineers such as project engineers; test engineers; human factors engineers; reliability, maintainability, and availability (RMA) engineers; safety engineers; and logistics engineers, modelers, and so forth that have SE experiential knowledge and competency levels as illustrated in Figure 5.

Unwitting industry and government executives, functional managers, and PMs are often unaware of the SE Management versus SE Competency issue. Even worse is the lack of awareness that the enterprise or engineers may be employing the SDBTF Engineering (Wasson, 2016, pp. xviii, Chapter 2) on projects *erroneously perceiving* it to be SE. Uninformed managers will contend they have five levels of SE personnel – SE Type I through SE Type V. The author's (Wasson) experience has been that these managers often have generic job labor categories that correlate with Kasser, et al (2009, p. 6) Types I through V scoping definitions. These managers *unwittingly* convert very competent discipline engineers (equivalent to Types I – V SEs) to SEs at the same competency level. The reality is a *Types III - V EEs, MEs*, or *others may only have a Type I or II SE competency concerning its proper implementation*. Likewise, having an MS or PhD in SE, which strengthens SE core knowledge (Figure 3), does not necessarily equate to SE core competency, which requires years of SE application knowledge (Figure 3).

Additionally, discipline engineers with *systems thinking* KSAs who have developed systems over many years with Type I or II SE competency are often "drafted" into SE positions to develop their enterprise's Systems Engineering Process. *If SDBTF Engineering is all they have ever known and perceive to be SE, guess what the infrastructure of the Engineering Process will be*? SDBTF Engineering! When system technical failures occur, Slegers, et al (2012, p. 78) panelists observe that a common reaction to failure is to add more processes. "… the panel is not suggesting that process is the source of the problem (failure). But rather misuse of the process to address a failure that the process cannot solve is the problem … The reason for failure is often not the process, but that the team didn't understand what they were doing in the first place."

Commercial Training Vendor Contributions. Due to the lack of an SE course requirement for all Engineering disciplines and a focus on performing SE Management practices *misperceived* to be SE, commercial training vendors have attempted to backfill a marketplace training need. Unfortunately, commercial training vendors sometimes contribute to the "group think" by promulgating the same outdated SE Management viewgraphs that have floated around for decades. The problem is exacerbated by labeling training courses "Systems Engineering" when in fact, the scope of their presentations is about "System Acquisition and Management" for Types I and II SEs.

Professional Society Contributions. Professional organizations *unwittingly* contribute to the SE Management "groupthink" through their standards, handbooks, and certification practices. Most notably is the *INCOSE SE Handbook: A Guide for System Lifecycle Processes and Activities* derived from and traceable to *ISO 15288, Systems and Software Engineering: System Lifecycle Processes*. Rhetorically, if INCOSE intends to be the "flag bearer" for SE among professional societies, it should reestablish its roots in SE core competency, not SE Management?

To illustrate the preceding point, INCOSE in its Vision 2025 (2014, p. v) expresses six (6) imperatives for the organization. From a system analysis perspective, sometimes it isn't the *whats* – goals and objectives – but rather the "gerunds" that tell the real story. INCOSE's imperatives focus on *expanding (twice), embracing, applying (twice), advancing, and enhancing*. Based on the words, there is no apparent recognition or understanding of the SE Management versus SE Core Competency issue.

Lastly, INCOSE "certifies" both engineering and non-engineering candidates as System Engineering Professionals (SEPs) based on its SE (Management) Handbook processes and activities. Given that US states, for example, govern the registration of professional engineers, it raises concerns about: (1) certifying engineering and non-engineering candidates based on management processes and activities as System **Engineering** Professionals (SEPs) and (2) misperceptions by executives and managers of their CSEP personnel as being equivalent to Types III – V SEs with SE core competency KSAs.

Standards Organizations Contributions. Standards organizations such as ISO, IEEE, etc. are the "headwaters" for influencing the heading of an Engineering discipline. Typically, members of professional societies deemed to have expertise in a given subject area write standards. In the case of the International Organization of Standards (ISO), INCOSE contributed to the writing of ISO 15288 (2015). Unfortunately, continuing to write SE Management process standards are a *necessary* but *insufficient* condition for solving the SE core competency issue that contributes to project cost, schedule, and technical performance issues.

Summarizing the contributions of the "influencers" in Figure 8, the momentum of the SE Management "group think" rotational mass (Figure 9) continues to run *unabated*. Referring to our discussion of Figure 8, we said the two interacting influence chains – SE Education and SE Standards – needed *stability* in the influence chain "thought processes" to prevent confusion and conflicts. In that context, the "groupthink" paradigm (Figure 9) moves *powerfully* in one direction and is slow to recognize the need for change.

Summary and Recommendations

In summary, this paper provides an in-depth perspective as to why projects continue to exhibit technical, cost, and schedule performance issues. So, *how do we fix this problem*? Our discussions of Figures 8 and 9 highlighted issues and opportunities for the "Influencers" in the SE Education and SE Standards Supply Chains to correct the SE core competency maturity indicators (Figure 5) to Level 10. This represents a culture change.

A consultant once observed that if you shrink a culture back to one person, it will return to its original state over time. The typical human systems response is to form more committees of the same people who oversaw the evolution of the current, outdated paradigm. Reiterating an earlier quote by Slegers, et al (2012, p. 78) in a panel discussion "… The reason for (technical) failure is often not the process, but that the team didn't understand what they were doing in the first place." Interactions between the influence supply chains require harmonization in thought to ensure stability and avoid conflicting direction. Collaboration must begin with a new SE core competency paradigm.

SE today is consumed with Model-Based Systems Engineering (MBSE). The irony is MBSE is often being performed by SDBTF engineers lacking SE core competency KSAs (Figure 3). MBSE requires more than Types I and II SEs performing ad hoc, dragging and dropping of boxes at various levels of abstraction with interconnecting lines mimicking life cycle processes for compliance. Prematurely launching into MBSE without SE core competency is analogous to deciding to forgo basic math and move into calculus. SE, like EE, ME, SwE, and other disciplines has an instructional taxonomy of concepts, principles, and practices structure that corrects SDBTF SE ad hoc interpretations and implementations. Learn to recognize the difference and take action to shift the culture within your enterprise and accountability.

In conclusion, will the "influencers" in Figures 8 and 9 recognize that SE Management is not SE Core Competency and take action to correct the imbalance? As Ryschkewitsch et al. (2009, p. 4) quoted earlier observe: "History has shown that many projects dominated by only one of these cultures suffer significant ill consequences." It is time to shift this outdated, 60+ year-old paradigm to reestablish SE Core Competency as the basis for correcting SE contributions to project performance technical, cost, and schedule performance issues.

References

ABET, 2016, *Criteria for Accrediting Engineering Programs – Effective for Reviews During the 2017 - 2018 Accreditation Cycle*, Accreditation Board of Engineering and Technology (ABET) Engineering Accreditation Commission, Baltimore, MD (US).

AFSCM 375-5, 1966, Air Force Systems Command Manual No. 375-5, *System Engineering Management Procedures*, Andrews AFB – Headquarters Air Force Systems Command (AFSC). Washington, DC (US).

Armstrong, James and Wade, Jon, 2015, *Learning Systems Engineering by Teaching It*, 25th Annual INCOSE International Symposium, Seattle, WA (US).

Blanchard, Ben, 2017, phone interview on 11/4/17 with permission, Blacksburg, VA (US).

Blanchard, Benjamin S, and Fabrycky, Wolter J., 2011, *Systems Engineering and Analysis*, 5th ed. Pearson Education, Inc., Upper Saddle River, NJ (US).

Cowper and A. Smith, *Selling systems engineering—how do you show a positive return on investment*?, INCOSE UK Spring Symposium, 12–14 May 2003, 1–8.

Dixit, Indrajeet and Valerdi, Ricardo, 2007, *Challenges in the Development of Systems Engineering as a Profession*, 17th Annual INCOSE International Symposium (2007).

Brill, James, 1999, "Systems Engineering – A Retrospective View," *Systems Engineering – The Journal of the International Council on Systems Engineering* (INCOSE), John Wiley and Sons, Inc., Hoboken, NJ (US).

CMMI, 2000, *CMMI for Systems Engineering/Software Engineering, Version 1.02, Staged Representation (CMMI-SE/SW, V1.02, Staged)* (CMU/SEI-2000-TR-018), CMMI Product Development Team, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA (US).

Drucker, Peter F., 1974, *Management: Tasks, Responsibilities, and Practices*, Harper & Row, Publishers, New York.

EIA/IS 632–1994, 1994, Interim Standard: *Processes for Engineering a System*, Electronic Industries Alliance (EIA), Arlington, VA (US).

Emes, Michael; Smith, Alan; and Cowper, Douglas, 2005, "Confronting an Identity Crisis – How to Brand" Systems Engineering," *Systems Engineering – The Journal of the International Council on Systems Engineering* (INCOSE), John Wiley and Sons, Inc., Hoboken, NJ (US).

Einstein, Albert, Insanity quote, ThinkExist.com

FM-770-78, 1979, US Army Field Manual: *System Engineering Fundamentals*, Department of Defense (DoD), Washington, DC (US).

GAO 17-77, 2016, *Defense Acquisition: Detailed Systems Engineering Prior to Product Development Positions Programs for Success*, U.S. Government Accountability Office (GAO) Report to Congressional Committees, Washington, DC (US).

Gelosh, Don; Heisey, Mimi; Snoderly, John; and Nidiffer, Ken, 2016, *The Path to Version 0.75 of the Proposed INCOSE Systems Engineering Competency Framework*, 26th Annual INCOSE International Symposium (2016), Edinburgh, Scotland.

Goode, H. H. and Machol, R. E., 1959, *Systems Engineering*, McGraw-Hill, New York (US).

Grady, Jeffrey, 2011, *Is Systems Engineering Overdue for a Fix*?, Proceedings of the 21st Annual INCOSE International Symposium, Denver, CO (US).

Griffin, Michael, 2010, *How Do We Fix Systems Engineering*, 61st International Astronautical Congress, Prague, Czech Republic.

Hall, A. D., 1962, *A Methodology for Systems Engineering*, D. Van Nostrand Company Inc (US).

Honour, Eric C., 1998, "INCOSE: History of the International Council on Systems Engineering," *Systems Engineering – The Journal of the International Council on Systems Engineering* (INCOSE), John Wiley and Sons, Inc., Hoboken, NJ (US).

IEEE Std 1220-1994, 1995, IEEE Trial-Use *Standard for the Application and Management of the Systems Engineering Process*, Institute of Electrical and Electronic Engineers (IEEE), New York, NY (US).

INCOSE, 2015, *Systems Engineering Handbook: A Guide for System Life Cycle Process and Activities* (4th ed.). Walden, D.D.; Roedler, G. J.; Forsberg, K. J.; Hamelin, R. D.; and, Shortell, T. M. (Eds.), International Council on Systems Engineering (INCOSE), San Diego, CA (US).

ISO/IEC 15288:2015, 2015, *System Engineering - System Life Cycle Processes*, International Organization for Standardization (ISO), Geneva.

INCOSE-PMI STP, 2017, *Changing the Acquisition 'Game': Alleviating PM-SE Integration Risks,* INCOSE-PMI SE-PM Integration – Strategic Technical Planning Working Group (STPWG), INCOSE, San Diego, CA (US).

INCOSE, 2014, *Vision 2025: A World in Motion*, International Council on Systems Engineering, San Diego, CA (US).

Johnson, Stephen B., 2002, *The Secret of Apollo: Systems Management in the American and European Space Programs*, The Johns Hopkins University Press, Baltimore, MD (US).

Johnson, Stephen, 2013, "Technical and Institutional Factors in the Emergence of Project Management," *International Journal of Project Management*, ISSN: 0263-7863, Vol: 31, Issue: 5, Page: 670-681, Elsevier, Amsterdam.

Kasser, Joseph; Hitchins, Derek; and Huynh, Thomas V., 2009, *Reengineering Systems Engineering*, Proceedings of the 3rd Annual Asia-Pacific Conference on Systems Engineering (APCOSE), Singapore.

Kelly, Mervin J., 1950, "The Bell Telephone Laboratories – An Example of the Institute of Creative Technology," *Proceedings of the Royal Society*, B 1950 137, doi: 10.1098/rspb. 1950.0050 published 28 November 1950, The Royal Society, London.

Mil-Std-499, 1969, Military Standard: *Systems Engineering Management*, Department of Defense (DoD), Washington, DC (US).

Mil-Std-499A, 1974, *Engineering Management*, Department of Defense (DoD), Washington, DC (US).

Mil-Std-499B Draft, 1994, Military Standard: *Systems Engineering*, Department of Defense (DoD), Washington, DC (US).

Ring, Jack, 2017, phone interview on 11/4/17 with permission, Gilbert, AZ (US).

Perry, William J. (1994), *Specifications and Standards – A New Way of Doing Business*, US Secretary of Defense (SecDef), 29 June 1994, Department of Defense (DoD), Washington, DC (US).

Ryschkewitsch, Michael; Schaible, Dawn; and Larson, Wiley, 2009, The Art and Science of Systems Engineering, Systems Research Forum, Vol. 3, Issue 2, *December 2009, Forum* **03**, 81 World Scientific Publishing Co. https://doi.org/10.1142/S1793966609000080 , Hackensack, NJ (US):

Slegers, Nathan J,; Kadish, Ronald T.; Payton, Gary E. Thomas, John; Griffin, Michael D.; and Dumbacher, Dan (2012), "Learning from Failure in Systems Engineering: A Panel Discussion," *The Journal of the International Council on Systems Engineering* (INCOSE), John Wiley and Sons, Inc., Hoboken, NJ (US).

Wasson, Charles S., 2008, *Systems Thinking: Bridging the Educational Red Zone Between Systems Engineering and Project Management*, 2nd Annual INCOSE Great Lakes Conference, September 7 - 9, 2008, Mackinac Island, MI (US).

Wasson, Charles S., 2010, *System Engineering Competency: The Missing Element in Engineering Education*, Proceedings of the 20th Anniversary of the INCOSE International Symposium (2010), Chicago, IL (US).

Wasson, Charles S., 2012, System Engineering Competency: *The Missing Course in Engineering Education*, American Society for Engineering Education (ASEE) National Conference, San Antonio, TX (US).

Wasson, Charles S., 2016, *System Engineering Analysis, Design, and Development: Concepts, Principles, and Practices*, 2nd Ed., John Wiley & Sons, Inc., New York, NY (US).

Whitcomb, Clifford; White, Corina; Kahn, Radia; Grambow, Dana; Velez, Jose; and Delgado, Jessica, 2017, *The U. S. Department of Defense Systems Engineering Competency Model*, 28th Annual INCOSE International Symposium, Adelaide, Australia.

Wymore, A. W., 1993, *Model-Based Systems Engineering*, CRC Press, Boca Raton, FL (US).

Biography

Wasson, Charles, INCOSE Fellow, Certified ESEP, is the Founder and Principal Consultant for Wasson Strategics, LLC (WSL). His professional affiliations include the INCOSE, IEEE, ASEE, PMI, and Tau Beta Pi. His educational background includes a BSEE and an MBA from Mississippi State University, a SE Certificate from Stevens Institute of Technology, and additional graduate studies. His professional experience includes over 38 years of leadership in project management; system, hardware, and software development; integration and test with Lockheed Martin Corporation; Teledyne Brown Engineering; U.S. Army Missile Research & Development, Command; Planning Research Corporation; Spacecraft, Inc.; and eight years with Wasson Strategics, LLC. Charles' first SE text *System Analysis, Design, and Development: Concepts, Principles, and Practices* won the International Academy of Astronautics' 2006 Engineering Sciences Book of the Year Award. In 2016, John Wiley published the 2nd Edition of his *System Engineering Analysis, Design and Development: Concepts, Principles, and Practices* text used world-wide by practitioners and universities.