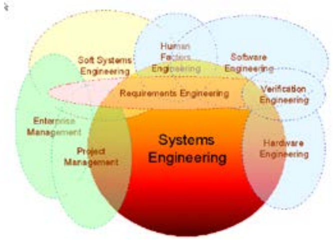
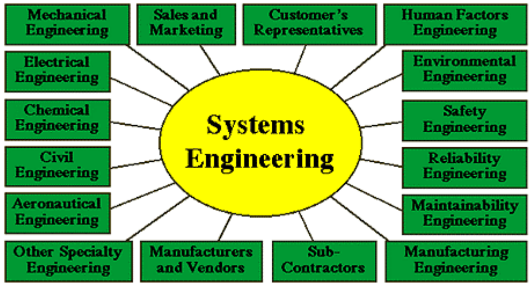




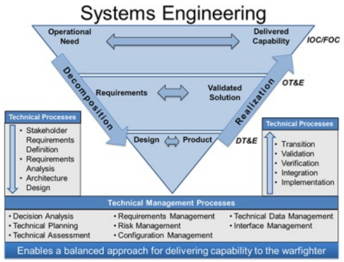


Figure 4. DoD SE Process Model of 2014

Why Systems Engineering – And INCOSE – Is Becoming Irrelevant

Dave Hall

ESEP/CISSP

Hall Associates LLC

Dave.hall@hallassociateshsval.com

Systems Engineering Maturity Model
Initial Perception

Over the past several years, I have developed a perception of SE from the viewpoint of a company or organization trying to implement and then working with Systems Engineering. A lot of DOD customers are requiring SE be in place in their RFPs and more companies are trying to put SE in place to meet that requirement. However, it is starting to look a lot like when Total Quality Management (TQM) was the current silver bullet - almost all of the capabilities and processes are being laid in with no regard for the needs of the organization and being "implemented" by people with very little overall SE knowledge.

Systems Engineering Maturity Model Initial Perception

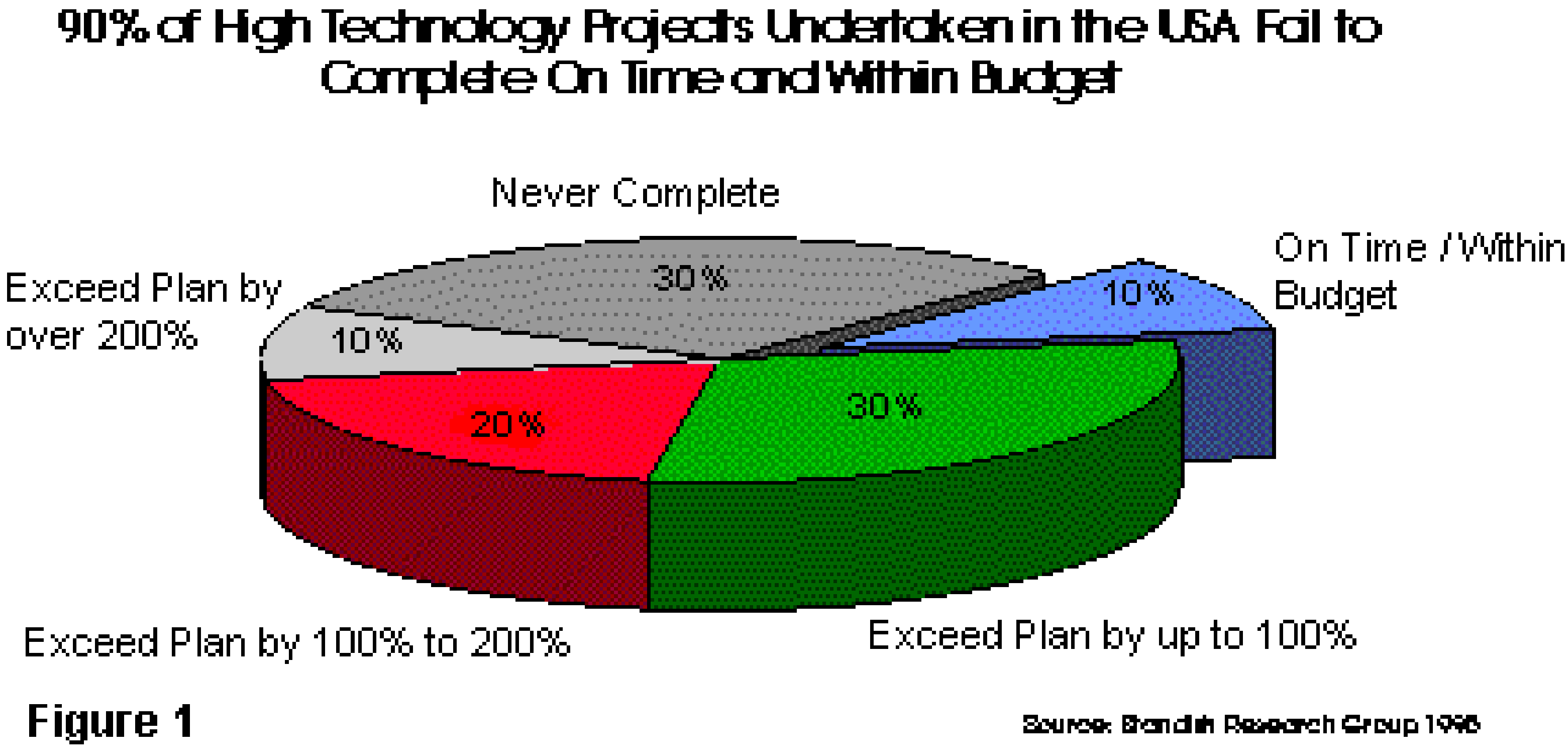
So when it comes to trying to develop proper overall SE for an organization, it doesn't happen. Usually even if the person is a CSEP - and with many ESEPs. The silo effect continues because there are processes that are "more important" to the person or the organization. And with the problem of determining SE Return on Investment (ROI) without historical data and no real metrics to determine SE efficiency and effectiveness, we are getting into shaky territory. And once an organization implements SE, they seldom go back and try to determine if their implementation actually accomplished what they were trying to do.

Problem Statement

The inability to properly define, implement, use and measure appropriate systems engineering processes and activities significantly contributes to schedule delays and cost increases, thus significantly reducing the supposed ROI of Systems Engineering.

Almost all of the organizational and company Systems Engineering capabilities and processes are being implemented and sustained in a non-integrated way with little regard for the needs of an organization, company or program and have been implemented by people with very little **overall Systems Engineering Discipline** experience or knowledge.

Actuality Versus Perception - 1996



Actuality Versus Perception - 1996

In industry today, there is considerable emphasis on attempting to quantify, qualify, and define systems engineering and the systems engineering process.

Systems Engineering Process Assessments (SEPAs) and the like, are being performed to identify the relative maturity of the process (e.g., how well do processes match up with a macro model and how well are they institutionalized and used). While these assessments can be powerful tools, they do not provide insight into the methodology of systems engineering or the effectiveness of the systems engineering process. Only when the maturity level, methods, tools, and effectiveness are viewed in the aggregate, can the systems engineering process be truly assessed.

Actuality Versus Perception - 1996

The following list describes several key benefits of performing a SE benchmarking effort:

- **Identifies the state of practice of systems engineering:**
  - Determines levels of maturity of the organization
  - **Demonstrates how organizations compare**
  - **Provides a baseline to investigate alternates in systems engineering**
  - Provides the means to assess the changes due to the use of alternate systems engineering methodologies, tools, and processes
- Identifies the present systems engineering methodologies:
  - **Provides a basis to develop new methodologies**
  - **Provides the means to measure improvement**
- Identifies the predominant systems engineering tools
- **Correlates practices, methodologies, and tools with process maturity level and marketplaces**

Other benefits of benchmarking the systems engineering process include:

- Provides systems engineering process model exposure
- Identifies expertise in the process areas
- **Provides a tool for process improvement**
- **Provides a means to compare groups and organizations against each other**

Actuality Versus Perception - 2015

On average, large IT projects run 45 percent over budget and 7 percent over time, while delivering 56 percent less value than predicted (A study of 5,400 large scale IT projects)

PricewaterhouseCoopers, which reviewed 10,640 projects from 200 companies in 30 countries and across various industries, found that only 2.5% of the companies successfully completed 100% of their projects. -

A study published in the Harvard Business Review, which analyzed 1,471 IT projects, found that all but one in six projects had a cost overrun of 200% on average and a schedule overrun of almost 70%. And we all have heard about large construction projects — the Channel Tunnel, Euro Disney, and Boston's "Big Dig" — that ended up costing almost double their original estimate. -

57% of projects fail due to "breakdown in communications

39% of projects fail due to lack of planning, resources, and activities

Actuality Versus Perception - 2015

RESOLUTION

	2004	2006	2008	2010	2012
Successful	29%	35%	32%	37%	39%
Failed	18%	19%	24%	21%	18%
Challenged	53%	46%	44%	42%	43%

Project resolution results from CHAOS research for years 2004 to 2012.

Reported Organizational Project Management Maturity Level		% of on-time projects	% of on-budget projects	% of projects meeting original goals and business intent
	High	67%	68%	73%
	Medium	55%	58%	67%
	Low	39%	44%	53%

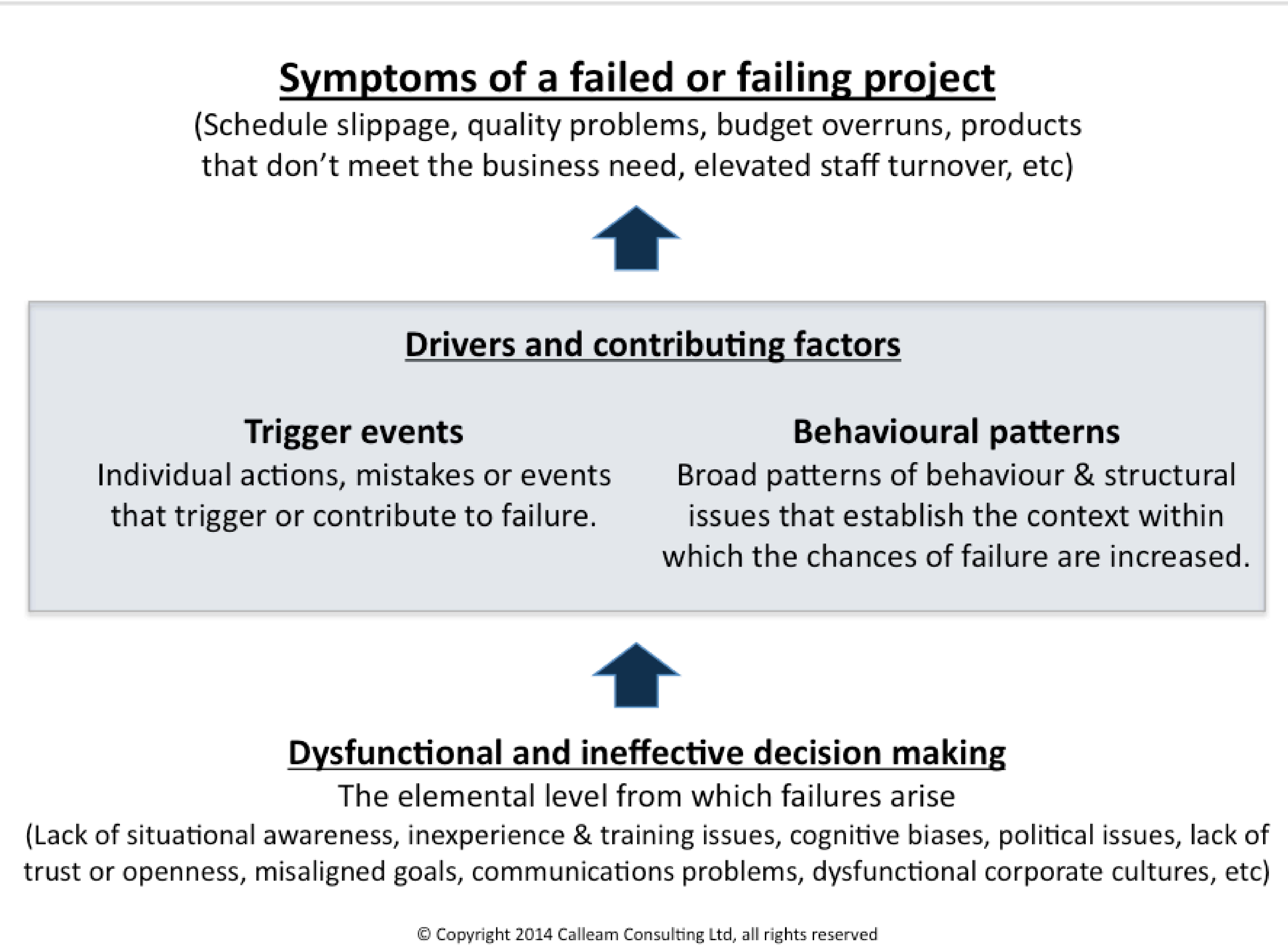
Average Rework in DOD Programs – 88% (INCOSE Handbook/MIT report)

Actuality Versus Perception - 2015

Only **64%** of projects meet their goals.
70% of companies report having at least one failed project in the last year.
Organizations lose $109 million for every $1 billion invested in projects and programs.

Most Common Causes of Project Failure:
Changing priorities within organization – **40%**
Inaccurate requirements – 38%
Change in project objectives – **35%**
Undefined risks/opportunities – 30%
Poor communication – 30%
Undefined project goals – 30%
Inadequate sponsor support – **29%**
Inadequate cost estimates – **29%**
Inaccurate task time estimate – **27%**
Resource dependency – 25%
Poor change management – 25%
Inadequate resource forecasting – 23%
Inexperienced project manager – **20%**
Limited resources – **20%**
Procrastination within team – 13%
Task dependency – 11%
Other – **9%**


Market & strategy failures
Organizational & planning failures
Engagement, teamwork & communications failures
Leadership & governance failures
Common classes of project failure
Skills, knowledge & competency failures
Underestimation & analysis failures
Risk failures
Quality failures


Symptoms of a failed or failing project
(Schedule slippage, quality problems, budget overruns, products that don't meet the business need, elevated staff turnover, etc)
Drivers and contributing factors
Trigger events
Individual actions, mistakes or events that trigger or contribute to failure.
Behavioural patterns
Broad patterns of behaviour & structural issues that establish the context within which the chances of failure are increased.
Dysfunctional and ineffective decision making
The elemental level from which failures arise
(Lack of situational awareness, inexperience & training issues, cognitive biases, political issues, lack of trust or openness, misaligned goals, communications problems, dysfunctional corporate cultures, etc)
© Copyright 2014 Calleam Consulting Ltd, all rights reserved

What Is Systems Engineering?

Systems Engineering **is the discipline encompassing** the entire set of scientific, technical and managerial processes needed to conceive, evolve, verify, deploy and support an integrated Systems of Systems capability to meet user needs across the life cycle.

The purpose of Systems Engineering is to increase a system's probability of success and reduce the risk of failure

What Is Systems Engineering?

*www.incose.org/AboutSE/WhatIsSE*C

Systems Engineering is an **interdisciplinary approach and means** to enable the realization of successful *systems*. ... ***Systems Engineering*** **integrates all the disciplines and specialty groups** into a team effort forming a structured development process that proceeds from concept to production to operation.

At NASA, “systems engineering” is defined as a methodical, **multi-disciplinary** approach for the design, realization, technical management, operations, and retirement of a system. A “system” is the combination of elements that function together to produce the capability required to meet a need.

What Is Systems Engineering?

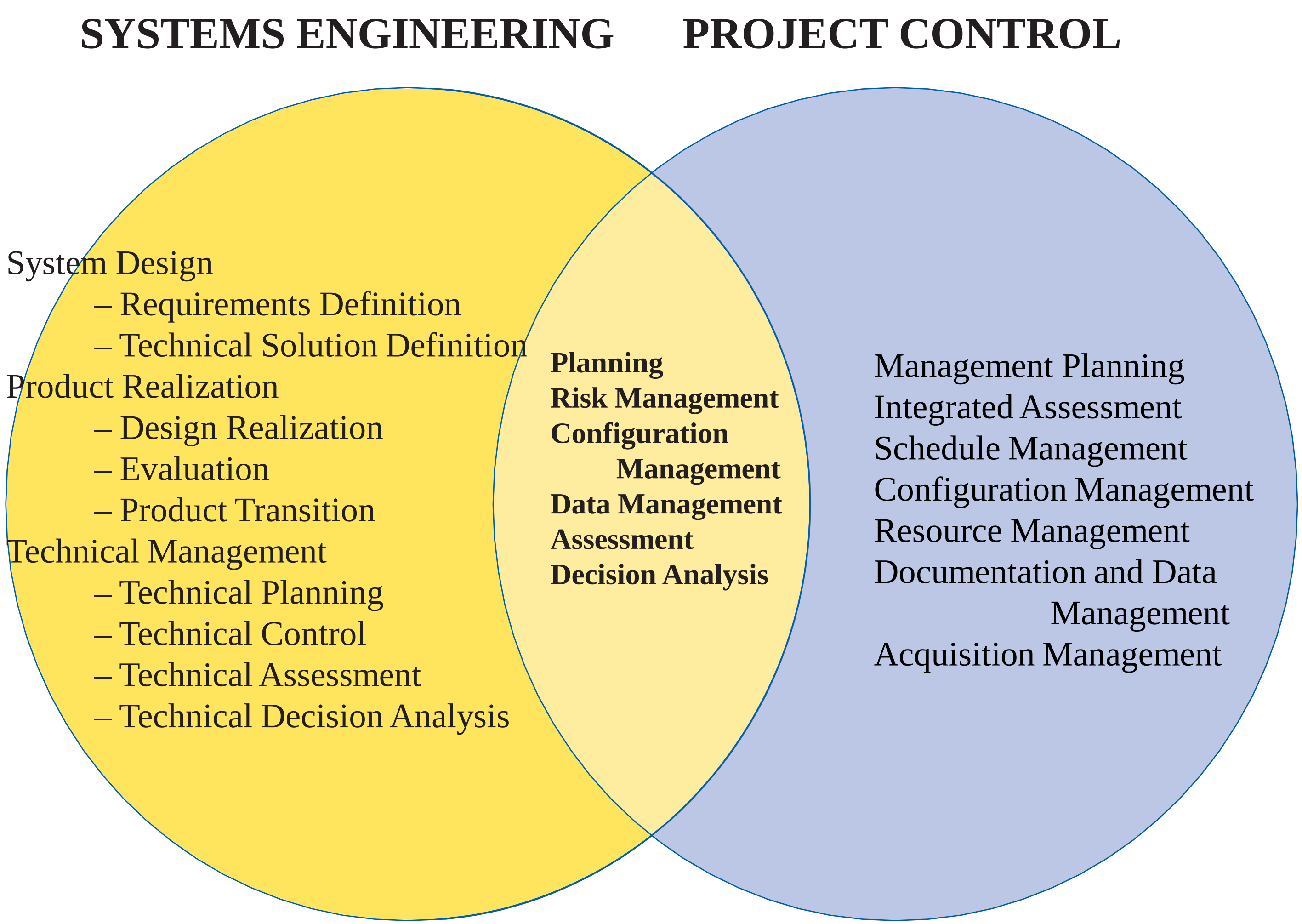
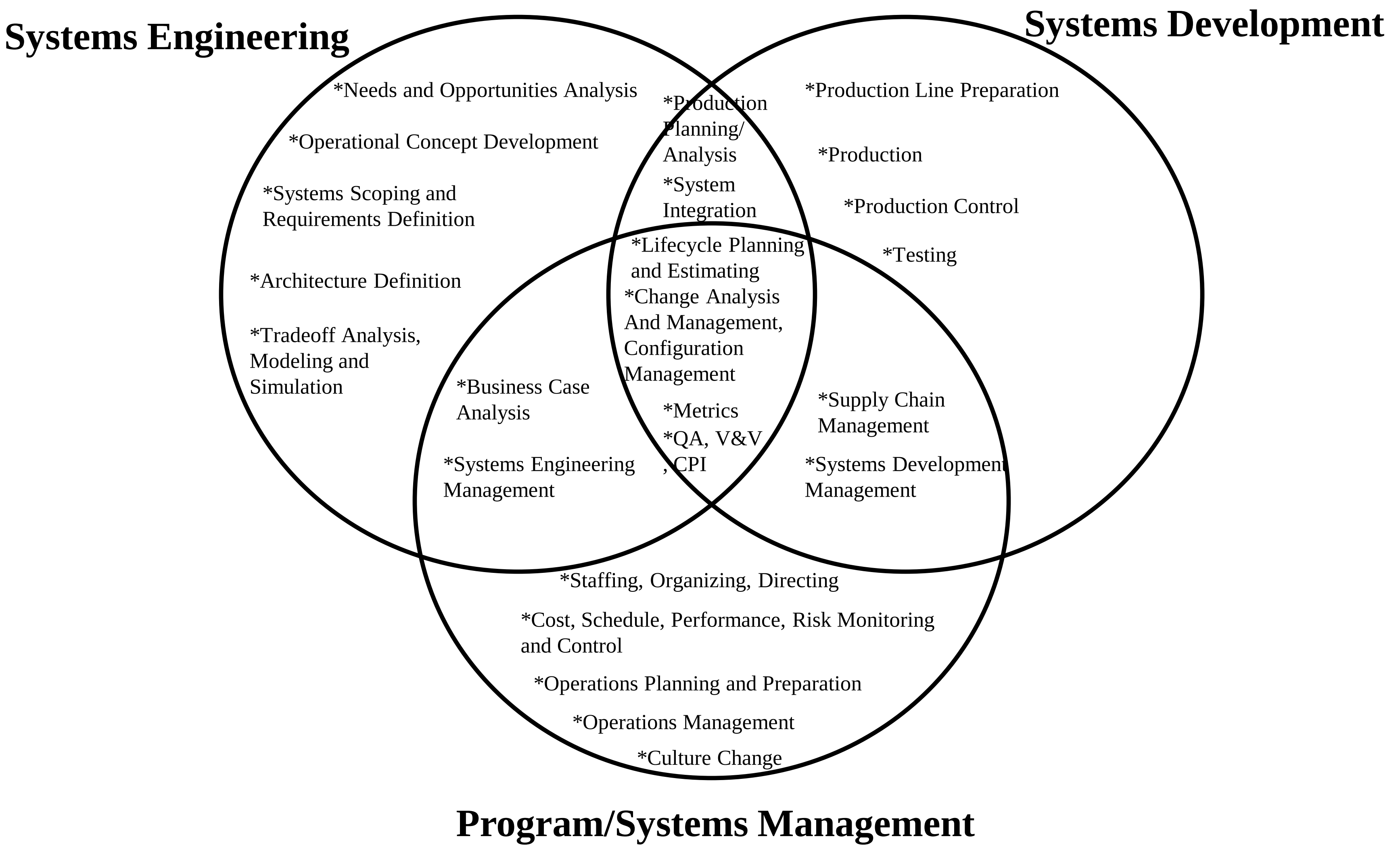


Figure 2.0-1 SE in context of overall project management (NASA)

What Is Systems Engineering?



What do you mean by "Systems Engineering"

INCOSE And Other Thoughts

Performance Assessment Measures (from the Handbook and SEBOK):

1. Effectiveness of the SE Process
2. Quality of SE Process Outputs
3. Timeliness of SE process Outputs
4. Organization's SE Capability Development
5. Individuals SE Competence Development
6. Productivity of Systems Engineers

I see objective evidence of this everyday in my work. My only comment is that industry, government, and academia have developed an incestuous, 1980's - 90's SE paradigm that limits their ability to progress. You see this manifested in the students they graduate - textbooks they used taught by instructors typically with no true SE experience and how they think/or don't; industry and government people who evolve through the same process, INCOSE decisions, et al. Since INCOSE is a reflection of its membership, it becomes the melting pot for industry, government, and academic "groupthink" SE.

In recent years, people around the country have gotten smarter, learned to see through the abstract facade – ABET, ISO, CMMI, INCOSE, et al - and recognize the current SE paradigm for what it is. It produces mediocre results at best as reflected in project performance. An organizational hero ends up bringing projects to delivery completion by brute force corrective actions via nights, weekends, and holidays work! Then, assign blame to "Systems Engineering" as the culprit rather than what it is – outdated 1980's – 90's industry, government, and academia "Groupthink" SE.

Could We Use CMMI?

The Standard CMMI® Appraisal Method for Process Improvement (SCAMPISM) A is designed to provide benchmark quality ratings relative to Capability Maturity Model® Integration (CMMI) models. It is applicable to a wide range of appraisal usage modes, including both internal process improvement and external capability determinations. SCAMPI A satisfies all of the Appraisal Requirements for CMMI (ARC) requirements for a Class A appraisal method.

The SCAMPI v1.2 Class A Method Definition Document describes the requirements, activities, and practices associated with **each of the processes that compose the SCAMPI A method.** It is intended to be one of the elements of the infrastructure within which SCAMPI Lead Appraisers conduct a SCAMPI A appraisal. Precise listings of required practices, parameters, and variation limits, as well as optional practices and guidance for enacting the method, are covered. An overview of the method's context, concepts, and architecture is also provided.

So the CMMI method also looks at ONLY the specific processes, not the interdisciplinary process or how to implement .

Areas Of Concern

The areas of concern (risks to properly implementing, using, measuring and analyzing Systems Engineering processes and activities) within your enterprise or organization (or for individual programs) are:

- **People:** Who are your systems engineers? Is systems engineering a job title, or does it describe anyone who wants to think about the larger system that a product fits into, or only people with "Systems Engineering" degrees, or something certifiable by INCOSE?
- **Culture:** What is your current management and work culture and how resistant is it to change? How much change is going to be required?
- **Value:** What is the value to your organization or company of performing systems engineering? What are the benefits of systems engineering you are expecting?
- **Training:** How should your system engineers and other personnel be educated? What classroom and on-the-job training is important?
- **Tools:** What tools do systems engineers need/use? What tools can provide support for everything systems engineering does in an integrated manner?
- **Measurement and Assessment:** How do you measure systems engineering processes/activities? How do you assess a research and development organization, a maintenance organization, or an order fulfillment organization against a systems engineering model?
- **Standards:** Who should use systems engineering standards (or domain best practices) and how should they use them? Do the various standards apply differently to different implementations of systems engineering? How do systems engineering standards apply to a small company making piece parts, consumer goods or services? Can you define effectiveness of an SE Interdisciplinary Process ?
- **Future:** How is your systems engineering capability expected/required to change in the future?

What is the Systems Engineering Process Addressing an Interdisciplinary Approach?

Goals/Objectives (Define)

Define
Implement
Sustain
Measure
Improve

Requirements (Define)
Risk
Product
Standards/Best Practices
Culture
Skills
Past Performance
Validation/Verification

Implementation
Training
Tools
Management Champions
Culture Change

Operational/Sustainment
Standards
Support
Training
Tools
Personnel/Skills
Culture Change
Enforcement

Measurement
Validation Metrics
Verification Metrics
Personal
Process
Organizational
Customer

Continuous Improvement
Each requirement achieved
What rework is still ongoing
What areas need to be more efficient
What areas need to be more effective
Sufficient metrics being collected
Culture Change
Additional enforcement required
Additional training required

How Should You Implement SE?

Examples of Systems Engineering Goals

- Systems Engineering (SE) must establish the technical framework for delivering materiel or service capabilities to the customer and assure that the design addresses the actual problem.

- SE must provide the foundation upon which everything else is built and support program success. The desired design is technologically possible.

- SE must ensure the effective development and delivery of capability through the implementation of a balanced approach with respect to cost, schedule, performance, and risk using integrated, disciplined, and consistent SE activities and processes regardless of when a program enters its life cycle.

Summary

No one is using the Systems Engineering Process (interdisciplinary), they are simply working with each individual process within the SE definition.

Not understanding the interdisciplinary nature of Systems Engineering means that each individual process is being implemented as if it was the only one being considered. This may optimize specific processes but suboptimizes SE. So SE does not provide the promised ROI even tho the individual processes do provide some ROI.

No one is collecting (or using) metrics on SE performance, so no one is improving SE. Individual processes are sometimes being improved.

There is no current methodology addressing the Systems Engineering Process nor does any SE Handbook, Guidance, Etc. discuss the SE Process except to “define” it.