Evaluation and Synthesis of Methods for Measuring System Engineering Efficacy within a Project and Organization

By

Timothy Daniel Flynn

B.S. Aerospace Engineering
Pennsylvania State University, 1994

Submitted to the Systems Design and Management Program
in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Engineering and Management

At the

Massachusetts Institute of Technology

February 2007

Signature of Author ______________________________
Timothy Daniel Flynn
System Design and Management Program
January 2007

Certified by ______________________________
Donna H. Rhodes
Thesis Supervisor
Lean Aerospace Initiative

Accepted by ______________________________
Patrick Hale
Director
System Design and Management Program

Evaluation and Synthesis of Methods for Measuring System Engineering Efficacy within a Project and Organization

By

Timothy Daniel Flynn

Submitted to the Systems Design and Management Program
in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Engineering and Management

ABSTRACT

The need for robust systems engineering in product development has been understood by those developing product in the aerospace and defense industries since the days of the Atlas ballistic missile program. In recent times industries developing systems of similar complexity have come to respect the value of systems engineering. Systems engineering is the glue which binds a large technical team and focuses the engineering effort towards satisfying a set of realizable customer needs. EIA/IS-632 definition of systems engineering is as follows; "Systems engineering is an interdisciplinary approach encompassing the entire technical effort to evolve and verify an integrated and life-cycle balanced set of system people, product and process solutions that satisfy customer needs."

To control and improve a process a viable set of measures must be in place. Existing measures of the strength of the systems engineering process in a specific project address only project execution (e.g. earned value) and technical performance. When applied properly these metrics provide valuable insight into the status (cost and schedule) of a project and a products ability to meet customer needs. However, few of these existing measures are progressive in nature and as such fail to provide early warnings of systems engineering process failure. What are needed are prognostics for the systems engineering effort; gauges to provide predictions of future events which impact product cost, schedule and/or performance. The Lean Aerospace Initiative (LAI), working with the International Council on Systems Engineering (INCOSE), released a guide (in Beta form) in December of 2005 outlining a progressive set of thirteen leading indicators to address this need. This set of metrics has yet to be been verified against an active or historical project but provides a starting ground for additional research.

Thesis Supervisor: Dr. Donna H. Rhodes
Title: Principal Research Engineer, Lean Aerospace Initiative

Acknowledgements

I would like to thank my thesis advisor Donna Rhodes for her guidance and patience. I would like to thank the SDM cohort and teaching staff, without their abundant insights and willingness to share knowledge I would not have gained so much through the SDM experience.

I would like to thank those within the organization where I am employed who provided continued support for my degree pursuit.

I would like to thank my children, Sean and Tara, for allowing me to smile when stress was at a peak.

Most of all, I would like to thank my wife Melanie for her insights as well as her continual support and understanding of the many late nights and weekends I spent in pursuit of knowledge.

Table of Contents

List of Figures

List of Tables

Acronyms

ANSI	American National Standards Institute
ARINC	Aeronautical Radio Incorporated
BCR	Benefit Cost Ratio
CA	Contract Award
CDR	Critical Design Review
CE	Concurrent Engineering
CM	Configuration Management
CMMI®	Capability Maturity Model® Integration
CPM	Composite Performance Measure or Critical Parameter Management
CSBA	Center for Strategic and Budgetary Assessments
DFSS	Design for Six Sigma
DFX	Design for X, where X is a superset of possible 'ilities'
DoD	Department of Defense
DQA	Design Quality Assurance
EIA	Electronics Industries Alliance
ESD	Engineering Systems Division
EVA	Earned Value Analysis
EVM	Earned Value Management
FCA	Functional Configuration Audit
FCS	Future Combat Systems
FY	Fiscal Year
GAO	General Accountability Office
GQM	Goal Question Metric
HSR	Hardware Specification Review
ICD	Interface Control Document
IEEE	Institute of Electrical and Electronics Engineers
INCOSE	International Council Of System Engineers (previously known as NCOSE)
IPPD	Integrated Product and Process Development
IPT	Integrated Product/Process Team
JIT	Just In Time
LAI	Lean Aerospace Initiative
LeanSE	Lean Systems Engineering
LPD	Lean Product Development
LRU	Line Replaceable Unit
MIT	Massachusetts Institute of Technology
MOE	Measure of Effectiveness
MOP	Measure of Performance
NASA	National Aeronautics and Space Administration
PCA	Physical Configuration Audit
PD	Product Development
PDP	Product Development Process
PDR	Preliminary Design Review
PM	Project Manager, Project Management or Program Manager

PO	Purchase Order
PSM	Practical Software Measurement
RFC	Request for Change
RFP	Request for Proposal
RSI	Requirements Stability Index
SAGE	Semi-Automatic Ground Environment
SDM	Systems Design and Management
SE	Systems Engineer or Systems Engineering
SECM	Systems Engineering Capability Model
SECOE	Systems Engineering Center Of Excellence
SEI	Software Engineering Institute
SEMP	Systems Engineering Master Plan
SOW	Statement of Work
SPS	Systems Performance Specification
SRR	System Requirements Review
SSCI	Systems and Software Consortium
SSR	Software Specification Review
SVT	System Verification Test
TPM	Technical Performance Measure
TRL	Technology Readiness Level
TRR	Test Readiness Review
USD	United States Dollars
VOC	Voice of the Customer
VOP	Voice of the Process
VSM	Value Stream Mapping

1 Introduction

1.1 *Motivation*

Organizations developing products in the aerospace and defense industry have recognized value in the deployment of systems engineering practices in product development since the days of the Atlas ballistic missile program. Over the past two decades several other industries developing systems of similar or greater complexity have espoused systems engineering practices. Systems engineering is the glue which binds a large technical team and focuses the engineering effort towards satisfying a set of realizable customer needs. The International Council of Systems Engineers (INCOSE) defines systems engineering as *"An interdisciplinary approach and means to enable the realization of successful systems"*

During the 1990s the aerospace industry focused on ways to achieve the later part of the industry mantra "faster, better, cheaper". This was driven by an ever shrinking defense budget and ever increasing price competition in the commercial market. In the last few years there has been elevated governmental spending[1] for complex system of systems endeavors such as the U.S. joint services Future Combat Systems (FCS) and NASA's return to the moon and ensuing mission to Mars. In the commercial sector, two "bet the company" level of investment aircraft programs are in development at Boeing and Airbus which could lend to changes in the global air transportation system. Yet there is still a call in the industry to decrease overall systems cost while vastly improving the time to market. Most continue to view systems engineering as a key enabler to attain these goals. However, there is a desire for greater control of the systems engineering process.

To realize improvement in process efficiency and value delivered a viable set of measures must be engaged which permit process control. The teachings of Stewart, Deming and others have been employed to develop process measurements in context of elevating the production process with success. Existing product development measures in a specific

---

[1] The Center for Strategic and Budgetary Assessment (CSBA) estimates the DoD budget will increase, reaching $502 billion USD in fiscal year (FY) 2011. The FY 2006 DoD budget requested was $419 billion, a 3% (inflation adjusted) increase from the previous fiscal year.

project predominantly address project execution (e.g. earned value analysis). A subset of project development projects extend measurement programs beyond project execution and track a select set of critical technical performance measures (e.g. Technical Performance Measures (TPM)). When applied properly, these metrics provide valuable insight into the cost and schedule of a project and a products ability to meet a subset of critical customer needs. Both of these are employed in a reactionary mode to a project already off course. Few existing measures are progressive in nature and as such fail to provide early warnings of systems engineering process failure. Greater value is obtainable from a set of prognostics for the systems engineering effort; gauges to provide predictions of future events which impact product and system cost, schedule and/or performance. The Lean Aerospace Initiative (LAI), working with the International Council on Systems Engineering (INCOSE), released a guide, in beta form in December 2005, outlining a progressive set of thirteen leading indicators to address this need. Although this set of metrics has yet to be verified against an active or historical project, it has provoked increased interest in developing, verifying and institutionalizing a system for systems engineering effectiveness prognostics.

1.2 *Objective*

The objective of this work is to study the value of employing a set of the right prognostics of the systems engineering process in a project setting and across the enterprise. The onset of this is to begin to validate use of the systems engineering leading indicators as recommended by LAI. The next higher goal of this thesis is to extract measurement processes from LAI's work and other systems engineering and product development metrics, such as used in design for six sigma (DFSS), in order to suggest a systems engineering effectiveness measurement system which can be used as part of a decision system to more effectively manage systems engineering in systems or product development.

1.3 *Scope*

The author's area of expertise is primarily in development of avionics, both on vehicle and ground support network aspects, i.e. a total system perspective. Avionics crosses industry boundaries; suppliers of avionic are part of the defense, electronics and software industries. Avionics often includes both the underlying embedded systems employed on the vehicle of interest and complex networks of ground support systems. Certain types of avionic systems qualify as software intensive systems as defined by Keating[2]. Figure 1-1 is a top level view of a system of systems employed for vehicle fleet management. The complexity of the onboard portion of this system, which is interconnected to essentially every line replaceable unit in the avionic suite on the vehicle and includes a comprehensive human vehicle interface, wanes in comparison to the complexity of the network centric ground based systems which feed on data from the onboard subsystem.

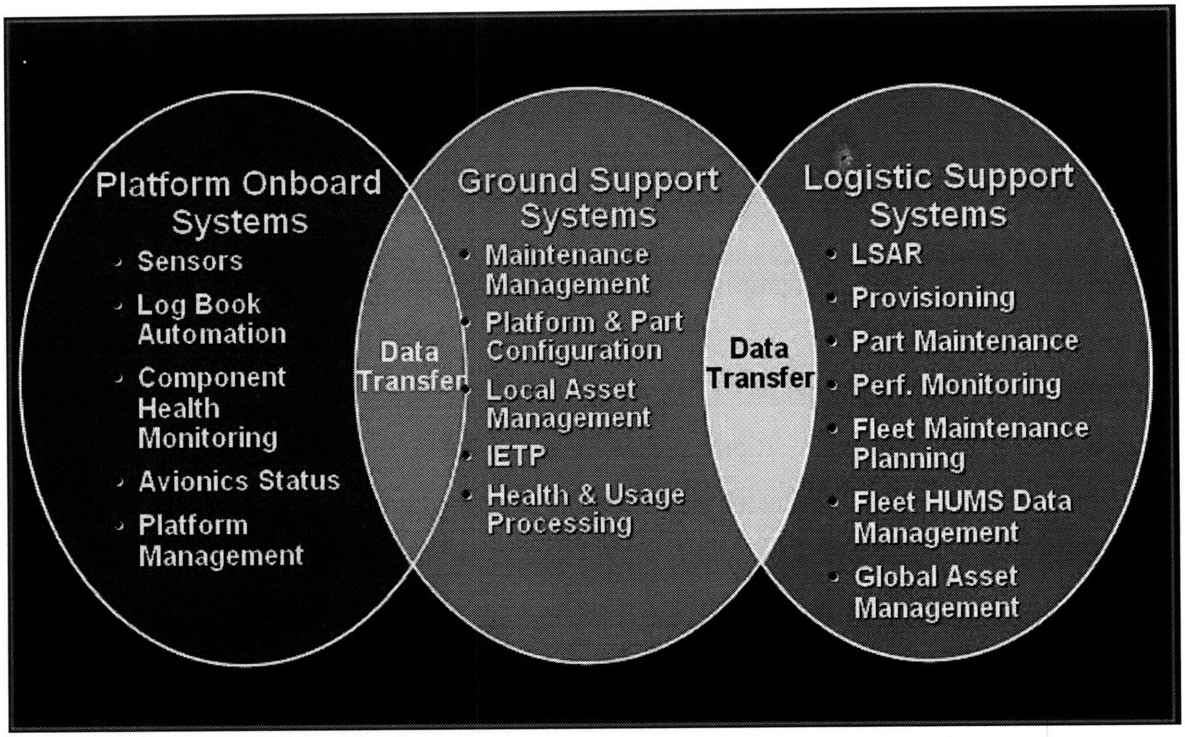


Figure 1-1 Vehicle Fleet Management System Architecture View

---

[2] Keating, Charles, et. al. "System of Systems Engineering", Engineering Management Journal, Vol 15 No 3 Sept 2003

This thesis will focus on the application of systems engineering to the industries served by the avionics community. Systems engineering processes have been employed in a vast number of industries from aerospace to pharmaceutical. A focus on avionics should not imply the methods presented are not applicable outside of this industry, but tailoring and validation in other industries are left to the reader or for future research.

1.4 *Research Methods*

First, a qualitative approach was taken to devise a set of prognostics of the effectiveness of systems engineering within a project and the organization. This synthesis of a set of prognostics relied on the outline of leading indicators per LAI's Systems Engineering Leading Indicator Guide. Next, a case study was performed which applied the measurement system on historical project data. Finally, through a series of interviews with practicing project participants, systems engineers and other engineering managers, an initial assessment of the perceived value of employing systems engineering prognostics at the project and organizational level is extrapolated.

1.5 *Thesis Overview*

Chapter 1 specifies the motivation, scope and objective of this thesis.

Chapter 2 provides an overview of the current state of systems development, the systems engineering role and metrics. It provides a view of what is being measured, how the measurement is applied, who is involved and why firms should measure systems engineering practices. This chapter highlights issues in systems development which support a need for additional control mechanisms such as systems engineering process prognostics.

Chapter 3 provides the suggested systems engineering prognostics. This is predominantly a summary of the leading indicators suggested by LAI/INCOSE in the recently published leading indicators guide book. Additional indicators and suggestions for tailoring of a minimal set of systems engineering leading indicators on a project bases are also discussed.

Chapter 4 presents a case study of the application of the measurements from the previous chapter.

Chapter 5 surveys the perceived value of and barriers to implementing a system of prognostics in a product development enterprise. Interview results are discussed in relation to the systems engineering prognostics outlined in Chapter 3 and applied in Chapter 4.

Chapter 6 presents the conclusions and suggestions for future research

2 Current State of Systems Engineering and Metrics

This chapter provides a brief review of the evolution of systems engineering, the perceived value of systems engineering and the use of measurement for assessment and control in complex human organizations.

2.1 *Foundations of Systems Engineering*

Systems engineering emerged onto the product development environment in the 1950s. Complex projects of this time employed what became the foundation of systems engineering practices. These projects existed both within the commercial sector, e.g. the telephone system, and the defense sector, e.g. the Semi-Automatic Ground Environment (SAGE) air defense system and the Atlas Intercontinental Ballistic Missile program. Systems engineering evolved in the organizations developing these highly complex systems because of the need for rapid progression from inception to system operation with high system reliability.

Systems engineering has evolved into a standard and effective practice for the development of complex systems. Systems engineering gained professional recognition in 1990 with the recognition of the National Council on Systems Engineering (NCOSE) which five years later became the International Council on Systems Engineering (INCOSE).

INCOSE defines systems engineering as;

An interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem. Systems Engineering considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs.[3]

Since the professional recognition of systems engineering, several organizations, including INCOSE, have published systems engineering guides, frameworks, and other forms of data

---

[3] INCOSE Systems Engineering Handbook, Version 3, June 2006

on the systems engineering practices and processes. Table 2-1 lists a sampling of the definitions of systems engineering in print.

Table 2-1 Systems Engineering Definitions in Practice

Definition	Source
Systems engineering is the application of scientific and engineering efforts to: 1. Transform an operational need into a description of system performance parameters and a system configuration through the use of an iterative process of definition, synthesis, analysis, design, test and evaluation. 2. Integrate related technical parameters and ensure compatibility of all related, functional and program interfaces in a manner that optimizes the total system definition and design. 3. Integrate reliability, maintainability, safety, survivability, human and other such factors into the total technical engineering effort to meet cost, schedule and technical performance objectives.	MIL-STD-499A, (1974)
An interdisciplinary approach encompassing the entire technical effort to evolve and verify an integrated and life-cycle balanced set of system, people, product, and process solutions that satisfy the customer needs.	EIA/IS 632, (1994) (Same used in 499B)
An interdisciplinary, collaborative approach that derives, evolves, and verifies a life-cycle balanced system solution which satisfies customer expectations and meets public acceptability.	IEEE P1220, (Draft 1994)
A robust approach to the design, creation, and operation of systems	NASA Systems Engineering Handbook, (1995)
An interdisciplinary approach and means to enable the realization of successful systems.	EIA/IS 731.1, (1998)
A multidisciplinary engineering discipline in which decisions and designs are based on their effects on the system as a whole	Rechtin and Maier, The Art of System Architecting, (2000)
An interdisciplinary engineering management process that evolves and verifies an integrated, life-cycle balance set of system solutions that satisfy customer needs	DoD System Management College, (2001)

Definition	Source
The interdisciplinary approach governing the total technical and managerial effort required to transform a set of customer needs, expectations, and constraints into a product solution and support throughout the products life. This includes the definition of technical performance measures, the integration of engineering specialties toward the establishment of a product architecture and the definition of supporting life-cycle processes that balance cost, performance, and schedule objectives.	CMMI Framework, (2002)
A process for designing systems that begins with requirements, that uses and/or modifies architecture, accomplishes functional and/or physical decomposition, and accounts for the achievement of the requirements by assigning them to entities and mandating oversight on the design and integration of these entities..."	MIT ESD Internal Symposium Committee Overview, (2002)
The function of Systems Engineering is to guide the engineering of complex systems. Systems Engineering is focused on the system as the whole – it emphasizes total operation. It looks at systems from the outside, that is, at its interactions with other systems and its environment, as well as from the inside.	Kossiakoff & Sweet, (2003)
The application of scientific and engineering efforts to: (1) transform an operational need into a description of system performance parameters and a system configuration through the use of an iterative process of definition, synthesis, analysis, design, test and evaluation, and validation; (2) integrate related technical parameters and ensure the compatibility of all physical, functional, and program interfaces in a manner the optimizes the total definition and design; and (3) integrate reliability, maintainability, usability (human factors), safety, producibility, supportability (serviceability), disposability, and other such factors into a total engineering effort to meet cost, schedule and technical performance objectives	Blanchard, Benjamin, (2004)
The multidisciplinary application of analytical, mathematical, and scientific principles to formulating, selecting and developing a solution that has acceptable risk, satisfies user operational need(s), and minimizes development and life cycle costs while balancing stakeholder interests	Wasson, Charles S. (2006)
SE is a discipline that concentrates on the design and application of the whole (system) as distinct from the parts. It involves looking at a problem in its entirety, taking into account all the facets and all the variables and relating the social to the technical aspect.	FAA Systems Engineering Manual, (2006)

A common theme sewn throughout these definitions is systems engineering embodies a robust framework for successful development of engineered systems. What constitutes a system is a matter of perspective[4]; e.g. an electrical engineer might consider a form of integrated circuit a system. However there is minimal value added to the product development process applying a traditional systems engineering framework in the development of low complexity products. Low complexity products generally incur limited development uncertainty and are manageable with ad hoc development processes which cost far less then traditional systems engineering practices. The systems engineering process adds value when applied to the development of complex human engineered systems[5]. Professor Ed Crawly defined a complex system as;

- having many interrelated elements and interfaces
- having many levels of elements and sub-elements
- requiring a great deal of information to specify.[6]

Some examples of complex human engineered systems include aircraft carriers, space craft, a production facility for building automobiles, personal computers, and missile systems. The success of super systems, system of systems and/or family of systems such as the central artery project ("Big Dig") in Boston, depend on the proper and diligent application of systems engineering concepts, practices and processes.

2.2 *Systems Engineering Management and Leadership*

Other aspects of systems engineering prevalent in Table 2-1 are technical orchestration of the systems development phase as well as cross functional domain design engineering

---

[4] INCOSE Systems Engineering Handbook, Version 3, June 2006

[5] Kossiakoff, Alexander, et. al. "Systems Engineering Principles and Practice", Wiley, 2003

[6] Crawley, Ed, System Architecture Course Lecture Notes, MIT, Fall 2004

leadership. The Venn diagram, Figure 2-1[7], provides a view of the technical management process aspects in scope of the systems engineering domain. These include;

- management of project risk,
- management and fosterage of customer interaction and
- definition of and control of the work breakdown structure required to successfully deliver the technical aspects of the system desired by the customer, within cost and schedule constraints.

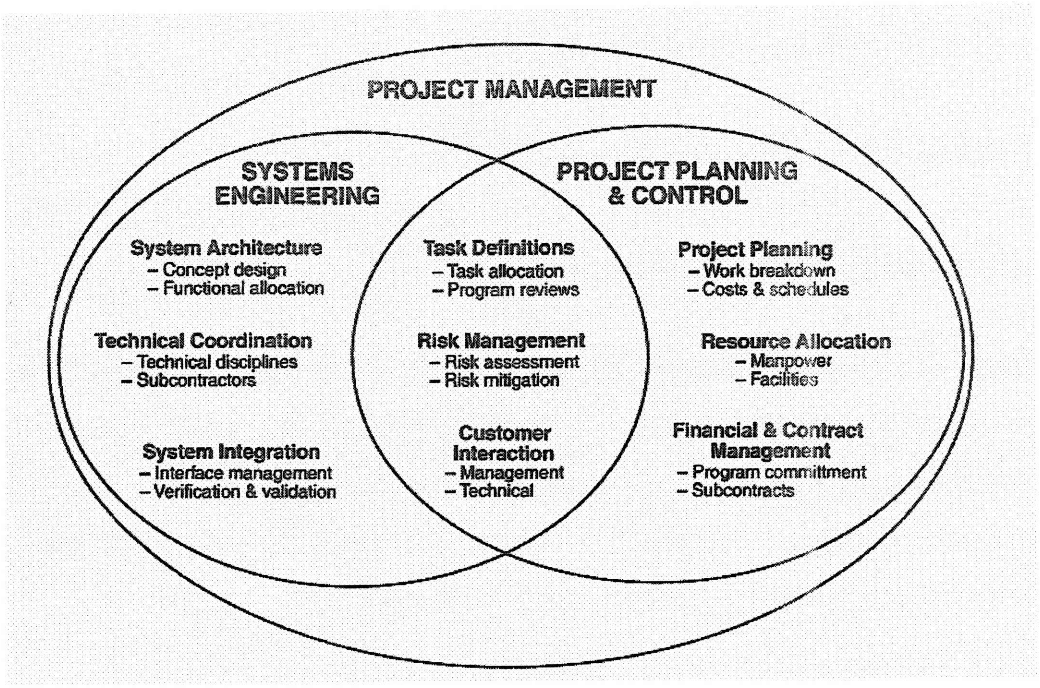


Figure 2-1 PM SE Overlap

Dasher[8] decomposed product development management into three elements; technical management, business management and contract management. In product development organizations the systems engineer (SE) is traditionally responsible for project technical management reporting to the program manager (PM), who is responsible for the overall

---

[7] Kossiakoff, Alexander, et. al. "Systems Engineering Principles and Practice", Wiley, 2003

[8] Dasher, George "The Interface Between Systems Engineering and Program Management" Engineering Management Journal, Vol 15 No. 3, Sept 2003

program and/or project. The systems engineers' focus is primarily within a project context, Figure 2-2. In order to effectively and efficiently delivery value in a balanced manner inputs and outputs to the project environment and those at the next level up, the enterprise, must be well understood.

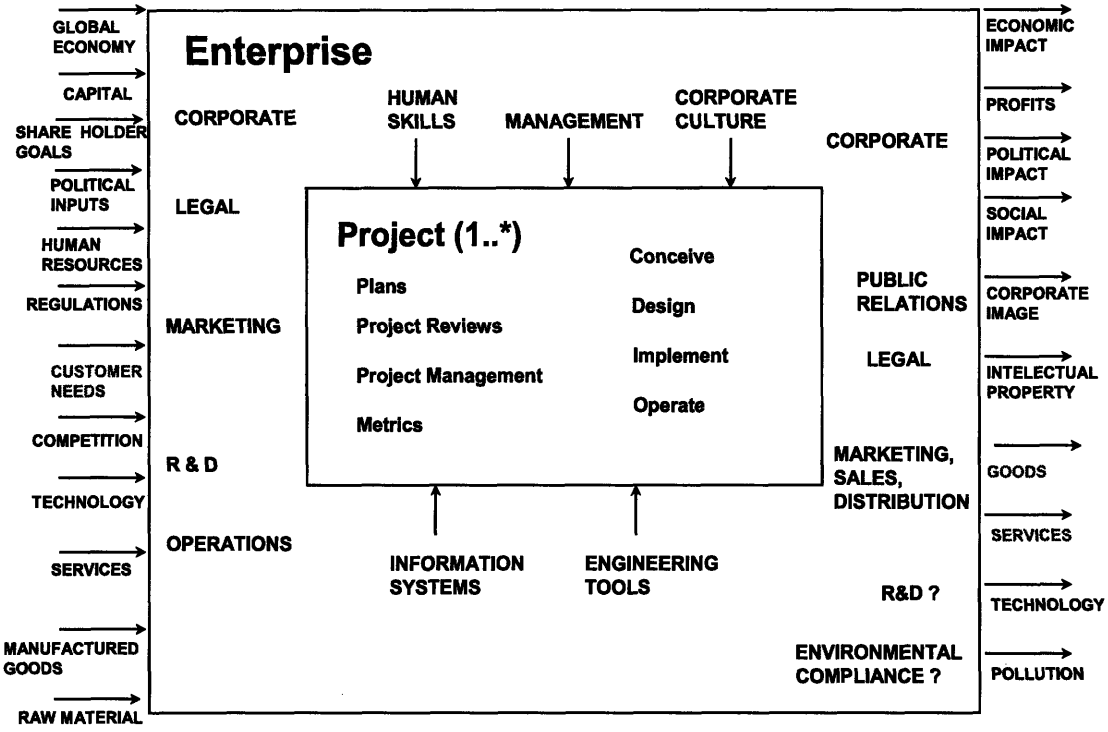


Figure 2-2 System/Product Development Project Context[9]

The roles the systems engineer and program manager play in the domain of projects often is blurred. The systems engineer tends to focus on leadership of the technical aspects of the project including acting as the primary engineering interface to external and internal stakeholders. To effectively lead in this one domain the systems engineer must also consider the overall balance with the needs of the other domains. Dasher noted "Fundamentally, the health of any contract-based business depends on meeting performance requirements on schedule and within costs targets. PM and SE share responsibility for these goals and only through close interaction and open communications

---

[9] Adapted from slide "Enterprise Context", Crawley, Ed, System Architecture Course Lecture Notes, MIT, Fall 2004

can success be assured."[10] Dasher also indicates this communication as especially in the early phases of project planning.

2.3 *Systems Thinking*

Systems engineering focuses on viewing a complex system from a "30,000 ft" perspective; the whole system, its desired operation, in the intended environment, as well as the equivalent perspective of the systems to which it is interconnected. Bahill and Gissing wrote "systems engineering is a grand unified theory of making things work better."[11] The application of such holistic thinking to aspects of human existence has been deemed "System Thinking." Senge parallels system thinking to a "fifth discipline", stating "system thinking as a framework for seeing interrelationships rather than things, for seeing patterns of change rather than static "snapshots"."[12] It has been proposed that top tier senior systems engineers are the system thinking archetype.[13]

Jay Forrester contributed an invaluable tool for the practice of system thinking when he leveraged practices common in control theory for engineering electronic and mechanical systems to develop System Dynamics[14]. System Dynamics is a system modeling tool which provides a means to model the dynamic nature of complex systems, e.g. urban social systems, and gain insight into system behavior. System Dynamics models have been employed to understand what drives human organizations, predict outcomes of adjustments to improve ecologic systems and economic systems. System Dynamics models of product development organizations show promise as useful tools for management to gain an understanding of the effect of organizational decisions.[15] These models have been

---

[10] Dasher, George. "The Interface Between Systems Engineering and Program Management", Engineering Management Journal, Vol 15, No 3, Sept 2003

[11] Bahill, Terry and Gissing, Bruce. "Re-Evaluating Systems Engineering Concepts Using Systems Thinking", IEEE Transactions on Systems, Man, And Cybernetics—Part C: Vol. 28, No. 4, Nov 1998

[12] Senge, Peter. The Fifth Discipline: The Art and Practice of Learning Organization, Currency Doubleday, 1994

[13] Davidz, Heidi, "Enabling System Thinking To Accelerate The Development of Senior Systems Engineers", MIT Thesis 2006

[14] Sterman, John. Business Dynamics: System Thinking and Modeling for a Complex World, McGraw-Hill, 2000

[15] MacInnis, Daniel. "Development of a System Dynamics Based Management Flight Simulator for New Product Development", MIT Thesis 2004

exercised to highlight the impact to a project of effective team communications, value of having the right people and benefit of getting design right early to reduce rework.

2.4 *Systems Science*

At the root of the concepts of several theories in complexity and complexity management is an area of study known as cybernetics. Cybernetics originated in the 1940s as an evolution from information theory and general systems theory as proposed by Shannon and von Bertalanffy. Cybernetics is been defined as;

"the science of effective organization, of control and communication in animals and machines. It is the art of steersmanship, of regulation and stability. The concern here is with function, not construction, in providing regular and reproducible behavior in the presence of disturbances. Here the emphasis is on families of solutions, ways of arranging matters that can apply to all forms of systems, whatever the material or design employed. It is the science of the black box, in which the how is irrelevant and only the what matters (similar in one way to behavioral thinking in psychology)."[16]

Louis Couffignal[17] in his 1956 paper on the subject summarized cybernetics as "the art of ensuring the efficacy of action." A noteworthy axiom from this field is the "Law of Requisite Variety" which Ashby[18] states as;

"Only variety in R [regulator] can force down the variety due to D [disturbance]; only variety can destroy variety."

This states in order to control a complex system the variety of the responses available to the regulator, i.e. the control mechanism, must be at least as numerous as the variety of the system itself. The degree of variety sufficient to control an instance of a complex system is known as the systems "requisite variety". Researchers of cybernetics also noted that as the complexity of a system increased the variety of behaviors proliferates.

---

[16] Lucas, Chris. "Cybernetics and Stochastic Systems", http://www.calresco.org/lucas/systems.htm, paper V1.0 June 1999

[17] Couffignal, Louis, "Essai d'une définition générale de la cybernétique", *The First International Congress on Cybernetics*, Namur, Belgium, June 26-29, 1956, Gauthier-Villars, Paris, 1958, pp. 46-54.

[18] Ashby, W.R. *Introduction to Cybernetics*. Methuen, London, UK, 1956

The theory of requisite variety plays part in the development of measurement systems aimed at controlling with some confidence a set of sub activities within the complex and stochastic nature of a project development system which acts on a set of requirements and other inputs to output a complex system. It drives the fundamental understanding that measurement must be diverse in order to stand a chance at viable control. It also indicates a measurement system that does not account for the variety in the project development "system" supports a suboptimal controller and might become a forcing function for undesirable outcomes.

2.5 *System/Product Development Process*

An enterprise's system and/or product development process is its recipe for conceiving, developing, producing and sustaining a system in a manner which optimizes stakeholder value. It is common for the enterprise system development process to provide a generic sequencing of the system development phase, guidance on standard inputs and standard interim expectations. The process is a means by which an enterprise propagates lessons learned and best practices. It defines a framework which supports a common set of quality and management checkpoints of each projects developmental activity the organization undertakes.

Figure 2-3 depicts a system lifecycle typical to the aerospace or defense industry. An acquirer, such as a government, uncovers a need for a product or system.

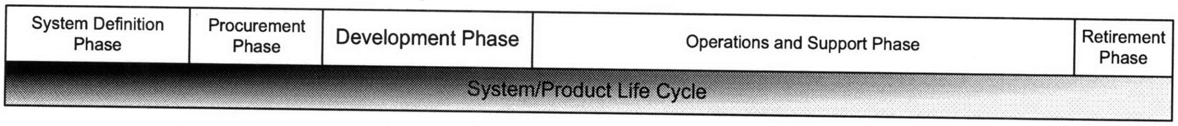


Figure 2-3 System/Product Lifecycle

The acquirer develops a definition of the needs the system will satisfy and the environment it will be expected to operate within. This needs definition or problem space definition is provided to potential developers for bid, often in the form of a request for proposal (RFP). Upon receipt of a RFP each potential developer evaluates the business case to determine the applicability and value to their enterprise of perusing the business. This evaluation might include developing a set of feasible solutions, estimating recurring and non recurring costs and assuring fit into the firm's core competencies. At the later part of the procurement phase each potential developer interested in the business provides the acquirer with a proposal which generally includes technical, relative historical, financial and schedule data related to their proposed solution or solution space. The acquirer reviews the proposals and selects the best value, where value includes attributes beyond cost such as risk, performance, and schedule. Contract award is often the phase gate for starting the development phase. Developers might choose to accept cost risk and enter development prior to a formal contract award as a means to reduce schedule risk if the deliverable is a core business objective.

Systems engineering leads the development team (software, hardware, the other applicable engineering practices, design support from production, etc) throughout the Development Phase. This phase includes a set of critical checkpoints in which the developer, acquirer and often other key stakeholders, review design artifacts. Each checkpoint includes entrance and exit criteria which are to be satisfied in order to continue development into the next lower level of design. These checkpoints, when used as intended, increase the likelihood the system design solution properly balances the needs of the key stakeholders. Figure 2-4 depicts the basic outline of the development phase including associated checkpoints, often considered project milestones. During these upfront system life cycle phases all aspects of the downstream activities are considered, including verification & validation, production process, operational modes and states and product or system retirement/disposal. The "Vee" product development process, Figure 2-5, indicates the overlapping and iterative nature of the subphases within the development phase of the system lifecycle.

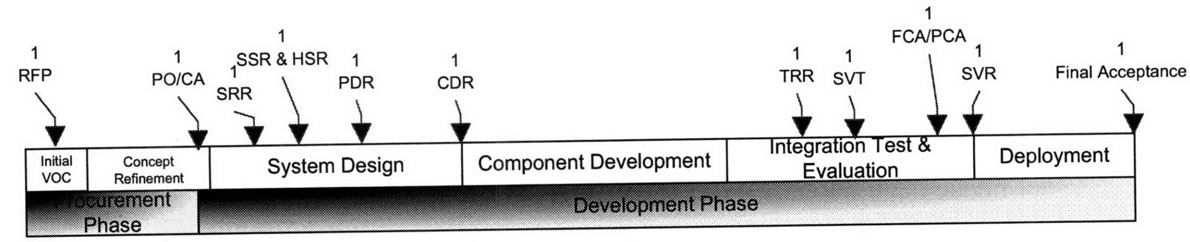


Figure 2-4 Typical Time Horizon SE Activity in Avionics Project

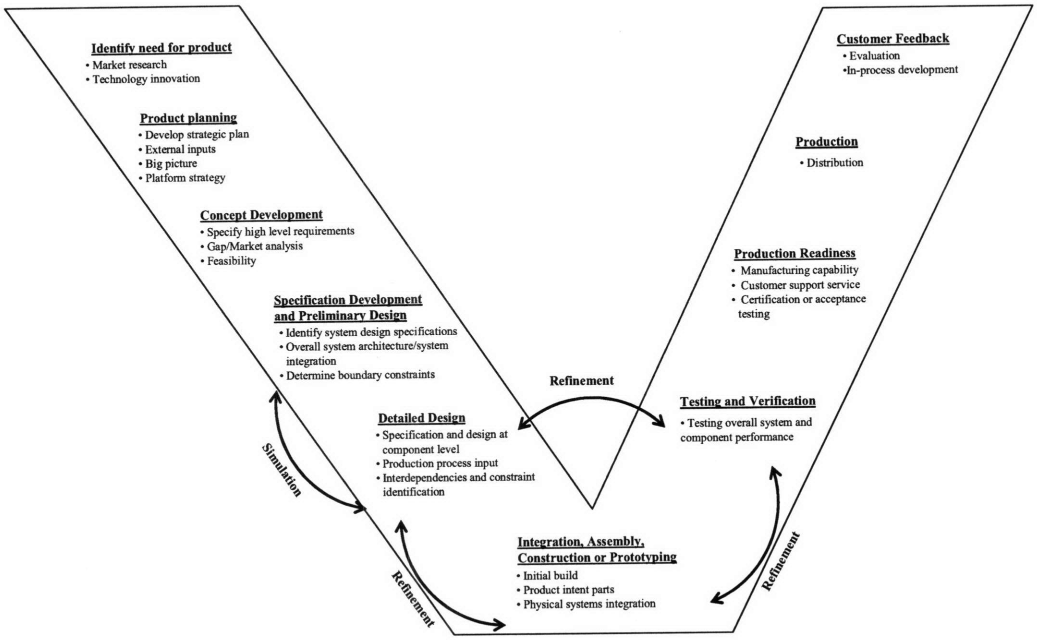
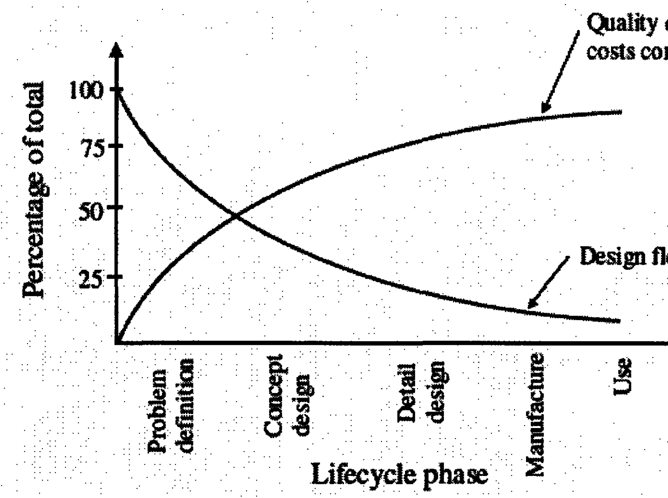


Figure 2-5 Generic "Vee" Product Development Process

The development phase culminates with the customer acceptance of the verified system solution, commonly referred to as the product baseline. The operation and support phase of avionics systems may stretch across decades. During this period, operators generally interact with the system developer through a customer support organization. This organization is responsible for technical manuals, repair, replacement parts and user support. Systems engineering is called upon to aid in correcting errors in product documentation or lend expertise in resolving field issues. When a system or product has expended its useful life, it is either retired or provided some form of life extension, e.g. refurbished. When the retirement phase is well planned during development phase, the expectation is minimal effort will be required by systems engineering during the retirement phase. Exceptions might occur in cases where the system of interest employed hazardous material, such as explosives in munitions or special material in unique electronic systems, where the effort to develop a process of disposing of the system might parallel the development effort.

It is in the early phases of a systems life cycle where many of the critical decisions are made which drive downstream quality. Figure 2-6 depicts observed trends for design flexibility and quality through the product development phases.



Source:
Russell B.
Ford and
Philip Barkan

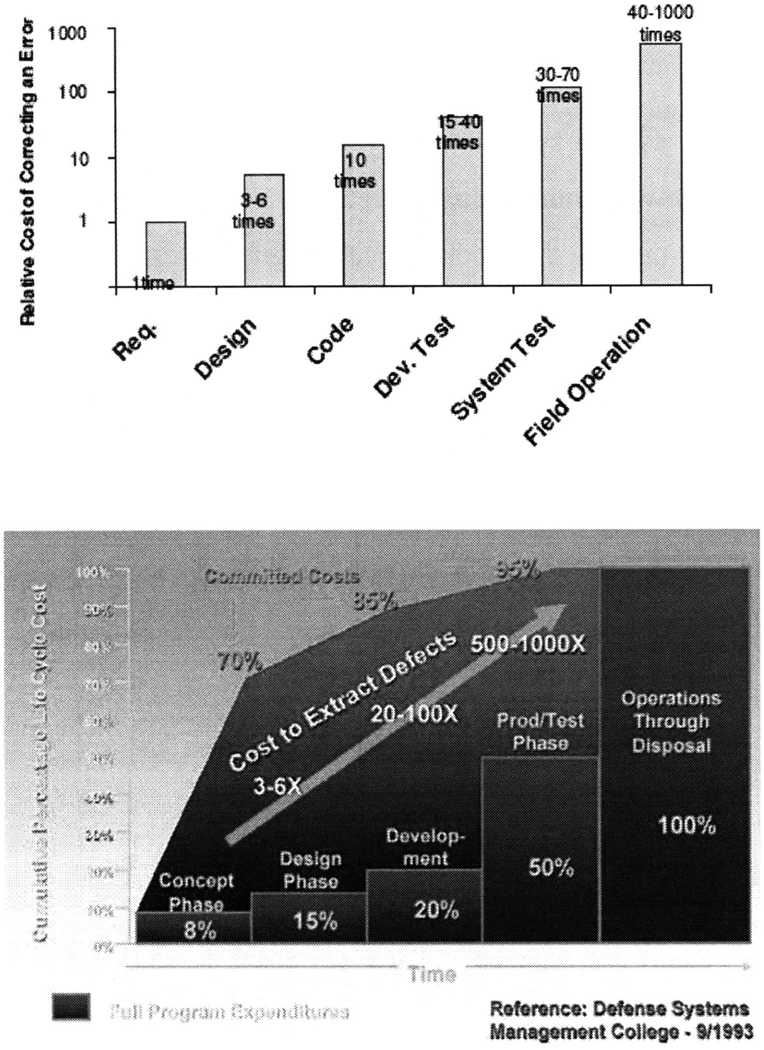
Figure 2-6 Typical Trend of Design Flexibility and Quality[19]

Several groups have estimated the cost associated with changes over the system lifecycle; two perspectives are shown in Figure 2-7. Reasons for changes during a system lifecycle include; inaccurate requirements, defects in design, changing customer needs, system environment changes, regulatory changes, tooling issues, inability to verify design or parts obsolesce. Avoidable defects and untapped opportunity left undiscovered in the early development phases can significantly increase costs or reduce the delivered value of the

---

[19] Ford, Russell B., and Barkan, Philip. "Beyond Parameter Design --A Methodology Addressing Product Robustness at the Concept Formation Stage", DE-Vol. 81, Design for Manufacturability, ASME, 1995

end solution. As the development process progresses the impact of change increases as the number of affected work products requiring rework increases.



Reference: Defense Systems Management College - 9/1993

Figure 2-7 Cost of Defects Over the System Lifecycle (Top – source Blanchard and Fabrycky[20], Bottom – source DSMC)

2.6 *Perceived Value of Systems Engineering*

Michael Wynne, Acting Under Secretary of Defense, wrote in the Feb 2004 memorandum "Policy for Systems Engineering in DoD",

---

[20] Blanchard, B.S., Fabrycky, W. J. Systems Engineering and Analysis 2nd Edition, Englewood Cliffs, NJ: Prentice-Hall, Inc. 1990

> *"Application of rigorous systems engineering discipline is paramount to the Department's ability to meet the challenge of developing and maintaining needed warfighting capability. This is especially true as we strive to integrate increasingly complex systems in a family-of-system, systems-of-systems, net-centric warfare context. Systems engineering provides the integrated technical process to define and balance system performance, cost, schedule, and risk. It must be embedded in program planning and performed across the entire acquisition life cycle."*[21]

The memorandum established a DoD policy which attempts to ensure a robust systems engineering approach is applied by projects managed by the defense department. A similar policy published by the U.S. Air Force[22] highlighted a need for incentives to assure contractors also adhere to sound systems engineering practices. Those in the commercial sector have expressed similar capability to discern the contribution of systems engineering to the success of complex engineering projects.

Honour surveyed previously published works on systems engineering effectiveness and statistical information gathered by the INCOSE Systems Engineering Center of Excellence (SECOE) as a means to evaluate the return on investment, i.e. the value, provided by systems engineering in product development[23]. This work indicated the systems engineer primarily added value by increasing focus on the early design phase which significantly reduced downstream development activity typically incurred in a more ad hoc development approach. The results reported vindicate the hypothesis that systems engineering improves development quality while reducing costs for complex systems development. The study concluded the optimal systems engineering effort (SE Quality * SE Cost / Total Project Cost) is 15 to 20%. The study noted the quality of the systems engineering effort, which was more of a subjective measurement, significantly impacts the value of systems engineering to the overall project. Costs attributed to systems engineering effort differ by organization; one organization might account the entire test effort to systems engineering where another might only account test planning and oversight.

---

[21] Wynne, Michael. Memo. "Policy for System Engineering in DoD", Feb 2004

[22] US Air Force policy memo 03A-005 "Incentivizing Contractors for Better Systems Engineering", 2003

[23] Honour, E.C., "Understanding the Value of Systems Engineering," Proceedings of the INCOSE International Symposium, Toulouse, France, 2004.

2.7 *Systems Engineering Frameworks*

There are several published works on the systems engineering framework from government and commercial organizations which provide views of the systems engineering process as applicable to target audience. One of the first and basic views of this framework was published in EIA/IS 634, Figure 2-8. This view is much like the original waterfall model for software development, leaving much to the imagination of the reader. The 2001 version of DoD Systems Management College Systems Engineering Fundamentals, Figure 2-9, expanded on the basic process view providing valuable detail to each step of the framework. The INCOSE endorsed EIA 632, Figure 2-10, revamped the original Systems engineering process to break the process out into technical management, acquisitions, design, product realization and evaluation.

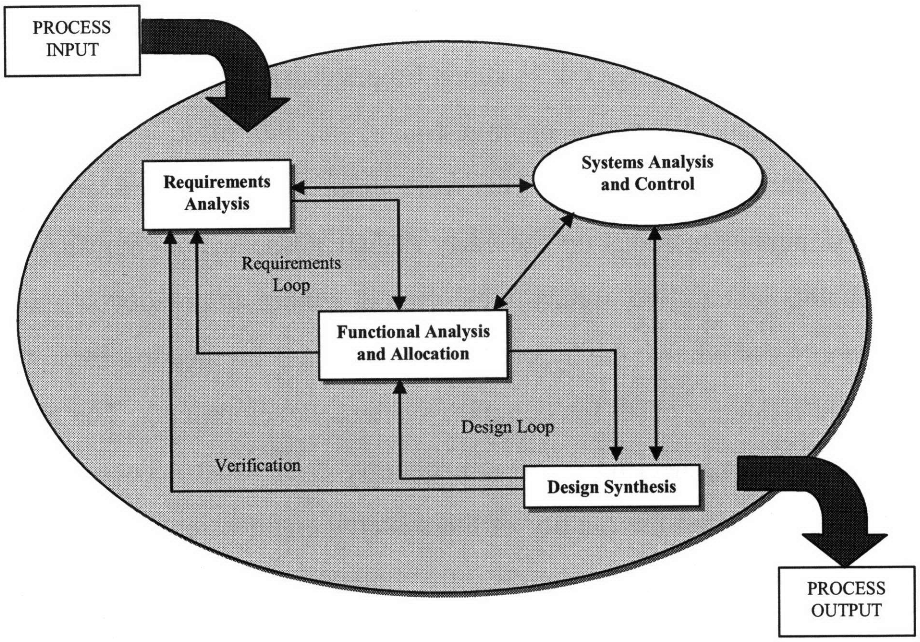


Figure 2-8 EIA/IS 634 (1994) Systems Engineering Process

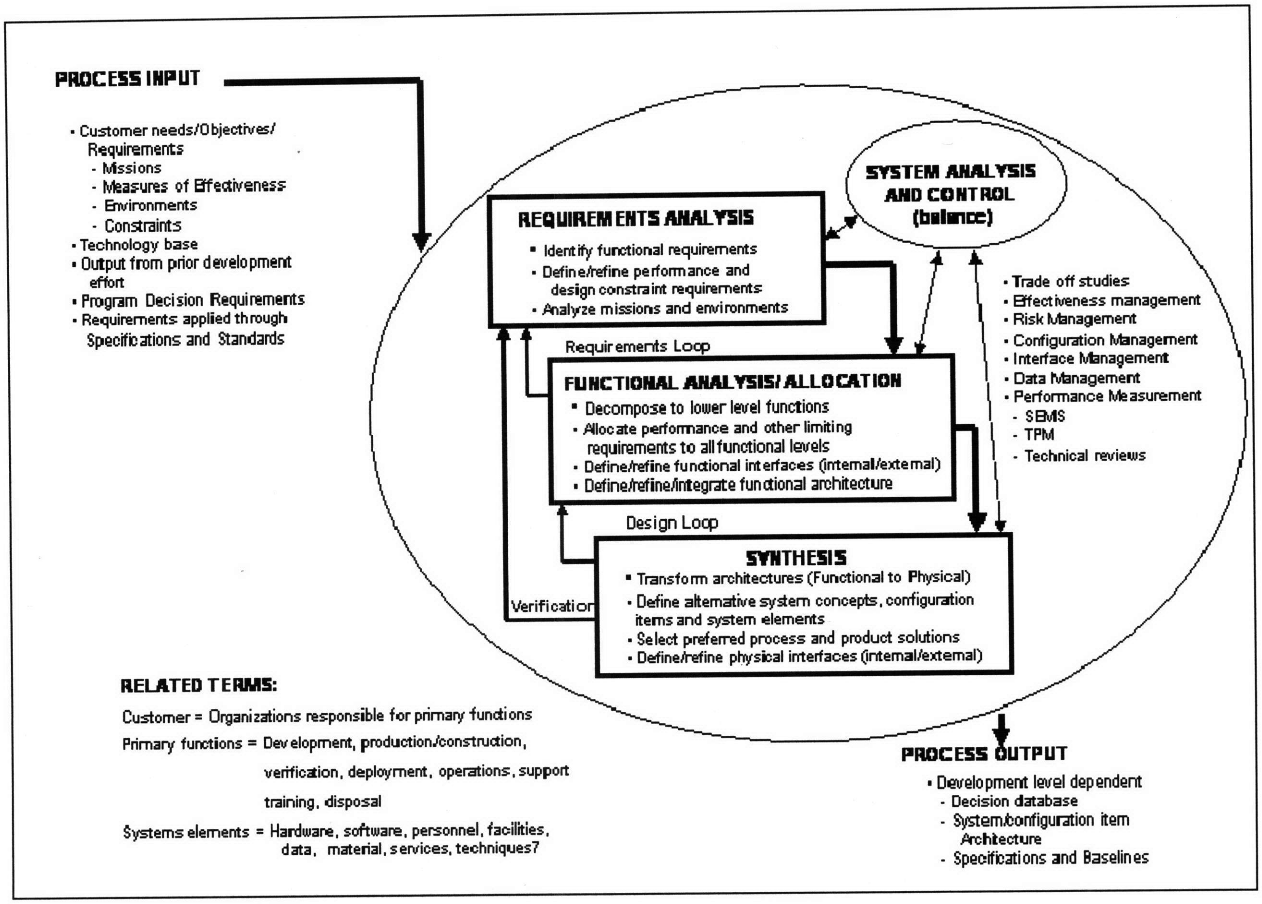
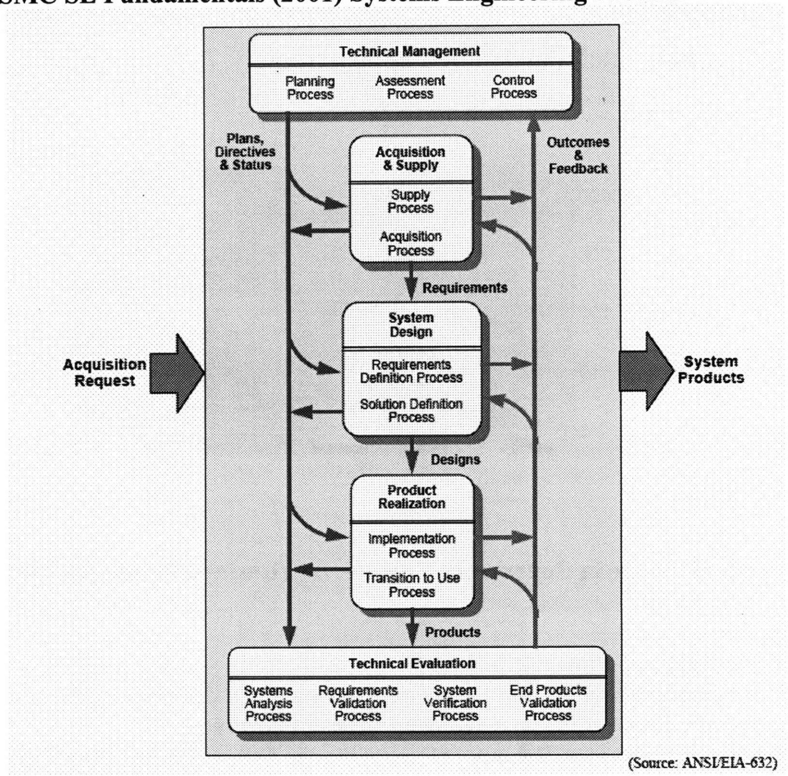


Figure 2-9 DoD SMC SE Fundamentals (2001) Systems Engineering Process



(Source: ANSI/EIA-632)

Figure 2-10 ANSI/EIA 632 (2000) (also employed by INCOSE) Systems Engineering Process

2.7.1 Concurrent Engineering

The defense industry first uttered the term “Concurrent Engineering” in the mid 80s. The Institute for Defense Analysis defined Concurrent Engineering (CE) as “a systematic approach to the integrated, concurrent design of products and their related processes, including manufacture and support. This approach is intended to cause the developers, from the outset, to consider all elements of the product lifecycle from concept through disposal, including quality, cost, schedule and user requirements.”[24] By the 90s several authors suggested concurrent engineering as the process to improve time to market, gain leaps in quality, develop flexible products and better meet true customer needs. The “engineering” in concurrent engineering is misleading[25] as this process is a product development process which incorporates the voice of the enterprise and the voice of the customer early in the development process in an effort to greatly increase the likelihood the right solution, one which meets the customer need, can be produced rapidly and efficiently, etc., is fostered into production. A focus on eliciting active participation of stakeholders during the upfront phases of the product lifecycle allowed concurrent engineering to affect the area of development which determined approximately 70% of the lifecycle cost[26].

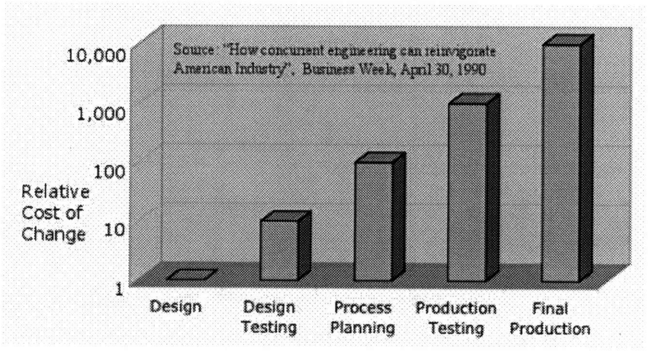

Source: "How concurrent engineering can reinvigorate American Industry", Business Week, April 30, 1990

Figure 2-11 Value in Concurrent Engineering (Business Week, Apr 30, 1990)

---

[24] Institute for Defense Analysis, Report R-338, 1986

[25] Gardiner, Geoff. “Concurrent and System Engineering: same thing, different name, or are they both just new product introduction”, Engineering Management Journal, Feb 1996

[26] Flint, Lynne Thompson and Gaylor, Dean A., “Expanding the Effectiveness of the Conceptual Design Phase: An Industrial Application of the Stanford Design for Manufacturability Method,” Design for Manufacturability, Vol. 81, 1995, pg. 99-104

Design for *X* (DF*X*), where the *X* represents the various downstream processes of concept selection and design, is a key outcome of development of concurrent engineering. Design for Manufacturing is perhaps the oldest and most publicized DF*X* entities; Figure 2-12 outlines several other entities of DF*X*.

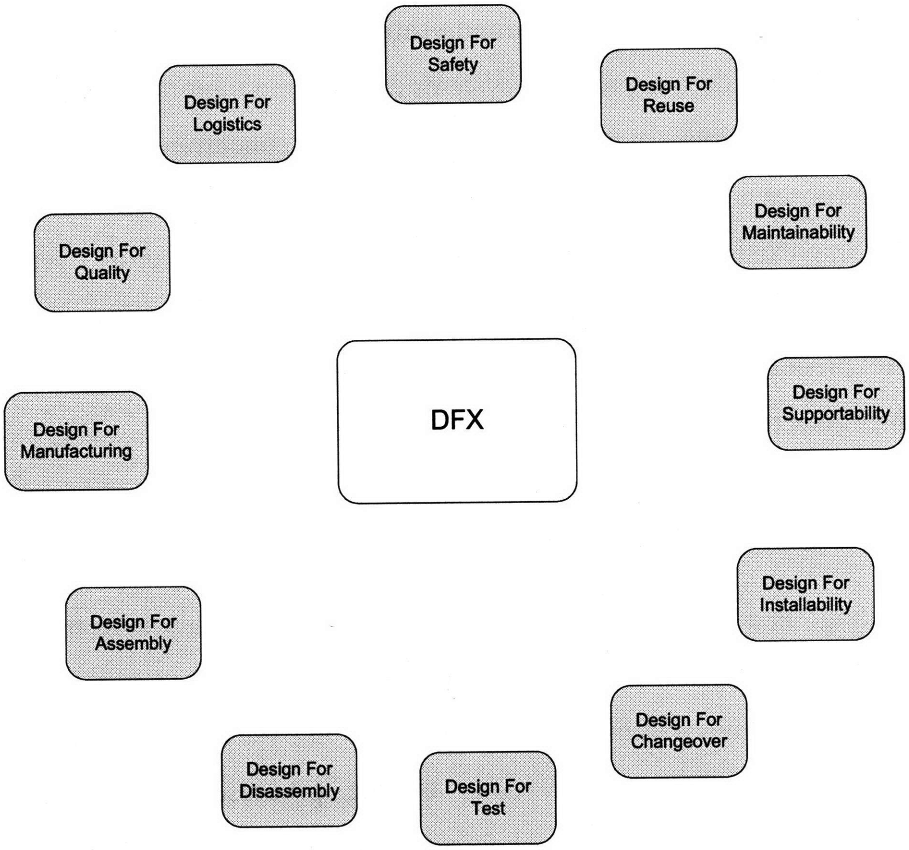


Figure 2-12 Concurrent Engineering, Design For X

Another version of concurrent engineering espoused by the Department of Defense in the 90s was integrated product and process development (IPPD). At the core of IPPD are the integrated product/process teams (IPT), which "simultaneously optimize the product, product manufacturing, and supportability to meet system cost and performance objectives."[27] IPTs brought together experts from various functional domains and

---

[27] DoD Regulation 5000.2R, "Mandatory Procedures for Major Defense Acquisition Programs and Major Automated Information Systems Acquisition Programs," Chapter 5, Paragraph C5.1, April 2002.

organizations outside of the developing entity early in the development as a means to assure voices from the upstream and downstream processes effected the early phase development and concept selection. Through these teams a development organization made certain the initial design took into account DF*X*.

Concurrent engineering is arguable an extension of systems engineering, it can be classified as a specialization of systems engineering applied to product development with a great focus on cross functional team communications. The efforts of concurrent engineering in the last decade provided systems engineering important insight into the value of early interdisciplinary communications. Today, concurrent engineering is for all intents and purposes a label of the past, however many of the sub processes have been incorporated into existing systems engineering practice. Those fostering Lean Product Development and Lean Systems Engineering are moving the industry's systems development processes forward, but not without incorporating lessons learned under concurrent engineering such as IPTs.

2.7.2 Lean Systems Engineering

"Lean Thinking" is a term readily identified with the automobile industry as it was first encountered by most readers in the book "The Machine That Changed the World"[28] which dissected its origins from within the Toyota Production System. The premise of "lean" in manufacturing comes from observations of the processes employed by Japanese automobile manufactures. The central focus of lean is the continuous minimization of process waste as a means to maximize the value delivered to all stakeholders. Value is enhanced primarily through an increased responsiveness to customer demand while simultaneously improving product quality to a world class level. Womack and Jones introduce Five "Lean Principles"[29] which are central to the lean framework; specify value, identify value stream, make flow continuous, let customer pull value and pursue perfection. Womack and Jones further reduce product generating activities into three classes; value added, required non-value and non-value added. Murman categorizes production process

---

[28] Womack, James and Jones, Daniel. The Machine That Changed the World, HapperPerennial, 1990

[29] Womack, James and Jones, Daniel. Lean Thinking, Free Press, 2003

waste into seven fundamental types; defects, overproduction, transportation, movement, waiting time, inventory and processing.[30]

One of the tenets of lean is involving those who do the work in process improvement activities. During one such activity, Value Stream Mapping (VSM) events, representatives from various stakeholder groups map out the process of delivering value to the stakeholders, looking to root out all non-value added activities. The output of VSM, the "future state" map, represents the perceived least waste incurring means to deliver value. Authors from the Lean Aerospace Initiative (LAI) provide additional insight into the application of lean in the aerospace and defense industries, where production quantities are considerably less then experienced in the auto industry and significant cyclic product demand exists[31]. Figure 2-13 provides the progression of value in lean as depicted in Lean Enterprise Value.

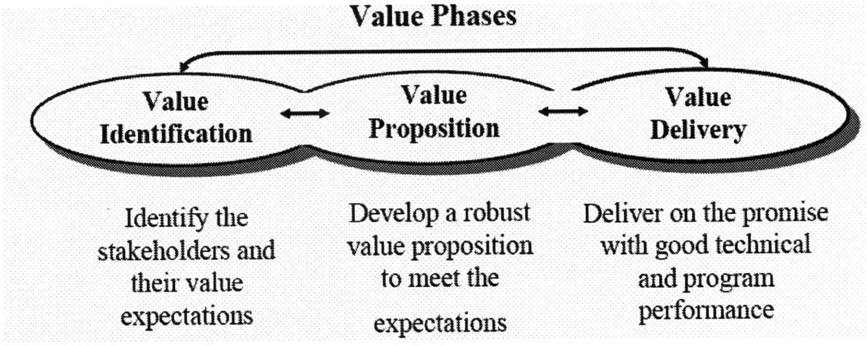


Figure 2-13 Value Phases

Recent applications of Lean Product Development (LPD) apply select methods and nomenclature prevalent in Lean Thinking to the product development process. The uncertainty which exists naturally in the PD process drives the need for different lean applications from those used in the production process. Those proposing LPD are carefully evaluating how to add value without erroneously tagging value added or required waste in the processes as waste, e.g. planned rework cycles. McManus stated "Iteration is not always a waste in PD processes, in fact managed iterations may be more desirable than a

---

[30] Murman, Earl, et. al., Lean Enterprise Value, Insights from MIT's Lean Aerospace Initiative, 2002
[31] Murman, Earl, et. al., Lean Enterprise Value, Insights from MIT's Lean Aerospace Initiative, 2002

slower 'right the first time' process especially early in PD."[32] This is not to imply that there are not opportunities for lasting continuous improvement to be gained in PD through the elimination of unneeded repetition. The product of a PD process is design information and it is the processing of information for which the five lean principles must be applied.[33] Millard translated the seven production wastes into PD wastes;

Table 2-2 Production Wastes Restated as PD Wastes

Waste	Description in PD Context
Overproduction	Creating unnecessary information
Inventory	Keeping more information than needed
Transportation	Inefficient transmittal of information
Movement	Related to movement to access information
Waiting	Approvals, releases, information, etc
Defects	Insufficient quality of information, requiring rework
Overprocessing	Working more then necessary to produce the outcome

---

[32] McManus, PDTTL Roadmap, 2005

[33] Oppenheim, "Lean Product Development Flow", Systems Engineering, Vol. 7, No. 4, 2004

Morgan[34] expanded on the PD wastes, to include "Lack of System Discipline" and "Ineffective Communication" which are direct reflections of inadequate systems engineering practices.

At the company interviewed for the case study, the lean product development process being phased into practice focuses on the voice of the customer (VOC) through a series of structured question and answer sessions with the key stakeholders from the customer and end user community. The companies VOC process specifies the type and scope of the questions to be asked as a VOC activity progresses as well as the proper seating arrangement for optimal value added customer – development team interaction. Figure 2-14 shows the spectrum of stakeholders for a typical system development project which must be considered during a VOC activity.

---

[34] Morgan, G. "Government perspectives on engineering systems", MIT Engineering Systems Symposium, 2004, //esd.mit.edu/symposium/pdfs/day1-2/morganslides.pdf,

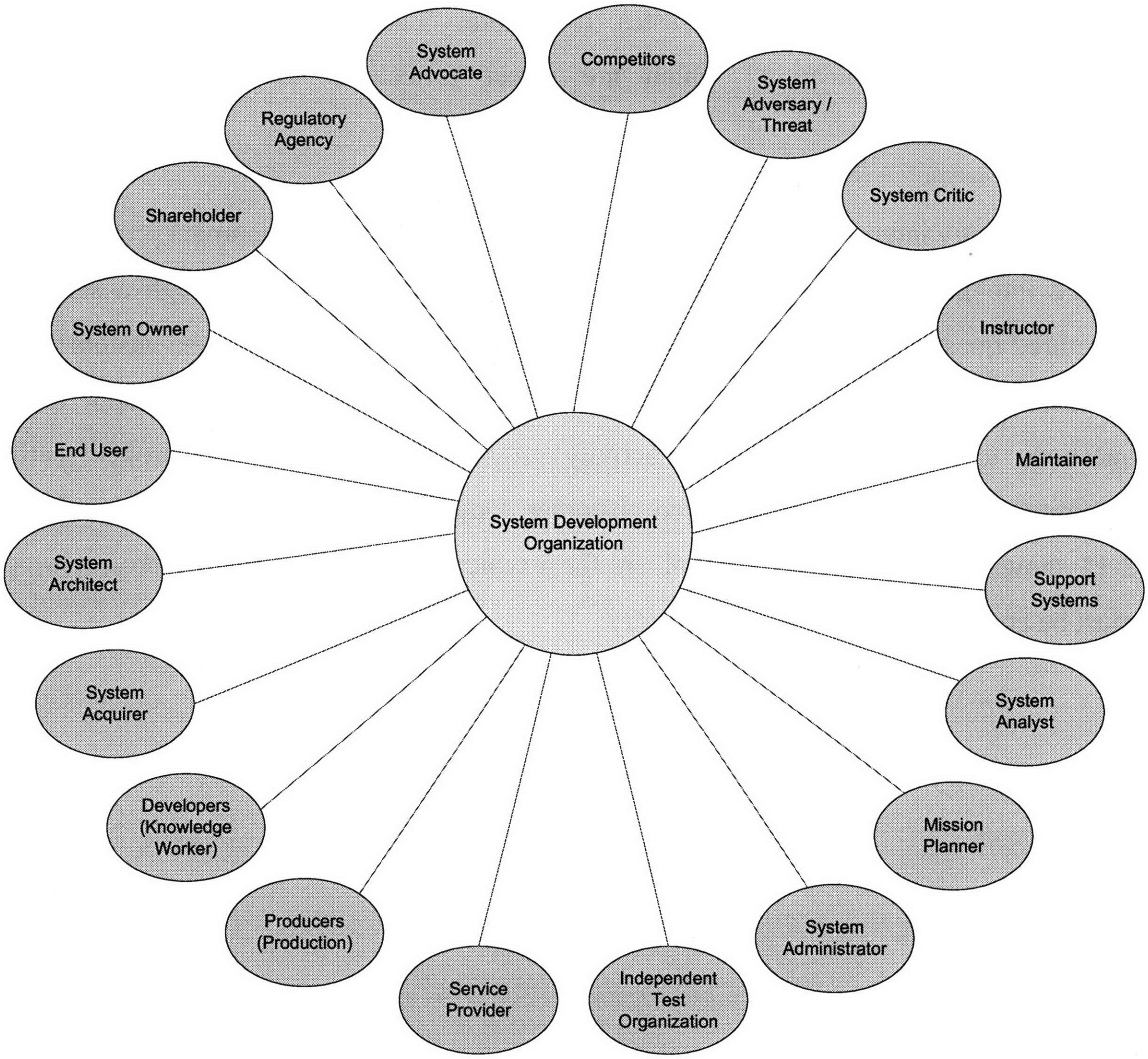


Figure 2-14 System Development Project Stakeholders

The intent of the VOC activities is to elicit intent statements and from those establish a set of key success factors for a program. Work activities are centered on delivering the value proposition identified in the VOC activities in the least waste way. The project schedule is developed following the VOC activity working back from the key customer need dates, such that development work is done in manner equivalent to the Just In Time (JIT) methodology employed under lean for production processes. To provide transparency to project activities to employees in the organization project schedules, key issues, risks and improvement activities with assigned leads and due dates are posted on large poster boards available in public areas, e.g. walls in common hallways. Project teams, including representatives from all internal stakeholder groups, meet daily in front of these visible controls to go over status and update the public project data. Conducting short duration team meetings with increased frequency instead of weekly, the preexisting company norm, reduces the likelihood of unwarranted inter-task delays. If a project incurs a critical issue, an andon light on the team's area is lit to assure management and others in engineering are aware the team is fire fighting and shifting effort off the planned or standard work activities. The enterprise has reported gains in customer satisfaction, employee satisfaction, project schedule adherence and project cost adherence at one division where the new process has been followed for over a year.

Rockwell Collin's instituted "lean electronics" which has reduced product development cycle time by 40%, reduced knowledge workers frustration and reduced overtime improving quality of life.[35] In lean electronics, lean thinking and knowledge management go hand in hand, as both are about optimizing the work flow, reducing cost and reducing cycle time. The firm extends the IPT concept to "communities of common practice" where knowledge workers gather to discuss a shared aspect of there work, e.g. a group for systems engineering practice, to assure tribal knowledge is disseminated across a more diverse group then in traditional product development organizations. Clayton Jones, Rockwell Collins CEO, indicated that lean is a journey not a destination and partaking in this journey is a requirement to remain competitive in the industry.

The Lean Aerospace Initiative proposes Lean Systems Engineering (LeanSE) as the application of the fundamentals of lean thinking to systems engineering with the objective of delivering best lifecycle value for complex systems and products. An example of lean thinking applied to systems engineering is the use of Integrated Process and Product Teams (IPPT).[36]

---

[35] Clayton M Jones, CEO of Rockwell Collins, "Leading Rockwell Collins Lean Transformation", Presentation to LAI in fall of 2006.

[36] Murman, Earll. Lecture from 16.885J, MIT 2003

Shah and Ward[37] wrote that a successful lean production process requires not the fundamental lean practices alone, but also a supporting structure which includes performance measurement and associated incentives which align with the goals of the lean practice. As industry continues to evolve lean practices in the design factory, we must also learn from lean experiences on the factory floor, including integrating a lean performance measurement system for the knowledge workers.

2.7.3 Design for Six Sigma

Design for Six Sigma (DFSS) "seeks to avoid manufacturing/service process problems by using systems engineering techniques to avoid process problems at the outset"[38] (e.g., product development (PD) fire prevention instead of PD fire fighting). DFSS promotes the use a set of systems engineering best practices such as the use of tools like the House of Quality and Design of Experiments to increase the success of the PD process to meet customer expectations. Similiar to Lean Product Development, DFSS elicites the voice of the customer as a means to define the value proposition. It relies on some of the concepts and nomenclature of Six Sigma for Quality and a has a strong reliance on measurement of the development process.

Six Sigma for operations follows a Measure, Analyze, Improve and Control (MAIC) roadmap. DFSS experts have recognized the same statisical control focus which has proven fruitful on the manufacturing floor is not ideal for product development[39]. DFSS per Creveling should follow a phase/gate PDP, with two versions of DFSS roadmaps for use based on the type of product development. The Concept, Design, Optimize, Verify (CDOV) roadmap is applicable to product design. For research projects Invent and Innovate replaces the Concept gate in the roadmap.

DFSS proposes a strong focus on technical management through measurement. DFSS Critical Parameter Management (CPM) espouses tracking of key measures of the project,

---

[37] Shah, R., Ward,. "Lean manufacturing: context, practice bundles, and performance" Journal of Operations Management, Vol 21, Page 129, 2003.

[38] Wikipedia, the free encyclopedia

[39] Creveling, et.al., Design for Six Sigma, Prentice Hall PTR, 2003

in what reads much like the systems engineering Technical Perfomance Measure (TPM) processes. One might differentiate the two through the fact that CPM attempts to draw from statistics including those employed in six sigma, e.g. the capability indexes Cp and Cpk.

2.8 *Measurement and Metrics*

"Measurement is a key element of successful management in every well established engineering discipline'"[40]

A metric is a quantitative periodic assessment of a process or product. Each metric is supported by an underlying procedure(s) for measurement or derivation as well as interpretation of the assessment. The goal of measurement is to provide information about status, quality and other attributes of a product or process as a means to enable the decision process which seeks to optimize the development process through tradeoffs in performance, cost and schedule. The use of metrics is a kin to feedback in a closed loop control system, a signal which can be measured now or in the past and is of value to improve a future outcome.

Putman and Myers state this as "The purpose of measurement (metrics) is to uncover reality ... and you uncover reality in order to do something real about it."[41]

Metrics are integral to risk and opportunity management. Metrics provide periodic quantification of areas of the project which are considered indicators of success as well as indicate the effect of risk mitigation measures instituted. Risk management extends into a larger space addressing concerns which are not easily quantifiable such as political risks.

Metrics or indicators used by economist are categorized by there temporal value.

---

[40] McCarry, John, et. al. Practical Software Measurement, Objective Information for Decision Makers, Addison-Wesley, 2001

[41] Putman, Lawrence and Myers, Ware. Five Core Metrics, The Intelligence Behind Successful Software Management, Dorset House Publishing, 2003

Lagging indicators follow the actual event of interest. These are valuable as confirmation an event did indeed occur.

Coincident indicators are real time indicators of an event. These are often used to provide insight into the real time status of related events but the indicators are more readily measurable then the event of interest.

Leading indicators predict future outcomes with some level of certainty. The certainty level of leading indicators generally decreases as the lead time, time horizon before the event of interest, increases.

Metrics programs are prevalent in the management of software projects. This is a product of the reoccurring issue of software development efforts overrunning cost and schedule estimates.

Practical Software Measurement[42] (PSM) provides the following as two key characteristics of successful measurement implementation:

- The collection, analysis, and reporting of measurement data that related directly to the information needs of the decision makers.
- A structured and repeatable measurement process that defines measurement activities and related information interfaces.

PSM prescribes two measurement concepts to address these areas; the Measurement Information Model and the Measurement Process Model, Figure 2-15.

---

[42] McCarry, John, et. al. Practical Software Measurement, Objective Information for Decision Makers, Addison-Wesley, 2001

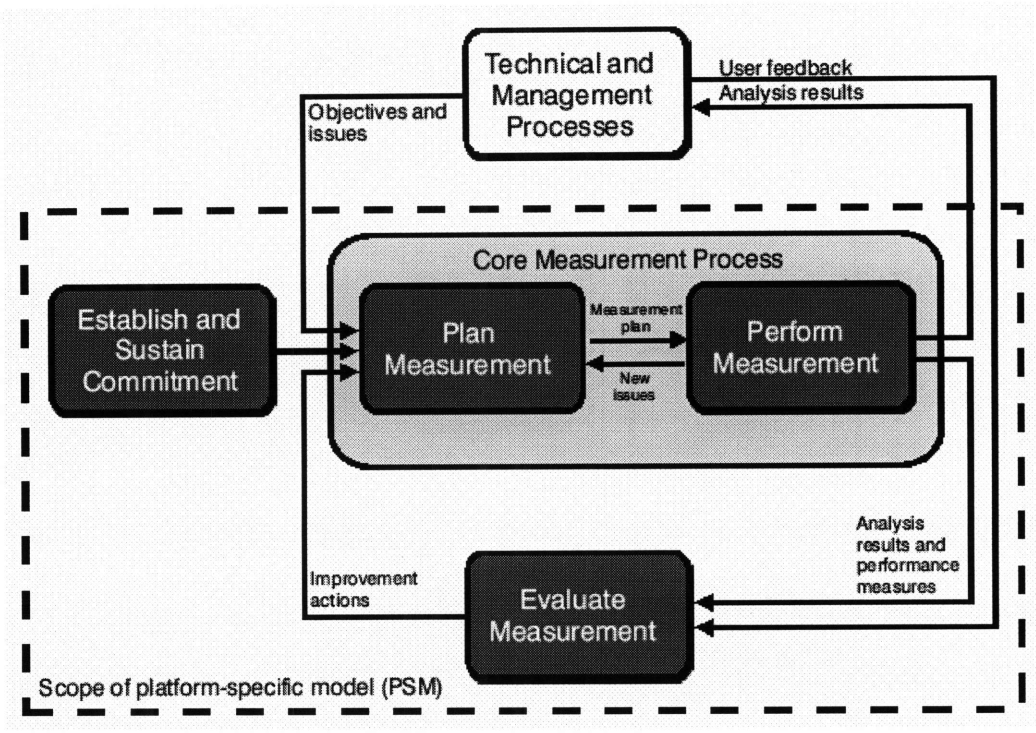


Figure 2-15 PSM Measurement Process Model

Within the PSM measurement information model there are seven common software information categories; schedule and progress, resources and costs, product size and stability, product quality, process performance, technology effectiveness and customer satisfaction. These information categories are leveraged to facilitate the identification and prioritization of a programs information needs. The measurement information model uses measurement constructs. A single measurement construct may involve three types, or levels of measures; base measures, derived measures and indicators, Figure 2-16.

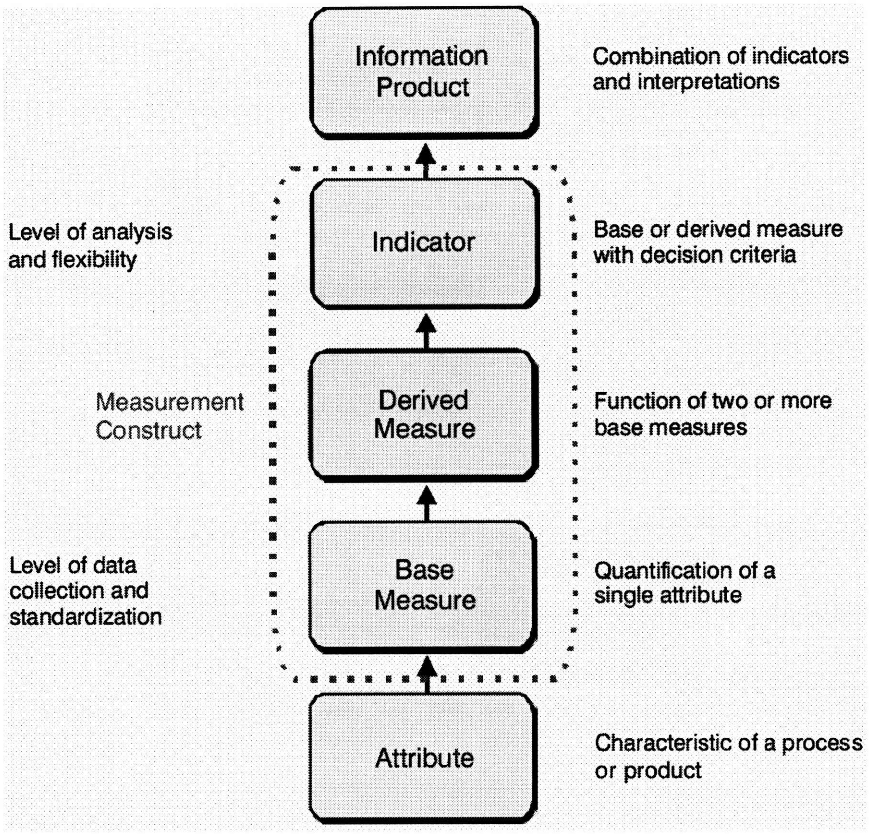


Figure 2-16 PSM Levels of Measurement Construct

Figure 2-17 is an example measurement construct for a productivity indicator from PSM.

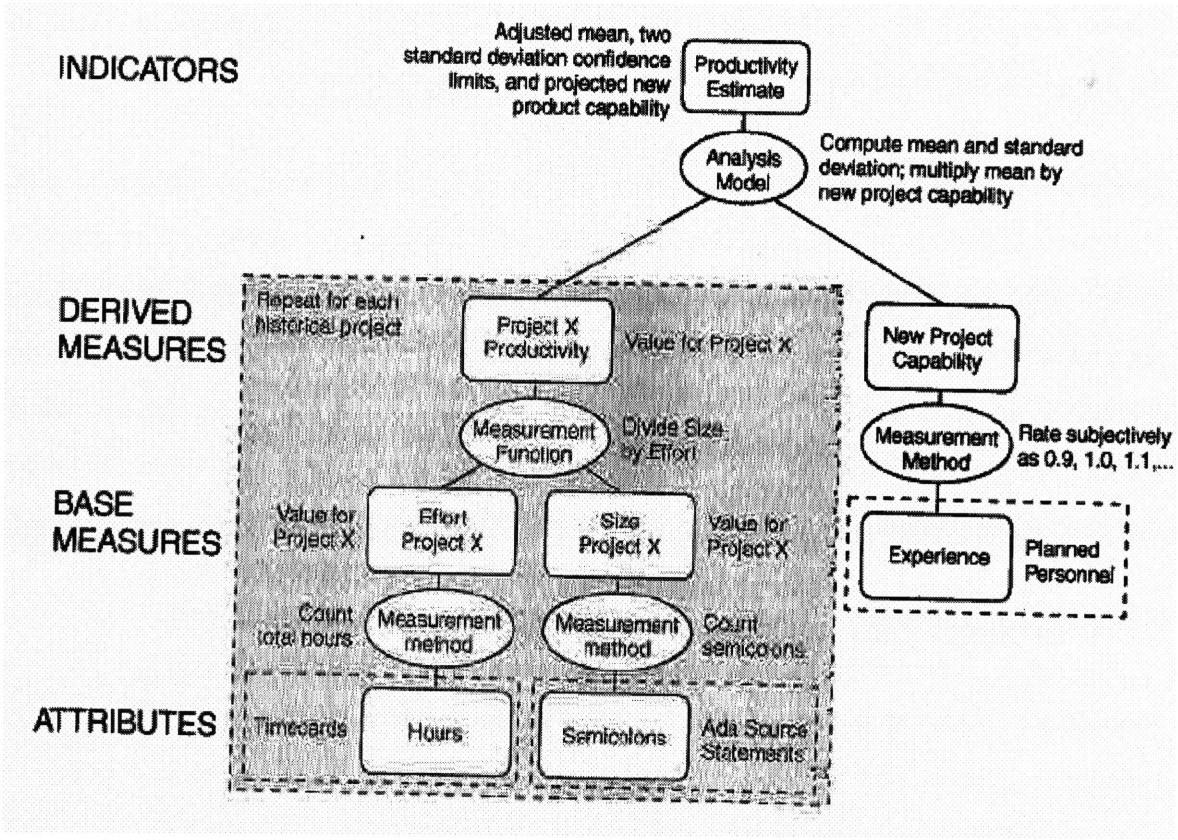


Figure 2-17 Example Indicator from PSM

Equations 1 through 3 provide the process of generating indicators in mathematical notation.

$$Y_i = f_1(X_i)$$

Equation 1 Base Measurement Function/Method

$$Z_i = f_2(Y_i, Z_a, ...)$$

Equation 2 Derived Function

$$W_i = f_3(Y_i, Z_i, \ldots)$$

Equation 3 Analysis Model for Indicator

Computation of a confidence value is not indicated in the measurement construct, but PSM recognizes "Measurement is always based on imperfect information, so quantifying the uncertainty, accuracy or importance of an indicator is an essential component of presenting the actual indicator value."[43]

Leaders of the LAI initiative to develop systems engineering leading indicators noted leaders within development organizations "measure" implicitly throughout the product life cycle. The formalization of this activity through a structured measurement program disseminates the information for use in improving decision analysis, prediction and control.[44] Measurement in engineering development is vital in creating a mechanism for creating a corporate memory. A well followed measurement program provides a historical database which can be leveraged to assure future projects gain from the lessons of the past.

2.8.1 Project/Program Management Metrics

Metrics are employed in program management to assess status of a program with respect to schedule and cost. The most widely used measurement system for program management is Earned Value Management (EVM), also known as Earned Value Analysis (EVA). EVM integrates schedule and cost tracking into a simple set of metrics useful for tracking project adherence to plan. In 2005 the DoD revised its acquisition policy to require contracts valued in excess of $20 million then year dollars to apply EVM as outlined in the EVM standard ANSI/EIA 748.[45] This action affirmed the DoDs belief "Earned Value Management integrates the scope of work, schedule, and cost to create an aggregate picture of performance, which helps ensure that day-to-day decisions on performance for development efforts are consistent with program objectives. EVM can help to mitigate cost and/or schedule overruns and provides a forecast of final cost and schedule outcomes."[46]

EVM has been criticized because it is valuable only when the initial plan is valid and fails to address measurement of quality. Often the initial plan is improperly assembled by a few

---

[43] McCarry, John, et. al. Practical Software Measurement, Objective Information for Decision Makers, Addison-Wesley, 2001

[44] LAI Knowledge Event on Leading Indicators, January 10th 2007

[45] Wynne, Michael. Memorandum from, Under Secretary of Defense, "Revision to the DoD Earned Value Management Policy", Mar 7 2005

[46] Waldron, Roger D. Memorandum from, Deputy Chief Acquisition Officer Office Of The Chief Acquisition Officer "Implementation of Earned Value Management System (EVMS) Policy in GSA" Aug 19 2005

or single individual developing a set of goals without eliciting stakeholder buy in or the overall business resource plan. An executive interviewed during the case study noted this style of project planning, which he referred to as the "Goal without a Plan (GWAP)" approach was often a major contributor to project cost overruns. Browning, et. al. note planners often omit accounting for unplanned rework during the development cycle, where rework is often inevitable as new information unearthed after design begins drives change. Rework and late information often drive the last 10% of the planned schedule to take half of total planned project time.[47] The Project Management Institute (PMI) indicates through practitioners following good project and program management processes, as those outlined in PMI's PMBOK, planners can account for the "last 10%" phenomena, the rework cycle and improve the veracity of the EVM process.

EVM has become an industry standard for project management, however there are other PM metrics in practice which provide significant value and don't risk dilution of information content by merging schedule and cost data. The 45° chart, a chart of plan versus actual for each major task, has recently seen revitalization as a simple but telling means to indicate the status of activities within a project schedule, as some struggle with inadequacies in the application of EVM. The key, per one company's continuous improvement manager, is the organization institutionalizing a concise set of PM metrics that fit the needs of those working the job and those managing the organization and use the metric set consistently across the organization.

2.8.2 Organizational Systems Engineering Capability Measurement

The Systems Engineering Capability Model (SECM), discussed in EIA/IS-731[48], provides a measurement system to evaluate a systems engineering organization. The Capability Maturity Model® Integration (CMMI) provides a similar assessment path, with additional focus on the software engineering and IPPD processes. Both of these use a series of questions (CMMI also includes interviews) focused around key process areas which are

---

[47] Browning, Tyson, et. al. "Adding Value in PD by Creating Information and Reducing Risk", IEEE Transactions on Engineering Management, VOL 49, No 4 Nov 2002

[48] I. Minnich, "EIA/IS 731 Compared to CMMI" Systems Engineering (INCOSE) V5N1, pg 62-72

used to classify the systems engineering organization into one of six rankings. Figure 2-18 and Figure 2-19 provide the key areas for each of the models.

Technical Category	Management Category	Environment Category
Define Stakeholder and System Level Requirements	Plan and Organize	Define and Improve the Systems Engineering Process
Define Technical Problem	Monitor and Control	Manage Competency
Define Solution	Integrate Disciplines	Manage Technology
Assess and Select	Coordinate with Suppliers	Manage Systems Engineering Support Environment
Integrate System	Manage Risk	
Verify System	Manage Data	
Validate System	Manage Configurations	
	Ensure Quality	

Figure 2-18 SECM Focus Areas

Process Management Category	Project Management Category	Engineering Category	Support Category
Organizational Process Focus	Project Planning	Requirements Management	Configuration Management
Organizational Process Definition	Project Monitoring and Control	Requirements Development	Process and Product Quality Assurance
Organizational Training	Supplier Agreement Management	Technical Solution	Measurement and Analysis
Organizational Process Performance	Integrated Project Management	Product Integration	Decision Analysis and Resolution
Organizational Innovation and Deployment	Risk Management	Verification	Causal Analysis and Resolution
	Quantitative Project Management	Validation	

Figure 2-19 CMMI Process Areas

These rankings progress from a base level of initial (incomplete in CMMI) to optimized. Both of these systems were developed as a means for an organization to benchmark its systems engineering capability versus other organizations.

2.8.3 Systems Engineering Technical Performance Measures

A Technical Performance Measure (TPM) is a critical technical parameter a project tracks to ensure technical requirements of a product are realized.[49] The TPM process is the primary existing project technical control mechanism managed by the systems engineering

---

[49] INCOSE, Systems Engineering Measurement Primer, V1, 1998

team. The process is a vehicle to continually assess the adequacy of the architecture and later the design to satisfy the stated critical customer needs. "TPMs are the "key" parameters that allow the program manager and systems engineer to determine the "health" of the design process."[50]

Tracking and reporting TPMs can easily become a very time consuming and cost bearing activity. TPMs should be selected by the lead systems engineer for few key elements of a system for which adherence to requirements are critical or perceived as critical by the customer. Wasson[51] suggests four to six TPMs per each development specification with traceability to a critical top level system performance specification measure of effectiveness (MOE). A logical TPM for the development of a cell phone might be power consumption per unit time or battery life. A logical TPM for an aircraft development program might be empty weight.

Typically each TPM consists of a threshold, upper limit, lower limit or both, which the attribute can not exceed at time of delivery. TPM plots will often provide a design to target line which allows for more deviation from the target at the start of the development process and approaches the target with a narrowing allowable range as the phase proceeds. TPMs are intended to be updated at a reasonable interval, often monthly but this varies depending on the planned period for the development cycle. In short development cycles weekly or biweekly reporting might be more appropriate. TPMs are generally reviewed with the customer at each milestone review. In development programs where IPTs are in place, TPMs are assigned to IPTs and reviewed at each IPT meeting. At first, updates are estimates based on engineering analysis until actual measurements can be ascertained. At each major update a predicted trend line is projected into the future, often to the next scheduled milestone, based on engineering judgment. Other implementations of TPM tracking indicate best, nominal and worst case point estimates instead of more common single point estimates, as a means to indicate the uncertainty at each measurement.

---

[50] Blyler, John, "Interface Management, Managing Complexity at the System Interface", IEEE Instrumentation & Measurement Magazine", March 20004

[51] Wasson, Charles. System Analysis, Design, and Development, Wiley, 2006

Solomon promotes integration of TPMs with EVM[52]. This includes tracking target values and measurement on the program plan for TPMs. It is suggested these go beyond tracking of technical milestones and become integration of the TPM plan with the project schedule.

Browning, et. al. propose a Composite Performance Measure (CPM) which attempts to merge a set of TPMs into one index which leverages utility theory to improve information content.

2.8.4 The Importance of Selecting the Right Measurements

"Be careful here - just because you can measure something does not mean it is a useful metric". Rear Adm. Dave Antanitus[53]

There are several attributes of value added metrics which must be considered when instituting a measurement plan. INCOSE suggests seven attributes of good metrics[54];

- Relevance – free of multiple interpretations and pertinent to a desired end result
- Completeness – measurement set should be balanced
- Timeliness – information provided in time to provide value in the future
- Simplicity – easy to collect, analyze and understand
- Cost Effective – the benefits of the indicator must out way the measurement collection and processing costs
- Repeatability – provides the ability to compare across time and program boundaries

---

[52] Solomon, Paul. "Integrating Systems Engineering with Earned Value Management", Defense AT&L, May June 2004

[53] Solomon, Paul. "Integrating Systems Engineering with Earned Value Management", Defense AT&L, May June 2004

[54] INCOSE, Systems Engineering Measurement Primer, V1, 1998

- Accuracy – measurement is accurate and resulting indicator accurately serves the intended purpose.

The cost associated with management of a set of metrics can exceed the value added if focus expands beyond the critical aspects of a project. The realization of this is what has driven TPMs to focus on a limited set of key parameters linked to critical MOEs instead of expanding to measuring all system attributes.

Hauser and Katz address the issue of the measurement system driving organizational behavior, for good or bad.

> *The link is simple. If a firm measures a, b, and c, but not x, y, and z, then managers begin to pay more attention to a, b, and c. Soon those managers who do well on a, b, and c are promoted or are given more responsibilities. Increased pay and bonuses follow. Recognizing these rewards, managers start asking their employees to make decisions and take actions that improve the metrics. Soon the entire organization is focused on ways to improve the metrics. The firm gains core strengths in producing a, b, and c. The firm becomes what it measures*[55]

Hauser and Katz go on to define seven pitfalls to counterproductive metrics and seven steps to effective metrics, summarized in Table 2-3.

---

[55] Hauser and Katz, "You are What You Measure!", European Management Journal, Vol. 16 No. 5, pp 516-528, 1998

Table 2-3 Seven Steps toward Effective Metrics and Seven Pitfalls that Lead to Counterproductive Metrics (Hauser & Katz)

Seven Steps Toward Lean, Effective Metrics	Seven Pitfalls that Lead to Counterproductive Metrics
1. Start by listening to the customer	1. Delay rewards
2. Understand the job	2. Using risky rewards
3. Understand the interrelationships	3. Making metrics hard to control
4. Understand the linkages	4. Losing sight of the goal
5. Test the correlation and test manager and employee reaction	5. Choosing metrics that are precisely wrong
6. Involve managers and employees	6. Assuming your managers and employees have no options
7. Seek new paradigms	7. Thinking too narrowly

It is not uncommon for metrics to be collected as a means to an end, or to restate in the terms of "lean thinking" they become a non-value added activity. This occurs perhaps as part of a bureaucratic need for information without a plan for use or to collect a large set of numbers with the intent only to propagate those which shine a positive light on the upstream process owner. These types of misuse of metrics serve as obstacles in instituting a successful measurement program and ultimately effects project success as management relies on the rhetoric. Kerr[56] indicated that people modify their behavior or actions to ensure positive performance even if this means inappropriate course of action.

---

[56] Kerr, S. "HBR Case Study: The Best-laid Incentive Plans", Harvard Business Review, 2003

Metrics are often a statistic and when developing a metrics system one must keep in mind the infamous saying "There are three types of lies; lies, damn lies and statistics" which is attributed to Mark Twain. The wrong metrics support a belief network which can drive the wrong behavior.

2.9 *The Value Proposition for Improved Feedback Mechanism in the Systems Engineering Practice*

"Increasingly, I'm convinced that the systemic problem is in the field of systems engineering", Secretary of the Air Force Dr. James D Roche.[57]

A half a century of active use of systems engineering in the product development landscape seems a considerable epoch to optimize the application of systems engineering. What evidence exists in the 21st century that methods for improved control of system or product development are needed? The answer might seem evident for those directly involved in complex system development. For those distant from this landscape or just entering, there are numerous case studies and articles addressing project budgets exceeded, schedules overrun and non complaint product performance. A significant number of troubled development activities attribute their state to inadequate management of requirements, ill defined interface management, lack of internal and/or external communications, incomplete product validation and verification and other activities within the domain of the systems engineering practice. There are examples across several industry boundaries. A sampling of major system and/or project failures attributed in part or whole to poor systems engineering are listed in Table 2-4.

Table 2-4 Sampling of System Failures with SE Contribution

Project / System	Description
Therac 25 (1985)	Medical device used for radiation treatment. Machine over radiated some patients leading to several deaths

---

[57] Roche, James. Air Force Times, 24 Jun 2002, statement made by Secretary of the air force Dr. James D Roche

	Root Cause: Error in embedded software but ... questionable system level decision to migrate to software control for critical safety control and inadequate system testing are at heart of failure in delivered product.
Ariane 5 Launcher (1996)	Lost guidance and attitude information shortly after start of the main engine ignition sequence. Error in software reused from previous program. Integration and final system test should have caught this issue. It was also note inadequate requirements specification practices obscured the ability extract rationale underlying critical design decisions.
Hubble Space Telescope (1990)	Error in production process for main mirror but error should have been identified in test. Systems issue was inadequate system test methodology.
FAA Advance Automation System (1990s)	Proposed replacement for the existing air traffic control system, well existing in the early 1980s. Of the several primary causes of the eventually cancellation of the project was unmanageable requirements creep inserted by the customer, the government.

In 2003 the DoD reported on the status of the Tri Service Assessment Initiative Systematic Analysis[58]. Of the critical program performance problems identified 61% occurrences were attributed to systems engineering. Of 16 programs reporting requirement issues, 11 had systems engineering issues. 43% of the assessed projects had interoperability issues. Reported systems engineering deficiencies included lack of the application of systems engineering practice where appropriate, lack of required systems engineering expertise, poor systems engineering implementation and dispersion of systems engineering responsibility. The systems with the highest technical complexity suffered the greatest gaps between the expected and actual effectiveness of the systems engineering practice.

Perhaps one of the most complex system of systems undertaken by the DoD in development at the present time is the Army's Future Combat Systems (FCS). In late 1999 Army leadership introduced a strategy to transform the Army to become lighter, more

---

[58] McGarry, John U.S. Army TACOM-ARDEC, "Systemic Analysis of Software Intensive System Acquisition Issues", Software Technology Conference – 2003, April 29, 2003

modular and more deployable. The vision was deployment of a brigade (3,000 to 4,000 soldiers and supporting equipment) in four days, a division (10,000 to 18,000 soldiers and supporting equipment) in five days and five divisions in 30 days.[59] The vision was called Objective Force and at the heart was FCS, a networked symphony of war fighting systems, including manned and unmanned air, space and ground platforms, as well as networked soldiers. In 2002 the Army awarded a contract to a Boeing and SAIC team to serve as lead system integrator of a proposed family of 18 FCS systems. In 2003 the FCS program entered the system design and development phase against the recommendation of the General Accountability Office (GAO) which warned the program was proceeding "with more risk than recommended by best practices or DoD guidance."[60] GAO concerns focused on systems engineering of a project of the magnitude of FCS. The Army restructured the program in 2004 in response to these concerns. After restructuring program cost rose from $99 million to $160 million but schedule was reduced in order to bring some systems online earlier.

Indications are the FCS team is adhering to robust systems engineering as defined by the DoD, employing systems engineering metrics such as MOEs, MOPs and TPMs. The program employs EVM as defined by the DoD to track schedule and cost as well as a set of Program Performance Measures (PPM) such as tracking the number of technologies at TRL 6 or greater versus plan on a yearly basis.[61]

In early 2006 both the GAO and congress again raised concerns over the team's ability to deliver the complex system on time and within budget. Again these concerns centered on the technical management aspects of a complex system of systems. In written testimony submitted to the House Armed Services Tactical Air and Land Forces Subcommittee the director of acquisition and sourcing management stated "FCS has all the markers for risk that would be difficult to accept for any single system, much less a complex, multi-program

---

[59] Tiboni, Frank, "Army's Future Combat Systems at the Heart of Transformation", Federal Computer Week, Feb 9, 2004

[60] GAO-03-1010R, "Issues Facing the Army's Future Combat System", Apr. 2003, p. 39.

[61] ExportMore.gov, Detailed Information on the Future Combat Systems/Modularity Land Warfare Assessment, http://www.whitehouse.gov/OMB/expectmore/detail.10003202.2005.html

effort."[62] Boeing and the Army responded placing even greater emphasis on program risk management in these early phases of design and development. In August the FCS team reported the program to be on schedule and within budget as well as successfully completing the system of systems functional review, a major development milestone.[63]

2.10 *Chapter Summary*

This chapter provided an overview of the systems engineering practice and the value proper application of systems engineering delivers to the stakeholders. Lean systems engineering and lean product development were reviewed and these concepts represent the future state of the role of systems engineering in PD. The goals of a performance measurement system as a means to lend to control to the stochastic project development system were introduced. Several measurement techniques in practice were reviewed. Finally, information on the issues systems engineering and product development face which call for improved control mechanisms were provided.

The addition of a set of systems engineering prognostics at the project level, with a clear connection to the organization level would permit the systems engineer to proactively tune the systems engineering activities and effort to seek an optimal output of information during the design phase. These prognostics also support an enterprise goal of balancing systems engineering effort across multiple projects. These coupled with the addition of a set of systems engineering organizational measurements have the potential to provide data to support systems engineering continuous improvement activities as part of an enterprise level goal of instituting lean throughout all business processes.

---

[62] Singer, Jeremy, "Boeing Says FCS Risk Management Plan Will Quell Concerns", Space News, Page 18, June 12, 2006

[63] FCS program web site, http://www.army.mil/fcs/, as available 28 Dec 2006

3 Prognostics for Systems Engineering

This chapter introduces the system of systems engineering prognostics which will be applied to historical data in a case study. The foundation is the LAI Leading Indicators as proposed in the Leading Indicators Guide[64]. The goal of systems engineering prognostics or leading indicators is to provide foresight to the effectiveness with which the systems engineering activities are applied within a project context. The data collected to support measurement across multiple projects also supports prediction of the systems engineering process effectiveness in an organizational context. This foresight lends to adjustment of the controllable inputs which influence systems engineering efficiency before the prognostication of poor efficiency is realized.

Prognostic is taken from the Greek prognostikos (of knowledge beforehand). It combines pro (before) and gnosis (a knowing). Webster's defines prognostic as "something that foretells". Prognostic is another term for "leading indicator" as used by the U.S. Air Force, LAI and INCOSE.

In economics leading indicators provide information on the predicted state of aspects of an economic system. Economic leading indicators have some understood and accepted correlation to the parameter of interest at some future point in time. For example, the number of housing starts in the U.S. is used to foretell the U.S. economic outlook. It is generally accepted that housing starts and the economic outlook have a positive correlation. Economist don't rely on this indicator alone to predict, as the indicator is not 100% reliable; a strong increase in housing starts could lead to a hike in inflation and ultimately a down turn in the economy. A more complete prediction of the future state of the economy can be gleaned from viewing a collection of leading indicators with some understanding of the interrelationships between the indicators. Such leading indicators with well understood behavior are invaluable in providing a means for adjustment prior to a process going off course.

---

[64] Roedler, G and Rhodes, D. *Systems Engineering Leading Indicators Guide*, LAI & INCOSE, Beta Release Dec 12, 2005

3.1 *Proposed Prognostics*

An approach to guide the development of a viable set of metrics is the Goal Question Measurement (GQM) methodology[65]. Basili originated GQM as a method for developing software metrics and it has since been applied successfully to software and enterprise level performance measurement systems. GQM is a top down method of developing process measurements which begins by evaluating the goals a group seeks to achieve by instituting measurement. The process is as follows;

- Define the overarching goals of the organization. What is the objective of the business, organization and/or process of concern?
- Generate questions which interrogate how well those goals are being met. There are often well known obstacles or issues related to an organization reaching each goal. In the PSM plan measurement activity process this aligns with the *Identify* in *Identify and Prioritize Information Needs* step.
- Analyze each question to isolate appropriate measurements to address each question.

A short coming of GQM is the framework neglects to address bounding the quantity of measurements. PSM extends GQM in the Plan Measurement Activity and addresses some of the methods shortcomings. The following are a suggested set of constraints for selection of systems engineering prognostics.

First, if the measurement system is to be effectively used the critical information content must be provided in a transportable way. Transportable here means the information provided is of a quantity which the target consumer can cope with.

---

[65] Victor R. Basili, et. al., "The Goal Question Metric Approach", Encyclopedia of Software Engineering, Wiley 1994

Second, implementation of the measurement project, continuous tracking and reporting must be cost effective, the benefit-cost ratio (BCR) must exceed 1 and the payback period must be accepted by management. In the context of this work this constraint drove avoidance of measurement requiring unique tools or new methods requiring months of training per employee.

Finally, in order to deliver a tractable measurement system the selected prognostics should be minimized to a degree permitting a systems engineer or program manager of average skill to retain the top level measurement information chunks on a daily basis. Miller's work on the limits of human short term memory[66] suggests seven plus or minus two (7 ± 2) information chunks is the limit of the average persons capacity.

3.1.1 Systems Engineering Leading Indicators Guide

The December 2005 Beta release of the LAI / INCOSE Systems Engineering Leading Indicators Guide[67] is the foundation of this research and the source for thirteen of the leading indicators described within.

LAI initiated the leading indicators project following the 2004 request from Dr. Marvin Sambur, Assistant Secretary of Air Force for Acquisition, to establish leading indicators for measuring the "goodness" of systems engineering. The original objectives of the ensuing LAI Leading Indicators Project were as follows;

- Gain common understanding of DoD needs and be in tune to industry needs
- Identify a set of leading indicators for systems engineering effectiveness
- Define and document measurable constructs for highest priority indicators
- Identify challenges for implementation of each indicator and recommendations for managing implementation

---

[66] Miller, G. A. "*The Magical Number Seven, Plus or Minus Two: Some Limits on our Capacity for Processing Information*". Psychological Review, 63, 81-97, 1956

[67] Roedler, G and Rhodes, D. *Systems Engineering Leading Indicators Guide*, LAI & INCOSE, Beta Release Dec 12, 2005,

- Establish recommendations for piloting and validation the new indicators before broad use.

The team formed through the LAI consortium is comprised of engineering measurement experts from industry, government and academia. Team leadership is balanced between industry and academia with co-leads Garry Roedler of Lockheed Martin and Donna Rhodes of the Massachusetts Institute of Technology. The project is fostered through collaborative partnerships with INCOSE, SSCI and PSM. This effort builds off the measurement foundations developed through PSM and employs the PSM template for defining each measurement construct.

3.1.2 Linking Goals, Questions and Measurements

The high level goal is to provide foresight which enables improvement in efficiency of the system development process as a means to increase delivered value to the stakeholders. In a defense industry value to the customer includes high quality and reliability at reasonable cost, meeting stated and tacit needs, and a minimal cost means for future system growth. The value delivered through the application of systems engineering on the system development process was established by the study by INCOSE as discussed in Chapter 2, therefore improving systems engineering efficiency is a path to improving system development efficiency.

The goal of developing and managing a system of systems engineering prognostics is to support preemptive control of systems engineering efficiency on a project.

Table 3-1 provides a mapping from the information question to the measurements for the thirteen initial leading indicators as proposed by LAI plus one additional suggested systems engineering "goodness" prognostic. There are 72 base measures and 60 derived measures listed within the table.

Table 3-1 Question to Measurement Mapping

Questions	Measurable Concepts	Indicator Name	Measurement
Is the SE effort driving towards stability in the system definition (and size)? Is the system definition maturity aligned with the scheduled/planned maturity?	System requirements and definition stability	Requirement Trend	# Requirements # Requirement Incomplete # Requirement Defects # Requirement Changes by Type # Requirement Changes by Cause Impact of Requirement Changes Requirement Change Request Initiated Requirement Change Request Approved % Requirements Approved % Requirements Growth Known unknown closure actual vs. plan % Requirements Modified Estimated Impact of Requirement Changes for time interval Defect Profile Defect Density Defect Leakage Cycle time for requirement changes Requirement stability / volatility

Questions	Measurement Concept	Indicator Name	Measurement
Are changes to the baseline or current system definition being processed in a systematic and timely manner?	System definition correctness	System Definition Change Backlog Trends	# Request for Change RFC Impact RFC Type RFC Cause # changes per approval disposition Start/Approval/Incorporated Time Approval Rates Cycle Time (Approval - Initiation)
Is the SE effort driving towards correct and complete definition and design of the interfaces? Is the interface definition maturity aligned with the scheduled/planned maturity?	System definition stability System Definition correctness	Interface Trends	# Interfaces # Interface Known Unknowns # Interface defects # Interface changes by type # Interface changes by cause Impact of interface changes Start/Approval/Incorporated Time % Interfaces approved % Interfaces growth Interface known unknown closure actual vs. plan % Interfaces modified Estimated impact of interface changes for time interval Interface defect profile Interface defect density Interface defect leakage Cycle time for interface changes Interface convergence of interfaces

Questions	Measurement Concept	Indicator Name	Measurement
Is the SE effort assuring in a timely fashion the right system has been specified to meet the needs of the key stakeholders?	System requirements correctness Process efficiency Process effectiveness	Requirements Validation Trends	# Requirements validated actual # Requirements validated planned Requirement validation effort with external stakeholders % Requirements validated Requirements validation rate
Is verification of the system requirements proceeding as planned? Will the product meet the customer needs?	Solution suitability Process efficiency Process effectiveness	Requirements Verification Trends	# of Requirements verified actual # of Requirements verified planned % Requirements validated Requirement verification rate
Is the quality of the system definition work products sufficient? Is the error rate and rework associated with work products above the expected/planned?	Process efficiency Process effectiveness Schedule	Work Product Approval Trends	# Work Products In Review # Work Products per approval disposition Work Product Approval Rates Distribution of dispositions Work product first pass approval rates Work product approval actual over plan
Are action items being addressed and impact to the baseline being incorporated in a timely fashion? Is the presented system definition meeting the needs of the customer? Is the system definition stability as expected for the current phase of development?	System definition correctness Solution suitability Customer satisfaction Process efficiency Process effectiveness	Review Action Closure Trends	# Action Items # Action Items by status (Open, Closed, Overdue, etc) # Action items by priority (Critical, Major, Minor) Impact of Action Item Closure rate

Questions	Measurement Concept	Indicator Name	Measurement
	Schedule		Action items closed actual over plan Variance from threshold
Can the technology meet the requirements or will additional/new technology be needed? What is the risk impact of technology insertion? What risks exist of technology obsolesce?	Technology suitability Technology stability	Technology Maturity Trends	# Technology obsolescence candidates identified # Critical/beneficial technology opportunities identified TRL for each opportunity # Technology obsolescence candidates realized # Technology opportunities realized Technology expected realization time Technology actual realization time Technology expected cost for realization Technology actual cost for realization Probability of technology insertion/phase out Expected impact of technology insertion/phase out Actual impact of technology insertion/phase out Technology opportunity exposure Technology obsolescence exposure Technology mean time to impact Technology mean error of impact estimate
Are the project risks being managed by the SE team effectively? Does the risk exposure indicate likely issues with the current plan (cost, schedule, performance)?	Risk Recognition Effectiveness	Risk Exposure Trends	# Risks Identified Risk Probability Risk Impact Criticality Planned Mitigations

Questions	Measurement Concept	Indicator Name	Measurement
			Executed Mitigations Risk Status (Open, closed, etc.) Risk Type (Cost, Schedule, Performance) Risk Exposure
Are the projects risks being managed by the SE team effectively? Are risk mitigations executed as planned? Are risk mitigations effective?	Risk management effectiveness	Risk Handling Trends	Risk Funding Planned Risk Funding Actual Risk Mitigations Risk mitigation status (open, closed, etc) Risk status (red, yellow, green) % Risk mitigations on time % Risk mitigations overdue % Risks mitigated per plan
Is the SEMP effective? Is the SE effort being staffed effectively and per plan? Is the SE effort being staffed with the appropriate skills?	Personnel Effort	SE Staffing and Skills Trends	Project effort by task, activity, event planned Project effort by task, activity, event actual SE effort by task, activity, event planned SE effort by task, activity, event actual SE effort by skill and experience planned SE effort by skill and experience actual # of equivalent SE staff by task, activity, or event planned # of equivalent SE staff by task, activity, or event actual % SE Effort planned % SE effort actual % SE staffing per plan - planned % SE staffing per plan - actual Variance of SE Effort Variance of SE Staffing

Questions	Measurement Concept	Indicator Name	Measurement
			Variance of quantity of SE skills
How consistently are the defined SE processes implemented?	Process Compliance Process Efficiency Process Effectiveness	Process Compliance Trends	SE processes satisfied SE processes with discrepancies # of SE process discrepancies by severity # of SE process discrepancies by category % Processes with discrepancies Profile of discrepancies High risk processes
Are the critical technical aspects of the project feasible? Is the technical solution progressing toward the required capabilities as planned? Is there additional risk due to issues with meeting the technical goals?	Technology Effectiveness	Technical Measurement Trends	TPM planned values TPM actual/estimated values TPM Priority TPM Variance from plan TPM Variance from thresholds TPM Variance from objective
Is the interdisciplinary development team effectively working together, sharing information and communicating project issues? Does the development team believe product quality is high (i.e. customer will be satisfied)? Is the development team committed to the schedule per plan? What is the team moral? Does the development team have faith in the SE process?	Team effectiveness	Development Team Cohesion Trends	Team Communications Team Moral Schedule Commitment SE Process Adherence Perceived Customer satisfaction SE Effectiveness

3.1.3 Roles of Systems Engineering within the Project Context

To support discussion the prognostics are classified by the systems engineering responsibly which they seek to optimize. These classifications are provided in Table 3-2.

Table 3-2 Indicator to SE Role Mapping

Indicator Name	Associated SE Processes (LAI)	Associated SE Role
Requirement Trends	Stakeholder Requirements, Requirements Analysis, Architectural Design	Requirements Management
System Definition Change Backlog Trends	Stakeholder Requirements, Requirements Analysis, Architectural Design	Requirements Management
Interface Trends	Stakeholder Requirements, Requirements Analysis, Architectural Design	Interface Management
Requirements Validation Trends	Stakeholder Requirements, Requirements Analysis, Architectural Design	Requirements Management
Requirements Verification Trends	Stakeholder Requirements, Requirements Analysis, Architectural Design	Requirements Verification
Work Product Approval Trends	Review Process	Configuration Management
Review Action Closure Trends	Review Process	Customer Interface
Technology Maturity Trends	Planning, Decision Making, Architectural Design and Production	Risk Management
Risk Exposure Trends	Risk Management, Program Management	Risk Management
Risk Handling Trends	Risk Management, Program Management	Risk Management
SE Staffing and Skills Trends	Planning, Control	Technical Planning and Management
Process Compliance Trends	All	SE Process Adherence
Technical Measurement Trends	Risk Management, Requirements Analysis, Modeling, Design and Integration	Design Synthesis and Evaluation
Development Team Cohesion Trends	NA	Team Cohesion Management

3.1.3.1 Requirement Management

One of the fundamental roles of systems engineering is requirement management. Jim Hill, director of systems engineering for MITRE Corporation in 2004 stated "Systems engineering ... focuses on defining customer needs and required functionality early in the development cycle..."[68]

The objective of the systems engineering effort applied for requirement management is to assure a complete and accurate set of requirements is elicited from the customer, defined in the systems requirements specification(s) and validated as early as possible in the development cycle.

The primary question the systems engineer seeks to answer as the owner of requirements management is "Is requirements stability as expected for the current phase of development?"

Requirements stability is defined by the quantity of changes and the cumulative effect of the changes. One change which has far reaching impact on the established design often outweighs twenty changes which have little to no impact on existing architecture or design decisions. In order to effectively track stability of the requirements both number and change impact must be accounted for. One means suggested is through projection of a weighted average computation of a derived requirement stability or volatility measure. To compute the Requirement Stability Index (RSI) the following is proposed;

$$RSI = 100 * (W_1*(R_{delta}/R_{total}) + W_2*(C_{delta}/C_{max}) + W_3*V_{Variance}$$

Where W_1, W_2 and W_3 are weights which the team or organization will need to determine at the start of the project. R_{delta} is the count of changes in requirements in a reporting period. R_{total} is the total number of accepted system requirements. $C_{delta}/$ is the impact cost for the changes in the reporting period. C_{max} is an upper threshold on cost associated to changes which the team will need to determine at the start of the project. This should be

---

[68] "MITRE Views on Systems Engineering", Collaborations, Vol 2, Num 3, MITRE Corporation 2004

based on the cost the project can absorb without needing to renegotiate the contract or request additional funding from the enterprises executive management. $V_{Variance}$ is the variance from plan for requirements validation. The equation is a weighted average of the aspects of requirement change key to stability.

3.1.3.2 Risk Management

Risk management is the continuous commitment to identifying, assessing, handling and monitoring project risks. Project risk management, in many organization, is a shared responsibility of the systems engineer and project manager. These process owners must assure risks are identified in a timely manner throughout the product lifecycle. This involves fostering project team and external stakeholder communications to assure all issues impacting the projects likelihood of success are raised to the watchlist.

Each risk identified is added to a watchlist and assigned a probability and impact rating. Each risk is linked to the risk category; cost, schedule, performance. Risks are assessed and assigned a planned risk handling action. For some risks this action might be acceptance without further effort, but for many the handling action will be mitigation. Other risk handling actions are avoidance, i.e. change the course of the project to attempt to avoid the risk, or transference, shifting the responsibility to handle the risk to another entity. As the project proceeds risks are monitored to ensure each has been sufficiently quantified and risk handling actions are assuaging the impacts as expected.

The risk exposure measure should quantify the open risks at each reporting period of interest taking into account the effectiveness of active actions and confidence in planned handling actions. Accounting for the effectiveness of risk handling actions in the risk exposure time history increases the dependence between the perceived risk and risk handling actions. To support creation of a historical database of risk exposure to support trending and threshold setting a consistent method must be employed across the organization.

3.1.3.3 Interface Management

Interface management is a critical role of systems engineering as a leverage point to evaluate and manage the evolution of the system or product complexity. Complexity is measured by the information required at the interfaces, e.g. the number of interconnections (information in count and type), the sophistication of the interconnections (how much information to specify), the sensitivity and or robustness of the interconnections (how much information to describe).[69] The systems engineer manages both the externally perceived complexity based on the products external interfaces and the overall complexity which takes into account all of the interface within the product boundaries as well as external interface.

Traditionally systems engineering manages key interface requirements as part of formal and informal TPMs as well as tracking of the interface documentation process (approval cycles and peer reviews).[70] TPMs are an effective means to manage key aspects of the interface design, but systems engineering is better served by additional measurements which take a holistic view of interface management in the system of interest context.

3.1.3.4 Requirements Verification

Requirements verification assures the developed solution meets the stated requirements. Verification of each requirement is allocated to a verification method, e.g. test, analysis, inspection. Systems engineering is responsible for planning and leading the verification process.

Verification occurs at the end of each development iteration and measures of the process provide predominantly lagging indicators of the design iterations success. However, there is leading information in measuring this process related to the downstream processes.

---

[69] Crawley, Ed, Lecture notes from Systems Architecture course, MIT, Fall 2004.

[70] Blyler, John. "Interface Management, IEEE Instrumentation & Measurement Magazine", March 2004, Pg 32-37

3.1.3.5 Technical Planning and Management

Systems engineering develops the technical schedule within the Systems Engineering Master Plan (SEMP) for the non recurring effort of the development phase. Systems engineering manages the development effort by controlling costs and scheduling resources per the SEMP, adjusting the plan as required and agreed to by the stakeholders. Monitoring the planned staffing levels and skills levels versus planned is critical to predicting the path of the project, both from a cost and schedule perspective.

3.1.3.6 Design Synthesis and Evaluation

Systems engineering is responsible for leading the development of the systems architecture. This includes conducting trade studies to seek the optimal solution relative to the system requirements and ensures technology is appropriately selected to meet the needs of the project.

Systems engineering continuously evaluates design to assure its alignment with requirements and flexibility for growth. Modeling, simulation, set based design practices, prototyping and other techniques are employed by systems engineering to increase early confidence the product will function and interface with the external environment as expected.

3.1.3.7 Configuration Management

Systems engineering is the owner of managing the flow of technical information on the project and assuring stakeholders have the information required to make informed technical decisions. The quality of the work products released into the configuration management system is one reflection of the efficiency with which this task is addressed.

3.1.3.8 Customer Interface

The systems engineer is the lead point of contact for the technical aspects of the project. The views and needs of the customer must be understood to ensure they are respected throughout the development phase. The systems engineer ensures the customer - contractor relationship is executed in a profession and customer-friendly manner. The measurement of creation and processing of action items from customer design reviews is an indicator of efficiency in this role.

3.1.3.9 Team Cohesion Management

Sheard[71] enumerates twelve roles of systems engineering, in which the notion of team cohesion management aligns with the "Coordinator" role.

Measurement through team surveys is intended to provide information on the trends related to how well the team is communicating, sharing information, showing up for meetings, working together, its belief in SE practice/process, and working with the customer. As systems engineering adjusts to influence team dynamics this prognostic provides feedback into the effectiveness of the changes.

3.1.4 Leading and Lagging Content within the Indicators

Each indicator has value both as a leading and lagging indicator. Table 3-3 provides some insight into these two forms of information content available in each indicator.

---

[71] Sheard, Sarah A. "Twelve Systems Engineering Roles," Proceedings of INCOSE, 1996.

Table 3-3 Leading and Lagging Information in Indicators

Indicator Name	Leading	Lagging
Requirements Trends	Issues with maturity of system definition versus the planned maturity Impact on architecture and design forthcoming, i.e. rework Impact on V&V, i.e. need for new test equipment or facilities with long lead time or high costs Impact on production, rework and other late process issues (returns, repeating qual tests, etc) Impact on cost and schedule Increasing complexity of solution	SOO was not as mature as required to support requirements phase VOC did not elicit the right voices or experts Customer / end user uncertain what problem space is Customer / end user focused on added features post initial SOO Not providing adequate prototypes early in concept selection
System Definition Change Backlog Trends	System definition maturity Project maturity Development quality Development schedule Staffing adequacy Impact on architecture and design, i.e. rework, due to design request not processed in a timely manner Increasing cost and schedule risk exposure as backlog of changes increases	Significant backlog indicator that process not implemented effectively or is defective Staffing inadequacy Bottleneck in review/approval process
Interface Trends	System definition maturity and volatility Risk level in design and integration/implementation Development quality Development schedule Staffing and skills adequacy Impact on architecture and design, i.e. rework Effectiveness of collaboration between interface owner and consumer (customer, internal groups or third party) Cost and schedule risk exposure	Adequacy of effort on interface management in previous development phase Effectiveness of collaboration between interface owner and consumer (customer, internal groups or third party) Staffing and skill adequacy (i.e. contractors staff understanding of interface standards common to industry)
Requirements Validation Rate Trends	System definition maturity development quality development schedule staffing adequacy stakeholder availability or sense of urgency for solution stakeholder (customer/end user) satisfaction impact on architecture and design, i.e. rework risk of issues during system validation, i.e. solution not meeting needs increasing cost and schedule risk	Planning inadequacy: Rate significantly exceeding plan might indicate planning inadequacy, introduce risk of inefficiency, Parkinson's law rate significantly lower than plan also efficiency issue. Acquirer provided operation needs statement or system performance specification correctness

Indicator Name	Leading	Lagging
	exposure	SE requirements analysis process effectiveness
Requirements Verification Trends	Product quality development schedule staffing adequacy stakeholder (customer/end user) satisfaction impact on architecture and design, i.e. rework risk of issues during system validation, i.e. solution not meeting needs cost and schedule risk exposure additional iterations in development for follow on deliveries to fully meet needs Overproducing, perhaps too much effort is spent on creating work products if always approved in 1 cycle	System definition maturity, verification is generally late in development phase, issues drive rework or contract re-negotiation Planning inadequacy: Rate significantly exceeding plan might indicate planning inadequacy, introduce risk of inefficiency, Parkinson's law rate significantly lower than plan also efficiency issue. SE requirements analysis process effectiveness, verification allocated correctly? Facility requirements correctness, i.e. needed customer lab for demonstration or test
Work Product Approval Trends	Level of understanding of customer or other key stakeholder expectations Product quality Stakeholder (customer/end user) satisfaction Development quality, possible rework to address Planning adequacy, possible replanting to address Process efficiency, review process might be flawed Reviewer quality, internal approval trends might indicate staffing adequacy Schedule correctness, i.e. readiness for design reviews (PDR, CDR, etc)	Planning inadequacy: plan over allocated or under allocation to creating work products and addressing time spent in the review cycle Work Product Quality: Reflection of quality of each work product Work product or process complexity: reflection of the complexity of the work product or what the work product addresses
Review Action Closure Trends	System definition maturity Schedule viability and project maturity Process effectiveness Product quality Stakeholder (customer/end user) satisfaction Risk management accuracy Staffing and skills adequacy Project support infrastructure adequacy Funding adequacy (from customer) or cost risk (of contractor) Technical suitability or effectiveness	Planning inadequacy: greater then expected actions might indicate planning error, i.e. more iterations of design and/or production Project Maturity; review action might indicate a need to return to earlier design phase Process effectiveness; personnel changes introducing new people with old actions, wrong people at review resulting in less action items then expected, lack of preparation for design reviews by SE

Indicator Name	Leading	Lagging
Technology Maturity Trends	Technical suitability or effectiveness Product quality Architecture or design viability, i.e. designed for easy future incorporation of technology (appropriate incorporation of scalability and flexibility) Risk management accuracy Funding adequacy (from customer) or cost risk (of contractor) Aftermarket, recurring and warrantee costs	Planning inadequacy: increased obsolescence unplanned will drive rework Technology readiness: too many technologies required for integration into the product at low TRL levels could indicate project is more research then product development Architecture inadequacy: if architecture or design does not incorporate growth or scalability to incorporate new technology it might require rework.
Risk Exposure Trends	Schedule, cost and performance uncertainty System definition maturity Product quality Stakeholder (customer, end user) satisfaction Funding adequacy (from customer) or Cost risk (of contractor) Technical suitability or effectiveness Process efficiency	Process effectiveness contractors risk acceptance criteria lessons learned, i.e. what mitigation worked/did not work
Risk Handling Trends	Likelihood risks will be realized Process efficiency Management commitment to risk plan	Process effectiveness Cost, schedule and performance impacts from realized risks lessons learned, i.e. what mitigation worked/did not work
Systems Engineering Staffing Trends	Staffing levels on track to meet schedule Skill levels on track to meet schedule Risk level for staffing Potential cost overruns (overstaffing) Staff burnout risk (too much overtime)	Schedule slip and cost overrun Baselining the proper SE level of effort for a project Staff inadequacies
Process Compliance Trends	Process efficiency Production issues post "handoff" Product quality Stakeholder (customer/end user) Satisfaction Cost and schedule risk exposure Organization commitment to existing processes Process improvement opportunities	Process effectiveness Training gaps Areas requiring additional quality focus or more frequent audits Process shortcomings or errors
Technical Measurement Trends	System definition maturity requirements feasibility technical suitability or effectiveness stakeholder (customer, end user) satisfaction architecture or design viability risk management gaps possible trade offs in performance	Architecture and design issues, leading to possible rework Technology suitability

Indicator Name	Leading	Lagging
Project Team Cohesion Trends	Process efficiency Staffing or skills adequacy Product quality Stakeholder (customer/end user) satisfaction Cost and schedule risk exposure Interface issues due to miscommunications or no communication Architecture or design adequacy	Team morale Team communications Schedule commitment

3.1.5 Possible Pitfalls

In employing each metric there are pitfalls to avoid and hurdles to overcome in order to gain the full value of the measurement program. Table 3-4 lists a few possible issues with each indicator. The information in this table is based on notes taken at a LAI Knowledge Exchange Event on the leading indicators in January of 2007.

Table 3-4 Possible Pitfalls in Implementation, Interpreting and Misuse

Indicator Name	Implementation Issues	Pitfalls in Interpreting	Possible Misuses
Requirements Trends	Sampling: How often are requirements changes communicated? Relevance: Is this an adequate metric in early phases of the life cycle? Stability: Does the requirements baseline change over time?	Granularity: different levels of detail throughout systems hierarchy Quality of Work: no indication of workmanship	Progress Gauge: Early stability not necessarily indicator of maturity Placing Blame: Open issues in requirements might be out of the control of the organization developing the solution
System Definition Change Backlog Trends	History: Requires historical data to set thresholds, where historical data is unlikely to exist Design space: not all changes are created equal Investment: Could be significant investment in tools and training to collect data	False Hope: Depending on where in the development phase, a small backlog might provide a false sense that the SE process is efficient, when in fact the issue might be a lack of expertise or focus on the system definition	Faith in One: Could give false sense of reality if used alone, not in combination with downstream/component deltas Stale Indicator: Not reviewed on a timely basis/continuous update

Indicator Name	Implementation Issues	Pitfalls in Interpreting	Possible Misuses
Interface Trends	Sampling: How often are interface changes communicated, when are changes relevant ? Skewed view of reality: It is the last 10% of issues/errors which take 90% of time to identify, need to assure the thresholds are set to account for this Investment: Could be significant investment in tools and training to collect data	False Hope: If thresholds are not set correctly might indicate SE process is on track when few issues are left open, but these represent significant outstanding effort Sequencing: Project might have reason for delay of interface definitions or sequencing not visible in indicator	Progress Gauge: Early stability not necessarily indicator of maturity Placing Blame: Open issues might be out of the control of the organization consuming the interface definitions Management of Indicator: Could lead to focus on getting metric to target instead of understanding and solving the root cause the metric was intended to indicate
Requirements Validation Rate Trends	History: Requires historical data to set thresholds, where historical data is unlikely to exist Investment: Could be significant investment in tools and training to collect data Sampling: How does requirements validation proceed? may occur all at once at SSR. Acceptance: Often metrics focus on failure events, some belief measurement of success events not valuable	Lack of Information: Difficult to extract value without cause and effect understanding for out of bounds conditions	Progress Gauge: Meeting plan not necessarily indicator of progressManagement of Indicator: Could lead to focus on getting metric to target instead of understanding and solving the root cause the metric was intended to indicate
Requirements Verification Trends	History: Requires historical data to set thresholds, where historical data is unlikely to exist Investment: Could be significant investment in tools and training to collect data Sampling: How does requirements verification proceed? may occur all in short period (TRR to test report).	Lack of Information: Difficult to extract value without cause and effect understanding for out of bounds conditions	Progress Gauge: Meeting plan not necessarily indicator of progress Management of Indicator: Could lead to focus on getting metric to target instead of understanding and solving the root cause the metric was intended to indicate

Indicator Name	**Implementation Issues**	**Pitfalls in Interpreting**	**Possible Misuses**
Work Product Approval Trends	History: Requires historical data to set thresholds, both on the internal review process and each customers review process. Each customer/external reviewer has different behavior. Sampling: When are most work products submitted and what is the expected turn time, if submitted in chunks around milestone might be more difficult to interpret. Information Content: Logging rejections might be common but here need list additional detail to drive to cause of rejections (major, minor, reviewer bias, reviewer not included, etc). Collecting the additional data might cost more then the value added by the indicator	Lack of Information: Difficult to extract value without cause and effect understanding for out of bounds conditions False Hope: If thresholds are not set correctly might indicate SE process is on track when few issues are left open, but these represent significant outstanding effort. Sequencing: If easy/simple work products are up front (i.e. generic plans) and more difficult work products toward the end of the development cycle the higher reject rate later in development might trigger undo concern or assignment of blame.	Management of Indicator: Could lead to focus on getting metric to target instead of understanding and solving the root cause the metric was intended to indicate. Reviews can go quickly if the goal is to approve and not to review (easy to game the system). Also, groups might choose to approve providing comments to be incorporated at later revision just to avoid rejection.
Review Action Closure Trends	Standard Definitions: Often there is not a shared definition of action item criticality or when action items are closed across an organization. Sampling: Often many closed at design reviews, as closure is an entrance criteria.	False Hope: closure as planned is not alone a indicator of "goodness". Sequencing: Not equal across the SE lifecycle	Management of Indicator: Could lead to focus on getting metric to target instead of understanding and solving the root cause the metric was intended to indicate. Avoidance of opening action items until design reviews. Action items closed for measurement only, but issue still at large or reopened under fresh action item.

Indicator Name	Implementation Issues	Pitfalls in Interpreting	Possible Misuses
Technology Maturity Trends	Information Content: new technology not well understood and issue/opportunities are over/under stated.	Granularity: different levels of detail throughout systems hierarchy False Hope: technology will evolve and some will be unaccounted for by the knowledge pool in the project Stability: one major technology could have serious impact where many small ones might have none	"better is the enemy of good enough" - too often engineering is drawn toward new technology without considering all downstream impacts Too narrowly focused too soon - Might be a technology looking for a problem to solve. technology and obsolesce management must follow processes similar to concept selection, first expand to consider many options to the problem then reduce to select. Overselling - Technology advocates will sell new technology as the solution to all problems
Risk Exposure Trends	Investment: Training and tools if an active risk management program not in place. Sampling: How often do risks get updated?	False Hope: Low risk exposure might indicate a poor risk identification process, not low risk. Scale: Need to indicate impact and probability, raw counts not of value. Incomplete: Risk might only account for impact as seen by risk owner, often there are interdisciplinary impacts not accounted for in risk or mitigation actions.	Lack of Follow-up: Identification without mitigations, funding and tracking escapes is non value added post biding. SEs task is to identify, decide how to handle, acquire funding and manage mitigation. Management of Indicator: Could lead to focus on getting metric to target instead of understanding and solving the root cause the metric was intended to indicate. Closing risks before risk horizon ends. Not adding risks except of high impact and probability.

Indicator Name	Implementation Issues	Pitfalls in Interpreting	Possible Misuses
Risk Handling Trends	Investment: Training and tools if an active risk management program not in place. Sampling: How often do risks get updated?	False Hope: Only risks identified can be handled, so strong tie to Risk Exposure Trends. Granularity: Need to understand count of successful/unsuccessful mitigation and the associated risk exposure Unknown risks: Some fraction of risks will "slip through the cracks", so can't take this as complete statement of goodness	Management of Indicator: Could lead to focus on getting metric to target instead of understanding and solving the root cause the metric was intended to indicate. Closing risks before risk horizon ends. Not adding risks except of high impact and probability.
Systems Engineering Staffing Trends	Cross Organizational Boundaries: Measuring across companies, teaming arrangements, would be difficult due to different time reporting systems Standards: Lack/absence of standard skill definition, Lack/absence of standard estimating Historical data: forecast work, schedule	False Hope: The indicator relies on proper planning, if the plan is flawed it will look as if staffing trends are ok Information content: Might lend to seeking experienced personnel instead of training available staff	Management of Indicator: Could lead to focus on getting metric to target instead of understanding and solving the root cause the metric was intended to indicate. Use of metric to retain staff just incase. Micro management of indicator could lead to high project turn over as resource management react to short term needs with short term assignments. Turnover of key personnel is detrimental to a project, indicator should provide visibility into changes in personnel. Not Integrated with Resource Planning: No accounting for personnel growth.

Indicator Name	Implementation Issues	Pitfalls in Interpreting	Possible Misuses
Process Compliance Trends	Standards: Assumes processes are: standard, repeatable, effective Information Availability: Assumes reliable/accurate/timely data Information Accuracy: Assumes accurate interpretation of data Management Commitment: Personnel motivators for process compliance are a factor	Missing Link: Reporting indicators doesn't reflect process interdependencies Wrong behavior: May drive sub-optimization, force compliance with "wrong" process Information content: Measuring discrepancies only and not also compliances paints a different picture.	Placing blame: could lose sight and assume process is always right, which is not always true Management of Indicator: Could lead to focus on getting metric on target instead of understanding and solving root cause. Quality/auditor might start to take partial process adherence as acceptable to avoid the "wrath of management" which occurs due to process compliance out of tolerances.
Technical Measurement Trends	Standards: assuring that measurement is performed in a consistent manner Information Overload: assure tracking of only critical technical aspects, 4 to 6 per SPS	False Hope: stability of TPM is not enough, need subjective insight from experts for more reliable trend prediction. Historical data: thresholds set on historical data might not be applicable for new technology	Management of Indicator: Could lead to focus on getting metric on target instead of understanding and solving root cause. Focus on raw TPM value might miss bigger picture risks or opportunities.
Project Team Cohesion Trends	Investment: Requires surveys periodically, need tools to administer and report data as well as project time for each team member to complete (1/mo or as needed based on project duration) Historical data: unlikely any existing historical data	False Hope: Team "feeling good" might be more related to good motivational leader and not indicative of true project conditions Signal to Noise: Need to weed out the noise from effects of a few peoples bad day	Placing blame: management decides project has poor team dynamics and reorganizes the team when results are more reflection of one team member or leader.

3.1.6 Interrelationships between Indicators

As with leading indicators of the economy, an understanding of the relationships between the indicators is critical to the effectiveness of the decision process consuming this information. Table 3-5 suggests where interrelationships between indicators might exist.

The existence or strength of the relationships could differ in different organization or industries and should be evaluated in each of these contexts.

Table 3-5 Indicator Interrelationships

Indicator Name	**Has Relationship To**
Requirement Trend	Risk Exposure Trends, SE Staffing & Skills Trends Requirement Validation Trends Requirement Verification Trends
System Definition Change Backlog Trends	Risk Exposure Trends, Requirements Trends, SE Staffing & Skills Trends, Review Action Item Closure Trends, Technology Maturity Trends
Interface Trends	Risk Exposure Trends, Requirements Trends, SE Staffing & Skills Trends, Team Cohesion Trends
Requirements Validation Trends	Risk Exposure Trends, Requirements Trends, SE Staffing & Skills Trends
Requirements Verification Trends	Risk Exposure Trends, Requirements Trends, SE Staffing & Skills Trends, Interface Trends
Work Product Approval Trends	Risk Exposure Trends, Requirements Trends, SE Staffing & Skills Trends Team Cohesion Trends
Review Action Closure Trends	Risk Exposure Trends, Requirements Trends, SE Staffing & Skills Trends, System Definition Change Backlog Trends, Interface Trends Team Cohesion Trends
Technology Maturity Trends	Risk Exposure Trends, Risk Handling Trends, Requirements Trends, SE Staffing & Skills Trends, System Definition Change Backlog Trends, Interface Trends
Risk Exposure Trends	Risk Handling Trends, SE Staffing & Skills Trends,
Risk Handling Trends	Risk Exposure Trends, SE Staffing & Skills Trends, Technology Maturity

Indicator Name	Has Relationship To
SE Staffing and Skills Trends	Risk Exposure Trends, Risk Handling Trends, Requirements Trends, SE Staffing & Skills Trends, System Definition Change Backlog Trends, Interface Trends, Review Action Item Closure Trends Work Product Approval Trends, Process Compliance Trends, Technology Measurement Trends Requirement Validation Trends Requirement Verification Trends, Team Cohesion Trends
Process Compliance Trends	Risk Exposure Trends, SE Staffing & Skills Trends,
Technical Measurement Trends	Risk Exposure Trends, Requirements Trends, Interface Trends,
Development Team Cohesion Trends	Risk Exposure Trends, Requirements Trends, SE Staffing & Skills Trends, Interface Trends, Review Action Item Closure Trends Work Product Approval Trends,

3.2 *Measuring Efficiency of Systems Engineering within an Organization*

At an organization level, systems engineering enables the business to meet its technical goals. It ensures value is added though following a series of best practices and processes. It evolves through continuous improvement activities which leverage lessons learned from the enterprise.

To measure the efficiency at the organization level first requires analysis of the indicator data across the collection of projects which defines the organization. This will require significant historical data and likely follow the institutionalization of the project level prognostics by several years.

In concert with the indicator data from the collection of projects the organization should also measure the effectiveness of training, consistency of organizational systems engineering tools for all phases, commitment to process improvement and fostering of systems engineering knowledge management.

3.3 *Chapter Summary*

This chapter provides an overview of a proposed measurement system for systems engineering prognostics aimed at providing a means to forecast the efficiency of systems engineering within a project context. These set of prognostics were mapped to systems engineering tasks as a means to categorize the information content. Possible issues which could impede the implementation process or detract from the value provided were also listed. Addition information on the LAI developed thirteen initial leading indicators and specific measurement constructs are available in the LAI Leading Indicators Guide.

4 Case Study

This chapter presents a case study evaluated to investigate the applicability of a systems engineering prognostics measurement framework as presented in the previous section. A development organization was selected which followed a well defined system development process and maintained significant project related data. Historical data from the development phase was gathered from a program now entering the operational phase of its lifecycle. Where gaps existed in historical data interviews with the project team members were conducted. A few measurements from each systems engineering task are presented along with discussion. The names of the firms, project and people reported in this chapter do not reflect the actual identity of companies or people involved in the subject project.

4.1 *Hg3 Case*

The Alpha Systems is a division of Zed Company, an international supplier of aerospace and defense products with yearly sales exceeding 1 billion USD. Alpha Systems has been developing products in the industry for over half a century. Over that time the firm gained a favorable reputation for quality products and superior customer support.

Alpha Systems product development group was structured as a matrix organization, with strong reporting links to the functional managers and weak reporting links to program managers. Engineers were assigned to multiple projects, i.e. shared resources. The PDP documented in the company procedures is shown in Figure 4-1.

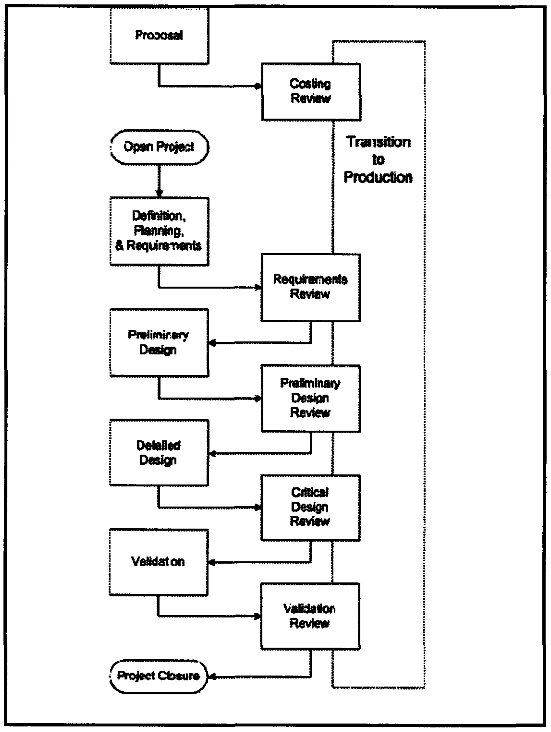


Figure 4-1 Alpha Systems PDP

In the mid 1990s the division expanded its product offerings, primarily consisting of special purpose electro-mechanical control systems, by acquiring a small company developing a new technology for aircraft flight and subsystem data acquisition and special purpose processing. First and second generation systems of the new product family which evolved from this acquisition were market successes. By the turn of the century new and existing customers were seeking capabilities beyond those of the current system with no indication of a willingness to accept an upward trend in pricing. In addition, the market niche the product filled for several years with minimal threat from other suppliers had experience significant demand increase, with strong customer pull. This pull enticed several other firms to seek entrance into the market. Alpha Systems decided to invest in a project to develop the 3rd generation of the system. This project will be referred to as the Hg3 Project.

Hg3 was initiated as an internally funded development project and preceded as such for several months without a specific, contracted customer. Although developing products for market without a contract or otherwise assigned customer is status quo in other industries, it was outside of the norm for the team at Alpha Systems. The firm's development process best served cases where a customer requested a solution to a well defined problem space. The concept development phase was lead by the resource manager of one functional group which included both hardware and embedded software engineers. This team focused on aspects of the system which were to be installed on the vehicle. The team was not allocated dedicated resources from the systems engineering group and designated many of the traditional system responsibilities to the functional groups manager. User needs were gathered by starting with the existing fielded systems requirements and augmenting these with business developments and the team leads perception of additional customer needs. A series of constraints were imposed on the concept generation process, typical of an nth generation system. There was an internal emphasis on reuse, both of hardware and software technologies from the previous generation systems and technologies in use in other areas of the division. There was also a push for the selected solution to permit retrofit of existing generation systems with minimal effort. Constraints were aimed at reducing cost and permitting a rapid time to market. Once a concept was selected the team developed a product interface control document (ICD) based on a set of product requirements for the on vehicle main line replaceable unit (LRU). This team brought the system into the initial product design phase. The ICD drove the early design, focused on initial hardware development.

Approximately nine months after the start of the internal project, Alpha Systems was awarded a contract to develop the system for a specific platform. The customer had the prior generation system installed on other platforms. The new generation system was to be installed as part of the customer's larger development program to provide a new airframe to the market. Key deliverables for the Hg3 schedule were aligned with overall milestone for the vehicle project.

The development team, with lead responsibility shifted to systems engineering, was increased to meet the needs outlined in the Systems Engineering Master Plan (SEMP). The primary objective stated in the project charter was to develop a solution which met or exceeded the needs of the customer as stated in the statement of work (SOW) while holding costs to within the allocated budget. A secondary goal was to develop a flexible system which supported the business goal of developing the beginnings of a product family. The project non recurring schedule outlined in the SEMP took into account the original teams estimates that hardware design was nearing the detailed design phase, approximately 80% of the embedded software was predicted to be reuse (design and source code) and ground content modification would be software only and held to less then 1% of the existing source code. The product baseline, signifying the end of the development phase, was scheduled to be ready in less than two years, with a Red Label unit, a pre production unit, due for flight test support in under a year. One of the primary risks in the program was the aggressive nature of the schedule. At a project management level, EVM was tracked and reported quarterly.

4.1.1 Requirements Management

The customer provided a system performance specification (SPS) at contract award based on systems level requirement specifications generated by Alpha Systems for the 2nd generation system with additions and modifications. A system specification was generated prior to SRR with approximately 800 requirements traced to the need statements from the initial revision of the SPS. Requirements grew by 400% from the initial requirements specified prior to contract award. Figure 4-2 through Figure 4-4 show requirements change through the development phase, up to the system test for the first deliverable software increment in which over 90% of functionality was provided. Initial product requirements developed during the concept generation phase were tracked in a less formal method and are not shown in the charts.

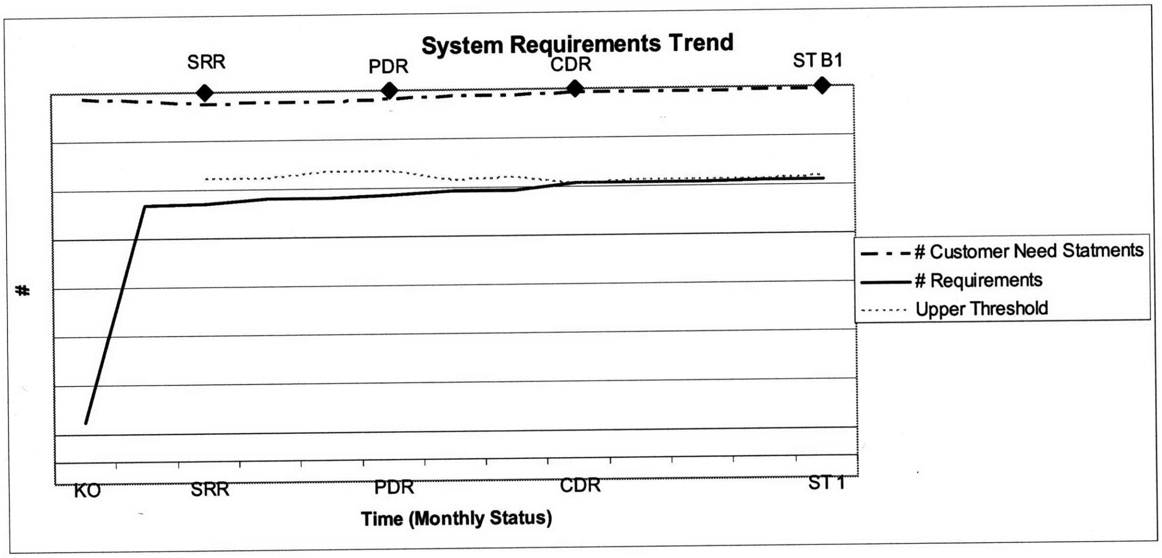


Figure 4-2 Hg3 Total System Level Requirements Time History

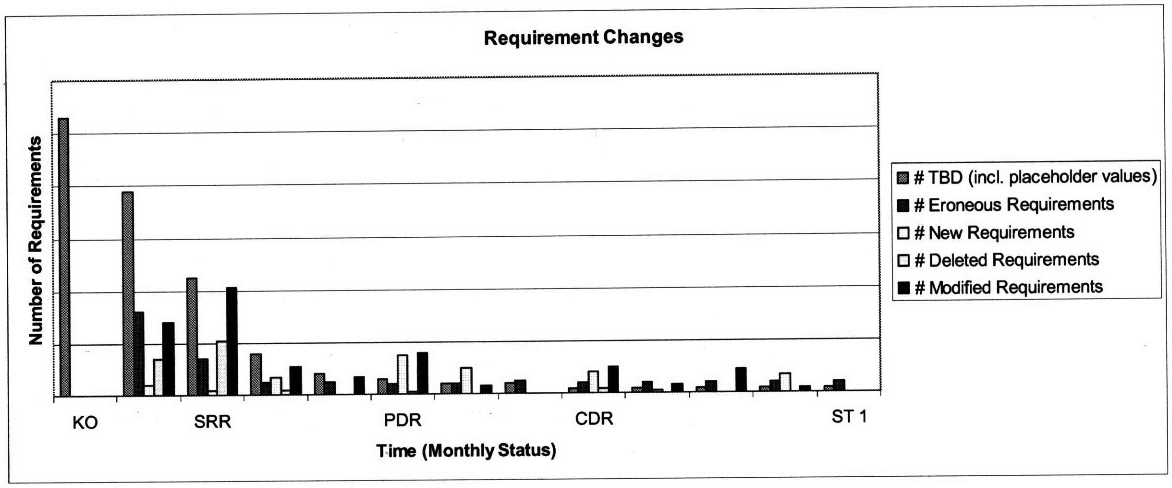


Figure 4-3 Hg3 System Level Requirements Change Time History – View A

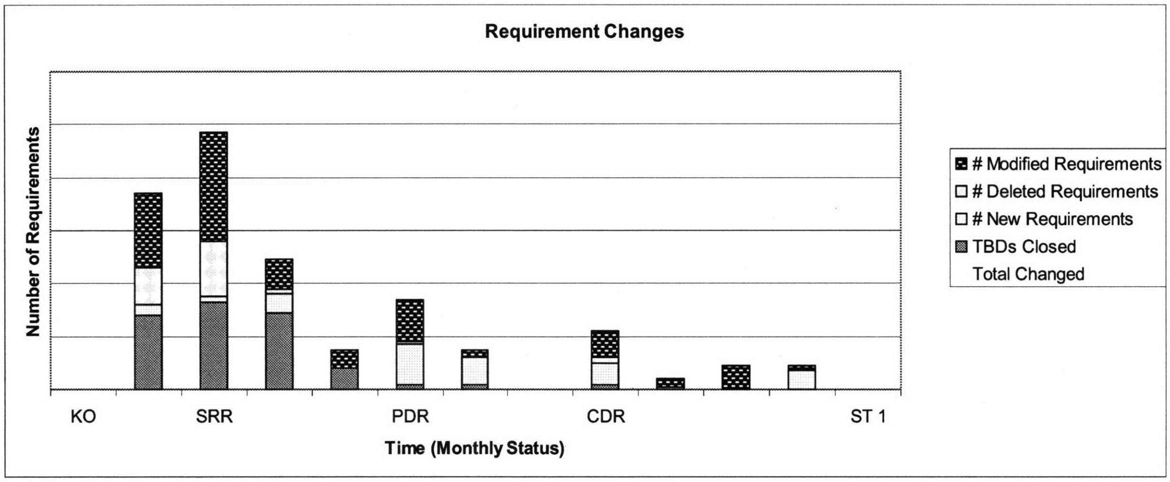


Figure 4-4 Hg3 System Level Requirements Change History - View B

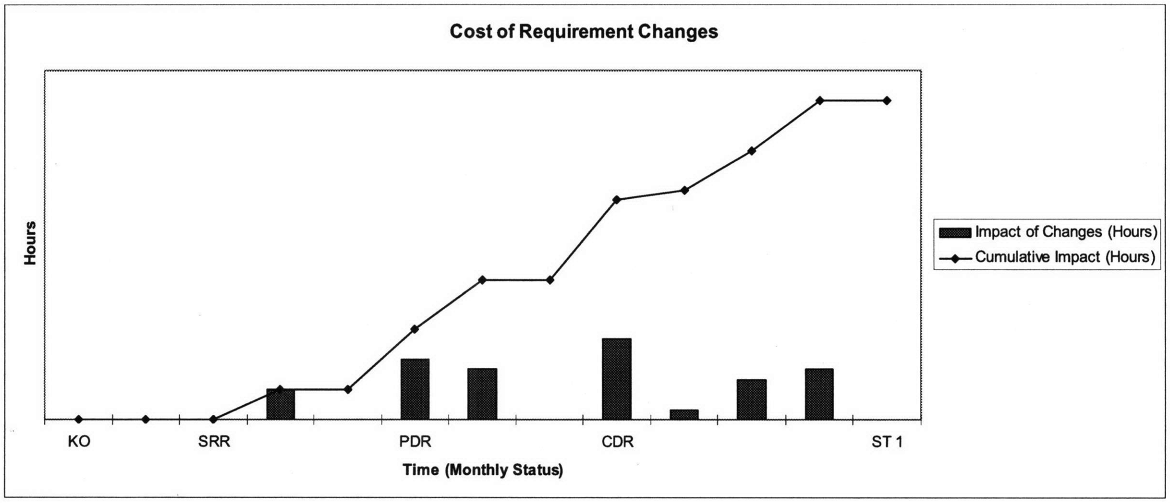


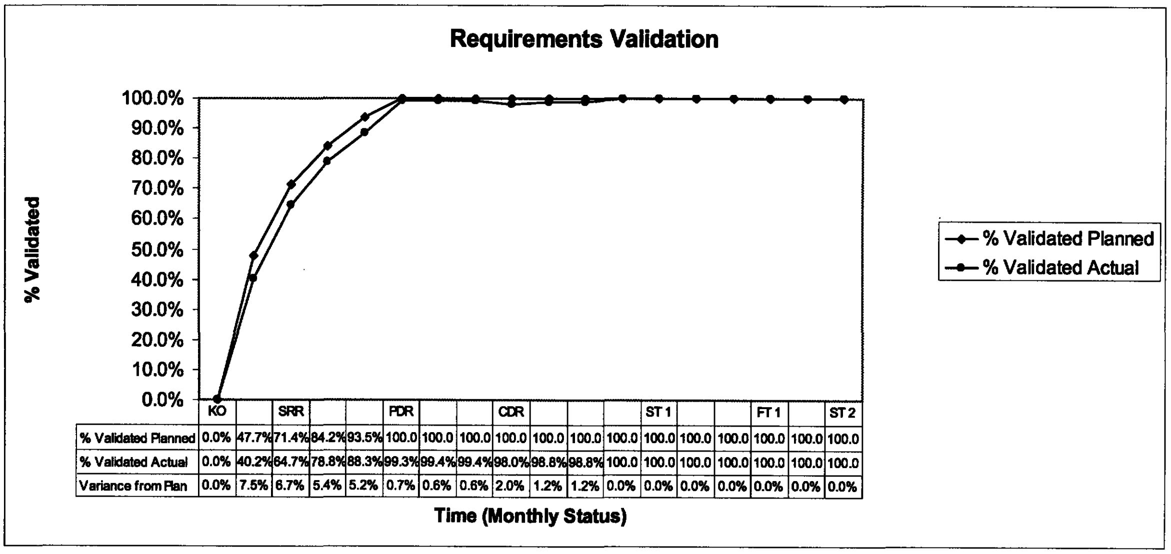
Figure 4-5 Hg3 Requirements Change Impact

Through discussions with the Alpha Systems SE team an upper threshold criteria was developed which represented the percent change from the previous reporting periods count. The SE team recommended the percent change decrease after each milestone review, in the case project; 8% up to SRR, 5% up to PDR and 1% beyond CDR. This threshold line represents the requirements stability expected by the project team from experience on similar projects. It is adjusted preceding each milestone to indicate the need of the development process to obtain a higher level of stability as the design develops towards toward final acceptance test. After CDR minimal requirements change is expected since changes at this late development phase are most likely going to cause architecture or design changes and create significant unplanned rework. The trend in requirements continued upward even past CDR, an indicator the system definition was not going to achieve the maturity expected post CDR.

Figure 4-5, Cost of Requirement Changes, provides a second perspective of the project requirements progression. It quantifies the impact of requirement changes in a reporting period. As shown on the chart several requirements changes were accepted after CDR. The impact of these were greater than larger quantities of changes prior to CDR, as might be expected the late changes drove rework across many areas of the design and some impacted production ramp up activities. The upward trend in predicted cost, person hours being a reflection of cost, for these changes should have triggered adjustments in the project. If most changes were customer driven the adjustment might have been to request additional funding and/or schedule relief from the customer.

The two middle charts lend insight into the type of requirement changes occurring over the development phase. Several new requirements were added late in this project. If an upward trend in requirements had been more visible to the entire project team and higher management it should have triggered an evaluation the voice of the customer process employed during the development phase. This would have highlighted a missing project risk; the team assumed acquirer and user needs were relatively stabile as represented in the SPS provided with the RFP. It became evident late in the project this was not the case.

Figure 4-6 shows the time history of requirements validation during development.



	KO		SRR			PDR			CDR				ST 1			FT 1		ST 2
% Validated Planned	0.0%	47.7%	71.4%	84.2%	93.5%	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
% Validated Actual	0.0%	40.2%	64.7%	78.8%	88.3%	99.3%	99.4%	99.4%	98.0%	98.8%	98.8%	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Variance from Plan	0.0%	7.5%	6.7%	5.4%	5.2%	0.7%	0.6%	0.6%	2.0%	1.2%	1.2%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%

Figure 4-6 Hg3 Requirements Validation

Validation remained close to on track through out the project, as indicated by the "Variance from Plan" row under the chart. Project team members interviewed established a trigger level of 10% up to PDR on Variance from Plan to assure validation remained near to schedule and downstream design activity impacts would be minimal. After PDR trigger levels should be reduced, but those interviewed stated this would be project dependent and was more a function of the nature of the requirements not yet validated.

Figure 4-7 depicts the progression of a three month moving average of the Requirement Stability Index (RSI). The raw RSI, also shown, was computed weighting requirement changes by 0.35, requirement impact by 0.5 and requirement validation by 0.15. Impact was normalized with an upper change threshold of 2 person years. The shaded area indicates an unacceptable volatility in the requirements and is intended to trigger action from the development team leadership, which would have occurred after PDR in this case.

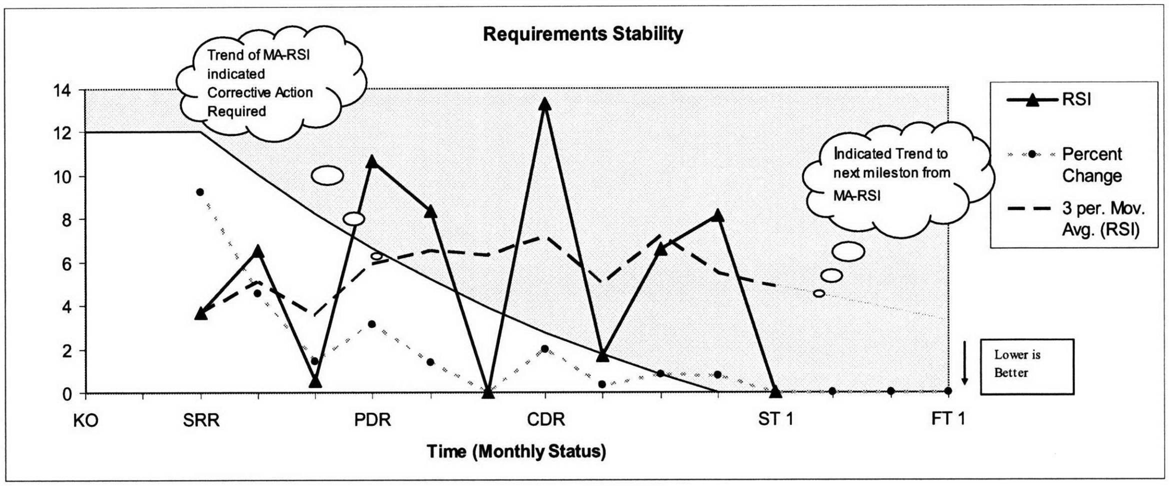


Figure 4-7 Hg3 Requirements Management Prognostic – Requirements Stability

Requirements percent change is also plotted to demonstrate tracking only the number of changes is not sufficient without some quantification of the impact of the changes in each reporting period.

Development team members indicated the trend line in Figure 4-7 is a telling indicator of one of the key issues faced. Alpha Systems planned the development based on management's view the project was a low risk extension of the existing generation system with well defined requirements. This contradicted the customer's goals, who desired design flexibility extending up through initial flight test (FT 1).

4.1.2 Risk Management

At project kick off seven risks were identified and entered into the risk management tool Risk Radar™.[72] These risks represented the set worthy of tracking based on discussions between the systems engineering team, program management and resource management; responsibility and risk handling actions were assigned as deemed necessary. The tool reported standard risk assessment matrix, with probability on the abscissa and impact on

---

[72] Risk Radar™ Users Guide, version 2.03, Integrated Computer Engineering Inc., June 2003

the ordinate, was reported as during initial program reviews.

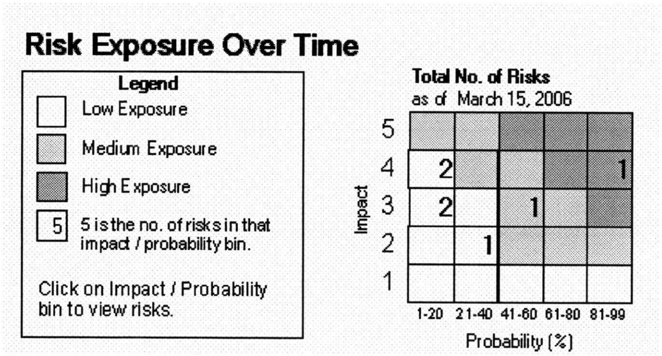
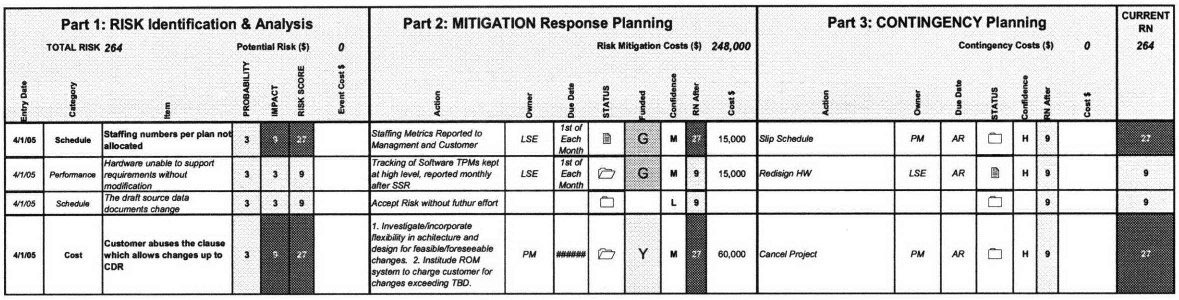


Figure 4-8 Hg3 Example Risk Assessment Matrix

Formal risk management changed to enumerated lists of issues and planned mitigations within program status presentations prior to PDR and was tracked in this manner for the remainder of the program.

Although risk assessment matrixes are valuable tools for risk management, they do not provide a visualization to support trending. Alpha Systems perceived a gap in the organizations risk management process and implemented an update to the process including introducing a new risk tool. As part of this update the business unit is also shifting to consistent use of the same process and tools for risk management on all projects.

To create a time history for the projects risk prognostics historical data and information from team members was transformed into a risk history. A risk management watchlist was generated in Alpha Systems new toolset, as shown in Figure 4-9. Stoplight color coding is used on each status cell in the watchlist for quick visual indication of risk status for current period.



																						CURRENT RN
Part 1: RISK Identification & Analysis							Part 2: MITIGATION Response Planning								Part 3: CONTINGENCY Planning							
TOTAL RISK 264			Potential Risk ($)			0	Risk Mitigation Costs ($)							248,000	Contingency Costs ($)						0	264
Entry Date	Category	Item	PROBABILITY	IMPACT	RISK SCORE	Event Cost $	Action	Owner	Due Date	STATUS	Funded	Confidence	RN After	Cost $	Action	Owner	Due Date	STATUS	Confidence	RN After	Cost $	
4/1/05	Schedule	**Staffing numbers per plan not allocated**	3	9	27		*Staffing Metrics Reported to Managment and Customer*	*LSE*	*1st of Each Month*	🗎	G	M	27	15,000	*Slip Schedule*	*PM*	*AR*	📁	H	9		27
4/1/05	*Performance*	*Hardware unable to support requirements without modification*	3	3	9		*Tracking of Software TPMs kept at high level, reported monthly after SSR*	*LSE*	*1st of Each Month*	📂	G	M	9	15,000	*Redisign HW*	*LSE*	*AR*	🗎	H	9		9
4/1/05	*Schedule*	*The draft source data documents change*	3	3	9		*Accept Risk without futhur effort*			📁		L	9					📁		9		9
4/1/05	Cost	**Customer abuses the clause which allows changes up to CDR**	3	9	27		*1. Investigate/incorporate flexibility in achitecture and design for feasible/foreseeable changes. 2. Institude ROM system to charge customer for changes exceeding TBD.*	*PM*	######	📂	Y	M	27	60,000	*Cancel Project*	*PM*	*AR*	📁	H	9		27

Figure 4-9 Slice of Hg3 Risk Management Watchlist

Time history of the number of risks tracked by impact category is shown in Figure 4-10

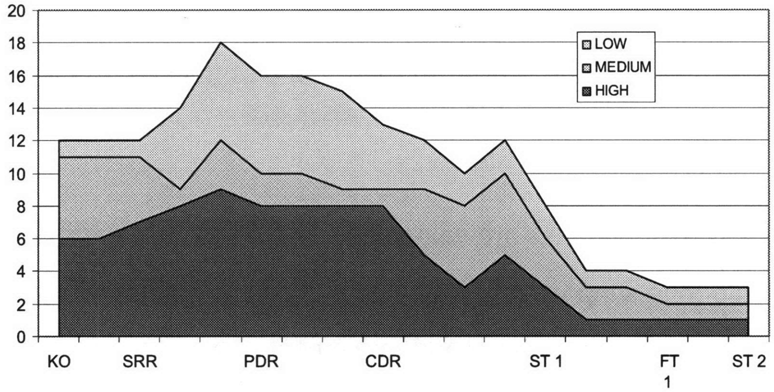


Figure 4-10 Hg3 Risk Impact Time History

Figure 4-11 shows a trend line extended as a means to predict future risk exposure. In the risk tool employed the total risk exposure is computed as the sum of all risks probability of occurrence times the impact of realization. The tool assigns a number to a low, medium and high in each of the categories (1, 3, 9). Current exposure takes into account the status and confidence in mitigations. Risk impacted the exposure as it was identified. The dashed line is the risk exposure upper threshold as determined by the systems engineering team based on experience with similar projects. A risk exposure lower threshold would have added valuable insight into the health of the identification process, indicating the minimum expected risks in similar projects.

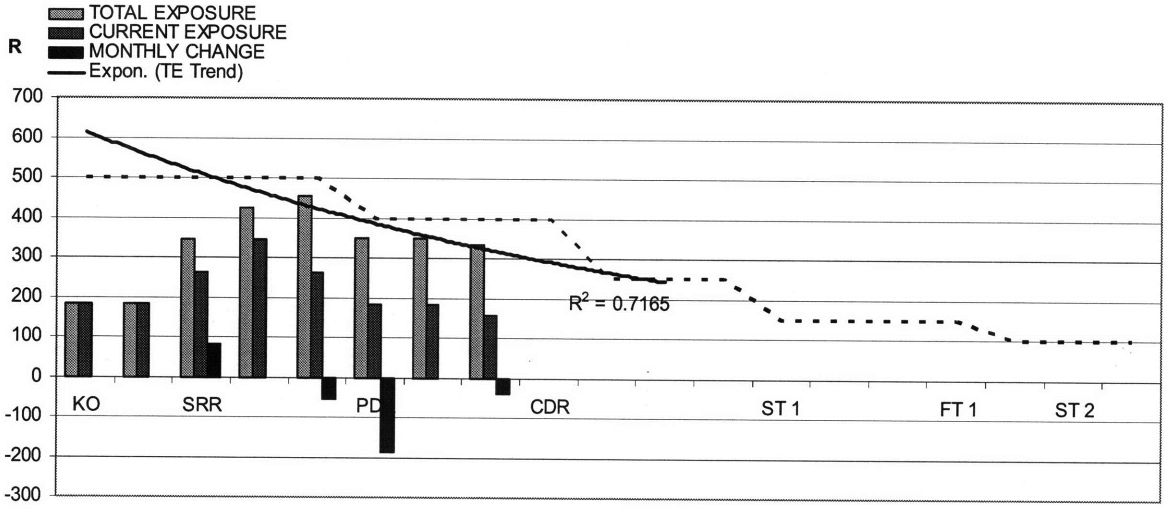


Figure 4-11 Hg3 Risk Exposure Prognostic Example

This visible control permits the systems engineering team to view current exposure and project likely risk trends. An exponential trend line was added to the chart which considered data from the maximum value forward in time. Based on experience, risk exposure tends to decay after the majority of risks are identified and managed. In successful projects this decay rate is fast but slows as the risk exposure approaches zero. Actions initiated in hopes of affecting the course of risk exposure when the trend indicate a likelihood of a future over limit conditions, as existed prior to CDR, are logged as part of the risk watchlist. Determination of a threshold line by the systems engineering organization should incorporate historical data of risk exposure in projects of similar complexity executed in the organization. In this case risk exposure is a new method for risk management to Alpha Systems and no historic data was available. No consistent risk management process had existed prior to 2006 when risk exposure tracking was implemented as the standard practice.

Figure 4-12 depicts Risk Exposure experienced during the project. As indicated, prior to CDR the SE and PM recognized a slow risk exposure decay rate was an issue and addressed changes required to mitigate the risk level during the formal CDR.

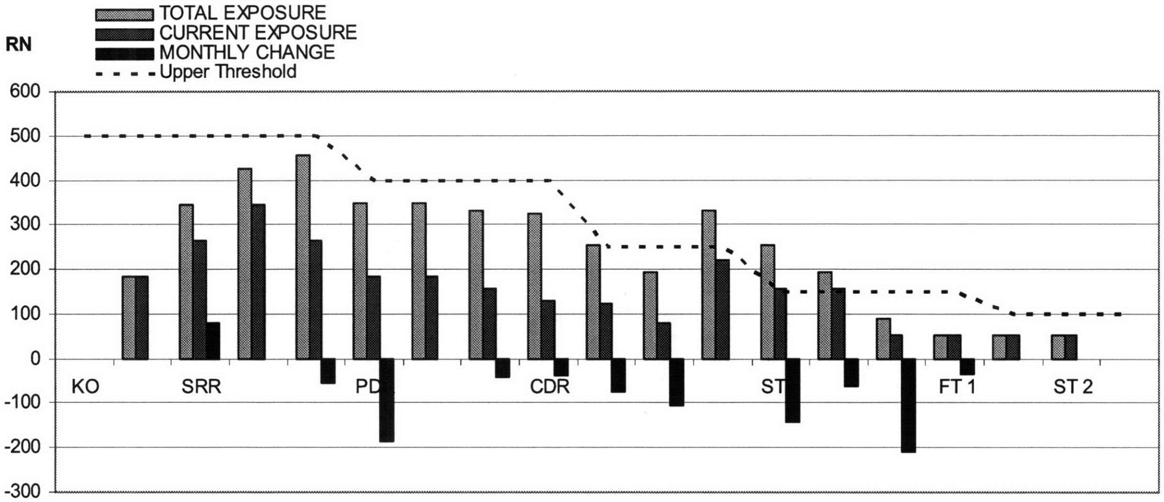


Figure 4-12 Hg3 Project Risk Exposure Actual

Figure 4-13 depicts the time history of the percentage of mitigations with funding gaps from the agreed to risk mitigation actions in the risk management plan. Mitigations exist for each risk classified as high or medium based on their risk number (impact * probability). A lack of funding to levels as planned for risk handling impedes the effectiveness of the risk management process. Beyond the simple relationship between funding and the ability to enact risk mitigations, the trend shows of significant funding gaps with no sign of decay after PDR and might indicate a lack of management commitment to the risk management approach which will impact other activities downstream. The trend might also indicate an inability to obtain funding for other reasons, e.g. inability to obtain staffing or other resources. In the Hg3 case, several of the gaps in risk mitigation funding remaining late in the project were ultimately deemed as acceptable without the planned mitigation, since the probability of occurrences had decreased.

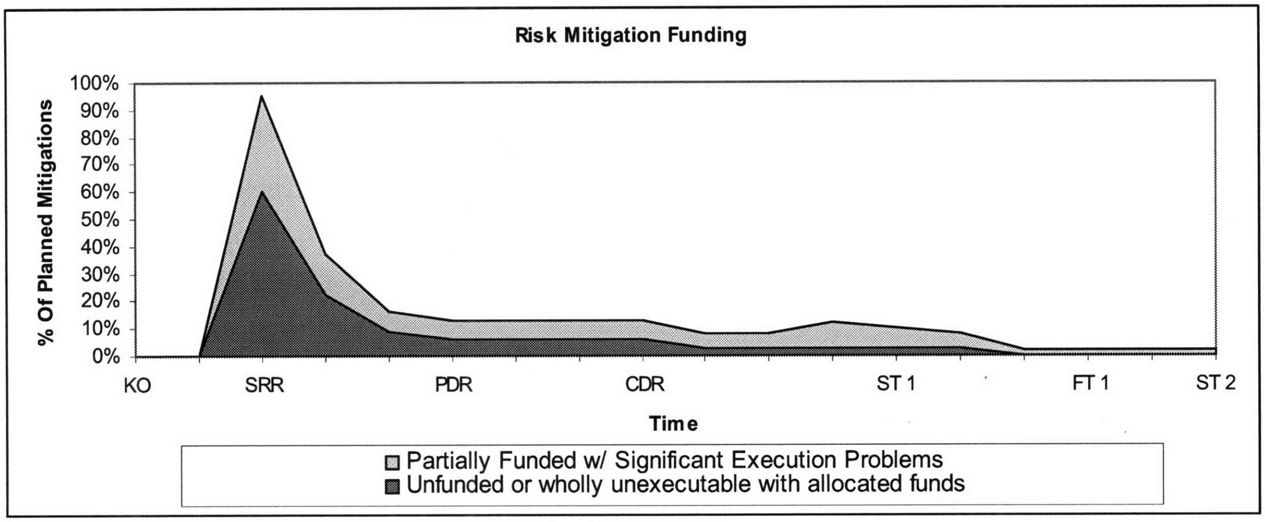


Figure 4-13 Hg3 Risk Mitigation Funding

Figure 4-14 indicates the mitigation action initiation aging. Any sign of late mitigation needs to be addressed. As a lagging indicator, mitigation actions not started for 90 days or greater should trigger re-evaluation of each risk, likely the risk can be marked as accepted at this point. The leading indicator from this could point to process or staffing issues. Based on the staffing history of the project the later is more likely. Leading or lagging, it does seem to highlight an issue with the efficiency of the applied risk management activity.

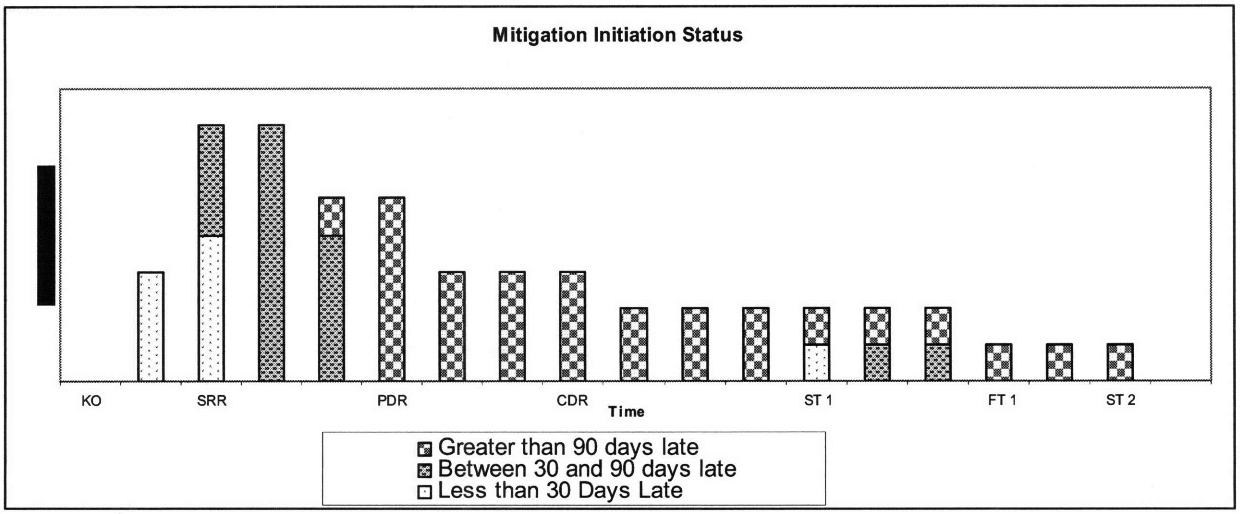


Figure 4-14 Hg3 Risk Mitigation Initiation Status

Figure 4-15 indicates the risks realized during the project. Although this is a lagging indicator of risks already realized, it is a prognostic of the effectiveness of the risk process applied by the project and perhaps more so, by the organization. It could also indicate issues with the inputs to the risk management process; funding, staffing, training and management commitment. The team attempted to quantify issues incurred which should have been managed under the SE risk process. Of the multiplicity of issues the project incurred over the development process, it was indicated a few might have been avoided through improved risk management practices.

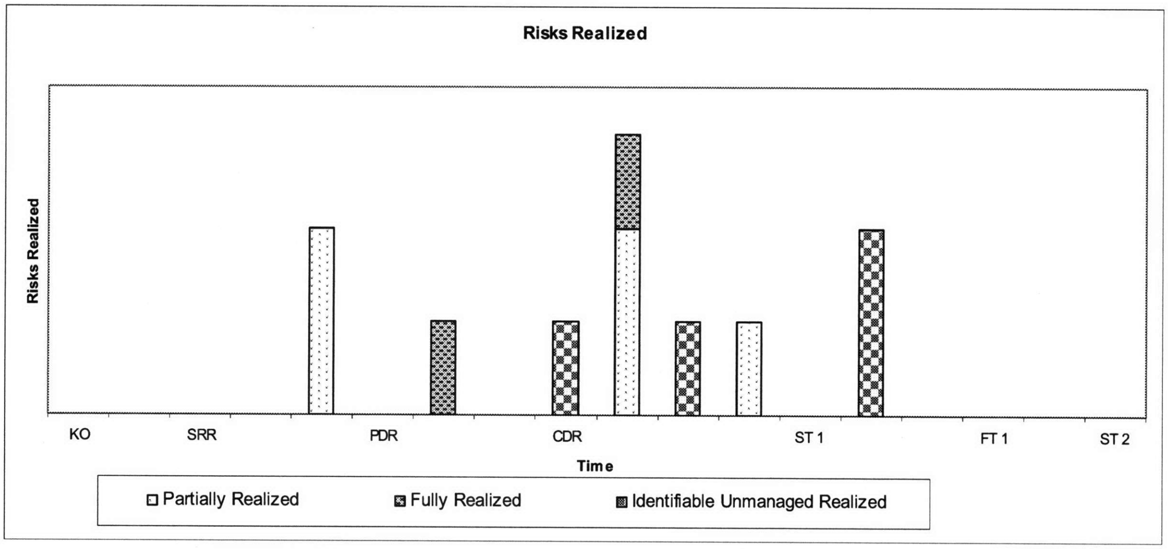


Figure 4-15 Hg3 Risks Realization

4.1.3 Interface Management

The Hg3 system was one of approximately twenty units of an electronics suite the customer outsourced for development as part of a next generation vehicle platform development project. Contract award for Hg3 fell late in the overall vehicle platforms development phase, which was perceived by Alpha Systems as beneficial since ~80% of the systems proposed to interface with Hg3 were completely defined. The onboard content of the Hg3 system interfaced with a significant number of on board electronic units to collect and record data. A large percentage of information was published on well established interfaces with which Alpha System had vast experience. The Hg3 also interfaced with a set of dedicated sensor which provided signals of the same characteristics as previous generation systems. One interface was a recent development and was added as a risk item for the project. At the onset of the project this interface only handled a fraction (less than 1%) of the overall information traffic.

Figure 4-16 through Figure 4-19 show the change and impacts for a portion of the development phase, similar to the requirements tracking charts.

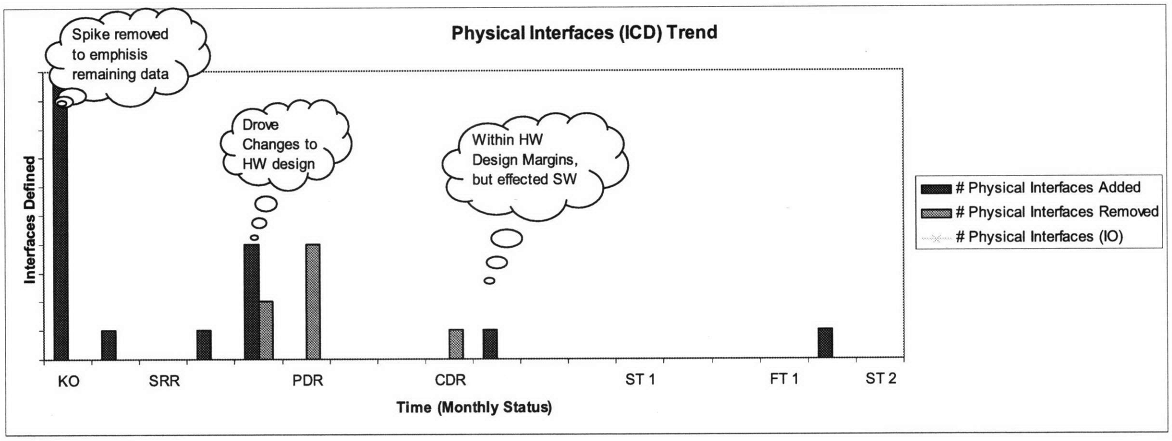


Figure 4-16 Hg3 Physical Interfaces Time History

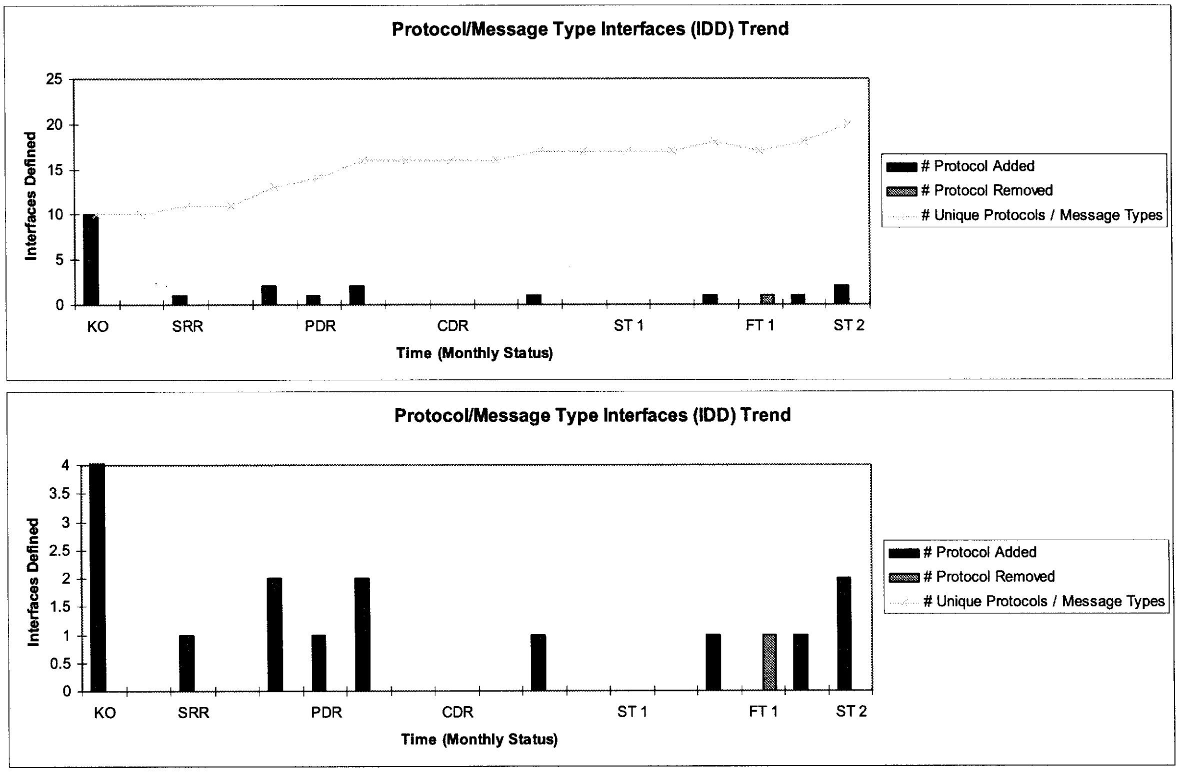


Figure 4-17 Hg3 Message Type Interface Time History

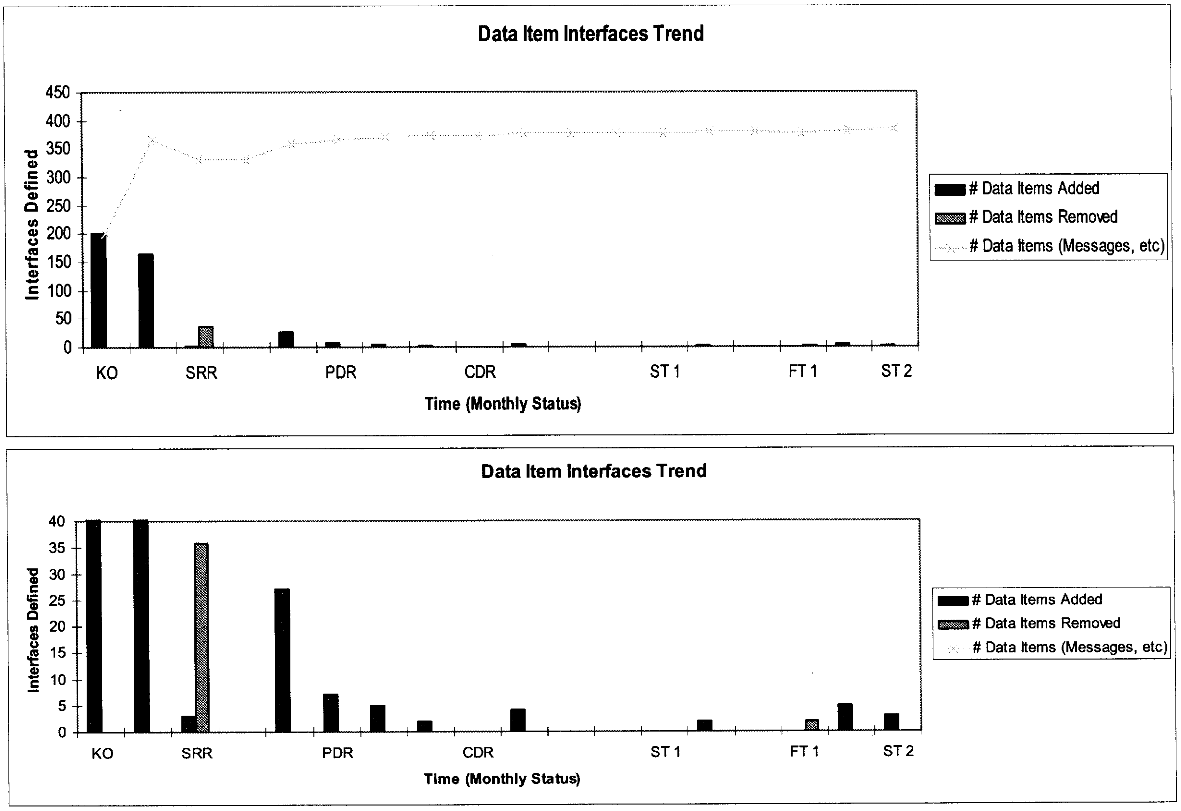


Figure 4-18 Hg3 Data Item Interface Time History

Trends for discovery of error in interface definitions, whether propagated from incorrect or ill defined source documentation or simply incorrectly defined in an internally developed ICD do not lend to regression lines. Errors are often uncovered during some form of testing. For Hg3, the first set of customer lab integration testing coincided with the CDR milestone and a series of errors were uncovered, refer to Figure 4-20. Later, system test uncovered additional interface issues. The costs accrue when the development team addresses these issues, which can be months or even years later. Projecting a trend line past ST2 in Figure 4-19 should trigger action from the systems engineering team, as costs are rising considerably. In reality the line flattened again as development activity subsided and later rose when the team started to address open interface issues, but the rise was not as sharp. A suggestion from the development team was the addition of the percentage of interfaces verified in the target environment. This would provide an indication of certainty in the system interface definitions and the implementation in the system. Remaining uncertainty in the interfaces after 100% of the interfaces had been verified in the target environment are changes to interface definitions outside of the purview of Alpha Systems. Several such instances of incorrectly defined interfaces in source documentation or changes not communicated were uncovered late in the development phase of Hg3 and drove unplanned rework. Assuring the customer or interface owners provide change information in a timely fashion is a responsibility of systems engineering, so this is a lagging indicator of issues in the systems engineering customer interface handling.

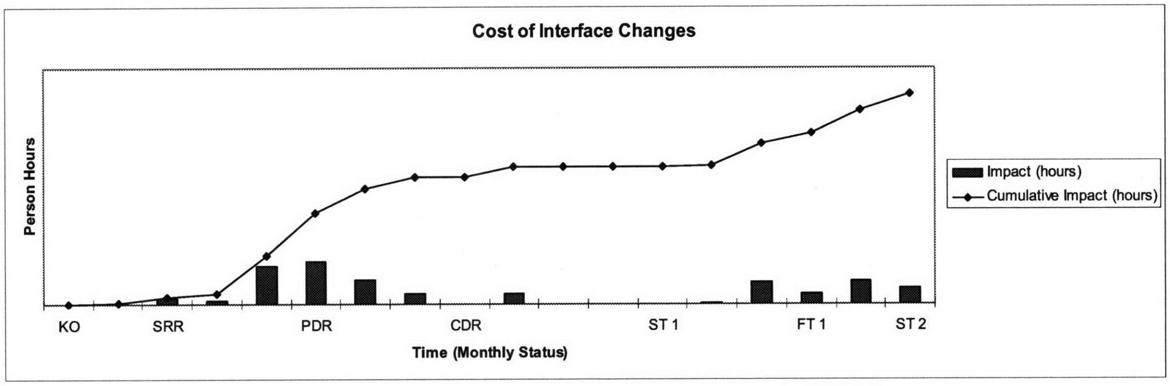


Figure 4-19 Hg3 Impact (cost) of Interface Changes

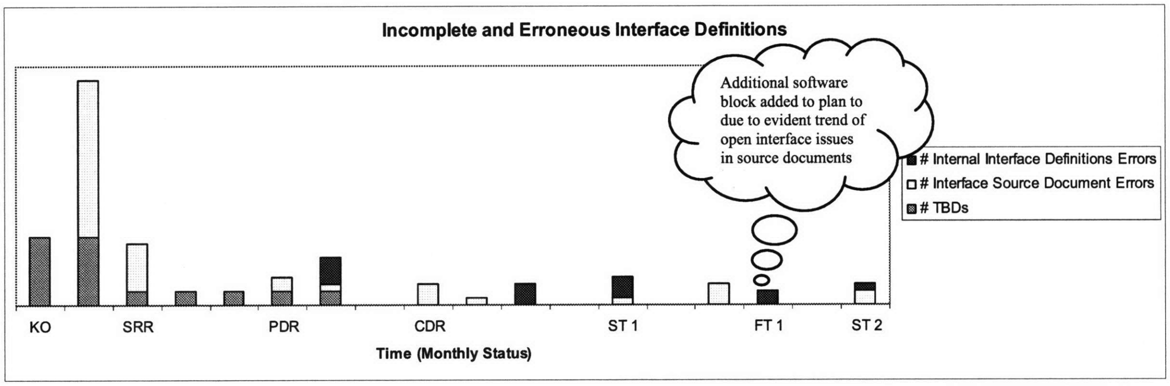


Figure 4-20 Hg3 Interfaces Definition Correctness and Completeness

4.1.4 Customer Technical Interface

The Hg3 project successfully completed three major design reviews in the development phase SRR, PDR and CDR. Visual controls for review action item closure are shown in Figure 4-21, indicating the items actually closed and the items planned to be addressed.

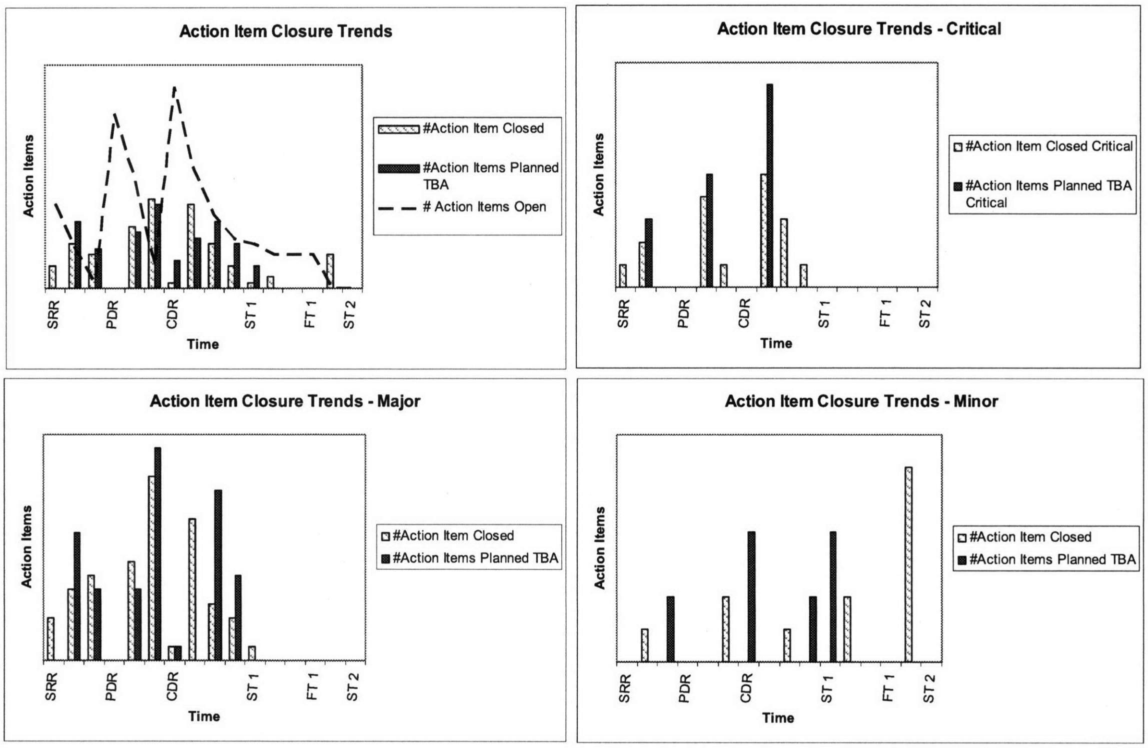


Figure 4-21 Hg3 Review Action Closures

The lagging of action item closure behind plan for all classes might indicate poor planning, process issues or staffing inadequacies. Action items defined as Critical and Major incurred a relatively minor slip from plan. Some of these issues required more effort to close then anticipated during planning and a few closures were delayed by the customer seeking approval for closure from other stakeholders. Cost and schedule impacts arose due to the underlying issues brought out by the action items, not from delay in response/closure. Minor action items lagged considerably, and this lag was deemed more a reflection of the unimportance of these versus other activities requiring the systems engineering team. Action item tracking for the development phase halted after action items for the final design review were closed.

Figure 4-22 is the action item closure index for the Hg3 project. Upper and lower thresholds are shown in orange and red respectively. During the project, closure rates declined below acceptable levels around periods of high systems engineering activity, design reviews and testing. After these activities the systems engineering team was able to refocus effort to aging action items and restore the closure index. The chart provides some insight into the closure rates and thresholds. However, it would likely be of minimal value to systems engineering, which generally is aware of the state of open action items during development.

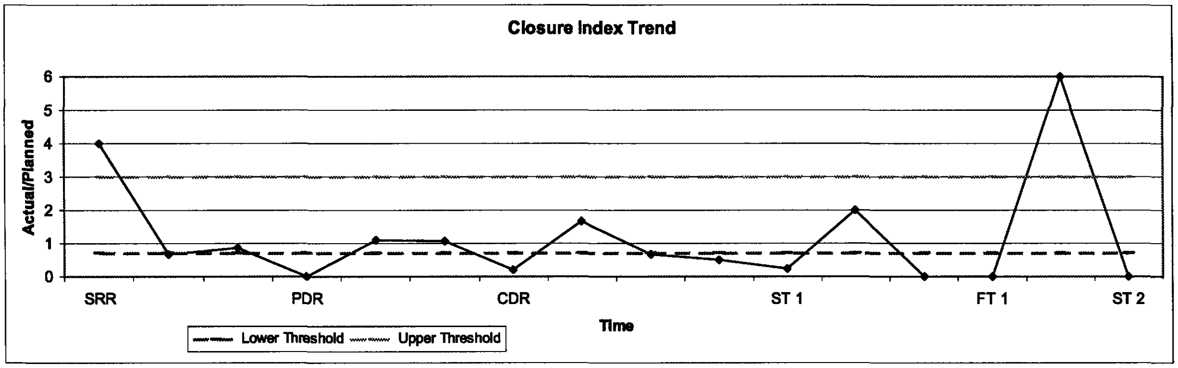


Figure 4-22 Hg3 Action Item Closure Index

An additional value added chart of action item closure trends would take into consideration the cost (person hours) of each closure action. This would lend to understanding the impact of action items on the cost and schedule as well as forecasting the maturity of the system definition compared to the needs of the customer. It also seems prudent to review action item closure trends against staffing trends, as these are coupled.

4.1.5 Technical Planning

Prior to the kick off meeting for the project the lead systems engineer generated a SEMP including a master plan for the non recurring effort (NRE) (engineering portion of the effort required to meet the SOW). This plan was approved by program management and the effected resource managers. Figure 4-23 and Figure 4-24 are charts of the SE staffing levels by task type and by experience level. Also shown is the planned and re-planned SE effort.

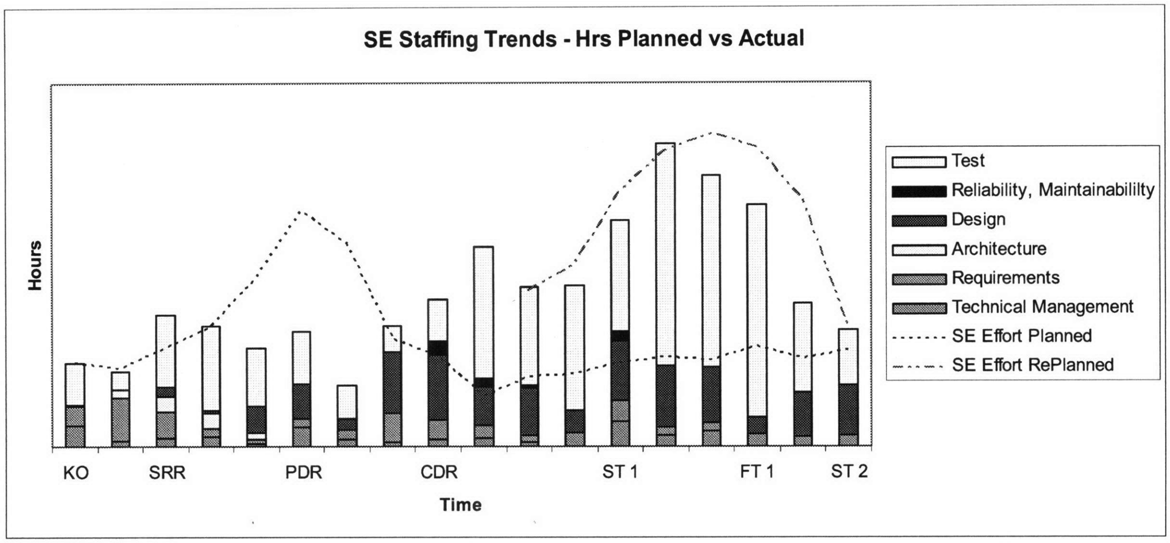


Figure 4-23 Hg3 SE Staffing by Task Type

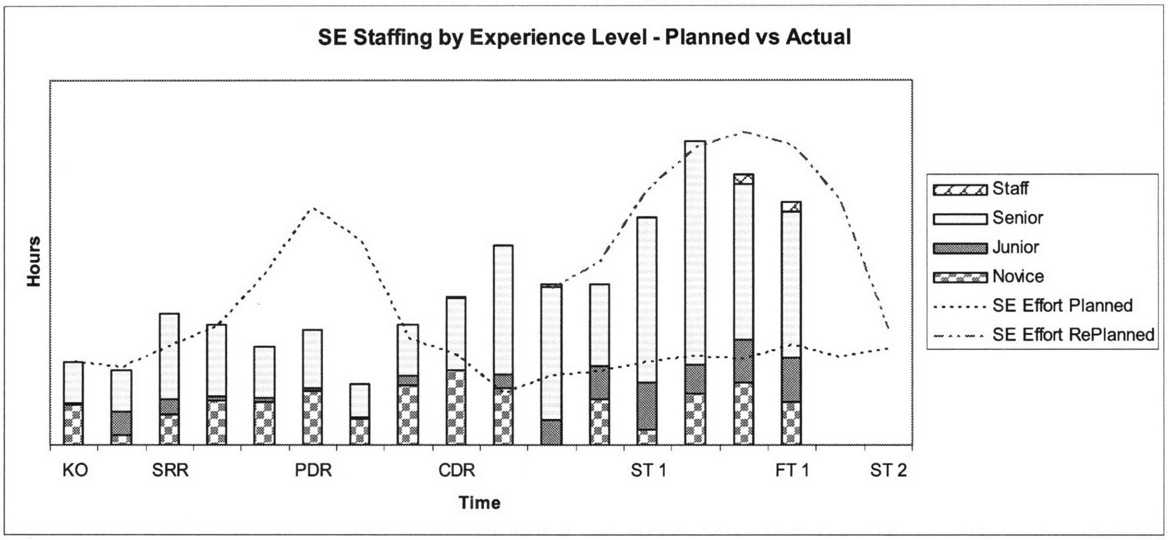


Figure 4-24 Hg3 SE Staffing by Experience Level

Re-planning required post CDR was attributed to several issues;

- Understaffing of the project in months prior caused by an Alpha Systems focus on other business activities which drew several key resources from the program. Understaffing trend is visible in the first three figures of this section.

- Poor planning, review of historic data indicated the testing effort allocated was not sufficient.

- Assignment of several novice engineers, who required training, to the project. These novice engineers also burned significant amounts of the design and requirements budget but did not produce at the level expected by the plan, which did not account for training of new engineering staff. The high percentage of novice staff is indicated in Figure 4-24.

- Scope creep.

Figure 4-25 shows the percent of deviation from plan. Threshold lines are shown on the chart, trends toward the outside of the green should trigger the SE and PM to act. The red dashed lines indicate staffing threshold which if crossed should trigger higher level management involvement to decide the proper course correction. Trends lines added on to this graph around PDR would have indicated the Hg3 project was heading off course and if resources could no be allocated the schedule would slip or other action was need to reduce scope.

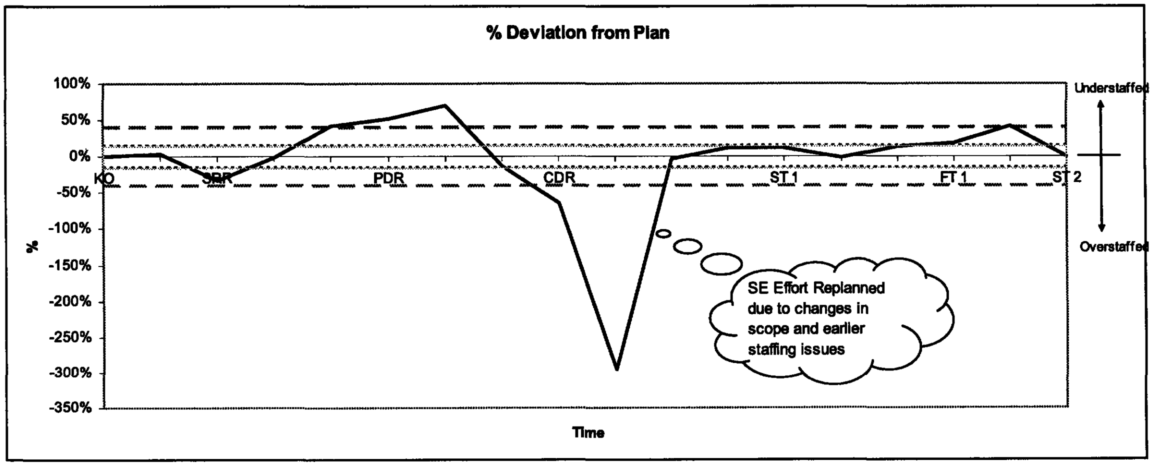


Figure 4-25 Hg3 SE Effort Prognostic - Deviation from Planned SE Effort

Figure 4-26 tracks the actual and planned level of systems engineer effort during the project. A review of this data prior to PDR should have raised alarm that early development phase effort was falling behind plan for the team which has its greatest influence on the outcome early in design. In this case, the project was re-planned post CDR with increased resources applied to meet the original schedule. The effort applied to complete systems testing exceeded plan and design issues identified during test drove an unplanned iteration of software development.

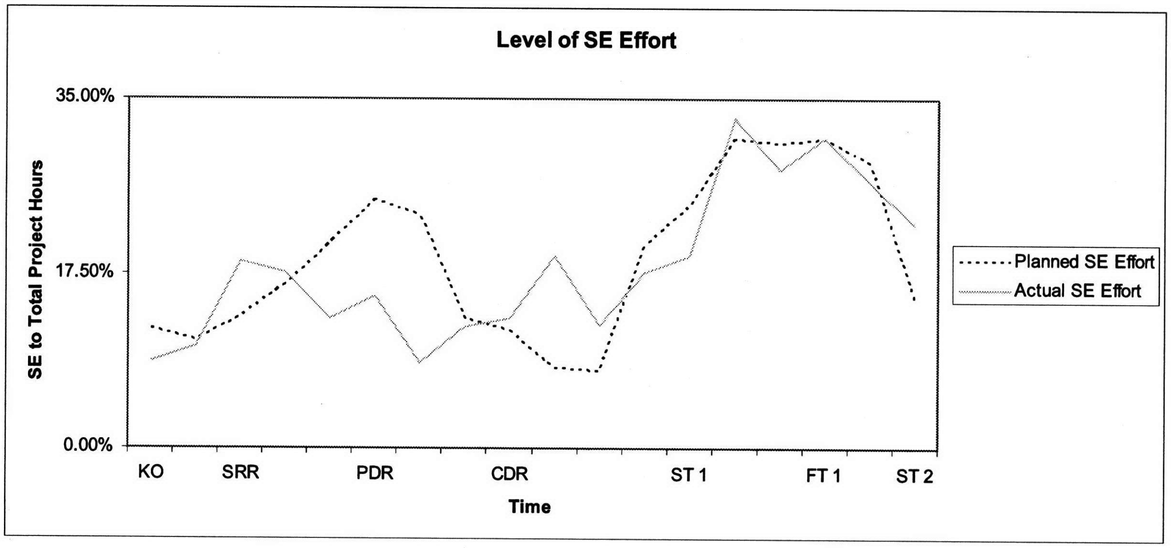


Figure 4-26 Hg3 Level of SE Effort

4.1.6 Design Synthesis and Evaluation

To measure the technical development of key requirements during synthesis of the architecture through design development several TPMs were tracked. Two TPMs tracked were charted against milestone time Figure 4-27 and Figure 4-28. Error bars were added to these modified TPM charts to indicate the level of uncertainty in the estimate or measurement. As design progressed and the models used for estimation improved the uncertainty was reduced. In both TPMs the error bars cross the thresholds several months in advance the mean, providing an early indication of impending design risk and immaturity of the design definition.

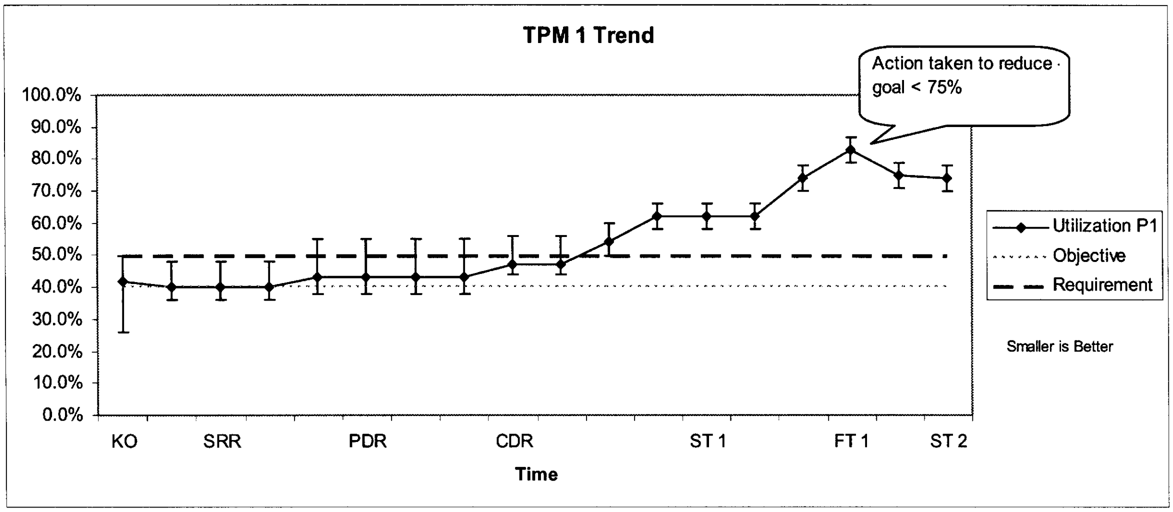


Figure 4-27 Hg3 TPM 1

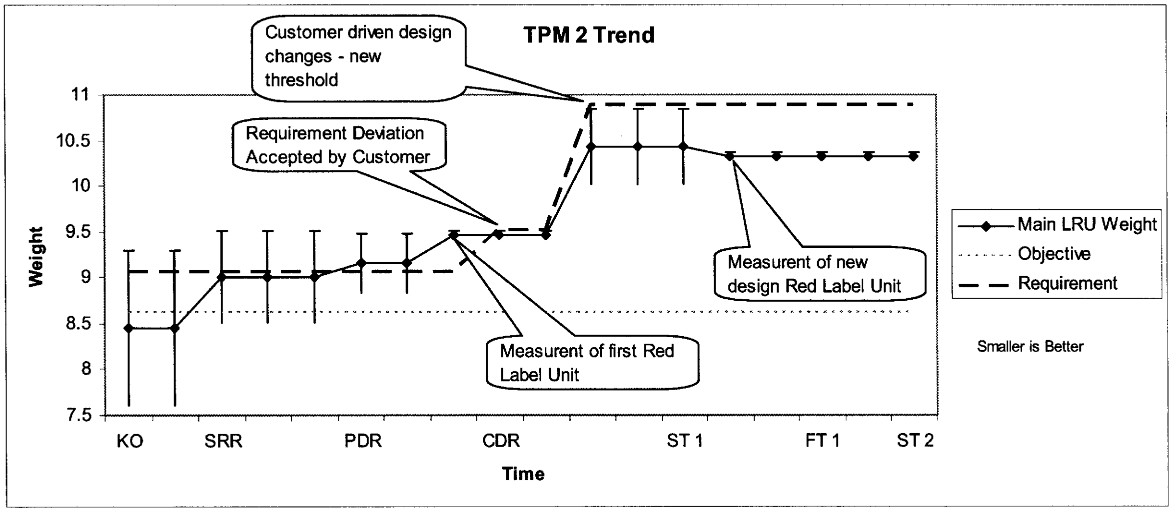


Figure 4-28 Hg3 TPM 2

4.1.7 Configuration Management

The project tracked customer deliverable work products through a formal CM process. A summary of the number of review cycles for one reporting period is shown in Figure 4-29. The systems engineering team agreed more than 2 internal cycles should raise concern. The first internal review generally ends in several comments which are classified as minor and are left to the author to remedy. Two review cycles is common for more complex work products, such as qualification procedures. Internal review cycles beyond three could indicate a need to review the review process being used by the team. Historically at Alpha

Systems 3 or more internal review cycles is an indication of piecemeal reviews, reviews occurring without the proper attendance, which represents unneeded PD waste and indicates likely schedule slip.

The customer of the Hg3 has an extensive history with Alpha Systems and based on this the company accepted a first version rejection was likely and often not a reflection of an issue with the technical content. A second rejection is a likely indication of problems uncovered during the customer review of the technical content. Follow on rejects are typically process issues, internal or external.

There is one instance of a work product taking four review cycles. This should have triggered the team to seek changes in the external approval process, since the cause was an inefficient review process where certain work products were reviewed by the customer and then by the end user in a linear fashion. Late in the development cycle, during testing, the time for external approval became a risk to meeting the users' vehicle level operational test schedule and onsite customer witnesses were provided for testing to speed the test report approval process.

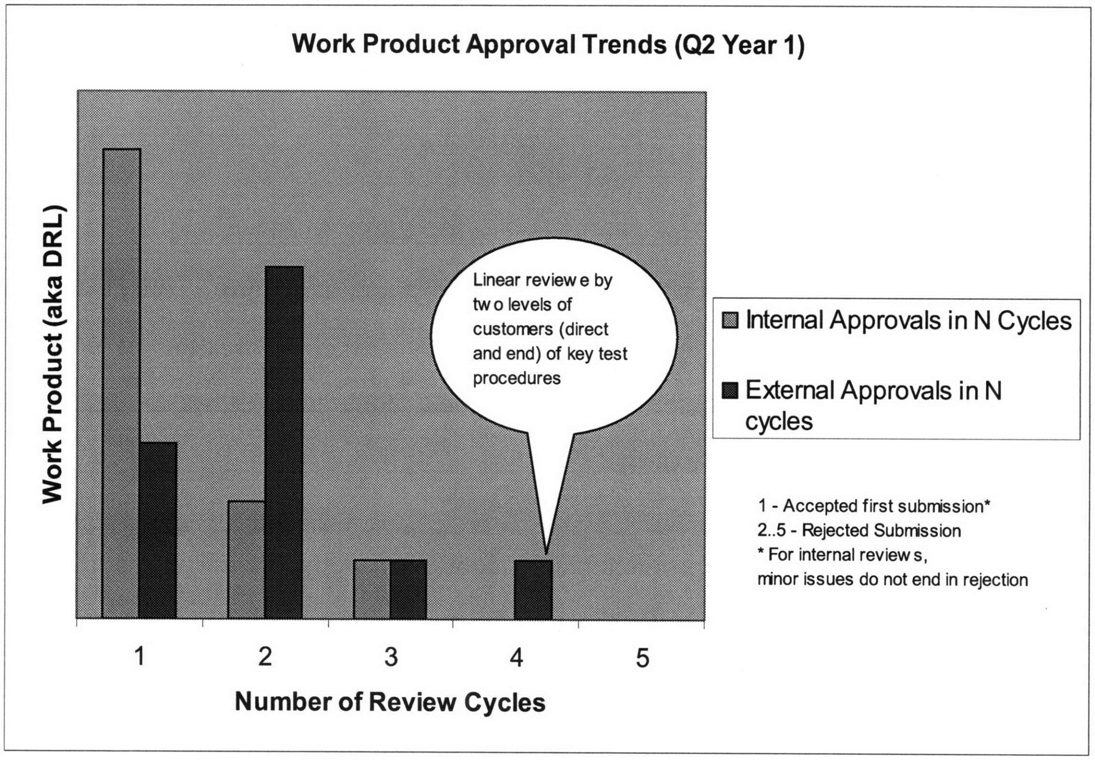


Figure 4-29 Hg3 Work Product Approval Review Cycles for Q2 Y1

4.1.8 SE Process Adherence

Figure 4-30 indicates which process areas incurred the highest number of finding during the development phase. The number of finding in the first three categories is a lagging indicator of process adherence issues with the project. It is also a leading indicator of possible issues with the existing process.

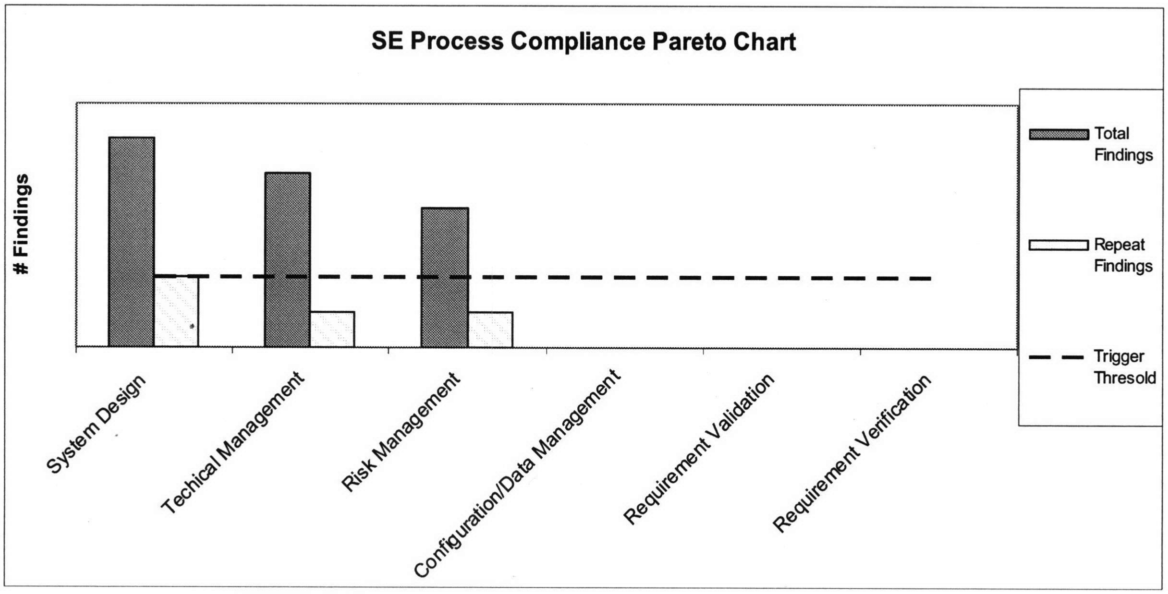


Figure 4-30 Hg3 SE Process Compliance - Pareto

Figure 4-31 indicates several minor finding at each milestone and a major finding during an internal ISO audit. The major finding was related to the lack of use of project risk management. Risks management had been reduced to an unacceptable level of formality. As noted in the risk management section, a consistent risk management process was not established, the finding here might have been a leading indicator of this process shortcoming. Alternatively it was an indication of ineffective application of systems engineering practices.

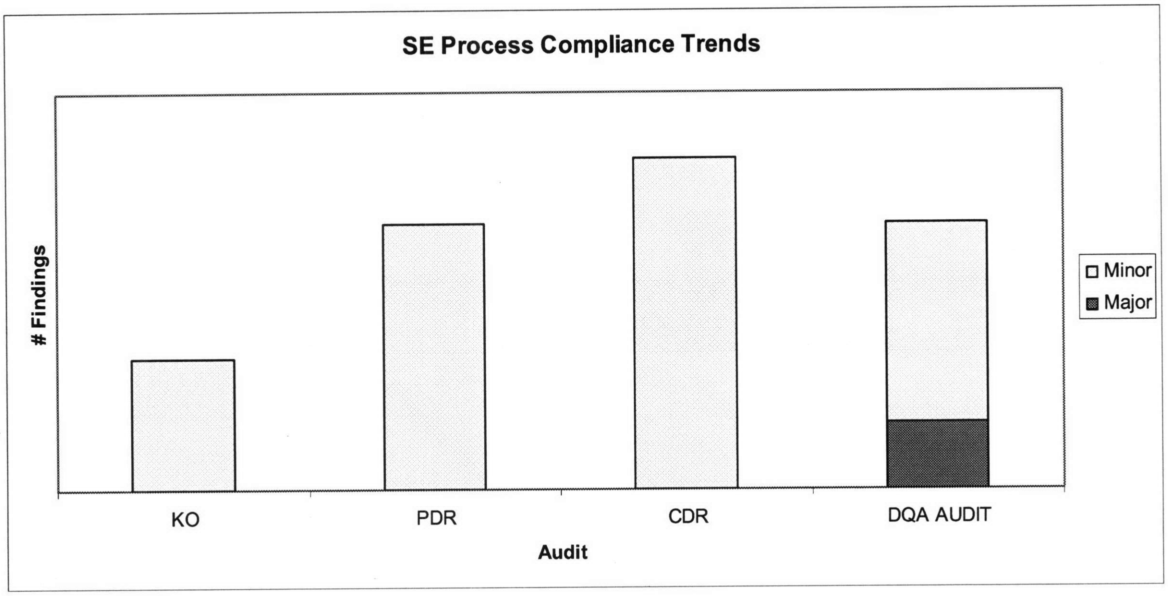


Figure 4-31 Hg3 SE Process Compliance Findings

4.1.9 Managing Team Cohesion

The intent for this measurement is to administer a brief survey addressing the team cohesion areas on a periodic bases, to be determined based on overall project length. Since this case study was based on application of the measurement constructs to historical data the time history for the measures had to be reconstructed through interviews with eight members of the development team. The likely error in the data due to the reconstruction method renders these as unusable to extract information related to the Hg3 project, but it provides a venue to display examples of the charts.

Figure 4-32 are the waterfall charts used to summarize team survey results. Red indicates a decrease and green an increase in value. The ratings ranged from 0 being the lowest and 5 the highest.

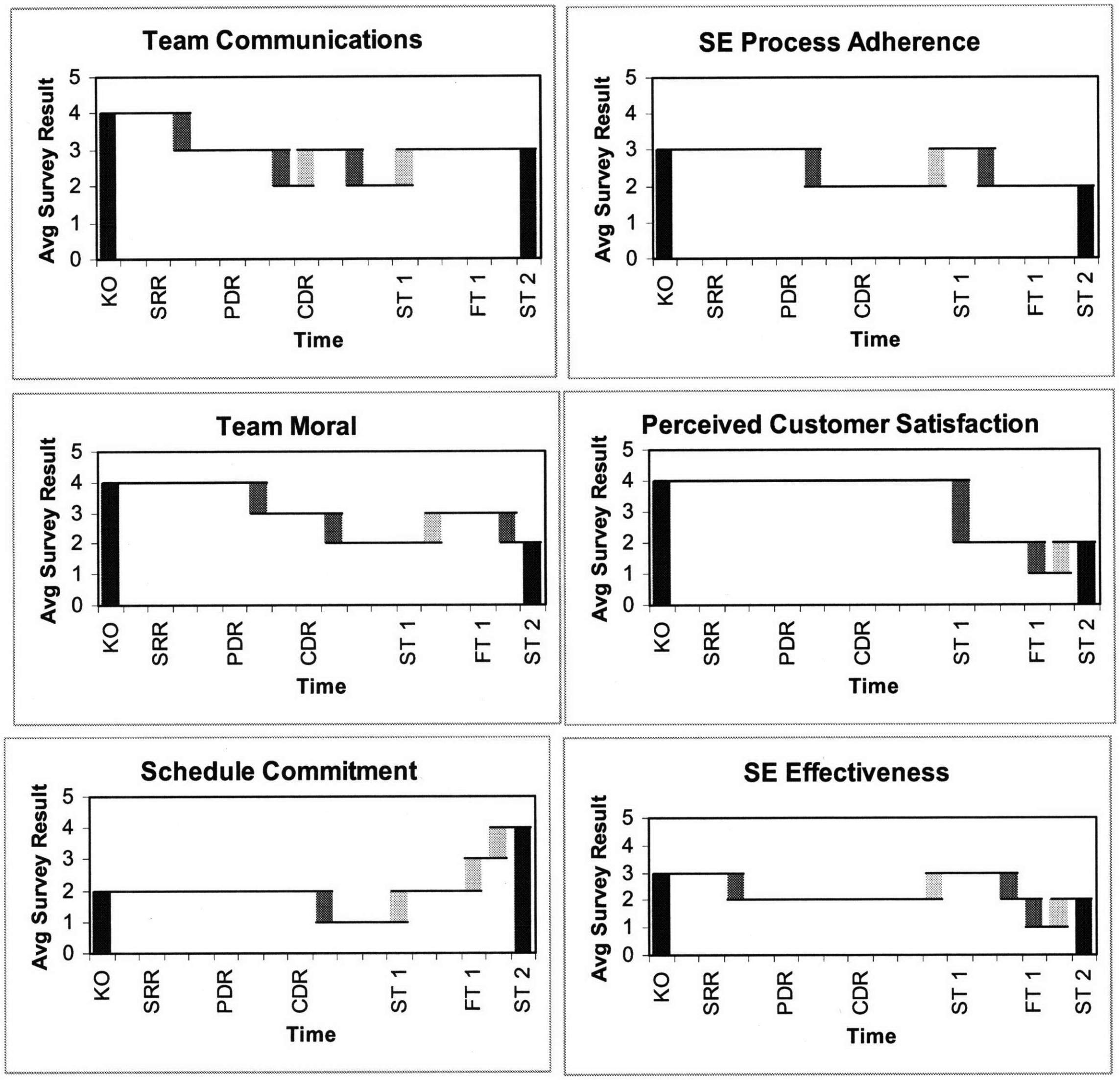


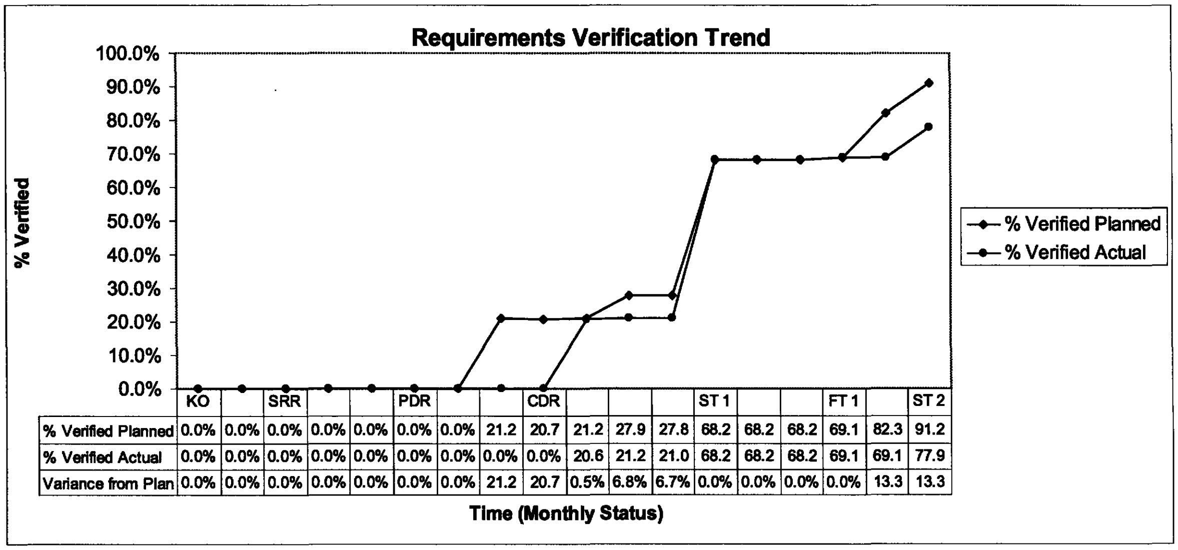
Figure 4-32 Hg3 SE Effectiveness Project Team Surveys

Of concern to the SE would have been the lagging indicator of an evident lack of team commitment to the SEMP schedule and a low confidence in the effectiveness and process adherence of the SE team. The leading indication in the downward trend of moral, team communications early in the design phase should have triggered action, since this is the critical stage for the design team where a high degree of inter team communications should exist. There is also indication that changes made in resource planning are improving the team's commitment to the schedule, as schedule pressure remained consistently high throughout the development phase investigated.

In discussions with the team it was noted the SEMP schedule was developed by the lead SE with input from the lead software and lead hardware engineers. However, the individual team members did not provide input or commit to the durations of there assigned activities. The team was not surprised by the low rating for systems engineering effectiveness. A lack of resource availability caused the systems engineering team to decide to forego planned activities and take on responsibility for non traditional activities, such as developing software modules, in an attempt to maintain schedule.

4.1.10 Design Verification

Requirement verification tracking, Figure 4-33, for trending and status is common practice at Alpha Systems. As the development process precedes slips in early phase tasks accumulate. As verification activities begin the project is often in risk of schedule slip and the late phase activities incur increased management oversight.



	KO		SRR			PDR			CDR				ST 1			FT 1		ST 2
% Verified Planned	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	21.2	20.7	21.2	27.9	27.8	68.2	68.2	68.2	69.1	82.3	91.2
% Verified Actual	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	20.6	21.2	21.0	68.2	68.2	68.2	69.1	69.1	77.9
Variance from Plan	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	21.2	20.7	0.5%	6.8%	6.7%	0.0%	0.0%	0.0%	0.0%	13.3	13.3

Figure 4-33 Hg3 Requirements Verification

In this chart, the observed value for prognostics is the variance from plan. The project gets off to a late start, but once the gap is realized additional resources are added to achieve the planned verification for System Test of block 1. Unfortunately a gap reappears prior to ST2 and the team is unable to adjust. The trend after ST2 implies the team will not meet plan and action is required.

4.2 *Chapter Summary*

This chapter presented a case study where several of the measurement constructs from previous chapter were applied to data from a historical project. Leading and lagging information content was discussed as well as the actual outcomes. The project is considered a successfull endeavor by Alpha Systems and the customer. Several offspring of this lead product are in development and fully funded. The project has seen value added through systems engineering and costs incurred as a result of the inefficient application of systems engineering over the entire development lifecycle.

The development phase of this project ran approximately 60% over budget and 45% behind schedule. Although this is not abnormal in complex software intensive systems development, it indicates significant opportunity for improvement. Some of the attributed causes of the cost and schedule overruns are; scope creep, late changes accepted and ensuing rework, staffing inadequacies, staff skills mix not as planned, insufficient early phase architecture reviews, less software reuse opportunity than planned, verification shortfalls and planning oversights.

5 The Value of Systems Engineering Prognostics

This chapter expounds on the value of the systems engineering prognostics as applied to the case study presented in the previous chapter. Some insights are reviewed from informal interviews regarding systems engineering prognostics with system development experts whose job functions included senior systems engineer, program manager, software engineer and resource manager.

A software engineer interviewed indicated the displayed measures for requirements management do produce valuable insights when short extensions are made through trending, as he had used similar approaches for software requirements with some success. However he also stated that "application of robust requirements management practices is critical to the success of systems development and is often held suspect in cases of faults in software intensive systems. To gauge the effectiveness of the system level effort which elicits needs, abstracts intent and translates external information into requirements which define the system to be designed the indictor should go beyond counting changes and customer validation."[73] This sentiment was furthered by another interviewed indicating a significant percentage of product issues, some driving multiple rework cycles, were introduced into design due to requirements which lacked clarity, contained sufficient supporting detail or were ambiguous for software and hardware engineering to interpret the intent. Perhaps a revealing sign of the proliferation of this issue in the embedded software industry is the insertion of the following statement in the commercial aviation software standard DO-178B[74];

Research into software development programs suggests that as many as half of the software errors actually result from missed or incorrect requirements or the incorrect interpretation of requirements.

It was indicated that software issues are conscientiously tracked and cases where root cause is determined to be the system level requirements are indicated on trouble reports. There

---

[73] Sr. Software Engineer interviewed

[74] RTCA, DO-178B, Software Considerations in Airborne Systems and Equipment Certification. 1992.

was no evidence this information retained in the software issues logs was reviewed by systems engineering or used in any existing reported metrics. Of course, an indicator based on software issue logs is unlikely to provide considerable foresight into future trends and therefore would be lagging not leading information. Known issues can remain off the logs or without root cause determination assigned for weeks or longer. What information could be used as a prognostic of the clarity of the requirements to the internal development team? Tracking issues from peer reviews of system requirements might provide leading insight. The peer review process in many development organizations has been applied in a formal manner to all deliverable or critical design artifacts. An indicator that considered planned versus actual and attendance of the right reviewers might permit trending to extract additional foresight to the health of the requirements management process.

Others interviewed noted the forms of measurement displays provided insight into trends of the systems engineering efforts effectiveness in the requirement management role beyond any metrics currently in use in their organization. A perceived issue with a focus on only requirements stability was voiced as "...can't view change in requirements as bad in all programmatic environments! The systems engineer must consider how the change affects cost, schedule, customer satisfaction, etc."[75] However, it is not the intent of the indicators to deterministically declare the systems engineering efficiency bad or good, instead these are inputs into a decision process which requires the right people involved from the project to determine cause and course of action.

It was noted that although the moving average applied to RSI in the charts from the case correctly indicated the state of volatility in the requirements was not settling rapidly as expected, such a derived measure requires additional validation before it could be placed into use. A participant at the LAI knowledge event on leading indicators stated more established requirements volatility indicators are in use at other organizations, which suggests additional research into more accepted methods of deriving such a measurement is appropriate.

---

[75] Systems engineer interviewed

The experts interviewed indicated trending of risk handling measures such as mitigations funded and initiated would be valuable. There was skepticism that a meaningful method to set thresholds on risk exposure for each project could be established. Acceptable levels of risk are dependent on several variables; problem complexity, product complexity, supplier relations, customer relations, project team, organizational stability, program managers risk process, teams experience level, etc. Setting levels for the organization without taking into account the influence of these variables would produce non value added limits. The other issue this organization faced was a lack of historical project data for risk management. Risk management data was often discarded when a project ended.

Systems and software engineers interviewed indicated errors not identified in the early phase of development in the interface definitions are indicators of issues in team communications and maturity of the systems which are interconnected to the system under development. If tracking an active project one would expect the Team Cohesion Management measures to confirm a communications issue existed. Inter-team communications is a strong influence on project success and it is a key attribute of effective systems engineering to "bridge the gaps between the different engineering tribes."[76] It was also indicated interface errors and changes, both inside the product boundary and external, were a contributor to the strength of the rework cycle late in development.

There were mixed reactions to the predictive nature of tracking of project team surveys. Most indicated project members could be a valuable source of information about the effectiveness of the systems engineering practices. It was explained that some of the more experienced project members develop reliable "radar" and seem to sense when a project is headed off track before any other measures might contain similar information content. This supports the hypothesis that there is predictive value in team cohesiveness as a prognostic. It is ideal to have one measurement process in use at an organization and since not all program managers are created equal the process cannot rely on each program manager's ability to select and read people. Several concerns were voiced about the sensitivity of survey methods to "circles of discontent." One program manager stated

---

[76] Interview comment from resource manager

"there are key people on each project I use to gauge the team effectiveness... in a small project team the noise from a few might attenuate the signal from those with real foresight."[77] This practice creates a built in bias, which lends one to question which is truly a better approach. Perhaps there is middle ground in requiring a minimum sample size for the survey data used to generate a set of unbiased prognostics.

A suggestion was made to add the actual and planned values by type and experience to each staffing time history. Being able see the expected level of experience versus actual as opposed to just actual to planned total effort would provide a better indication of how far off course systems engineering effort is. For skill codes, having some insight to the level of effort of one code might indicate gaps not otherwise visible, e.g. if test engineering did not staff as planned during test planning activities in the preliminary design phase.

There were general concerns voiced on tracking process adherence for use as a predictor of effectiveness. Several felt this would drive the wrong behavior. In order for this to be effective the measurement can't be used as a policing tool. Adhering to the open statements in process documents and following the right process can be vastly different activities. It was also noted the stability of the processes themselves contributed to capability of the process user to adhere. A suggested augmentation to the process adherence measurements was a measure of process stability. Trends across multiple projects linked to changing internal process might indicate need for additional training or adjustment to the processes.

Program managers questioned the lack of a measurement to track the health of the recurring costs committed in the development phase. One of the interviewees stated "... one of the missing links between the development phase and the production phase is stronger engineering accountability for recurring costs. Product cost is a factor during component selection and make/buy decisions, but think of the gain if engineering tracked recurring trends against each design decision and reported this weekly to the IPT."[72]

---

[77] Interview comment from program manager

Concern was expressed about the cost of collecting and processing the level of data required to implement the complete set of measurements. Without substantiation that the measurement and reporting process would add value it would be difficult to convince management to invest. Initiation would require a high level manager as a champion or a significant failure attributed to insufficient systems engineering. One suggestion was to prioritize the list of measures by best value and start with a limited set of measures.

The following are the key recommendations of the experts interviewed.

- A cost matrix and associated guidelines to tailoring should be developed to support a stepped approach to implementation in order to gain management commitment. Also needed is a rating of the value of each indicator relative to development phase. Discrete markings that the indicators apply in each phase are not sufficient information to support trade off decisions. Metrics have shelf life, tracking and reporting should stop before this expires.
- Additional measurements are recommended; measure of requirements clarity, measure of process stability and a measure of the designed recurring cost.
- A clear mapping between each prognostic and future trends of more tangible attributes of the project, such as cost, schedule and performance are needed.
- In implementing such a measurement system it needs to be clear the system exists to improve the process and allow for adjustments not to support policy policing. It is not advised all of the prognostics become standard status reporting to the acquirer or other external stakeholders. Measures used for external status should be a concise and limited set which takes into account the political environment the project exists in while providing an accurate view of the key project attributes of concern.
- Success of measurement programs of this magnitude requires a high level champion in the organization. Driving metrics practices from the bottom, or middle, historically fail.

- In general those interviewed agreed there was benefit in pursuing prognostics of systems engineering "goodness" and felt additional upfront effort would improve the likelihood of implementation success.

6 Conclusions and Future Research

6.1 *Conclusions*

The goal of this research was to evaluate the effectiveness with which a specified collection of prognostics would forecast the efficiency of the application of systems engineering in a project and within an organization. Evaluation was performed through application of measurements to a case study of historical project data.

Those interviewed agreed systems engineers and project leaders need improved early indicators that provide confidence the system will perform as specified and designed. There is subjective evidence from the case study the addition of systems engineering prognostics at the project level would provide new information content for systems engineering and program management to proactively tune the systems engineering activities and effort to seek an optimal output of information during the design phase. Further, it is recognized to extrapolate useful trends requires "smart people" people, people who know the program and know historically how the organization performs.[78]

Some measurement gaps in development phase which warrant addition to the systems engineering prognostics were identified. The concept generation phase lacks sufficient measurements to address the questions critical to determining the degree of concepts considered, the effectiveness of the selection process and the completeness of stakeholder needs elicitation. Some proposed measures include the number concepts considered versus historical norms for organization, IPT effectiveness, number of concepts considered versus historical norms for organization, use of early phase simulation or set based design, experience of concept generation team, concept generation team dynamics, number of non traditional concepts and measures of flexibility in the early architecture. Other gaps identified include assessment of design decision influences on recurring costs, team stability (i.e. turn over within the project) and requirements clarity.

---

[78] Notes from LAI Knowledge Event on Leading Indicators, Jan 10, 2007

One of the issues unintentionally uncovered in this work was the inadequacy of existing historical data, even in organization rated CMMI level 3 or greater, to support setting of thresholds or reconstructing time histories of the salient features of the development phase without extensive interaction with the original project team. Far too many decisions made by the project team are retained as tribal knowledge, not documented in project specific artifacts, stored for knowledge sharing purposes or shared outside of the project tribe. While the addition of a formalized measurement program addresses some of the issues related to capturing the project history, it does not address the fidelity at which design decisions should be documented or where this information is documented. More robust specification practices, such as intent specifications[79], provide a path to manage the design decisions. Management of this data is invaluable in reducing error in follow on engineering work, e.g. post production block upgrades, cases where designs are reused and in managing knowledge for lessons learned.

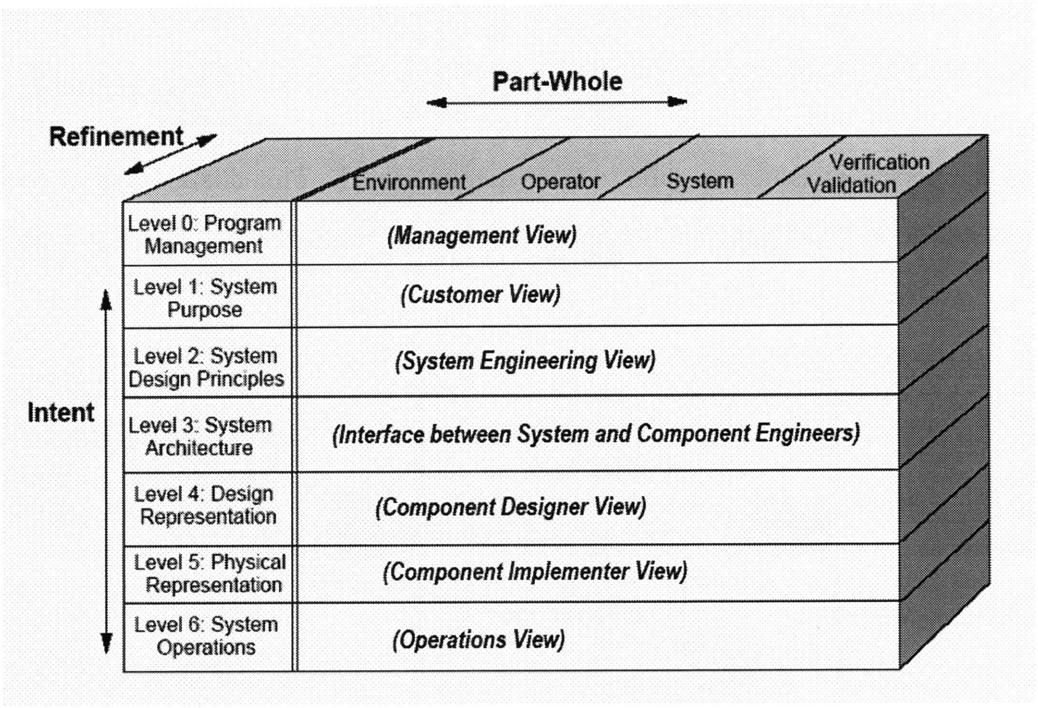


Figure 6-1 Intent Specification [Leveson]

---

[79] Leveson, N.G., Intent specifications: an approach to building human-centered specifications. Software Engineering, IEEE Transactions on, 2000. 26(1): p. 15.

The statement that organizations and people learn much more from failures than from success is a reflection of "single loop" learning[80], Figure 6-2.

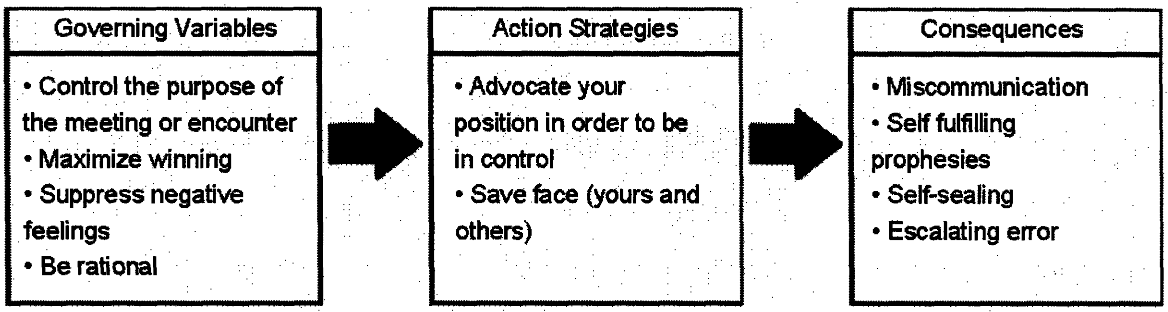


Figure 6-2 Single Loop Learning [Frey]

In this model success reinforces existing beliefs about cause and effect relationships. For example, a quality engineer is likely to conclude the success of a project is due to good process and adherence to that process. Minor faults result in a change based on the espoused cause and effect relationship, i.e. enact process policing activities or adjust the policy. Until the failures grow large, this self sealing process perpetuates. Only when faced with an extreme outlier in which the failing of existing mental models can not be avoided will the person consider revision to their mental model. This challenging of ones mental model based on experienced situational outcomes is "double loop" learning, Figure 6-3.

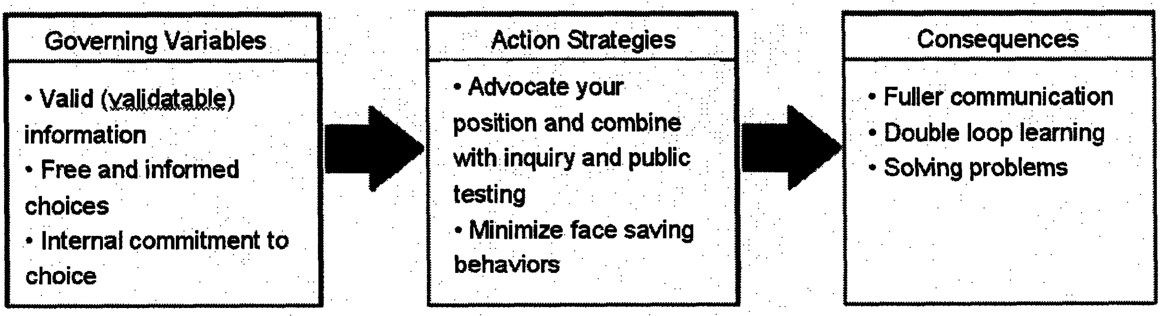


Figure 6-3 Double Loop Learning [Frey]

To gain the intended value from the systems engineering prognostics participants need to be trained to employ double loop learning. Prediction of the state of "goodness" of systems

---

[80] Frey, Dan. MIT Systems Engineering course, lecture 1, slides 24-25, Summer 2005

engineering based on the methods of chapter 3 should lead to questioning internal as well as others mental models on a recurring basis. Trends and other forms of predicted values should be reviewed both to understand what they mean to the project and what they imply about the fitness for use of the measurement constructs. Feedback must occur within the context of the measurement system to foster its continual improvement.

To assuage the concern raised by program management, measurement of a great many of the attributes of the development process can be automated and derivation of the data visualization to be done with little additional human effort. Metrics dashboards, such as presented in work by Selby[81], provide information content in a customizable format, allowing each stakeholder to view the information of concern to them. However, it is common for individuals in an organization to be pessimistic and cautious when faced with evaluating implementation of continuous improvement activities. This is especially true when those individuals are asked to fund activities. To succeed the measurement program must have a senior level champion who drives the implementation, its active use and continual improvement. Without the commitment of the organizational leadership the best laid intentions will fail.

Systems engineering prognostics, with the addition of clear connection to the organization level goals, could also support an enterprise goal of balancing systems engineering effort across multiple projects. These coupled with the addition of a set of systems engineering organizational measurements hold potential to provide data to support systems engineering continuous improvement activities as part of an enterprise level goal of instituting lean throughout the business processes. The value provided by a system of systems engineering prognostics will be dependent on the effort put into evolution, implementation and utilization of the measurements.

---

[81] Selby, Richard. "Measurement-Driven Dashboards Enable Leading Indicators for Requirements and Design of Large-Scale Systems", 11th IEEE International Software Metrics Symposium, 2005

6.2 *Suggested Future Research*

The application of these measures to historical project data facilitates learning the value proposition of the systems engineering prognostics. However, the value gained from validation in an active project is expected to be orders of magnitude greater. Future validation activity should apply the measurement constructs throughout the development phase. Gathering and sharing measurement data across a multiplicity of projects within same or similar organizations might also lend to investigating reoccurring interrelationships between the indicators. Data from application of the prognostics to an active project should lead to guides which provide example timelines to implement, effectiveness of each or subsets of the measurements and additional literature on the value of these prognostics to entice senior management to become champions.

Indicators, leading and lagging, of systems engineering are valuable, but as lead system thinkers systems engineers should also have tools to investigate both the causal relationships in the interface of the system under development and the development organization. Investigating the extension of the management system dynamics simulators into the realm of systems engineering with these indicators as influencers seems a sapient research effort. With a sufficient database of measurement data traversing several projects and organization researchers could work to substantiate such a systems dynamics model. Such a model would be an invaluable learning tool and allow simulation of PD organizational outcomes. A system dynamics model could be invoked to investigating the level of control obtainable by employing subsets of the suggested measurement system versus the complete set.

Effort is needed to identify leading indicators which provide greater insight into the systems engineering efficiency during early concept generation phase of the development process. Industry and academia need to continue to seek opportunities to leverage the design influence in the early phases of development, as discussed in chapter 2.

There is ongoing research in higher order leading indicators which are built on developments in operations research, systems dynamics and other related areas. Continued effort to apply these advanced methods to the systems engineering landscape in a manner which leads to consumption by the industry are paramount to continuing on the lean journey and removal of waste in product development practices.

Linking the disperse metrics programs in use throughout organizations is an area for continued research. Mahidar[82] suggests metric clusters as a construct to connect the many measurements within an enterprise, but do indicators such as those discussed in this thesis fit into this model?

During the LAI knowledge event on leading indicators held in early January of 2007 participants shared numerous insights into issues, hurdles, improvements and salient information content in the initial thirteen indicators defined in the leading indicators guide. Attending this event and listening to the many "smart people" involved in the LAI leading indicators project reinvigorated my interest in implementing predictive measurement in a lean systems engineering environment. It is through collaborative endeavors such as LAI, with diverse participants willing to invest toward improving the product development practice, that an evolutionary lean product development process will be realized in the near future.

---

[82] Mahidar, Vikram. Designing the Lean Enterprise Performance Measurement System, MIT Thesis, September 2005

7 References

ANSI/EIA 632, Standard - Process for Engineering a System, 2000

Ashby, W.R. Introduction to Cybernetics. Methuen, London, UK, 1956

Bahill, Terry and Gissing, Terry. "Re-Evaluating Systems Engineering Concepts Using Systems Thinking", IEEE Transactions on Systems, Man, And Cybernetics—Part C: Vol. 28, No. 4, Nov 1998

Blanchard, B.S., Fabrycky, W. J. Systems Engineering and Analysis 2nd Edition, Englewood Cliffs, NJ: Prentice-Hall, Inc. 1990

Clayton M Jones, CEO of Rockwell Collins, "Leading Rockwell Collins Lean Transformation", Presentation to LAI in fall of 2006.

Couffignal, Louis, "Essai d'une définition générale de la cybernétique", The First International Congress on Cybernetics, Namur, Belgium, June 26-29, 1956, Gauthier-Villars, Paris, 1958, pp. 46-54.

Crawley, Ed. Systems Architecture Course Lecture Notes, MIT, Fall 2004 17

Creveling, et.al., Design for Six Sigma, Prentice Hall PTR, 2003

Dasher, George. "The Interface Between Systems Engineering and Program Management" Engineering Management Journal, Vol 15 No. 3, Sept 2003

Davidz, Heidi. "Enabling System Thinking To Accelerate The Development of Senior Systems Engineers", MIT Thesis 2006

DoD Regulation 5000.2R, "Mandatory Procedures for Major Defense Acquisition Programs and Major Automated Information Systems Acquisition Programs," Chapter 5, Paragraph C5.1, April 2002.

DoD Systems Management College, Systems Engineering Fundamentals, Defense Acquisition University Press, 2001

EIA/IS 634, Systems Engineering, EIA 199428

ExportMore.gov, Detailed Information on the Future Combat Systems/Modularity Land Warfare Assessment, http://www.whitehouse.gov/OMB/expectmore/detail.10003202.2005.html

FCS program web site, http://www.army.mil/fcs/, as available 28 Dec 2006

Flint, Lynne Thompson and Gaylor, Dean A., "Expanding the Effectiveness of the Conceptual Design Phase: An Industrial Application of the Stanford Design for Manufacturability Method," Design for Manufacturability, Vol. 81, 1995, pg. 99-104

Ford, Russell B., and Barkan, Philip. "Beyond Parameter Design --A Methodology Addressing Product Robustness at the Concept Formation Stage", DE-Vol. 81, Design for Manufacturability, ASME, 1995

Frey, Dan. MIT Systems Engineering course, lecture 1, slides 24-25, Summer 2005

Gardiner, Geoff. “Concurrent and System Engineering: same thing, different name, or are they both just new product introduction”, Engineering Management Journal, Feb 1996

Hauser and Katz, “You are What You Measure!”, European Management Journal, Vol. 16 No. 5, pp 516-528, 1998

I. Minnich, “EIA/IS 731 Compared to CMMI” Systems Engineering (INCOSE) V5N1, pg 62-72

INCOSE Systems Engineering Handbook, Version 3, INCOSE, June 2006

INCOSE, Systems Engineering Measurement Primer, V1, 1998

Institute for Defense Analysis, Report R-338, 1986

Keating, Charles, et. al.. "System of Systems Engineering", Engineering Management Journal, Vol 15 No. 3, Sept 2003

Kosiak, Steven. "FY 2006 Defense Budget Request: DoD Budget Remains on Upward Trajectory", CSBA Update provided online (www.csbaonline.org), Feb 4, 2005

Kossiakoff, Alexander and Sweet, William. “Systems Engineering Principles and Practice”, Wiley, 2003

LAI Knowledge Event on Leading Indicators, January 10th 2007

Leveson, N.G., Intent specifications: an approach to building human-centered specifications. Software Engineering, IEEE Transactions on, 2000. 26(1): p. 15

Lucas, Chris. “Cybernetics and Stochastic Systems”, http://www.calresco.org/lucas/systems.htm, paper V1.0 June 1999

MacInnis, Daniel. “Development of a System Dynamics Based Management Flight Simulator for New Product Development”, MIT Thesis 2004

Mahidar, Vikram. Designing the Lean Enterprise Performance Measurement System, MIT Thesis, September 2005

McCarry, John, et. al. Practical Software Measurement, Objective Information for Decision Makers, Addison-Wesley, 2001

McGarry, John U.S. Army TACOM-ARDEC, “Systemic Analysis of Software Intensive System Acquisition Issues”, Software Technology Conference – 2003, April 29, 2003

McManus, PDTTL Roadmap, 2005

Murman, Earll. Lecture from 16.885J, MIT 2003

Oppenheim, “Lean Product Development Flow”, Systems Engineering, Vol. 7, No. 4, 2004

Putman, Lawrence and Myers, Ware. Five Core Metrics, The Intelligence Behind Successful Software Management, Dorset House Publishing, 2003

Rechtin, Eberhardt and Maier, Mark. The Art of System Architecting, CRC Press, 2000

Risk Radar™ Users Guide, version 2.03, Integrated Computer Engineering Inc., June 2003

Roche, James. Air Force Times, 24 Jun 2002

Roedler, G and Rhodes, D. Systems Engineering Leading Indicators Guide, LAI & INCOSE, Beta Release Dec 12, 2005

RTCA, DO-178B, Software Considerations in Airborne Systems and Equipment Certification. 1992

Senge, Peter. The Fifth Discipline: The Art and Practice of Learning Organization, Currency Doubleday, 1994

Sterman, John. Business Dynamics: System Thinking and Modeling for a Complex World, McGraw-Hill, 2000

U.S. Air Force policy memo 03A-005 "Incentivizing Contractors for Better Systems Engineering", 2003

Waldron, Roger D. Memorandum from, Deputy Chief Acquisition Officer Office Of The Chief Acquisition Officer "Implementation of Earned Value Management System (EVMS) Policy in GSA" Aug 19 2005

Wasson, Charles. System Analysis, Design, and Development, Wiley, 2006

Womack, James and Jones, Daniel. Lean Thinking, Free Press, 2003

Womack, James and Jones, Daniel. The Machine That Changed the World, HapperPerennial, 1990

Wynne, Michael. Memorandum from, Under Secretary of Defense, "Revision to the DoD Earned Value Management Policy", Mar 7 2005