A SYSTEMS ENGINEERING PRIMER
For Every Engineer and Scientist

First Edition
December 2001

Prepared by the LBNL Systems Engineering Staff
Bill Edwards, Editor

FOREWORD

The first complete formal text that I purchased on project management was entitled: *Project Management: A Systems Approach to Planning, Scheduling, and Controlling* (H. Kerzner). It is a considerable tome with an all-inclusive approach to project management. Though detailed, it closely follows a tailored approach to ensure a successful project. This should be our own approach to projects here at LBNL -- do only that which is necessary to ensure success and adopt those best practices that contribute to successful projects while keeping to the scientific and research mission and goals of the Laboratory.

A systematic approach to ensure that project requirements are well developed, monitored and maintained is an essential part of ensuring project success. Understanding the underlying assumptions that go into derived requirements or constraints is essential in optimizing the schedule and cost of a project. The solution of thorny technical issues often requires properly challenging those assumptions. Developing and controlling the scope and technical configuration of a project are crucial in avoiding *scope creep* or its more insidious twin *creeping elegance*. Communication between all parts of the project team, its sponsors, and stakeholders is essential to a successful project and many problems that arise on projects are the direct result of failures in communication.

The approaches and techniques outlined in this work contribute to successful projects by addressing these and other issues. The awareness and appropriate level of application of these approaches and disciplines are necessary. Not that every project *needs* a professional systems engineer on staff, but every project team member should have an awareness and familiarity with systems engineering. Just as an electrical engineer who designs a high power pulsed system without regard to the mechanical design and structure of the components is increasing the risk of failure, a project team that plans and executes a project without regard to systems engineering significantly increases the risk of failure.

Discussions as to what “belongs” to systems engineering versus what “belongs” to project management are of little value. If an approach, technique, or discipline, is necessary to ensure a successful project it must be incorporated into that project regardless of who “lays claim” to it. It is much more important to do what is right for a project and get on with the scientific business of the Laboratory. I believe that this work attempts to do that.

Kem Edward Robinson, Ph.D.
Berkeley, California
November 2001

ACKNOWLEDGMENTS

The following Systems Engineering Department staff contributed to the creation of this manual: Bill Edwards, Lisa Gullo, Ed Kujawski, Fritz Rene. They have benefited from discussions with Alan Biocca, Ken Chow, Peter Denes, Dick Digennaro, Rob Duarte, Dick Jared, Vic Karpenko, Daryl Oshatz, Kem Robinson, Ross Schlueter, Henrik Von Der Lippe, and Ron Yourd.

TABLE OF CONTENTS

1.0 INTRODUCTION 5
1.1 Scope and Intent 5
1.2 The Need for More Formal Systems Engineering 5
1.3 Defining Systems Engineering 6
1.3.1 General View 6
1.3.2 SE Practices and Principles 6
1.3.3 Relationship Between SE and Other Project Activities 7
1.3.4 Responsibility for SE Activities 7
1.4 Challenges of Implementing Systems Engineering 9

2.0 SYSTEMS ENGINEERING FUNCTIONS 10
2.1 Technical Coordination/Integration 10
2.2 System Architecting 10
2.3 System Analysis 11
2.4 Requirements Engineering 11
2.5 Systems Integration 11
2.6 Process/Performance Improvement (PPI) 11

3.0 TASKS & PRODUCTS CHECKLIST 12

4.0 "TOP TEN" FREQUENTLY ASKED QUESTIONS 17

5.0 ILLUSTRATIVE EXAMPLE - SE ON A SMALL PROJECT 20
5.1 Project Description 20
5.2 MuCoS Description 20
5.3 Conceptual Design Activities and Sample Outputs 20
5.3.1 Partial Workflow 21
5.3.2 Condensed System Specification 22
5.3.3 Risk Reduction Activities 26
5.3.4 General System Architecture and Some Design concepts 27
5.3.5 Selection Criteria 28
5.3.6 Error Budget 29
5.4. Concluding Remarks 29

6.0 REFERENCES & RECOMMENDED READING 31

1.0 INTRODUCTION

1.1 Scope and Intent

The Systems Engineering (SE) staff at LBNL has generated the following artifacts to assist projects with implementing a systems approach:

1. The present document that focuses on the "what", "why", and "when" of SE. It also provides a simple case-study to illustrate several SE tasks.
2. A web site with primary emphasis on the project life-cycle and workflow, (http://www-eng.LBNL.gov/Systems/index.html). It includes:

- SE guidelines and principles
- A list of in-house tools
- Templates
- Case studies with “how to” examples
- Links to useful SE material.

These sources are living documents to be updated as necessary.

The viewpoint adopted in this document is that what LBNL engineers and scientists need is a set of principles and guiding practices for developing R&D systems rather than a "cookbook". There are many excellent "how to" resources such as the "INCOSE Systems Engineering Handbook" to guide those in search of more details. The SE staff is another resource available to consult and support projects.

This document specifies SE principles and activities that are applicable to all LBNL projects independent of their specific differences. Each project should tailor the SE implementation to meet its individual needs and culture including project-specific resources, procedures, products, and tools.

1.2 The Need for More Formal Systems Engineering

Uncertainty and risk are intrinsic characteristics of R&D projects. A major challenge is to effectively manage performance, cost, schedule, technology, and risks. Most LBNL projects already implement some aspects of SE. For example, to quote from the STAR project:*" The team of integration and system level engineers and physicists was crucial to building the detector on time and on budget. The planning worked well and as a result the final mechanical and electrical environment for STAR was built as intended."*

Most projects can benefit from a more systematic approach to system design and integration.

1.3 Defining Systems Engineering

1.3.1 General View

Over the past fifty years, SE has evolved as a discipline with principles, methods, and techniques to deal with a broad spectrum of projects ranging from complex R&D to small commercial projects. SE is so wide and multi-faceted that as of yet there is no applicable single unified approach. Instead professional organizations (INCOSE, PMI, EIA, IEEE…), government agencies and contractors, commercial industry, and academic research have developed different models. But being models, they are at best approximate representations of the SE effort. Their usefulness depends on how well they help the practitioners understand and solve their problems.

SE has had its successes as well as its failures. Some of the lessons-learned are:

1. A formal SE process is necessary, but not sufficient for good SE implementation.
2. Successful SE requires:
   - An appreciation of systems thinking as a “good thing”
   - A sound project implementation and practices
   - A proven risk management process
   - A knowledgeable and receptive staff.
3. Each project must tailor the SE activities to match its specific needs. Tools and techniques that work in one situation will not necessarily work in another.

1.3.2 SE Practices and Principles

The LBNL SE staff has tailored an approach that addresses both the art and the mechanics of SE. It recognizes that successful projects require that the following three areas achieve an adequate level of maturity:

- **Environment** including organizational culture and leadership;
- **Process** including technology base; and
- **Enablers** including technical skills, thinking skills, tools, and organizational learning**.**

The LBNL SE staff approach embodies the following eleven key principles:

P1. Tailor the SE activities to the scope and complexity of the project.

P2. Ensure that the system design meets the needs of the customer and addresses the complete life-cycle for the system.

P3. Act as the glue for the different disciplines to ensure that (1) the hardware and software components meet their allocated requirements, and (2) there are no incompatibilities between subsystems.

P4. Maintain a "win-win" environment through (1) openness, trust, and communications, and (2) early identification of problems (and don't shoot the messenger).

P5. Establish and manage requirements. But plan for requirement changes as insight into the need and the "best" solution evolves.

P6. Take the time to innovate by generating a wide range of alternatives before converging on a solution.

P7. Understand the project risk/benefit trade-off strategy among performance, cost, and schedule.
P8. It's everyone's responsibility to manage risks and look for opportunities.
P9. Quality must be designed in; it cannot be tested in.
P10. Minimize the number of reports required; but important work must be recorded thoroughly.
P11. Institute continuous improvement.

These principles are not original. They are extracted from the writings of many experts who have shared their experiences of successful projects. Principles alone however are not sufficient. Effective system design also requires technical skills, systems thinking, and good judgment. But SE activities based on these principles will have a greater beneficial impact on projects than simply following a process. The LBNL SE website provides additional details and information.

1.3.3 Relationship Between SE and Other Project Activities

The SE activities are an integral part of the project life-cycle depicted in Figure 1. They complement the project management and design activities that are already in place by placing greater emphasis on iterative development, trade studies, uncertainties, and risk management to optimize project success including technical performance within cost and schedule constraints.

1.3.4 Responsibility for SE Activities

The nature of the SE organization and responsibilities for a given program should be a function of the project type and size. For a small project with few risks, the project manager and design team may handle all SE activities in a relatively informal manner. For a modest size program, the assignment of a part-time person with experience to coordinate and foster the SE activities is appropriate. For a very large program, a full-time person or a small team may be required to handle these activities. In all cases, the project team has responsibility for SE.

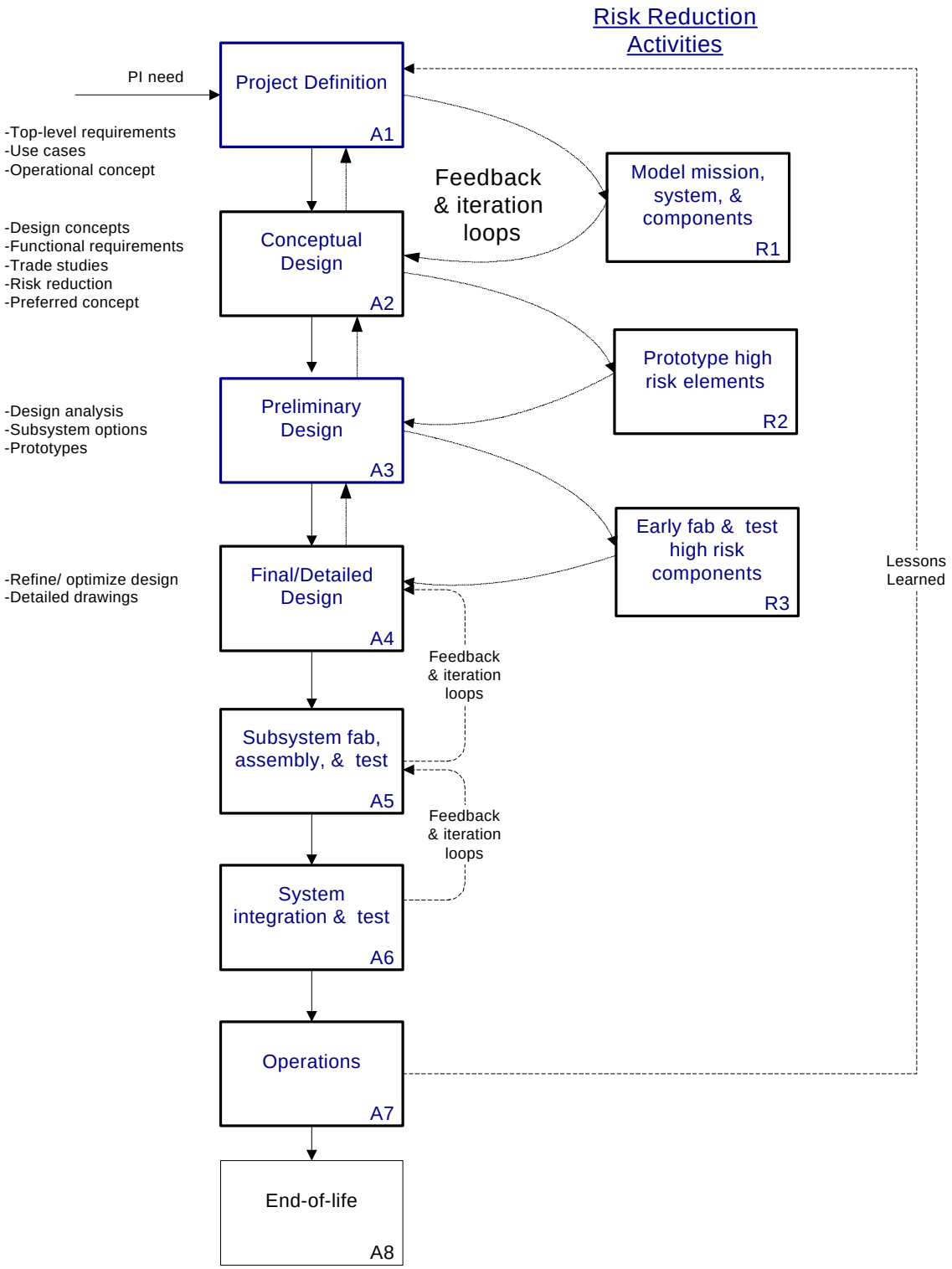


Figure 1. SE is integrated in the design process and project life-cycle

1.4 Challenges of Implementing Systems Engineering

The nature of SE and the LBNL culture pose challenges over and above those seen in other process improvements. As we proceed, it is important to be cognizant of the potential barriers to SE improvement at LBNL. These include:

- Thinking we're different.
- A "Two Cultures" problem of engineers and scientists.
- Successful project managers and principal investigators who base their decisions on intuitive approaches.
- Fear that SE would stifle creativity.
- A lack of hard numbers on the benefits of good SE on R&D projects.
- Concerns about the ripple effects that SE may have on projects and organizations.

The LBNL SE staff has designed an approach that addresses and overcomes these barriers. Improving SE is not offered as a quick-fix remedy to improve the performance of R&D projects at LBNL. But like any process improvement or change, it is a challenging project that requires practice and resources. A partial set of SE activities is already being done. A more systematic approach to SE and performing the appropriate additional activities should increase the efficiency of LBNL projects and increase the likelihood that they will meet technical performance within cost and schedule.

2.0 SYSTEMS ENGINEERING FUNCTIONS

For convenience we have classified the SE activities into the following six functions and/or roles:

1. Technical coordination/integration
2. System architecting
3. System analysis
4. Requirements engineering
5. Systems integration
6. Process/performance improvement.

The above categorization is not to be construed as representing a division of responsibilities. The functions and roles are strongly coupled and integrated into a coherent SE effort. Many of these functions and roles are performed jointly and in common. The emphasis is on ensuring a systems approach and not "who should do what". Weakness in any one area is likely to adversely impact the project.

2.1 Technical Coordination/Integration

The technical coordination/integration function is to ensure that the project accomplishes the tasks necessary to demonstrate technical readiness at project milestones. It involves:

- Planning and coordinating key design reviews.
- Coordination and communication throughout all technical levels.
- Providing leadership and ensuring that the interfaces between groups are running smoothly.
- Configuration management, change control, data and document management.
- Disseminating information as needed to ensure the success of the project.

2.2 System Architecting

The system architecting function is to develop system design strategies and priorities. It defines the form of the system (selection of the concept, types of system elements, their characteristics and arrangement) which meets the following criteria:

1. Satisfies the scientific and operational needs.
2. Is acceptably close to the true optimum within the constraints of time, budget, available knowledge and skills, and other resources.
3. Is consistent with the technical maturity and acceptable risks of the available components.
4. Accommodates system growth and introduction of new technologies.
5. Provides the base of information that will allow subsequent design and implementation to proceed.
6. Is robust, i.e., allows subsequent, more detailed system definition to proceed with minimum backtracking as additional information is uncovered.

2.3 System Analysis

The system analysis function is to analyze and model the system and mission to determine if they meet the stated science requirements and operational needs in an optimal or near optimal manner subject to performance, cost, schedule, and programmatic constraints. Typical activities are (1) evaluation of mission, system, and subsystem performance, (2) cost modeling, (3) trade studies, and (4) technical risk analysis. A subset of system analysis involves "specialty engineering" tasks such as reliability/maintainability/ availability analyses and trade-offs.

2.4 Requirements Engineering

The requirements engineering function is to develop a complete and accurate set of requirements that forms the basis for the design, manufacture, test, and operations of the system developed by the project. It makes sure that the scientific and operational needs are met. It involves defining, deriving, clarifying, modifying, and documenting the requirements. Requirements flow down from the science requirements to the system and subsystem level.

2.5 Systems Integration

The systems integration function is to ensure that (1) the hardware and software subsystems are integrated into the system and that the system is fully integrated into the mission, and (2) the implemented hardware and software conform to its requirements. System integration includes (1) interface management, and (2) verification and validation activities. The following big picture questions are answered: (1) Did we build the system right? (2) Did we build the right system?

2.6 Process/Performance Improvement (PPI)

The PPI function is to continually improve the individual, team, and organizational performance to ensure that LBNL delivers products that achieve the scientific goals and high quality within the cost and schedule constraints. This requires that the technical, management, and programmatic aspects develop successfully as an ensemble. SE is important to these aspects, but by itself it is not sufficient to ensure a successful project. PPI applies to all the aspects of a project. Lessons-learned are collected and disseminated to avoid repeating past mistakes and provide a common knowledge base for future projects.

3.0 TASKS & PRODUCTS CHECKLIST

The following table shows the planning, control, and technical tasks and the associated results that are typically necessary to develop a successful project. These artifacts are associated with the SE functions identified in Section 2.0. Many of these involve more than one function and reflect the interdisciplinary nature of SE. The following table is intended as a guide for use by every LBNL project. The specific tasks and their sequence will vary from project to project. The effort should be commensurate with the program and tailored accordingly.

Tasks	Benefits	Products by phases[1]
1.0 Technical planning ➢ Define SE implementation strategy. ➢ Plan, budget, schedule and organize SE effort. ➢ Coordinate preparation of technical plans for subsystems.	- An efficient and effective implementation of SE at the system and subsystem levels tailored to the project.	- CD: Drafts of SEMP[2], WBS[3], specification tree, and technical plans. - PD: drafts refined and published. - Other phases: plans and other documents revised as necessary.
2.0 Technical assessment ➢ Assess progress of the technical effort against applicable plans and schedules. ➢ Track technical performance progress. ➢ Conduct technical reviews.	- Status information to enable efficient use of technical resources. - Early identification and resolution of technical problems.	- Level of Effort (LoE) throughout all the phases of the project. - Technical review data package: specifications, drawings, trade studies, risk analysis, test methods & data, safety reports, specialty studies. - TPM[4] reports at key milestones.

[1] The project life-cycle phases are abbreviated as follows: PCD = Pre-Conceptual Design, CD = Conceptual Design, PD = Preliminary Design, FDD = Final/Detail Design, I&T = Integration & Test.
[2] Systems Engineering Management Plan
[3] Work Breakdown Structure
[4] Technical Performance Measures

3.0 Technical coordination ➢ Capture technical decisions and assumptions. ➢ Perform configuration and change control management. ➢ Manage the technical information database and technical documents.	- The outcomes of the technical effort are properly recorded and managed in accordance with the agreement and technical project plans. - Technical information is properly disseminated.	- LoE throughout all phases of the project. - PD: Change control board established. - FDD: Baselined and controlled versions of all requirements.
4.0 Mission analysis ➢ Identify, collect, and prioritize stakeholders' needs. ➢ Define concept of operations.	- The range of anticipated uses and operational profile are identified. - A validated set of requirements is established.	- PCD and CD: Draft Concept of Operations (ConOp) document. - PD: Released ConOp document.
5.0 Define system technical requirements ➢ Challenge questionable requirements. ➢ Ensure completeness and consistency of the system technical requirements. ➢ Prepare system requirements specification.	- A set of system technical requirements that are "done enough" to proceed with design. - Documented rationale and assumptions.	- PCD and CD: Released top-level science/mission requirements; Draft of system requirements. - PD: Released system specification. - FDD: Updates
6.0 Develop logical solution representation ➢ Analyze system behavior. ➢ Define states and modes of operation. ➢ Develop data /control flow. ➢ Assign requirements to appropriate functions, objects, data structures, etc.	- Provides foundation for defining the system architecture through the allocation of functions to hardware, software, and operations.	- PCD and CD: System models at functional level including data/control flow diagrams, timelines, state transition diagrams,… - PD: Subsystem models at functional level.
7.0 Develop system architecture solution alternatives ➢ Partition the system into hardware, software, and procedural components.	- A system architecture baseline and supporting documentation to demonstrate, within reasonable certainty, that: • It is adequately close to the theoretical	- PCD and CD: System architecture document including rationale for architecturally significant decisions. - PD and FD: Revised/updated system

➢ Assign functions to appropriate entities that will make up the solution. ➢ Evaluate alternative architecture solutions. ➢ Select preferred system architecture.	optimum. • It is robust. • The data (features and parameters) are adequate to support subsequent work.	architecture document.
8.0 Define external and internal interfaces ➢ Evaluate user needs to identify external interfaces. ➢ Define internal and external interfaces for all modes of operations.	- All physical and functional requirements for both hardware and software interfaces are clearly defined. - All interfaces documented in only one place.	- CD: Develop external interfaces. - PD: Initial Interface Control Document (ICD) including internal and external interfaces. - FDD: Final ICD.
9.0 Trade studies ➢ Plan trade studies. ➢ Define selection criteria and their method of application. ➢ Determine risk/opportunity for each alternative. ➢ Perform and record trade-off analyses.	- A sound basis for determining that the methodology and data collection were sufficient to support a "good" evaluation. - A set of criteria sufficient to distinguish the preferred solution from the contenders.	- PCD and CD: Documented significant trade studies for sensitive top-level requirements. - PD and FDD: Documented trade studies for subsystem designs and component selection.
10.0 Cost modeling ➢ Analyze the life-cycle cost of each alternative. ➢ Support cost-benefit and cost-effectiveness analyses.	- Realistic costs and cost-risks for development, fabrication, testing, and operations. - Cost is integral element for evaluating alternatives and proposed changes.	- PCD and CD: Preliminary system cost and life-cycle cost, cost-risks. - PD and FDD: Refined costs and cost-risks.
11.0 Technical Risk Analysis ➢ Identify and characterize technical risks. ➢ Define approaches for mitigating significant risks. ➢ Capture and communicate risk analysis	- Effective risk handling approaches are defined; significant risks are averted; surprises are minimized.	- PCD and CD: List of quantified technical risks and evaluation of potential risk response actions. - PD: Status of technical risks; risk management plans.

outcomes.		- Later phases: Status of technical risks; risk actions and plans refined.
12.0 Establish and control baseline ➢ Define the hardware, software, and procedural components for the selected design. ➢ Evaluate impact of proposed changes.	- Definition of an integrated system architecture including technology, schematics, data description, interfaces… - A specification tree including all configuration items.	- LoE initiated at CD and continued throughout all phases.
13.0 Perform SE at subsystem and lower levels ➢ Allocate system requirements to subsystems and components. ➢ Ensure completeness and consistency of the requirements flowdown. ➢ Prepare subsystem and component requirements specifications.	- A set of subsystem requirements that are "done enough" to proceed with component design, fabrication, and verification. - Risk reduction through the early development of high-risk components.	- CD: Development of high-risk components may start. - FDD: Released hardware and software subsystem and/or component requirements.
14.0 Establish a centralized database ➢ Record the technical requirements, interface definitions, and associated data. ➢ Make it available to the team.	- A validated set of requirements, interface definitions, and associated data is captured, maintained, and controlled throughout the life of the project.	- LoE initiated at CD and continued throughout all phases.
15.0 Safety and Quality/dependability ➢ Identify and assess safety hazards. ➢ Analyze quality/dependability. ➢ Perform Failure Modes and Effects Analysis (FMEA). ➢ Implement logistics support.	- System/product meets the specified safety goals and criteria. - System/product provides adequate quality/dependability including reliability, maintainability, human factors, etc.	- CD: Preliminary hazard analysis report. - PD: Preliminary safety analysis report; Functional/system-level FMEA. - FDD: Reliability analysis; Updated documents; Detailed FMEA; Draft technical manuals. - I&T: Final safety report;

		Qualification test reports; Updated documents; Training manuals.
16.0 Verification and Validation (V&V) ➢ Select appropriate V& V method for each requirement. ➢ Plan V&V effort. ➢ Evaluate V&V data. ➢ Identify and support resolution of variances.	- Demonstrated compliance of the design and end product with the scientific requirements and user needs.	- PD: Preliminary verification requirements matrix; Draft master test plan. - FDD: Updated verification requirements matrix; Released master test plan; System and subsystem test procedures. - Later phases: Verification report; Operational readiness & acceptance report; Operations performance report.
17.0 Process/Productivity improvement ➢ Conduct pre-project review of last projects data/results. ➢ Brief project team. ➢ Look for and capture opportunities to improve process/product. ➢ Conduct systematic post-project review. ➢ Analyze information and develop recommendations for PPI.	- Provide a lab-wide experiential database for individual and organizational learning. - Make LBNL engineering jobs more satisfying. - Make LBNL projects even more successful.	- Level of Effort (LoE) throughout all the phases of the project. - Post project: Documented lessons-learned.

4.0 "TOP TEN" FREQUENTLY ASKED QUESTIONS

1. How does "SE coordination/integration" differ from "project management"?

"SE coordination/integration" is more of an analytical, advisory, and planning function while "project management" is more of a decision-making function. Very often the distinction is irrelevant as the same individuals perform both roles. As indicated by Kem Robinson in the foreword, "Discussions as to what belongs to SE versus what belongs to project management are of little value…It is much more important to do what is right for a project and get on with the scientific business of the Laboratory."

2. How does SE apply to LBNL projects?

SE, as presented in this document, includes all the team members and is designed to help discover the system requirements and converge on an optimal or near optimal solution. It helps develop successful R&D projects that meet technical performance within cost and schedule. Achieving these objectives requires making the right trade-offs between simultaneous and often conflicting requirements such as product demands from scientists, engineering, budget, and schedule. Each project tailors the SE activities to best meet its needs.

3. What deliverables are typically required and when?

The activities integral to the development of the system/product should be documented. The emphasis should be on quality rather than quantity. The applicable deliverables often depend on the scope of the project and the SOW. Typical deliverables are conveniently listed in Section 3.0. Figure 2-3 of the DOE Program and Project Management Manual (Draft October 2000) depicts the typical stages of a DOE project and the technical documentation DOE may require to support moving to the next phase.

4. Is it necessary to formally document all these plans?

Formally documenting plans is of value; but it is not the primary intent. The important action is to adequately plan the technical effort and to make the relevant/necessary data available to those who need it in order to develop a successful end-product.

5. How does "System Architecting" differ from "Design Engineering"?

As defined in this document, "System architecting" deals with the relationships of the system or product being designed to its purpose, user needs, and existing components. "Design engineering" deals with the details of the subsystems and components. The system architect viewpoint is broad, rather than deep. It encompasses (1) all the system life cycle from conception to disposal, and (2) all of its functions from normal operation, to degraded operation, to failure.

6. What is the purpose of a system abstraction or logical models?

An abstraction or logical model is a simplified description of a system that emphasizes the system's functions and properties while suppressing design details of hardware and software components. It is a proven technique to support the one's creativity and thought process. The ease of use and usefulness of a system abstraction or logical model depends on the system/product and the inclination of the user. Numerous models, representations, techniques, and tools have been developed. The best way to appreciate their usefulness is to try using them on a real application.

7. How detailed should the analyses be?

Without addressing a specific problem, we can only give generic guidance, which unfortunately may be of rather limited value.

- The level of detail of the analysis should be commensurate with the specific project needs and requirements. The analyses should also be cost-effective and timely.
- Apply a healthy dose of common sense because models can only approximate the real world, not replace it.
- To quote Einstein: "A model should be as simple as possible and yet no simpler."

8. What happens when you don't really know what the requirement should be?

The early requirements need not be perfect. It is more important to have a starting point that can be proven wrong or not necessary than to overlook potentially very important aspects of the system. An important purpose of writing down and reviewing requirements is to give other interested parties a chance to see them and solicit ideas and criticism that can be used to improve them. Testing and modeling efforts can then be identified which can help resolve problems and reveal unexpected conditions.

9. When should requirements be put under configuration control?

Configuration control is a stepwise process. Requirements evolve commencing with those generated in the pre-conceptual phase. Only those requirements that are agreed to by the stakeholders are put under configuration control. Putting requirements under configuration control does not mean that the requirements "are done", but rather that the requirements "are done enough" to proceed with them. For most projects, configuration control starts during the conceptual phase when the science requirements are agreed to and continue throughout the project development.

10. What are some of the barriers to SE process improvement and how can we overcome them?

"Change is good. You go first." - a T-shirt

Barrier	Solution
Thinking "we're different".	Don't tell people how to work. Define functions to help them do their job and get support when needed.
No generally acknowledged	Do not insist that only people with the title of

definition of SE.	"systems engineer" should do SE. But, assess how well the project is performing the SE activities.
Assuming training is the answer.	Training is necessary but not sufficient. Engineers and scientists must get involved and Management must be committed to its success.

5.0 ILLUSTRATIVE EXAMPLE - SE ON A SMALL PROJECT

5.1 Project Description

MuCoS is a small project, under $100K. It is being designed and built by LBNL as a DesignWorks project. The SE staff is involved in the project to:

- Perform SE functions in direct support of the project.
- Use MuCoS as a pilot small project to evaluate the LBNL SE staff approach.
- Provide a case study of SE contribution on a small project.

5.2 MuCoS Description

The Multi-Cell Core Position Sensor (MuCoS) is an instrument designed to measure the position of the cores that make up the magnetic induction accelerator cells of the DARHT facility. Each cell has four cores of tape-wound Metglas enclosed in aluminum housing around an open central bore. Only six (6) acrylic shoes driven by setscrews in the aluminum housing support each core. Under gravity each core can then move relative to the beam tube. The resulting core movements can (1) induce unacceptable large transverse magnetic fields, and (2) damage the beam tube. MuCos measurements are taken to provide information on the long-term core movements. To minimize adverse impact on beam availability, the MuCoS enables taking the measurements without the need to disassemble the individual cells.

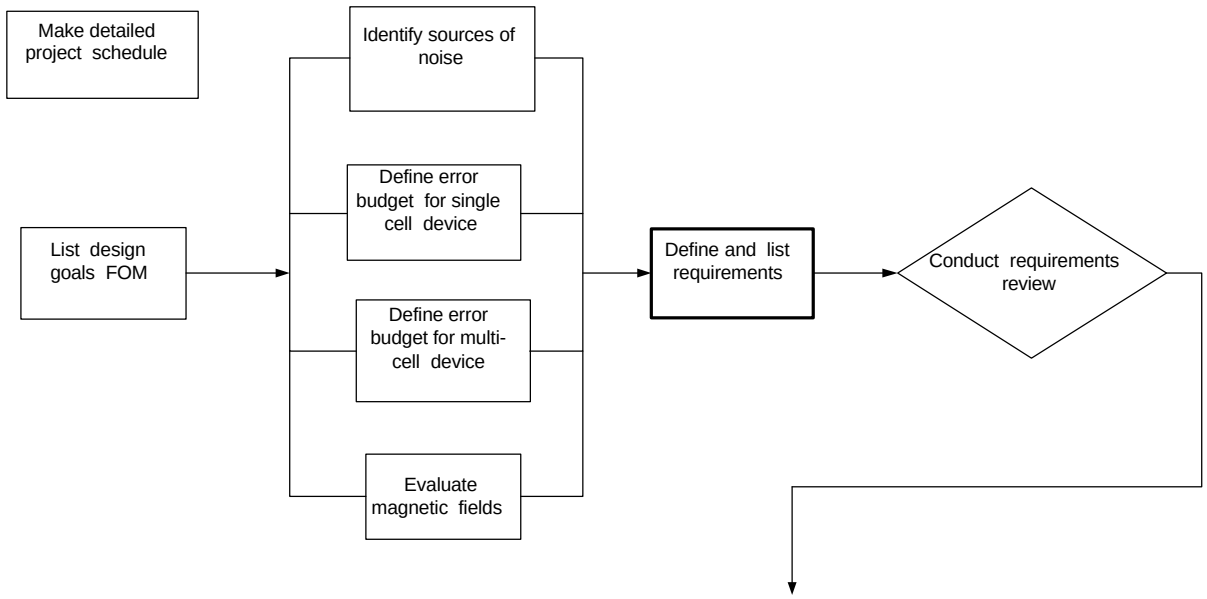
5.3 Conceptual Design Activities and Sample Outputs

The workflow was developed in a single meeting in less than one hour. Good synergy and brainstorming rules were important factors to the success of this meeting.

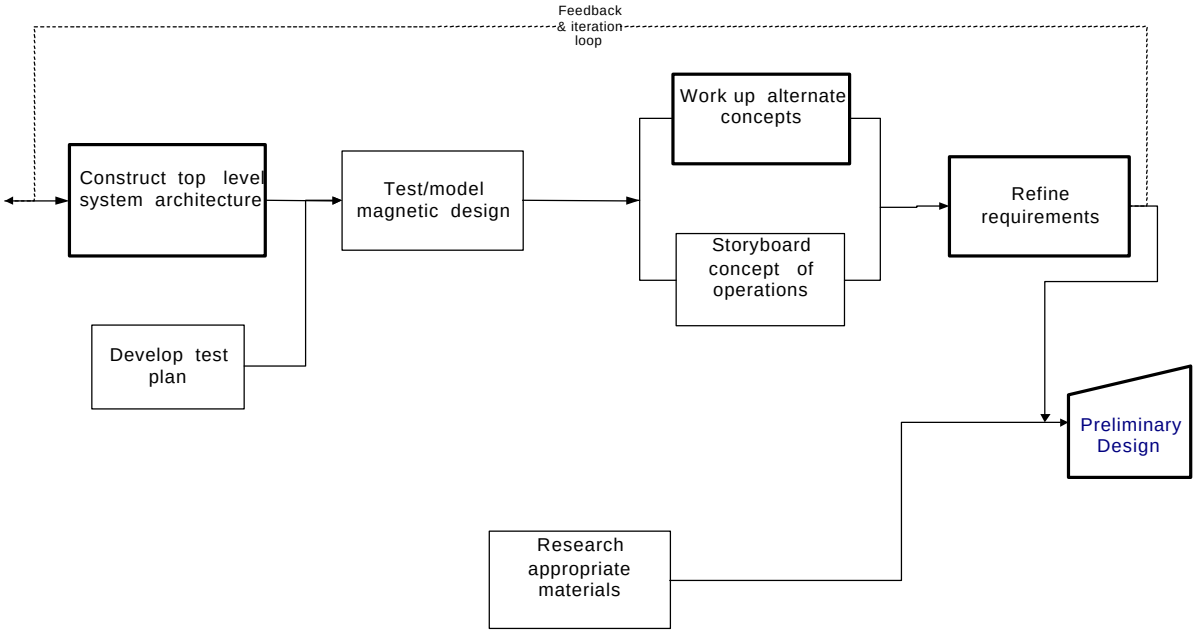
The reported results cover the initial phases including project definition, conceptual design, and preliminary design. The team members working on a part-time level developed them over a period of two months. The total man-week effort was approximately 3 weeks. The attachments represent the part of this effort that we think is applicable to all projects. Detailed analyses are not included. We emphasize that we do not specifically label any activity or output as SE. Instead, the MuCoS project used a SE process and the four team-members practiced SE.

5.3.1 Partial Workflow

Project Definition Activities



Conceptual Design Activities



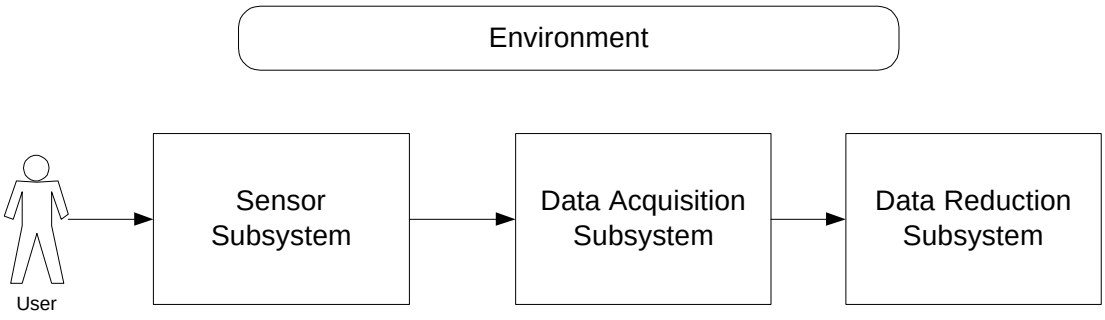
Note: Software is treated as an integral part of the system architecture, requirements, and error budget

5.3.2 Condensed System Specification

1.0 SCOPE
This specification establishes the performance requirements for the Multi-Cell Core Position Sensor System (MuCoS) for the DARHT facility. It also includes the rationale for the requirements. The rationale is not contractually binding; only the requirements are.

2.0 SYSTEM DESCRIPTION
The functional block diagram for MuCoS is shown below. The functions are allocated to three subsystems:

- The sensor subsystem and its support hardware
- The data acquisition subsystem
- The data reduction subsystem.



MuCoS Functional Block Diagram

3.0 SYSTEM REQUIREMENTS

3.1 PERFORMANCE CHARACTERISTICS
3.1.1 Position Accuracy
The positions of cores 2,3, and 4 for each of the 6 cells in a cell block shall be determined through a combination of direct measurements and analysis to the accuracy specified below.

3.1.1.1 Radial position accuracy
The radial displacement over a range of 0.25 inch shall be determined to accuracy of 0.1 mils (3 sigma) TBR[1].
Verification - Analysis or test

3.1.1.2 Horizontal position accuracy

<u>Rationale for paragraph 3.1.1 requirements</u>: Movements of up to 0.25 inch have been measured. Large core movements are unacceptable for cores 2, 3, and 4. Core 1 (the

[1] TBR: To Be Reviewed

core at the cell upstream end) is not of concern because it neither affects the transverse magnetic field nor risks damage to the insulator. The acceptable accuracy is based on analysis of the required magnetic field.

3.1.2 Maximum Measurement Time

3.2 ENVIRONMENTAL CONDITIONS
3.2.1 Natural Environment
3.2.1.1 Operating Temperature

3.2.1.2 Storage Temperature
Minimum: - 15^0C (+5^0F) (TBR)
Maximum: +60^0C (+140^0F) (TBR)
Verification - Test

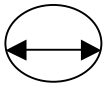
3.2.1.3 Operating & Storage Humidity

3.2.2 Induced Environment
3.2.2.1 Mechanical Vibration

3.2.2.2 Electrical Noise

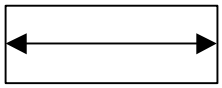
<u>Rationale for paragraph 3.2 requirements</u>: It is important to understand the natural and induced environmental conditions to ensure that the product meets the end-user needs and to avoid over-designing. The above conditions are relatively benign and should not limit operations.

3.4 PHYSICAL CHARACTERISTICS
The MuCoS shall fit within the following envelope:



14.0 cm (TBR)

<u>Transverse dimension</u>



65.0 cm (TBR)

<u>Longitudinal dimension</u>

Verification - Inspection
<u>Rationale</u>: Enable measurements of vertical and horizontal positions of cores 2,3, and 4 with only 1 or 2 inter-cells removed.

3.5 ELECTRICAL REQUIREMENTS
3.5.1 Supply Voltage
Nominal: 12.0 Vdc (TBR)

Minimum: 10.5 Vdc (TBR)
Maximum: 16.0 Vdc (TBR).
Verification - Test
Rationale: Permits the use of commercial sensors.
3.5.2 Over-Current Protection
Rationale: Good design practice.

3.6 MAINTENANCE

3.7 PACKAGING, HANDLING, AND TRANSPORTATION

3.8 STORAGE

3.9 PERSONNEL TRAINING

3.10 SUPPORT EQUIPMENT

3.11 TECHNICAL DATA

3.12 REQUIREMENTS ALLOCATION
3.12.1 Sensor Subsystem

3.12.2 Data Acquisition Subsystem

3.12.3 Data Reduction Subsystem

Rationale for paragraph 3.12 requirements: It is a good practice to state the performance and physical characteristics for each major component identified in paragraph 2.0.

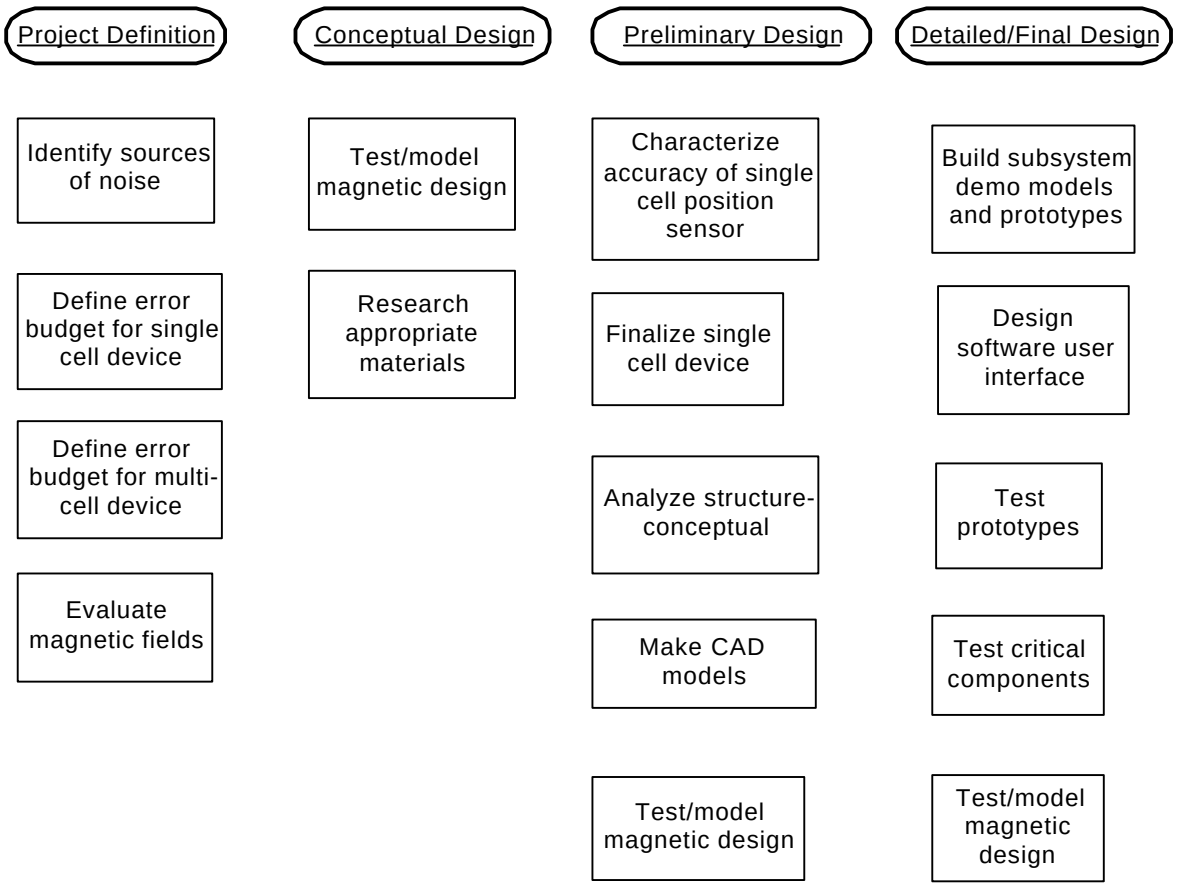
4.0 VERIFICATION
The requirements of Section 3, exclusive of Section 3.12, shall be verified by the methods for each requirement as shown in Table 1. The methods include test (T), demonstration (D), analysis (A), and inspection (I). The requirements of Section 3.12 shall be verified as specified in the respective specification for each of the subsystems.

Rationale: A test plan and specific test procedures will provide all tests necessary to insure accomplishment of the MuCoS verification requirements.

Table 1. Verification Matrix

Section 3 Paragraph	Title	Method			
		T	A	D	I
3.1.1.1	Radial position accuracy	X	X		
3.1.1.2	Horizontal position accuracy	X	X		
3.1.2	Maximum Measurement Time			X	
3.2.1.1	Operating Temperature	X			
3.2.1.2	Storage Temperature	X			
3.2.1.3	Operating Humidity	X			
3.2.1.4	Storage Humidity	X			
3.2.2.1	Mechanical Vibration	X			
3.2.2.2	Electrical Noise	X			
3.3.1	Installation			X	
3.3.2	Device Interconnection			X	
3.3.3	Cable Damage			X	
3.3.4	Bore Tube Protection	X			
3.4	Physical Characteristics				X
3.5.1	Supply Voltage	X			
3.5.2	Over-Current Protection	X			
3.6	Maintenance		X		
3.7	Packaging, Handling, and transportation		X		
3.8	Storage				X
3.9	Personnel Training			X	
3.10	Support Equipment			X	
3.11	Technical Data				X

5.3.3 Risk Reduction Activities



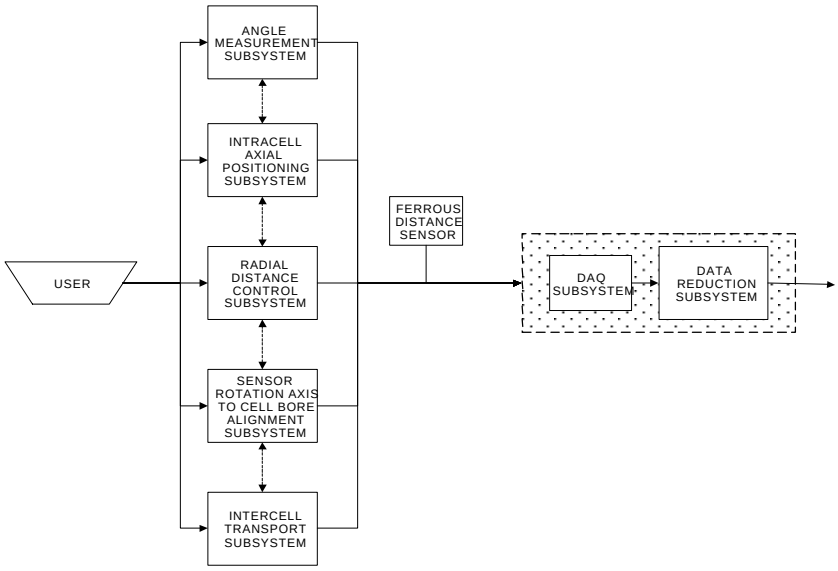
Notes: The above activities are specific responses to the identified risks. MuCoS is a first of its kind device. We use an evolutionary rather than a "big bang" approach. The plan proceeds in the following stages:

- Fully characterize the single-cell device
- Develop MuCoS concepts
- Develop, test and analyze prototype
- Design and build final MuCoS.

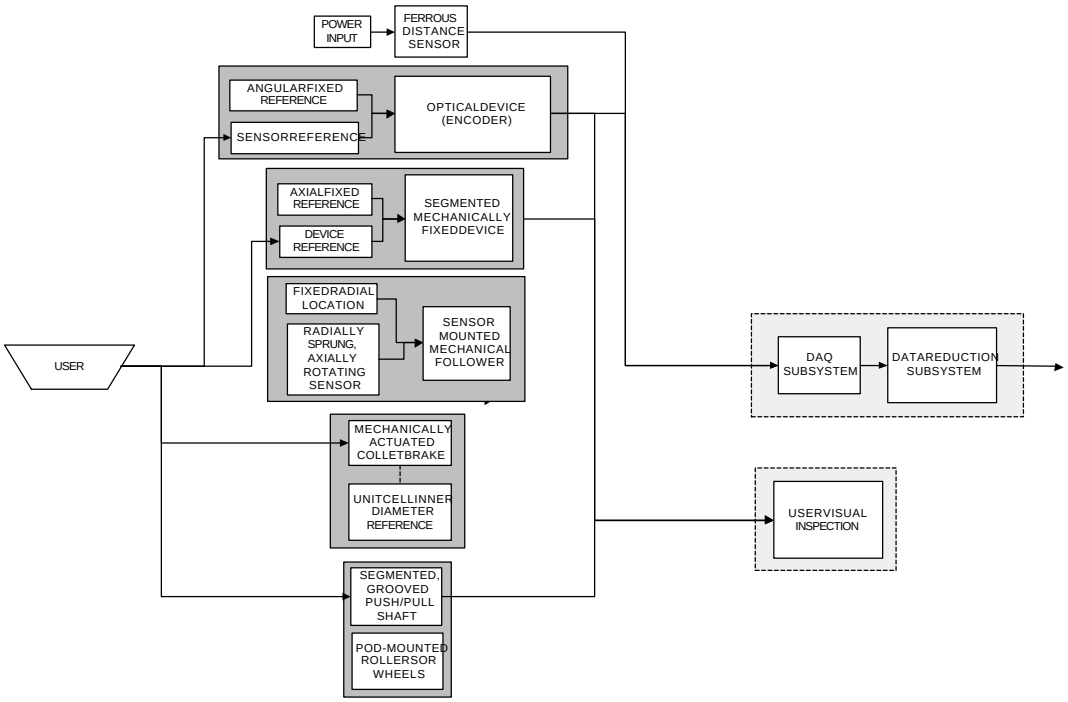
Such an approach reduces risk through mitigation, prevention, or anticipation. It is encapsulated in the above activities.

5.3.4 General System Architecture and Some Design concepts

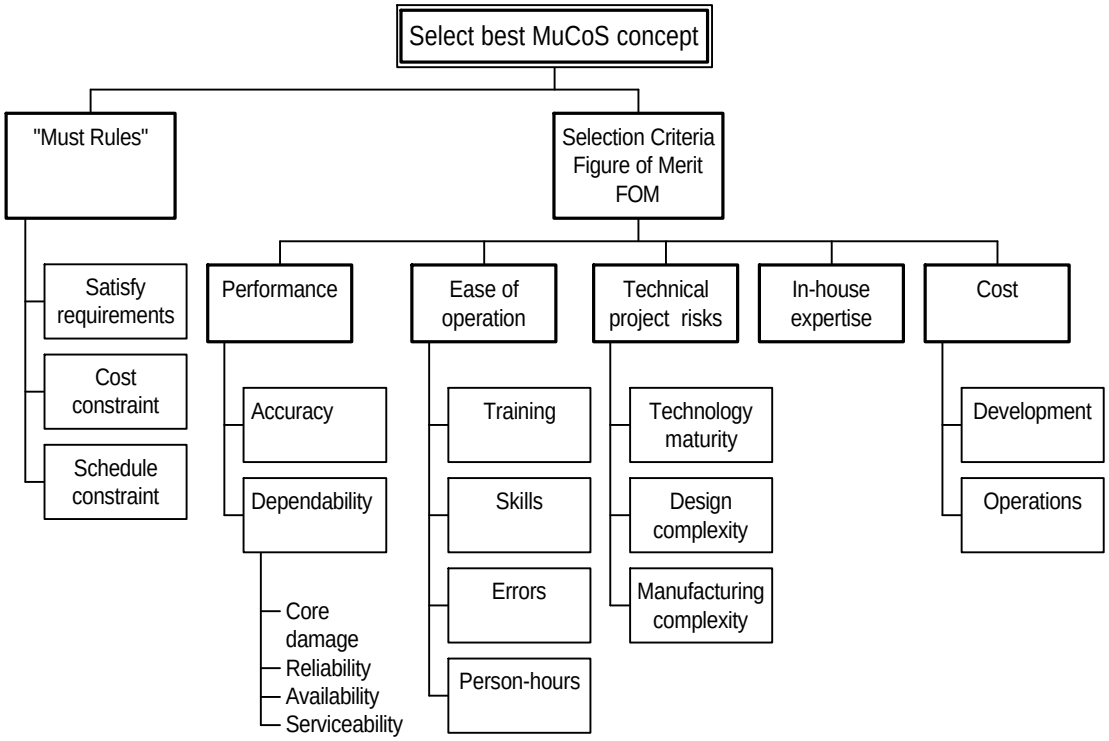
MULTICELL CORE SENSOR -- GENERAL SYSTEMS ARCHITECTURE



MULTICELL CORE SENSOR — IPOD CONCEPT

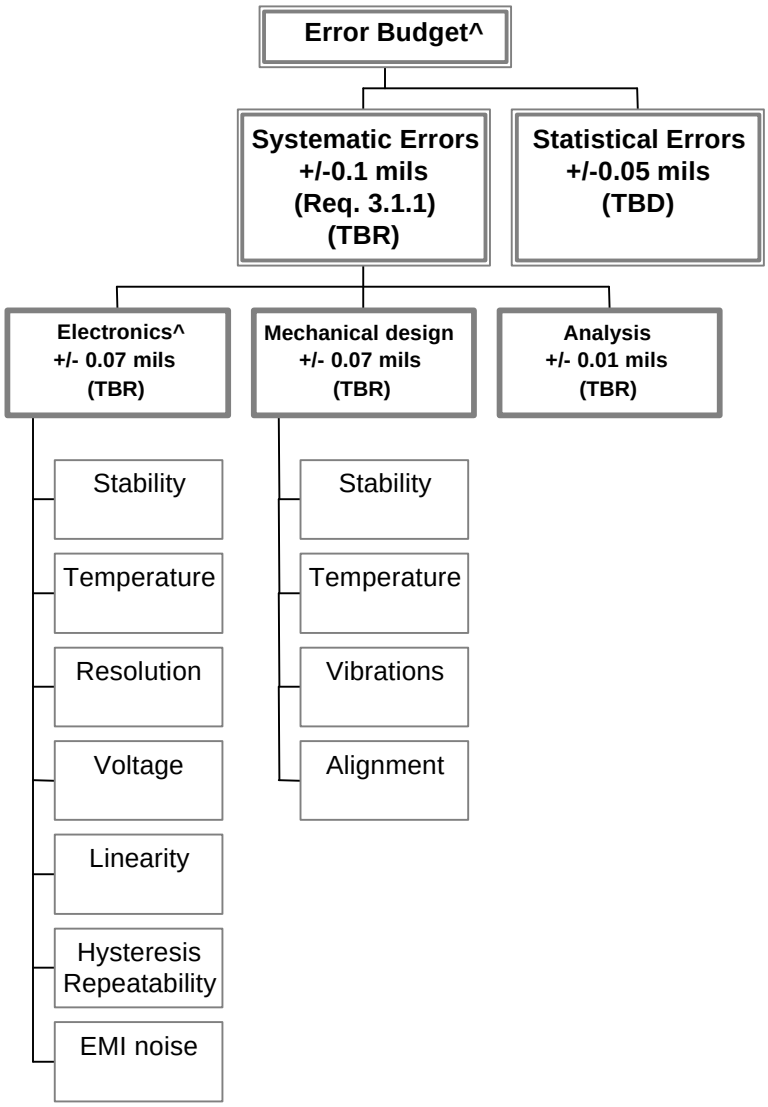


5.3.5 Selection Criteria



Notes: Each concept is evaluated against these criteria. Concepts that violate the "must rules" are eliminated up-front. The remaining concepts are scored for each Figure of Merit (FoM). Whenever possible, the FoMs are quantified using sound technical analysis. For example, cost should be quantified in $. The analyses and reasons for each score should be recorded. Popular decision-making techniques include Multi-Attribute Utility Theory (MAUT), the Analytic Hierarchical Process (AHP), Kepner-Tregoe (KT), and variations thereof.

5.3.6 Error Budget



^ The total error is computed as the RMS of the individual error contributions because they are statistically independent. The allocated errors are important design drivers.

5.4. Concluding Remarks

We think that by combining the SE and DesignWork techniques, the MuCoS project developed a set of activities and outputs that helped deliver a better product, faster, and cheaper than otherwise. The level of detail and formality of the activities and outputs were tailored to the needs of the MuCoS project. SE added a little effort in the early stages of the planning and analysis; but it probably paid-off by eliminating surprises in

the later stages. We think that when SE is not integral to a team's culture a SE presence and identity is necessary. It takes effort, but all projects can benefit from a SE approach and thinking. Experience has shown that the SE approach and generated outputs should facilitate the planned Conceptual Design Review (CDR) with LANL.

6.0 REFERENCES & RECOMMENDED READING

AIAA and INCOSE, Systems Engineering, 1997, http://www.incose.org/lib/aiaa/brochure.html.

ANSI/EIA-632-1998, Processes for Engineering a System.

B. Boehm, *Software Risk Management*, IEEE Computer Society Press, 1989.

F. Brooks, *The Mythical Man-Month*, Addison-Wesley, 1995.

C. Chapman and S. Ward, *Project Management Processes, Techniques and Insight*, John Wiley & Sons, 1997.

R. Clemen, *Making Hard Decisions: an Introduction to Decision Analysis*, Duxbury Press, 1996.

R. Cross and L. Baird, "Technology Is Not Enough: Improving Performance by Building Organizational Memory," Sloan Management Review, Spring 2000, pp. 69-75.

R. Davis, "Systems Engineering Experiences Growth as Emerging Discipline", Engineering Times, Nov. 2001, http://www.nspe.org/etweb/1!-01systems.asp.

C. Dym and P. Little, *Engineering Design A Project-Based Introduction*, John Wiley & Sons, 1999

K. Forsberg et al, *Visualizing Project Management*, John Wiley & Sons, 1996.

E. Hall, "Risk Management Return on Investment." Systems Engineering Vol. 2 (1999), pp. 177-180.

J. Hammond, R. Keeney, and H. Howard, *Smart Choices A practical Guide to Making Better Decisions*, Harvard Business School Press, 1999.

Y. Haimes and C. Schneiter, "Covey's Seven Habits and the Systems Approach: A Comparative Analysis", IEEE Transactions on Systems, Man, and Cybernetics - PartA, Vol. 26 (4), 1996, pp. 483-487.

M. Harris, "Process Improvement - A Management Primer", 35th Annual Engineering and Technical Management Conference & Symposium, GEIA, 2001.

I. Hooks and K. Farry, *Customer-Centered Products*, Amacom, 2001.

T. Hughes, *Rescuing Prometheus*, Vintage, 2000.

INCOSE Systems Engineering Handbook, January 1998.

INSIGHT Vol. 2, Issue 2, " Focus on Commercial Activities", INCOSE 1999.

T. Kelly, *The Art of Innovation: Lessons in Creativity from Ideo, America's Leading Design Firm*, Doubleday, 2001.

C. Kepner and B. Tregoe, *The Rational Manager*, Kepner-Tregoe, 1965.

H. Lewis, *Why Flip a Coin The Art and Science of Good Decisions*, John Wiley & Sons, 1997.

M. Maier and E. Rechtin, *The Art of Systems Architecting*, CRC Press, 2000.

S. McConnell, *Software Project Survival Guide*, Microsoft Press, 1998.

NASA Systems Engineering Handbook, SP-6105, June 1995.

H. Raiffa, Howard, *Decision Analysis Introductory Lectures on Choices under Uncertainty*, Addison-Wesley, Reading, 1970.

E. Rechtin, *Systems Architecting Creating & Building Complex Systems*, PTR Press 1991

D. Reinertsen, *Managing the Design Factory*, The Free Press, 1997.

J. Russo and P. Schoemaker, Decision *Traps - The Ten Barriers to Brilliant Decision-Making and How to Overcome Them*, Fireside 1990.

A. Sage, *Systems Engineering*, John Wiley & Sons, 1992.

M. Sampson, "The Allegory of the Humidifier: ROI for Systems Engineering", Computer, August 1997, pp. 102-104.
S. L Savage, *Insight.xla Business Analysis Software for Microsoft Excel*, Duxbury Press, Pacific Grove, 1998.
P. Senge, *The Fifth Discipline*, Doubleday, 1994.
M. Shaw and D. Garlan, *Software Architecting Perspective on an Emerging Discipline*, Prentice Hall, 1996.
S. Sheard, "Twelve Systems Engineering Roles", INCOSE Symposium 1996, http://www.software.org/pub/externalpapers/12ROLES.html.
S. Sheard et al, "Overcoming Barriers to Systems Engineering Process Improvement", http://www.software.org/pub/externalpapers/sepi_barriers.html.
K. Skytte, "Engineering a Small System", IEEE Spectrum March 1994, pp. 63-65.
U.S. DoE, Project Management Practices, Draft October 2000.
I. Sommerville, *Software Engineering*, Addison-Wesley, 1996.
R. Thayer and M. Dorfman (ed.), *Software Engineering*, IEEE Computer Society, 2000.
The Metrics Handbook, AFMC Pamphlet 90-102, 1995.
S. Wheelwright and K. Clark, *Revolutionizing Product Development*, The Free Press, 1992.
K. Wiegers, *Software Requirements*, Microsoft Press, 1999.