SPACECRAFT SYSTEMS ENGINEERING: AN INTRODUCTION TO THE PROCESS AT GSFC

by Tony Fragomeni and Mike Ryschkewitsch

Systems engineering means different things to different people. Some say it applies only to one spacecraft or a total mission. Others say it applies only to hardware and not to software, but that assumption is flatly wrong. Still others say it is electrically oriented while others say it is mechanically oriented; that depends upon whether you talk to an electrical or a mechanical engineer. Systems engineering is often equated with systems management and systems design. Some would reduce it to a purely analytical process and others would reduce it to mere hands-on physical integration.

Systems engineering is all of these and much more. It encompasses such terms as the system approach, system analysis and systems integration. It includes systems requirements analysis and functional analysis. The Goddard Space Flight Center's Code 400 *Project Manager's Handbook* says it is "one of the most important technical efforts of a project and . . . assures the design adequacy of the complete system to meet the stated user/experimenter requirements for a mission." These efforts include both the ground and flight segments, launch vehicle interface, and the end-to-end data system from collection of raw data on orbit to reduced data on the ground ready for analysis. The handbook says: "The Systems Manager of a project serves as Chief Engineer and provides a focal point for the systems engineering effort throughout all phases of the project."

As a succinct definition, that is as good as any but not really very helpful in understanding the systems engineering process, especially in the development of spacecraft. The concept becomes much clearer and richer when we ask why we need systems engineering, who a systems engineer is, what the systems engineer does and what are some of the products.

But first we can state what systems engineering is not. It is not one, single, isolated process. The whole process of systems engineering is better described as an attitude . . . a plan of attack . . . a way of thinking. Consider, for example, the difference between a chemist adding one ingredient to a fixed solution to achieve a predictable result, and a doctor who must consider a variety of uncertain and ever changing physical and emotional factors in the diagnosis and treatment of a patient.

As shown in Figure 1, systems engineering is not a process that is easily contained in a single manual or cookbook. Rather, it is the systematic use of many time-tested and experience-verified disciplines, tools and human resources needed to identify, define and solve problems. Which tools to use or expertise required depends not only on the mission under consideration but also the phase or stage of the project. The process thus demands a great deal of versatility and flexibility.

Finally, systems engineering is not always one individual or even one organization. Instead, it is a flexible process which makes the development and design meet the requirements and constraints imposed by the user and the system environment. It is a process characterized by multiple starts and stops, frequent shifts and alternate approaches, as opposed to a clear-cut path or a simple recipe for success.

Systems engineering is clearly a dynamic process that cannot and will not be pinned down into a simple procedural formula. This process, however, is generally the same for different kinds of projects. In these times of increasingly constrained budgets, it is

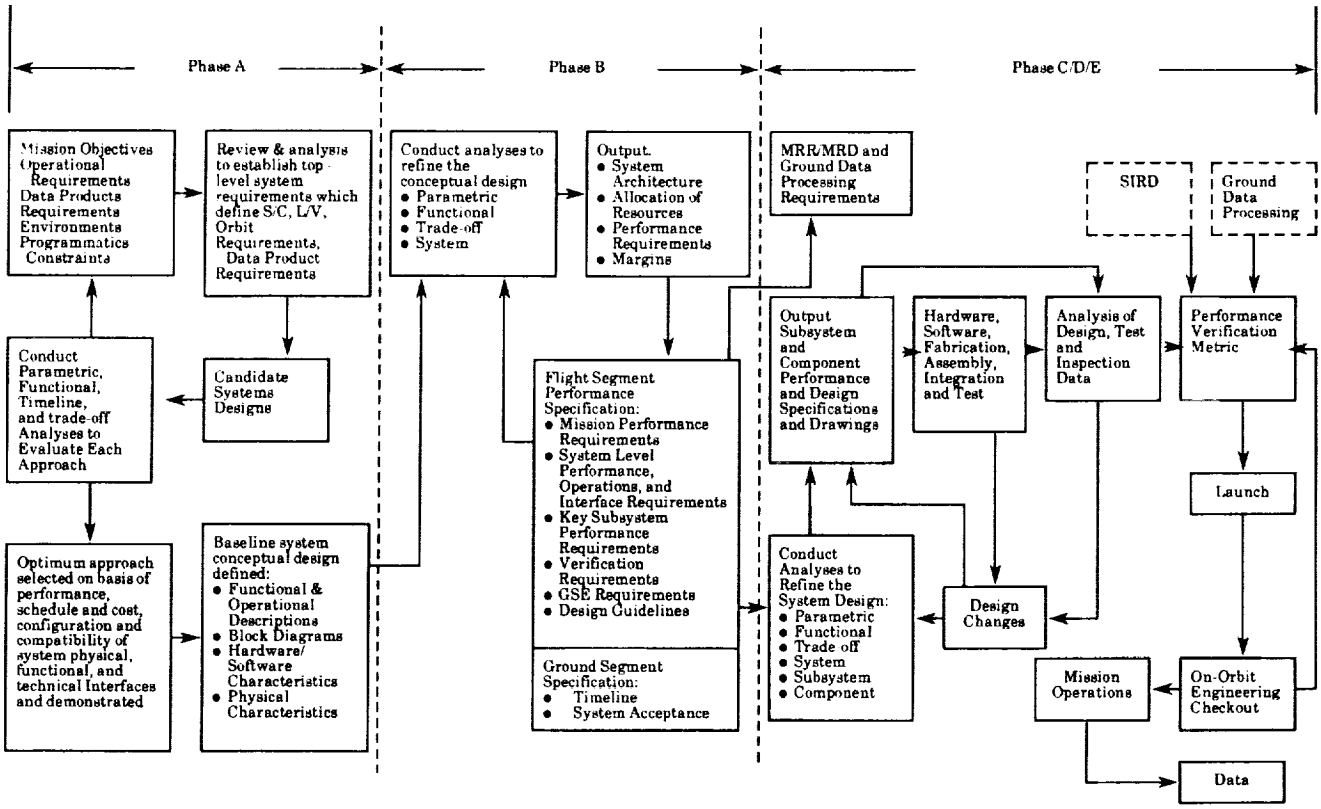

Figure 1 SE Process in the Evolution of a Mission

incumbent upon the systems engineer to optimize the systems design and to do things efficiently and *not* just effectively. Systems engineers are called upon to identify the risks in increasingly complex projects, and then attempt to minimize the impact of those risks. In very complex spacecraft, which are expected to perform delicate and ultrasophisticated functions, a minor intrasystem perturbation can have a major performance impact across multiple systems. Systems engineering is a disciplined technical approach that forces us to do our homework up front and early on, to uncover problems before they become showstoppers. Although we cannot conclusively test for everything, we are expected to identify and verify realities and adequate margins.

In a sense, we have always had systems engineering in NASA, but it may aptly be termed "informal." Certainly, we recall engineers and managers who had a big-picture perspective, looking at all functions and how they interrelate, but more often than not, their trade studies were on isolated scratch pads and the logic kept in their heads or in a desk drawer. You can almost hear them say: "This is the way we've always done it."

Sometimes this informal system worked, especially on small, relatively simple projects. But as the spacecraft became more complex and development time elongated, a more formal process of systems engineering emerged. In simple terms, it starts with functional analysis and leads to functional requirements and then design requirements. It starts at the top and works down, fully documented at each step and traceable. The greater the complexity and duration of a project, the greater the penalty for not catching errors early on, and the greater the need for a well understood and well documented process. The SE process should ensure that all fixes be made before the start of hardware fabrication when the cost of fixes is relatively inexpensive. To wait until later is costly, and

it can be prohibitive at the interval between acceptance testing and launch.

SE ROLES AND RESPONSIBILITIES

The main objective in systems engineering is to devise a coherent *total* system design capable of achieving the stated requirements. Requirements should be rigid. However, they should be continuously challenged, rechallenged and/or validated. The systems engineer must specify *every* requirement in order to design, document, implement and conduct the mission. Each and every requirement must be logically considered, traceable and evaluated through various analysis and trade studies in a *total* systems design. Margins must be determined to be realistic as well as adequate. The systems engineer must also continuously close the loop and verify system performance against the requirements.

The fundamental role of the systems engineer, however, is to engineer, not manage. Yet, in large, complex missions, where more than one systems engineer is required, someone needs to manage the systems engineers, and we call them "systems managers." Systems engineering management is an *overview* function which plans, guides, monitors and controls the technical execution of a project as implemented by the systems engineers. As the project moves on through Phases A and B into Phase C/D, the systems engineering tasks become a small portion of the total effort. The systems management role increases since discipline subsystem engineers are conducting analyses and reviewing test data for final review and acceptance by the systems managers.

REQUIREMENTS

The name of the game in systems engineering is requirements. The statement, traceability and eventual verification of requirements is probably the most important aspect of systems engineering. Requirements are initially derived from user needs, i.e., the customer. It is understood that for each requirement there is an associated margin that must continually be challenged. As the project nears completion, the amount of available margin is expected to decrease since the margins are updated based on "actuals."

- **Functional Requirements** provide a description of the functions and subfunctions required to conduct the mission. These are generally derived from functional analysis and allocation.

- **Performance Requirements** or source requirements define what the system must accomplish and how well the system must perform. These requirements are initially derived from user needs and requirements statements and refined through requirements analyses and trade studies. They are defined during each application of the systems engineering process based on outputs from previous iterations of the process, program decisions and updates to user requirements. They provide the metrics that must be verified through appropriate analyses, demonstrations and tests.

- **Derived Requirements** are lower level (subsystem and components) performance requirements resulting from an analysis of the user stated performance requirements and the definition of functional requirements. These derived requirements are used by subsystem discipline engineers in characterizing the subsystem performance requirements necessary to ensure the attainment of the user-stated performance or source requirements.

- **Reflected Requirements** are requirements placed on other subsystems or on the higher level systems which must be provided to each of the subsystems to ensure proper performance of the subsystem and the eventual attainment of the user

stated performance or source requirements.

- **Design Requirements** are described by drawings, material lists, process descriptions and other supporting documents for the fabrication, production or manufacturing of a system element. These are generally derived from the synthesis of a solution for one or more higher level requirements.

The systems engineer must be able to demonstrate the traceability of each requirement through each level, right up to the contractually binding source requirements. User requirements are determined and refined during Phase A studies. A host of considerations are made in order to produce the best set of "integrated performance requirements," considering technical performance, first as mitigated by cost and schedule. Systems engineers should not and do not make cost and schedule decisions, especially in the later phases, but in Phases A and B, cost and schedule are trade-off parameters that must be considered in determining the best course of action.

PHASE A - MISSION ANALYSIS

In Phase A Mission Analysis, systems engineers will translate user needs or goals into a quantifiable set of functional requirements that can be translated into design requirements. User requirements are defined as a "set of objectives" that are quantified in broad terms and basic functions. The user should also state performance measures in terms of preferences as well as trade evaluation criteria. The systems engineers will conduct functional, parametric and system analyses to define and refine mission requirements and to generate alternative candidate system designs. Baseline system conceptual designs should emerge as design drivers are identified, as well as high risk areas and offsets. Common system drivers include size, weight, power, data rate, communications, pointing, orbital altitude, mission operations coverage (geometry and timing) and scheduling. Trade-off studies are conducted to balance the requirements, but even the optimal technical approach may not be the best way when the design is evaluated in terms of cost, schedule *and* risks. Since all projects will undergo cost, schedule and technical perturbations during development, it is imperative that a good system be developed. However, contractual, legal and fiscal requirements dictate that the technical approach must be agreed to by the start of Phase C/D. The overall system architecture must be established during Phase A; this includes the apportionment of functions between the flight and ground segments. It is imperative that proper studies and analyses be done to result in the correct structure since this affects the remainder of the project up through the operations phase.

Phase A outputs or products include a Phase A Report, a Science Requirements Document, preliminary Instrument Interface Requirements Documents, cost, schedule and a Project Initiation Agreement (PIA). The Phase A Report includes functional and operational descriptions, hardware and software distribution, design requirements, system/subsystem descriptions, mission description, a preliminary work breakdown structure (WBS) and recommendations for Phase B. The Phase A Report must have sufficient data to answer questions such as these:

- Do the conceptual design and operational concept meet the overall mission objectives?
- Is the design technically feasible?
- Is the level of risks acceptable?
- Are schedules and budget within the specified limits?
- Do preliminary results show this option to be better than all others?

PHASE B - DEFINITION PHASE

Assuming that each crucial question is answered affirmatively during Phase A, the systems engineer will continue development of the system requirements by conducting more detailed analyses to refine the baseline system conceptual design. These Phase B tasks must result in technical requirements and operational functions that are reflected in Interface Control Documents, performance and design specifications and statements of work that are used to produce the hardware during Phase C.

Specifications are defined as "a description of the technical requirements for a material or product that includes the criteria for determining whether the requirements are met." Basically, there are four types of specifications:

- Functional - describes only the ultimate end use; contractor is responsible.
- Performance - describes quantitatively what it must do; contractor is responsible.
- Design - what to make and how to make it; buyer is responsible.
- Levels of Effort - used only for support services.

The statement of work (SOW) describes the work needed to carry out the entire mission as well as how and where the work is to be done. The work breakdown structure (WBS) is used for reporting progress, performance and engineering evaluations. The WBS will structure the family of specifications and drawings resulting from the progressive stages of systems engineering. The final result of the Phase B process is a system definition in sufficient depth of detail to allow beginning the detailed design process for each of the individual subsystems.

PHASE C/D - EXECUTION PHASE

During Phase C/D, systems engineering provides technical oversight during design, development, test and evaluation to ensure that timely and appropriate intermeshing of all technical disciplines are reflected in the overall design. Technical performance requirements and margins are continually reaffirmed through analyses and tests during this phase. Phase C/D outputs or products will also include a variety of analytical and test reports on hazards, faults, single-point failures and failure modes for "what-if" or worst-case scenarios. Trade-offs and other analyses continue but in greater detail at the subsystem and component levels to ensure proper conversion of performance requirements into the design and into the hardware.

PHASES E AND F - PRE-MISSION AND MISSION OPERATIONS

Phases E and F, Pre-mission and Mission Operations, also involve systems engineering, although to a lesser degree since the most important SE work is done early on. However, the final verification of a space flight, system can only be done in flight, on-orbit. The systems engineering team is full time with the flight operations team during initial on-orbit engineering checkout and on call during mission operations. The final product is the "On-Orbit Engineering Performance Report" which measures mission performance against requirements. This document becomes useful in subsequent projects, especially if it contains lessons learned. Finally, the systems engineer's job is only completed when the user has the final delivered product, e.g., scientific data, in hand.

SYSTEMS ENGINEERING ANALYSES

Systems engineering is a highly analytical process. Throughout the entire project (not just at the beginning) the systems engineer will conduct or review numerous analyses to establish strong performance and design parameters as well as to continually evaluate design approaches and options. A systems engineer is expected to establish

performance parameters and margins, verify them with test and inspection data, and compare the actual to the predicted. Everything must be "what-ifed" to the lowest necessary level, not just once but continually, so that there are few if any surprises.

One tool used by the systems engineer is functional analysis. This is a top-to-bottom effort done in all phases and at every hardware level. The systems engineer takes a performance requirement (function) at one hardware level of assembly and, after thorough analysis, determines the optimum distribution and implementation of the requirement at the next lower hardware level. Functional analysis is also used to determine whether a particular function is best accomplished in flight or on the ground. Functional analysis results in a hierarchical structure (i.e., architecture) that progressively divides and allocates how a function is to be accomplished, down to the lowest common denominator. This is extremely useful in deciding where to cut the interface, especially in view of verification, accountability and jurisdictional (i.e., contractual) boundaries.

Another top-to-bottom systems engineering analysis done in all phases is the requirements flowdown and allocation analysis. This can be described as an equitable, attainable and realistic distribution of system-level performance requirements and resources, including margins, to successively lower levels of hardware assemblies. To verify the validity and distribution of tolerances and margins, continued analysis and review are required throughout the project. This starts during Phase A and continues through every on-orbit checkout. Distribution should be compared to actuals, and estimates should be quantified as a function of design maturity.

Trade-off studies and analyses also define margins and identify potential problem areas. They are done on all systems and for all technical disciplines to select the configuration that best satisfies a user requirement. Alternative technologies are examined to satisfy functional and design requirements, including those with moderate to high risk. Trade-off studies also support make-or-buy decisions and help manage technical risk. In Phases A and B, they establish system architecture and configuration. In Phase C/D, they evaluate alternate solutions in system/subsystem/component design. After critical design review (CDR), however, trade-off studies are conducted only during the evaluation of design changes or responses to failures. All factors that affect the function or requirement must be studied: performance, reliability, safety, cost, risk, schedule, maintainability, servicing, power, weight, thermal, complexity, etc.

System parametric and sensitivity modeling and analyses are used to develop confidence that a design satisfies higher level requirements, and to provide traceability of functional, performance and design requirements. This is accomplished by varying a particular performance parameter between its established worst-case limits and as perturbed by worst-case environmental stresses to determine the resultant effect on successively higher assembly levels or performance parameters. These analyses can serve as a primary vehicle for conducting trade studies and to assess the whole system effectiveness of synthesized design options and alternatives. Like all other studies and analyses, these analyses are done during all phases and are updated based on actual test data.

RISK ASSESSMENT

Risk assessment is approached from different but related directions. During Phases A and B, the systems engineer will want to do sufficient analyses to ensure that the technical approach is valid and that any new developments or state-of-the-art items and their risk offsets have been identified. During Phase C/D, sufficient analysis must assure that performance requirements and margins are adequate and are in fact satisfied. Throughout the entire project life cycle, risk assessment and particularly Failure Mode Effects

Analyses and fault tree analyses should be used as design tools to enhance the overall system design and make it immune to failures, both hardware and human.

Risk assessment is the identification and evaluation of the impact upon the technical performance of those system elements that appear to possess an inherent probability of failing to meet some critical performance or design requirement essential for the successful accomplishment of the intended mission. Systems engineering identifies the potential failures, establishes margins and quantifies the risk. Risk taking gets down to knowing what your margins are and how they are distributed. How do you know what the margins are? By doing lots of analyses and backing them up with tests. Two of the best tools are Failure Mode Effects Analysis (FMEA) and hazards analyses.

The FMEA assures that the failure modes of a system are known and can be addressed in an orderly fashion. Initially the analysis must identify all critical functions and the effects of the impairment of those functions on mission success. Following this, a detailed component and system interaction study is conducted to determine all the ways a function could be impaired, the effect on mission success and how such an impairment could be detected. The impact of these failures and the probability of occurrence must be evaluated in light of the user requirements and the desired level of reliability.

The FMEA is also used in compiling the system-level fault tree used by the flight operations team (FOT) during mission operations. The fault tree is a listing of every plausible anomaly or failure that may occur on orbit. It starts out with the detection of the anomaly or failure as observed by the FOT via telemetry. It then provides a road map used by the FOT in isolating the cause of the anomaly and taking the required corrective action or operational work-around so that the mission can proceed. The fault tree analysis and the development of the FMEA should be done together.

Systems safety hazards analyses are also considered a systems engineering function. The intent of the systems safety hazards analysis is to identify design deficiencies that could directly — or indirectly through operator error — result in personnel injury or damage to the flight hardware. In this case, any potential hazards that could result in death, severe injury or illness must be eliminated. The impact of a major system loss or damage must be evaluated in light of user requirements.

Operations hazards analyses look at possible failures occurring during testing, handling and transportation that could jeopardize the hardware or personnel. All catastrophes and critical hazards resulting in death, severe injury or illness, or major system loss or damage must be eliminated. Marginal hazards may be tolerated if they can be rationally justified and accepted.

REVIEWS, PERFORMANCE ASSESSMENT AND VERIFICATION

The systems engineer is best advised to start early and stay late in reviewing and assessing performance requirements and the associated verification methods employed to prove the requirement has been satisfied. Reviews must be done at all levels. Non-advocate reviews (NARs) should be conducted at the end of Phase B to evaluate the technical, cost and schedule approach for accomplishing the mission. System-level reviews and lower-level hardware design and test reviews should be conducted continually. Peer reviews are vital at all levels and must be conducted by "looking at the drawings and not the viewgraphs." Trend analysis is needed on all critical performance parameters, from box level acceptance through on-orbit to enable the early identification of potential problem areas. Technical performance measurement (TPM) is one proven method of assessing compliance to requirements and the level of technical risk. TPM is defined as the continuing analysis, test and demonstration

of the degree of anticipated and actual achievement of selected technical measures and performance parameters. TPM involves analysis of the differences among the achievement to date, current estimate and the required or target value for the parameter.

SUMMARY AND SOME ADVICE

Systems engineering is much more than a one-person job. It is best described as "the technical conscience of a project." As such, systems engineering is a highly structured and disciplined engineering process that cuts across all technical disciplines to ensure interface design compatibility, both intersystem and intrasystem. It organizes at the system level — not at the subsystem level, where compromises may be made. It establishes performance requirements and margins. Systems engineering evaluates the validity of hardware through analysis and review of test data. It identifies risk and offers approaches for the project manager to eliminate or reduce the impact. One eye of the system engineer is on how the end product is used during mission operations; the other is focused on how analyses and tests can prove it can do the job within acceptable margins. Both eyes work in tandem, together, clearly and in focus. Remember:

1. Perform sound systems analyses and design; consider *all* options.
2. Don't box yourself in with unnecessary and undue constraints.
3. Exercise extreme care in system design, especially incorporating appropriate (to the risks) redundancy and provisions for late design changes and on-orbit operational work-arounds, and factor in testing ability.
4. Institute the discipline to ensure painstaking attention to details — great and small.
5. Maintain a total dedication to quality — quality is *designed in*, it does *not* accidentally happen.
6. Ensure rigorous pre-launch testing to establish that requirements are in fact satisfied, and any workmanship or marginal designs are uncovered.
7. Insist on inexhaustible diligence in testing — allow an unexplained or random failure only after all reasonable and practical steps to isolate are taken.
8. Attempt to design backwards — satisfy mission requirements first.
9. Conduct extensive reviews — look at the drawings, not viewgraphs.
10. Have adequate documentation to know where you are going, how you are getting there, where you have been and *when you are there.*
11. Have an open door policy to foster strong intra-project technical communications.
12. Ensure total openness regarding problem identification and resolution.