

N93-24685

THE SE ROLE IN ESTABLISHING, VERIFYING AND CONTROLLING TOP-LEVEL PROGRAM REQUIREMENTS

by Charles W. Mathews

158577
P-10

People working in the field of systems engineering have differing views as to the range and depth of this subject. Without venturing into the controversial arena of specific definitions, I will assert that systems engineering has much to do with the definition, evaluation and control of the technical effort aimed at achieving the objectives of a program. Efforts in the field of systems engineering may in fact go well beyond purely technical considerations, e.g., when cost or political considerations impact the technical approach to a program. In this context, the systems engineering process must function to maximize the probability that a program's technical requirements can be met while at the same time recognizing and including other program factors and constraints. New constraints as well as technical problems can be encountered at all stages of a program, often necessitating some adjustment to the program objectives and requirements. Such activities are part of the systems engineering process, which must begin immediately at the start of a program and continue throughout the life of the program.

Sometimes a program manager will concentrate on insuring that hardware elements perform well and all play well together, assuming that this alone will enable the program requirements to be met. Then on entering the operational phase, while the system may indeed perform, it may not do what was intended. This situation frequently occurs because many engineers, scientists, managers, and yes, even administrators tend to be intrigued by and want to concentrate on configuration selection and design problems. It is the responsibility of the top-level systems engineering professionals to be the conscience of all participants in making sure that program requirements are met or properly adjusted.

The need is to focus on program requirements during all phases and facets of a program, e.g. definition, development, manufacturing, testing, operations, growth and, most important, effective use or mission accomplishment. The effort just described involves the entire systems engineering task; however, the main emphasis of this paper is the interaction of the systems engineering process with the top-level program requirements. This aspect of systems engineering is often given inadequate attention during certain phases of a program. This paper will endeavor to answer such questions as:

What is meant by top-level program requirements, and who generates them?

How are these requirements validated, altered, and controlled by the systems engineering process?

What capabilities are needed to accomplish such efforts effectively?

WHAT ARE TOP-LEVEL PROGRAM REQUIREMENTS?

Top-level program requirements are directly related to program objectives or systems uses determined and stated early during the program definition. Probably the most remembered program objective of the past was to "land men on the moon and return them safely to Earth." The program requirements that emerged from early studies included, among others, one to two-week mission durations, lunar landing, extravehicular activities, launch from a remote site, rendezvous, and reentry from near escape velocity, all of which had never been accomplished at the time of President Kennedy's statement.

These requirements in turn highlighted the need to define and validate specific technical approaches—redundancy concepts, simple system interfaces, new technology

requirements (e.g., fuel cells), operational demonstrations such as Gemini, entirely new configurations such as the LM, and the nature of the flight program buildup. Incidentally, many of the program requirements for Apollo determined the mission objectives for the earlier Gemini program. In any event, program requirements must be established early and stated distinctly so that all necessary steps for meeting and validating them can be determined. This effort is a fundamental systems engineering function.

Types of Program Objectives and Requirements

The program objectives and requirements described in the preceding paragraphs emphasize mission demonstrations. Obtaining desired science or applications information is another type of program objective. The program requirements then state the need for specific data, usually specifying a particular instrument or instrument set; the operating conditions under which the data is to be obtained (e.g., orbit altitude, field of view, and pointing accuracy); and the data handling and use. Conversely, a new instrument may be conceived or created with the program objective to establish its use potential. The Multispectral Scanner employed in the Landsat program is an example.

Another space program category includes service functions such as Earth-to-orbit transportation or a space laboratory. In the first case, the program objective might be economical and an easy access to the space environment for the using community. Program requirements then include such parameters as dollars per pound to orbit, launch frequency and payload integration lead times. Conversely, in this case, the program objectives might also be stated in terms of capability demonstrations such as the reentry of a winged spacecraft, ground landing and reusability. The program requirements then are related to system

performance in accomplishing these mission and configuration demonstrations.

It is important to firmly establish which of the above two categories reflect the real program objectives because a capability demonstration has a higher potential for success than a tightly specified use commitment. The systems engineering organization should be providing top-level program management with the information to make such determinations. The program objectives may vary during program implementation because of early "selling" pressures or because of unforeseen technical problems. When this happens, the systems engineering organization should provide concrete evidence to management so that a strategy can be developed to properly inform the outside world, e.g., Office of Management and Budget (OMB), Congressional committees and the media; if the outside elements are not made to understand and accept such changes in a timely way, support can be alienated, placing extreme pressure on the program.

Establishing Priorities

When a large number of objectives and associated requirements are included in a given program, an additional complication occurs. Several past programs qualify including programs as early as Gemini and space station programs such as Skylab. Even Apollo, with its simply stated mission objective, had many secondary objectives associated with lunar exploration and lunar science. It is very important to establish priorities without precluding the accomplishment of objectives of lower priority. For example, the two top priorities in the Gemini program were demonstration of long duration flight and rendezvous, but large quick-opening hatches were incorporated to accommodate extravehicular activities (EVA) and the spacecraft structure was designed to permit the firing of a large propulsive stage once docked to it. Most of these secondary objectives were

accomplished. In fact, because of the way the actual flight program developed, EVA was one of the first accomplishments. The secondary program objectives also afforded some flexibility; the paraglider system planned for use in ground landing, for example, was dropped from Gemini in order to meet cost and schedule objectives.

To summarize what has been stated thus far, a number of classes of top-level program requirements exist. They can be associated with mission objectives, scientific investigations or space services, among others. In addition, different ways of looking at top-level program requirements include demonstrations as compared with tightly specified commitments. Many programs have multiple requirements. Nevertheless, it is important to 'zero in' on these requirements early in the systems engineering process, i.e., during Phase A. Most important, they must combine to realistically meet the stated objectives of the program; they must be prioritized when necessary; and they must be clearly stated and documented in the Program Requirements Document.

These requirements may have to be changed, adjusted or reprioritized as the program proceeds, and any changes must be carefully controlled and formally approved at the top level of the program throughout its life. If program objectives are affected, a decision by the administrator is required (at least for medium-to-large programs). The outside world needs to be kept abreast of significant changes in objectives or top-level requirements so that no sudden surprises occur that affect support.

The systems engineering function should provide the initial evaluation and validation of the top-level program requirements and should continue to evaluate proposals or events that would produce any change. The effort should occur at the top level of a distributed systems engineering function and guide upper level program management and the administrator.

WHO GENERATES TOP-LEVEL PROGRAM REQUIREMENTS?

A program objective can be conceived and stated initially by almost anyone working at any level, from the President, as in Apollo, to others on down. If considered seriously, such an objective is studied to determine its validity, practicality and usefulness. Sometimes it takes a short time to obtain a go-ahead; sometimes it takes many years, as on the Space Station. One of the fallouts of these efforts should be a clear statement of top-level program requirements.

The involvement of the right people in the generation of top-level program requirements is extremely important. Depending on the nature of the program, this involvement can include customers, users, operators and, of course, designers and developers. Program managers and directors, however, should guard against limiting involvement in this activity to just the latter two. Systems engineering, should be involved early to assure a reasoned and logical approach to the generation and iteration of program requirements.

In the space science and applications arenas, program requirements are frequently generated by a process that begins with a program objective or a flight system capability being stated in an "Announcement of Flight Opportunity." Investigators are then selected through evaluation of the responses obtained. The experiments selected determine the actual requirements of the flight program. Other inputs are often required, as adjustments may be needed in consideration of technical limitations or program costs, for example. The analysis and resulting output of the systems engineering group usually gives rise to an iteration of the program requirements, which again involves the science team. Frequently, a selected proposal provides for excellent science but does not deal adequately with other constraining technical considerations and the cost implications associated with the overall effort.

Hierarchical Consideration in Requirements Generation

In all classes of space flight programs, the systems engineering organization should work closely with groups having expertise in and cognizance over program requirements. In Apollo, because the primary program objective was oriented to the accomplishment of a specific mission demonstration, operational personnel—particularly those involved in flight operations—tended to be near the top of the program requirements hierarchy. Even though science requirements existed and science teams and advisory committees were active, the science requirements were of lower priority, at least until after the first lunar landing was accomplished. In contrast, a program such as Skylab always included the solar scientists and Earth resources investigators, among others, at the top of the requirements hierarchy, even though the engineering and operations personnel may have been somewhat confused by this arrangement.

The Space Shuttle involves still another situation. The operations groups can be perceived to be the customers for the system, but the real users at the top of the hierarchy are the scientists, commercial firms, industrial experimenters and NASA engineers who provide the payloads that fly on the Space Shuttle or conduct related experiments or other use functions. This is similar to the relationship between passengers and shippers, the airlines, and the commercial airplane developer in the air transport industry. In addition to general operating efficiency, consideration must be given to user accommodation from the start. Such needs are now quite successfully accomplished in commercial aviation. Naturally, expectations are less in the case of the Space Shuttle because of its experimental nature, but it is fair to say that user accommodation has not been accomplished to the degree desired.

The foregoing discussion is not meant to imply that successful hardware design,

development and systems integration is not an important facet of systems engineering. There are instances where these considerations are at the top of the requirements hierarchy. An instrument demonstration such as the Multispectral Scanner is one case in point, and the Advanced Communications Technology Satellite (ACTS) is another technology demonstration of this type. In most respects, the research airplanes such as the X-1 and X-15 fit into this category. However, this case does not fit the situations occurring in most NASA programs. It is therefore critical for top-level program management to examine its program, determine who the main contributors or generators of the program requirements are, and assure that they are interfacing adequately with the systems engineering function. This need exists at the outset of the program but should continue through the design and development phases, for as hardware and software systems problems are encountered, the tendency is to focus on them, and top-level program requirements can be altered or even disappear without due consideration.

WHO VALIDATES TOP-LEVEL PROGRAM REQUIREMENTS?

Activities that validate top-level program requirements are mostly of a systems engineering nature. This validation, is an important, though small, part of the total systems engineering job. In total, systems engineering, particularly during design and development, is a distributed activity. Spacecraft hardware systems such as electrical power, attitude control and communications all have to be systems engineered. Total systems elements (e.g., a launch vehicle stage, a checkout facility, a launch complex and a flight control center) all have to be systems engineered to correctly perform their functions. In the end, all elements involved in a program—the total flight system, the operational support facilities, the mission planning, and the user integration, among

others—need to be brought together in a timely fashion to meet the program objectives and requirements. An effort of this nature, even for a very modest program, is too complex to be handled in a purely top-down fashion. The cardinal rule is that all the interfaces at any particular level, both horizontally and vertically, should be as clean and simple as is practical.

Validation Efforts During Program Definition

At the start of program evolution, practically all of the mainstream effort is of a systems engineering character and is more top-down than later in the program. The validation effort begins in pre-Phase A, where options are examined for meeting the program objectives as well as certain initially stated program requirements. These requirements should endeavor to incorporate most of the major program factors but are usually general and often are quite optimistic. All aspects of the technical and programmatic approach should be studied. Although effort is limited in this phase, a determined attempt must be made to establish and to ascertain the feasibility of meeting the program requirements. This work should usually be accomplished by a team working at a single location, although supporting effort and information can be obtained from groups in other locations. There have been cases where alternative approaches are studied by separate teams, which has proved to be effective in some pre-Phase A efforts. In all likelihood, the program requirements will be changed and expanded to account for such factors as technology readiness, knowledge of the operating environment, mission complexity and similar factors. A need for additional technology development or operational verifications may be identified as well. Any pre-Phase A study that is completed with everything looking rosy should be viewed with caution.

Phase A efforts are aimed at selecting and analyzing a single programmatic and technical approach, at least in theory, to best meet the requirements of the program. Again, the Phase A activity is chiefly a systems engineering effort usually conducted by a single team at a single location. If a work breakdown structure with clear interfaces can be established at this time, then systems engineering at multiple locations may be possible. In any case, the group that worked during the pre-Phase A study needs to be augmented considerably, and the support of one or more contractors is frequently obtained.

In this phase, emphasis should be placed not only on hardware but on validating the mission design and other operational or use aspects of the program. This emphasis is particularly important where the operational life of the program is envisioned to be very long, e.g., Space Shuttle, Hubble Telescope, Space Station and the Earth Observing System (EOS). It is important to clearly establish what is required in the operational phase and to establish with adequate confidence the feasibility of accomplishing the programs with realistic operational costs and schedules.

At the time the program enters Phase B, a complete work breakdown structure should be established, including all facets of the program with simple and clear interfaces and as little overlap as possible. Program work assignments will be made. For moderate to large programs, these assignments may involve program groups at different geographic locations, including parts of the total systems engineering effort. The top-level program requirements should have been established in adequate detail, and each program organizational element should regard these requirements as program constraints.

The program requirements or even the objectives can be changed because of unforeseen events or other activities occurring

throughout the course of the program, but they should be subject to formal change control. Obviously this particular change control activity deals with top-level program requirements and must occur at the highest level in the program; in certain cases, the administrator should be informed of an impending change and must be informed when program objectives are significantly impacted.

Validation Efforts During Design, Development and Operations

Although the top-level systems engineering effort in the definition phases of a program is important, this function is critically important in Phases C/D, the design and development phases. It is during this time that most of the technical difficulties and other program limitations surface. There is a strong tendency to focus on the flight hardware and to get it delivered and flying. These situations sometimes allow the top-level requirements to "fall through the cracks," later producing surprises, embarrassments and undue pressures, which can contribute to the potential for accidents and failures in the operational phase.

Systems engineering must continue throughout the operational phases of a program. Although the character of the top-level activities change, there still is a need to deal with program requirements and their alteration. Some of the possible subjects are the rate and nature of the flight program buildup, working around performance deficiencies or failures, and adjustments to mission objectives. On the positive side, the top-level systems engineering in the operational phase involves the incorporation of new system capabilities and mission extensions, including the development and control of the associated program requirements.

Support to the activities just described is accomplished by a systems engineering group also operating at the highest level in the program's organizational structure. This

group is the guardian and conscience of the top-level program requirements but by no means includes the total systems engineering effort. The group should be composed of engineering personnel, each of whom has considerable technical experience in one or more of the applicable areas and possesses a natural talent and desire to deal with all aspects of the program. The individuals should be selected so the group encompasses as many of the technical, scientific and programmatic disciplines involved in the program as possible, but the group does not have to be large. By selecting people with the right backgrounds and talents, the work can be done in part by obtaining information from other elements of the program—in particular, other systems engineering groups.

HOW ARE TOP-LEVEL PROGRAM REQUIREMENTS CONTROLLED?

Control of top-level program requirements is extremely critical to program success. This is not to say that such requirements cannot be changed. Almost without exception changes will occur, but they must be carefully controlled by a well-defined process that establishes the change impact on the program, particularly its objectives. This process also must inform program participants inside and outside the program organizational structure, including those having responsibilities or scrutiny from above.

The program director is the individual who is personally responsible for the integrity and control of the top-level program requirements. As such, the program director must assure that a Program Requirements Document is produced during Phase A and that it is properly updated immediately following a change. This effort is supported mainly by the program director's systems engineering group described in previous sections. This group is responsible for analyzing any proposed change that could potentially impact the top-level program requirements.

The analysis can be done by the group itself or by support groups, including contractors. The analysis must specifically include in writing how the affected requirement(s) would be changed and the determination of other impacts such as cost or schedule, which could be either positive or negative.

Change Control of Program Requirements

Change proposals are brought before a standing committee, usually called a change board, selected by the program director. There will be other change boards throughout the program, but this one should deal only with top-level program requirements. Anyone who proposes a legitimate change in the program requirements should be able to come before this board. In general, individuals who have a significant input should also be invited. The proposed change is usually presented by its sponsor and is followed by a presentation of the analysis of the systems engineering group. Following discussion, the program director makes the disposition, which can include acceptance, rejection, or a requirement for further analyses or information. Following an acceptance, the Program Requirements Document should be changed immediately. Regardless of the nature of the decision, the affected elements of the program organization need to be informed immediately. Affected elements outside the program should also be informed in a timely manner but only after an appropriate strategy is developed.

One of the chief difficulties associated with this change control activity is that events that impact the top-level program requirements can occur at any place, at any time and at any level in the program, and there is a natural tendency to try to fix a problem at its source without passing on information. Several things can be done to alleviate this difficulty as it relates to the activities of the top-level systems engineering group. Individuals in the group must

attend design reviews and other program reviews associated with all the program elements. They must be able to have free information exchange with other program and project personnel and to accompany them on visits to contractors when the occasion demands. These activities are best accomplished if the group and its members operate with a low profile. They should not give or imply directions or conclusions in discussions with program and project people. All direction as a result of their work should come from the program director. Naturally, these individuals must be able to request and analyze program documentation, but all such activities should be done in a way to maintain good rapport with other groups working in the program.

TOP-LEVEL PROGRAM REQUIREMENTS IN PREVIOUS PROGRAMS

In general, most of NASA's past major programs have successfully met their program objectives and must have fulfilled their program requirements. Some brief observations of the results obtained during some of the previous manned programs may provide useful insight into future programs. Although the very early programs were not explicitly divided into program phases, in retrospect, it is possible to discuss them within a phased context.

The Mercury Program

The Mercury program objective was to place a manned spacecraft in orbit around Earth and return safely. In pre-Phase A, several winged (lifting) configurations were studied as well as the so-called "capsule." The capsule was selected on the basis of greater technical simplicity and limitations on launch vehicle payload capability. In Phase A, in addition to developing the spacecraft systems specifications, safety requirements were emphasized, including the proper positioning and support of the crew to handle

launch and reentry accelerations, which were demonstrated on a centrifuge; the concept of a system to escape from the launch vehicle if necessary; and the layout of a worldwide tracking and monitoring network. In Phase B, a full-scale demonstration of the reentry heat protection system was conducted, and the results produced minor design changes. The concepts of flight control and recovery were evolved, including a mission control center and flight controller deployment to remote sites, worldwide communication for near real-time surveillance of the missions, and recovery procedures involving ship deployment.

The spacecraft configuration and specifications proved to be satisfactory although considerable development problems were encountered. The biggest systems engineering problem was associated with the lack of appreciation of the difficulties in conducting factory and preflight checkout. The checkout required more or less continuous human presence in the extremely confining interior of the spacecraft, producing wire breakage and other damage. These conditions were severe enough to curtail the flight program, although six manned flights were made, building up to a duration of approximately one day in orbit.

The Gemini Program

The pre-Phase A activity concentrated largely on correcting some of the basic problems encountered in Mercury, i.e., a Gemini spacecraft design that had most of the equipment outside the pressure vessel and was also checkable from the outside, allowing a relatively clear cockpit area. The spacecraft was enlarged to provide for a two-man crew, but the basic external configuration and heat protection system of the Mercury spacecraft was retained.

Most of the Phase A activity involved defining the mission objectives, in support of Apollo, and the related program requirements associated with rendezvous and long

duration flight, e.g., the Atlas-Agena target vehicle, orbit maneuvering system, rendezvous radar, fuel cells, and the cryogenic storage of hydrogen and oxygen in a supercritical state. Again, considerable development problems emerged, largely associated with the newer systems, such as ablative thrusters and fuel cells. Problems were also encountered in the flight program. The initial rendezvous exercise revealed inadequate attention to mission design, which was later corrected, and several classes of rendezvous were successfully demonstrated. The extravehicular activities revealed deficiencies in training, and neutral buoyancy simulation was introduced late in the program.

One significant systems engineering achievement emphasized the checkout systems and checkout procedures, and the delivery of flight ready spacecraft. To gain confidence, many of the checkout personnel at the Cape were sent on temporary duty (TDY) to the factory to participate in the factory checkout of the early spacecraft. This approach allowed the ten manned flights to take place on about two-month cycles and contributed immensely to the on-time launches required for rendezvous.

The Apollo Program

The Apollo Program was characterized by a disjointed definition program. Because of the obvious schedule pressures, certain contracts involving Phase B-type effort were let before either the mission design or the lunar landing mode had been selected. For example, the command and service module contract was awarded while questions about the use of Earth orbit rendezvous, lunar orbit rendezvous, and the so-called direct ascent were still being debated. Sufficient pre-Phase A effort was completed to enable a decision to go with the lunar orbit rendezvous route in the spring of 1962, but the Phase A work on the lunar module, even when accomplished in-house on a highly accelerated schedule, did not allow the lunar

module contractor to be selected until nearly a year after the selection of the command and service module contractor. This situation proved to be very distracting to the latter and resulted in major inefficiencies in the contracted effort caused by premature work force buildup.

What saved the situation was the maintenance of simple interfaces between the two spacecraft. In fact, not much more than a docking interface existed; however, there was also an important structural interface recalling that service module propulsion was used to place the docked configuration in lunar orbit. No support was required between the two spacecraft except status monitoring, and no commonality of systems was specified, although by some rationales, this approach appears inefficient. The simple system organizational and programmatic interfaces obtained greatly benefited the program. It was also the approach taken in connection with other elements of Apollo.

The operational phase of the Apollo program provides good illustrations of systems capability extension and mission extensions. The major extensions to the lunar surface stay-time of the lunar module is an example. The decision to accomplish this was made about the time of the first lunar landing, and a Headquarters systems engineering group provided the impetus for the validation. Another capability extension was the addition of the lunar rover contract, awarded about six years after the Apollo start but before the first lunar landing. Both these added capabilities greatly enhanced the lunar surface science and exploration aspects of the Apollo program.

The Skylab Program

The definition activities of the program that ultimately became Skylab proceeded in what must be described as a highly confused state; most of the program objectives and user-oriented program requirements, however, remained stable for the entire duration of the

program. The program first known as Apollo Applications started out as a series of single-mission flights involving a larger number of scientific and technical experiments. This program concept was the basis for approval in the President's budget for FY 1968. About the same time, a command decision was made to incorporate these experiments in a concept known as the "wet workshop," in which a spent upper stage of the Saturn V would be left in orbit, purged, occupied and outfitted to perform the experiments. Many believed the concept could not work, but the program proceeded to preliminary design and, in many cases, detailed design. In the spring of 1969, a decision was made to go to a "dry workshop" wherein all the flight hardware elements would be assembled and checked out on the ground and launched using the first two stages of the Saturn V as the launch vehicle. It took another four years of design and development to bring the program to flight readiness. The flight program was quite successful in the accomplishment of the many experiments. The data obtained from a large solar telescope, for example, the Apollo Telescope Mount (ATM), was regarded as outstanding by the scientists involved. This capability was included in the earliest program requirements.

CONCLUDING REMARKS

This paper has endeavored to highlight the importance of generating top-level program requirements at an early stage in the program evolution or Phase A definition phase. These requirements should include all the factors involved in meeting the program-objective(s) and should be stated with clarity so a determination can be made as to whether they can or are being met. Depending on the nature of the program, these requirements can relate to uses of a capability, a mission objective or other factors, including a simple hardware demonstration such as a test of a new instrument. It is critical to understand whether specific performance

requirements are to be met or only a demonstration of capability is entailed, for the latter provides more flexibility for program adjustments.

The establishment of program requirements usually requires input and involvement of people both inside and outside of the program organization. Determination of just what disciplines are involved is important, particularly for the users and operators.

Validation of the top-level program requirements is a systems engineering function. At the outset, the systems engineering organization works with entities responsible for generating the requirements in an iterative process to assure their validity. This activity continues throughout the life of the program because of unforeseen events that impact the program effort. At times, this will necessitate changes to top-level program requirements. Changes should be under formal change control, and the systems engineering organization operating at the top of the program organization structure should be responsible for the validation effort. Systems engineering is a program-

distributed activity that allows the top-level systems engineering organization to be relatively small because it depends on others for most of the required analysis. It should operate with a low profile.

Past programs serve to illustrate the range of program requirements considerations and the associated systems engineering effort. In the early manned programs, safety was a dominant consideration. Experience in these programs showed that preflight checkout is an important consideration, as is mission design, training, and simulation, all of which can impact the hardware design.

The top-level program requirements and the associated systems engineering activities should obtain and maintain simple interfaces between program elements, even though this produces some apparent program inefficiency. At least one past program, Skylab, has shown that top-level program requirements can be maintained even when considerable fluxing occurs with regard to the hardware and mission design.