MANAGEMENT ISSUES IN SYSTEMS ENGINEERING

by Robert Shishko and Robert G. Chamberlain
with contributions by
Robert Aster, Vincent Bilardo, Kevin Forsberg, Hal Mooz, Lou Polaski and Ron Wade

When applied to a system, the doctrine of successive refinement is a divide-and-conquer strategy. Complex systems are successively divided into pieces that are less complex, until they are simple enough to be conquered. This decomposition results in several structures for describing the *product system* and the *producing system* ("the system that produces the system"). These structures play important roles in systems engineering and project management. Many of the remaining sections in this chapter are devoted to describing some of these key structures.

Structures that describe the product system include, but are not limited to, the requirements tree, system architecture and certain symbolic information such as system drawings, schematics, and data bases. The structures that describe the producing system include the project's work breakdown, schedules, cost accounts and organization. These structures provide different perspectives on their common *raison d'etre*: the desired product system. Creating a fundamental harmony among these structures is essential for successful systems engineering and project management; this harmony needs to be established in some cases by one-to-one correspondence between two structures, and in other cases, by traceable links across several structures. It is useful, at this point, to give some illustrations of this key principle.

System requirements serve two purposes in the systems engineering process. First, they represent a hierarchical description of the buyer's desired product system as understood by the systems engineer. The interaction between the buyer and systems engineer to develop these requirements is one way the "voice of the buyer" is heard. Determining the right requirements — that is, only those that the informed buyer is willing to pay for — is an important part of the systems engineer's job. Second, system requirements also communicate to the design engineers what to design and build (or code). As these requirements are allocated, they become inexorably linked to the system architecture and product breakdown, which consists of the hierarchy of project, systems, segments, elements, subsystems, etc.

The work breakdown structure (WBS) is also a hierarchical structure that contains the pieces of work necessary to complete the project. Each task in the WBS should be traceable to one or more of the system requirements. Schedules, which are structured as networks, describe the time-phased activities that result in the product system in the WBS The cost account structure needs to be directly linked to the work in WBS and the schedules by which that work is done.

The project's organizational structure describes clusters of personnel assigned to perform the work. These organizational structures are usually trees. Sometimes they are represented as a matrix of two interlaced trees; one for line responsibilities, the other for project responsibilities. In any case, the structure should allow identification of responsibility for each WBS task.

Project documentation is the product of particular WBS tasks. There are two fundamental categories of project documentation: baselines and archives. Each category contains information about both the product system and the producing system. The baseline, once established, contains information describing the current state of the product system and producing system resulting from

all decisions that have been made. It is usually organized as a collection of hierarchical tree structures, and should exhibit a significant amount of cross-linking. The archives should contain all of the rest of the project's information that is worth keeping, even if only temporarily. The archives should contain all assumptions, data and supporting analyses that are relevant to past, present and future decisions. Inevitably, the structure (and control) of the archives is much looser than that of the baseline, though cross references should be maintained where feasible.

The structure of reviews (and their associated control gates) reflect the time-phased activities associated with the realization of the product system from its product breakdown. The status reporting and assessment structure provides information on the progress of those same activities. On the financial side, the status reporting and assessment structure should be directly linked to the WBS, schedules and cost accounts. On the technical side, it should be linked to the product breakdown and/or the requirements tree.

MANAGING THE SYSTEMS ENGINEERING PROCESS: THE SYSTEMS ENGINEERING MANAGEMENT PLAN

Systems engineering management is a technical function and discipline that ensures that systems engineering and all other technical functions are properly applied.

Each project should be managed in accordance with a project cycle that is carefully tailored to the project's risks. While the project manager concentrates on managing the overall project cycle, the project-level or lead systems engineer concentrates on managing its technical aspect. This requires that the systems engineer perform (or cause to be performed) the necessary multiple layers of decomposition, definition, integration, verification and validation of the system, while orchestrating and incorporating the appropriate concurrent engineering. Each one of these systems engineering functions requires application of technical analysis skills and tools to achieve the optimum system solution.

The techniques used in systems engineering management include baseline management, requirements traceability, change control, design reviews, audits, document control, failure review boards, control gates and performance certification.

The Project Plan defines how the overall project will be managed to achieve the pre-established requirements within defined programmatic constraints. The Systems Engineering Management Plan (SEMP) is the subordinate document that defines to all project participants how the project will be technically managed within the constraints established by the Project Plan. The SEMP communicates to all participants how they must respond to pre-established management practices. For instance, the SEMP should describe the means for both internal and external (to the project) interface control.

Role of the SEMP

The SEMP is the rule book that describes to all participants how the project will be technically managed. The responsible NASA Center should have a SEMP to describe how it will conduct its technical management, and each contractor should have a SEMP to describe how it will manage in accordance with both its contract and NASA's technical management practices. Since the SEMP is project- and contract-unique, it must be updated for each significant programmatic change or it will become outmoded and unused, and the project could slide into an uncontrolled state. The NASA Center should have its SEMP developed before attempting to prepare a "should-cost" estimate, since activities that incur cost, such as technical risk reduction, need to be identified and described first. The contractor should have its SEMP

developed during the proposal process (prior to costing and pricing) because the SEMP describes the technical content of the project, the potentially costly risk management activities, and the verification and validation techniques to be used, all of which must be included in the preparation of project cost estimates.

The project SEMP is the senior technical management document for the project; all other technical control documents, such as the Interface Control Plan, Change Control Plan, Make-or-Buy Control Plan, Design Review Plan, Technical Audit Plan, etc., depend on the SEMP and must comply with it. The SEMP should be comprehensive and describe how a fully integrated engineering effort will be managed and conducted.

Contents of the SEMP

Since the SEMP describes the project's technical management approach, which is driven by the type of project, the phase in the project cycle, and the technical development risks, it must be specifically written for each project to address these situations and issues. While the specific content of the SEMP is tailored to the project, the recommended content is listed below.

Part I — Technical Program Planning and Control. This section should identify organizational responsibilities and authority for systems engineering management, include control of contracted engineering; levels of control established for performance and design requirements, and the control method used; technical progress assurance methods; plans and schedules for design and technical program reviews; and control of documentation.

This section should describe:

- The role of the project office
- The role of the user
- The role of the Contracting Office Technical Representative (COTR)
- The role of systems engineering
- The role of design engineering
- The role of specialty engineering
- Applicable standards
- Applicable procedures and training
- Baseline control process
- Change control process
- Interface control process
- Control of contracted (or subcontracted) engineering
- Data control process
- Make-or-buy control process
- Parts, materials and process control
- Quality control
- Safety control
- Contamination control
- EMI/EMC
- Technical performance measurement
- Control gates
- Internal technical reviews
- Integration control
- Verification control
- Validation control.

Part II — Systems Engineering Process. This section should contain a detailed description of the process to be used, including the specific tailoring of the process to the requirements of the system and project; the procedures to be used in implementing the process; in-house documentation; the trade study methodology; the types of mathematical and/or simulation models to be used for system cost-effectiveness evaluations; and the generation of specifications.

This section should describe the:

- System decomposition process
- System decomposition format
- System definition process
- System analysis and design process
- Trade study process
- System integration process
- System verification process
- System qualification process
- System acceptance process
- System validation process
- Risk management process

- Life-cycle cost management process
- Use of mathematical models
- Use of simulations
- Specification and drawing structure
- Baseline management process
- Baseline communication process
- Change control process
- Tools to be used.

Part III — Engineering Specialty Integration. This section of the SEMP should describe the integration and coordination of the efforts of the specialty engineering disciplines into the systems engineering process during each iteration of that process. Where there is potential for overlap of specialty efforts, the SEMP should define the relative responsibilities and authorities of each.

This section should contain the project's approach to:

- Concurrent engineering
- The activity phasing of specialty disciplines
- The participation of specialty disciplines
- The involvement of specialty disciplines
- The role and responsibility of specialty disciplines
- The participation of specialty disciplines in system decomposition and definition
- The role of specialty disciplines in verification and validation
- Reliability
- Producibility
- Human engineering
- Maintainability
- Safety
- Survivability/vulnerability
- Integrated logistics
- Quality assurance.

Development of the SEMP

The SEMP must be developed concurrently with the Project Plan. In developing the SEMP, the technical approach to the project, and hence the technical aspect of the project cycle, are developed. This becomes the keel of the project that ultimately determines the length and cost of the project. The development of the programmatic and technical management approaches of the project requires that the key project personnel develop an understanding of the work to be performed and the relationships among the various parts of that work. (See sections on work breakdown structures and network schedules.)

SEMP Lessons Learned from DoD Experience

- A well-managed project requires a coordinated SEMP that is used through the project cycle.
- A SEMP is a living document that must be updated as the project changes and kept consistent with the Project Plan.
- A meaningful SEMP must be the product of experts from all areas of the project.
- Projects with little or insufficient systems engineering discipline generally have major problems.
- Weak systems engineering, or systems engineering placed too low in the organization, cannot perform the functions as required.
- The systems engineering effort must be skillfully managed and well communicated to all the individuals.
- The systems engineering effort must be responsive to both the customer and the contractor interests.

The SEMP's development requires contributions from knowledgeable programmatic and technical experts from all areas of the project that can significantly influence the project's outcome. The involvement of recognized experts is needed to establish a SEMP that is credible to the project manager and to secure the full commitment of the project team.

Managing the Systems Engineering Process: Summary

Systems engineering organizations, and specifically project-level systems engineers, are

responsible for managing projects through the technical aspect of the project cycle. This responsibility includes managing the decomposition and definition sequence, managing the integration, verification and validation sequence and controlling the technical baselines of the project. Typically, these baselines are the functional, "design-to," "build-to" (or "code-to"), "as-built" (or "as-coded"), and "as-deployed." Systems engineering must ensure efficient and logical progression through these baselines.

Systems engineering is responsible for system decomposition and design until the design-to specifications of all lower level configuration items have been produced. Design engineering is then responsible for developing the build-to and code-to documentation that complies with the approved design-to baseline. Systems engineering audits the design and coding process and the design engineering solutions for compliance to all higher level baselines. In performing this responsibility, systems engineering must ensure requirements traceability and document the results in a requirements traceability/verification matrix.

Systems engineering is also responsible for the overall management of the integration, verification and validation process. In this role, systems engineering conducts Test Readiness Reviews and ensures that only verified configuration items are integrated into the next higher assembly for further verification. Verification is continued to the system level, after which system validation is conducted to prove compliance with user requirements.

Systems engineering also ensures that concurrent engineering is properly applied through the project cycle by involving the required specialty engineering. The SEMP is the guiding document for these activities.

THE WORK BREAKDOWN STRUCTURE

A WBS is a hierarchical breakdown of the work necessary to complete a project. The WBS should be a product-based, hierarchical division of deliverable items and associated services. As such, it should contain the project's product breakdown structure (PBS), with the specified prime product(s) at the top, and the systems, segments, subsystems, etc. at successive lower levels. At the lowest level are products such as hardware items, software items and information items (e.g., documents, databases, etc.) for which there is a cognizant engineer or manager. Branch points in the hierarchy should show how the PBS elements are to be integrated. The WBS is built from the PBS by adding, at each branch point of the PBS, any necessary service elements such as management, systems engineering, integration and verification (I&V), and integrated logistics support (ILS). If several WBS elements require similar equipment or software, then a higher level WBS element might be defined to perform a block buy or a development activity (e.g., "System Support Equipment"). Figure 1 shows the relationship between a system, a PBS and a WBS.

A project WBS should be carried down to the cost account level appropriate to the risks to be managed. The appropriate level of detail for a cost account is determined by management's desire to have visibility into costs, balanced against the cost of planning and reporting. Contractors may have a Contract WBS (CWBS), which is appropriate to the contractor's needs to control costs. A summary CWBS, consisting of the upper levels of the full CWBS, is usually included in the project WBS to report costs to the contracting agency.

WBS elements should be identified by title and by a numbering system that performs the following functions:

- Identifies the level of the WBS element
- Identifies the higher level element into which the WBS element will be integrated
- Shows the cost account number of the element.

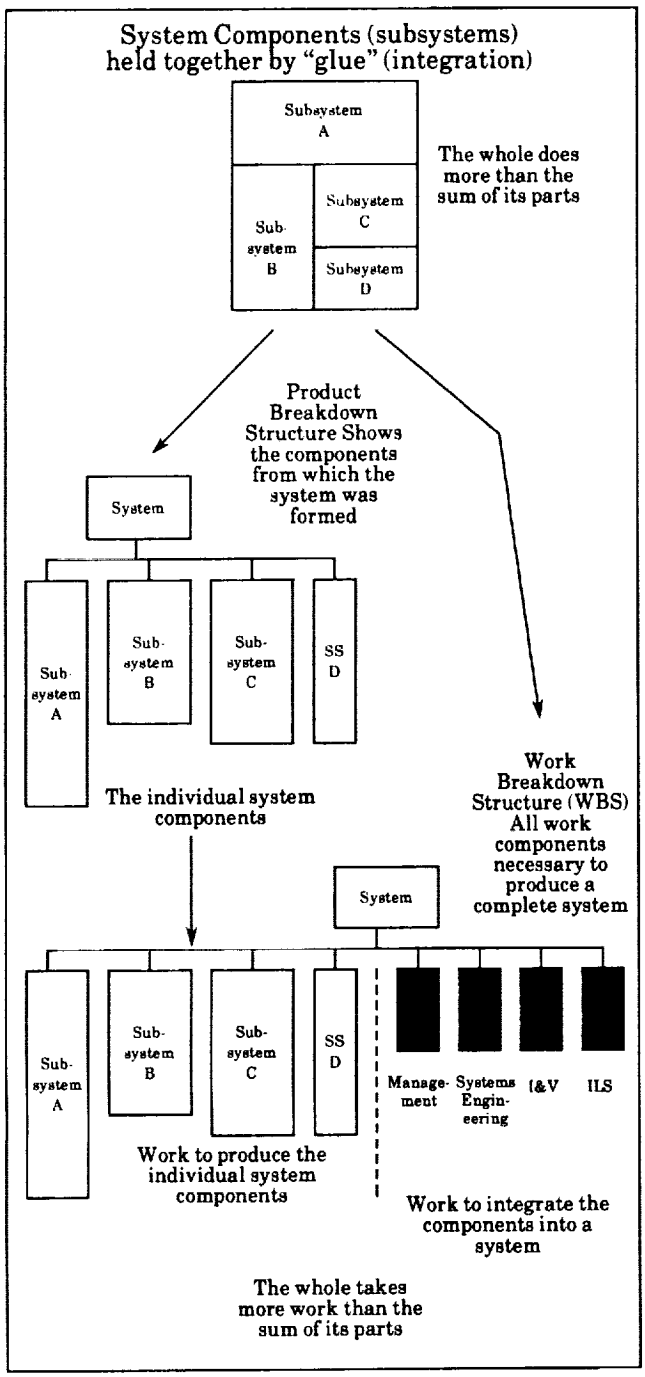


Figure 1 The Relationship between a System, a Product Breakdown Structure, and a Work Breakdown Structure

A WBS should also have a companion WBS dictionary that contains each element's title, identification number, objective, description, and any dependencies (e.g., receivables) on other WBS elements. This dictionary provides a structured project description that is valuable for orienting project members and other interested parties. It fully describes the products and/or services expected from each WBS element.

This section provides some techniques for developing a WBS, and points out some mistakes to avoid.

Role of the WBS

A product-based WBS is the organizing structure for:

- Project and technical planning and scheduling
- Cost estimation and budget formulation (In particular, costs collected in a product-based WBS can be compared to historical data. This is identified as a primary objective by DoD standards for WBSs.)
- Defining the scope of statements of work and specifications for contract efforts
- Project status reporting, including schedule, cost and work force, technical performance, integrated cost/schedule data (such as earned value and estimated cost at completion)
- Plans, such as the SEMP, and other documentation products, such as specifications and drawings.

It provides a logical outline and vocabulary that describes the entire project and integrates information in a consistent way. If there is a schedule slip in one element of a WBS, an observer can determine which other WBS elements are most likely to be affected. Cost impacts are more accurately estimated. If there is a design change in one element of the WBS, an observer can determine which other WBS elements will most likely be affected, and these elements can be consulted for potential adverse impacts.

Techniques for Developing the WBS

Developing a successful project WBS is likely to require several iterations through the project cycle since it is not always obvious at

the outset what the full extent of the work may be. Prior to developing a preliminary WBS, there should be some development of the system architecture to the point where a preliminary PBS can be created.

The PBS and associated WBS can then be developed level by level from the top down. In this approach, a project-level systems engineer finalizes the PBS at the project level, and provides a draft PBS for the next lower level. The WBS is then derived by adding appropriate services such as management and systems engineering to that lower level. This recursive process is repeated until a WBS exists down to the desired cost account level.

An alternative approach is to define all levels of a complete PBS in one design activity, and then develop the complete WBS. When this approach is taken, it is necessary to take great care to develop the PBS so that all products are included, and all assembly/integration and verification branches are correct. The involvement of people who will be responsible for the lower level WBS elements is recommended.

A WBS for a Multiple Delivery Project. Some of the terms for projects that provide multiple deliveries, are "rapid development," "rapid prototyping" and "incremental delivery." Such projects should also have a product-based WBS, but there will be one extra level in the WBS hierarchy immediately under the final prime product(s) that identifies each delivery. At any point in time there will be both active and inactive elements in the WBS.

A WBS for an Operational Facility. A WBS for managing an operational facility such as a flight operations center is analogous to a WBS for developing a system. The difference is that the products in the PBS are not necessarily completed once and then integrated, but are all produced on a routine basis. A PBS for an operational facility might consist of information products or service products provided to external customers. However, the general concept of a hierarchical breakdown of products and/or services would still apply.

The rules that apply to a development WBS also apply to a WBS for an operational facility. The techniques for developing a WBS for an operational facility are the same, except that services such as maintenance and user support are added to the PBS, and services such as systems engineering, integration and verification may not be needed.

Common Errors in Developing a WBS

There are three common errors found in WBSs:

Error 1: The WBS describes functions, not products. This makes the project manager the only one formally responsible for products.

Error 2: The WBS has branch points that are not consistent with how the WBS elements will be integrated. For instance, in a flight operations system with a distributed architecture, there is typically software associated with hardware items that will be integrated and verified at lower levels of a WBS. It would then be inappropriate to separate hardware and software as if they were separate systems to be integrated at the system level. This would make it difficult to assign accountability for integration and to identify the costs of integrating and testing components of a system.

Error 3: The WBS is inconsistent with the PBS. This makes it possible that the PBS will not be fully implemented, and generally complicates the management process.

Some examples of these errors are shown in Figure 2. Each one prevents the WBS from successfully performing its roles in project planning and organizing. These errors are avoided by using the WBS development techniques described above.

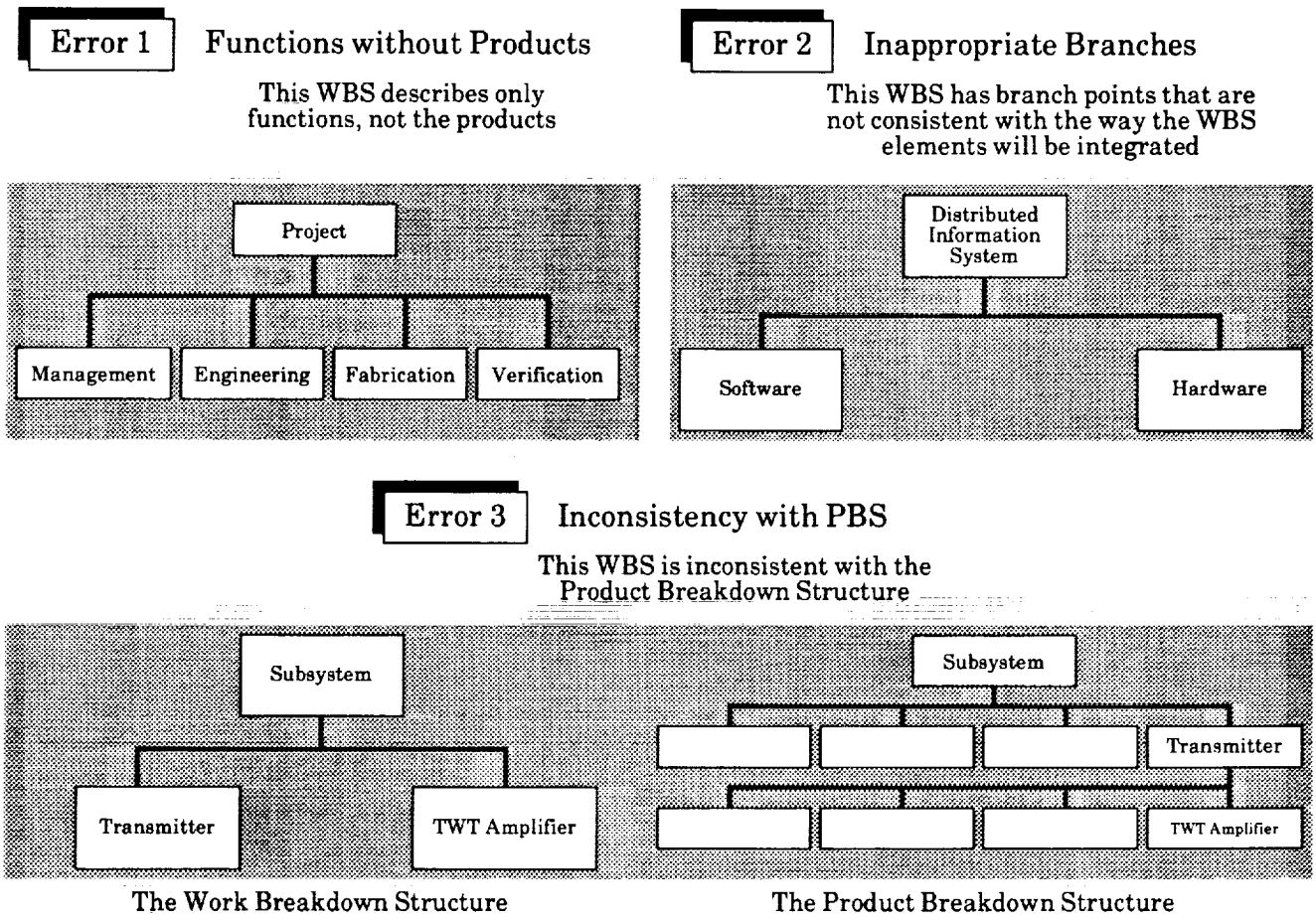


Figure 2 Examples of WBS Development Errors

NETWORK SCHEDULING

Products described in the WBS are the result of activities that take time to complete. An orderly and efficient systems engineering process requires that these activities take place in a way that respects the underlying time-precedence relationships among them. This is accomplished by creating a network schedule, which explicitly takes into account the dependencies of each activity on other activities and receivables from outside sources. This section discusses the role of scheduling and the techniques for building a complete network schedule.

Scheduling is an essential component of planning and managing the activities of a project. The process of creating a network schedule can lead to a much better understanding of what needs to be done, how long it will take, and how each element of the project WBS might affect other elements. A complete network schedule can be used to calculate how long it will take to complete a project, which activities determine that duration (i.e., critical path activities), and how much spare time (i.e., float) exists for all the other activities of the project. An understanding of the project's schedule is a prerequisite for accurate project budgeting.

Keeping track of schedule progress is an essential part of controlling the project, because cost and technical problems often show up first as schedule problems. Because network schedules show how each activity affects other activities, they are essential for predicting the consequences of schedule slips or accelerations of an activity on the entire project. Network scheduling systems also help managers accurately assess the impact

Critical Path and Float Calculation

The *critical path* is the sequence of activities that will take the longest to accomplish. Activities that are not on the critical path have a certain amount of time that they can be delayed until they, too are on a critical path. This time is called *float*. There are two types of float, path float and free float. Path float is where a sequence of activities collectively have float. If there is a delay in an activity in this sequence, then the path float for all subsequent activities is reduced by that amount. Free float exists when a delay in an activity will have no effect on any other activity. For example, if activity A can be finished in 2 days, and activity B requires 5 days, and activity C requires completion of both A and B, then A would have 3 days of free float.

Float is valuable. Path float should be conserved where possible, so that a reserve exists for future activities. Conservation is much less important for free float.

To determine the critical path, there is first a "forward pass" where the earliest start time of each activity is calculated. The time when the last activity can be completed becomes the end point for that schedule. Then there is a "backward pass," where the latest possible start point of each activity is calculated, assuming that the last activity ends at the end point previously calculated. Float is the time difference between the earliest start time and the latest start time of an activity. Whenever this is zero, that activity is on a critical path.

of both technical and resource changes on the cost and schedule of a project.

Network Schedule Data and Graphical Formats

Network schedule data consist of:

- Activities
- Dependencies between activities (e.g., where an activity depends upon another activity for a receivable)
- Products or milestones that occur as a result of one or more activities
- Duration of each activity.

A *work flow diagram* (WFD) is a graphical display of the first three data items above. A network schedule contains all four data items. When creating a network schedule, graphical formats of these data are very useful. Two general types of graphical formats, shown in Figure 3, are used. One has *activities-on-arrows*, with products and dependencies at the beginning and end of the arrow. This is the typical format of the Program Evaluation and Review Technique (PERT) chart. The second called *precedence diagrams*, has boxes that represent activities; dependencies are then shown by arrows. Due to its simpler visual format and reduced requirements on computer resources, the precedence diagram has become more common in recent years.

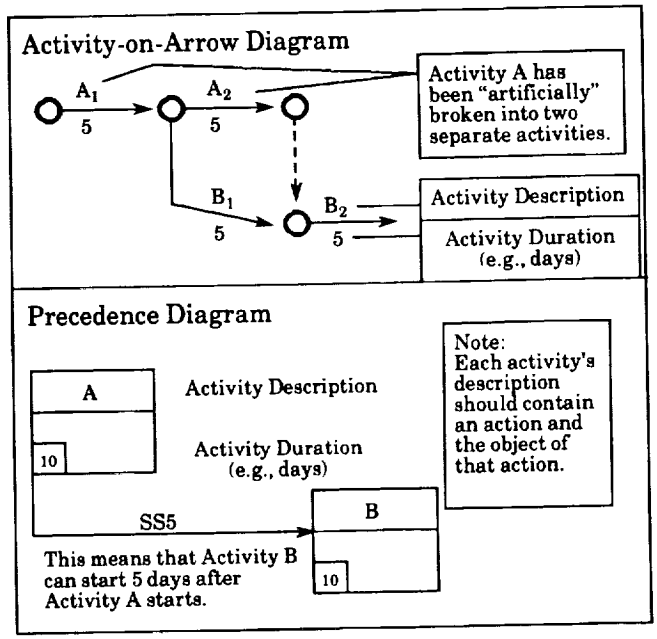


Figure 3 Activity-on-Arrow and Precedence Diagrams for Network Schedules

The precedence diagram format allows for simple depiction of the following logical relationships:

- Activity B begins when Activity A begins (Start-Start, or SS)
- Activity B begins only after Activity A ends (Finish-Start, or FS)

- Activity B ends when Activity A ends (Finish-Finish, or FF).

Each of these three activity relationships may be modified by attaching a lag (+ or −) to the relationship, as shown in Figure 3.

It is possible to summarize a number of low-level activities in a precedence diagram with a single activity. This is commonly referred to as hammocking. One takes the initial low-level activity and attaches a summary activity to it using the first relationship described above. The summary activity is then attached to the final low-level activity using the third relationship described above. Unless one is hammocking, the most common relationship used in precedence diagrams is the second one mentioned above. The activity-on-arrow format can represent the identical time-precedence logic as a precedence diagram by creating artificial events and activities as needed.

Establishing a Network Schedule

Scheduling begins with project-level schedule objectives for delivering the products described in the upper levels of the WBS. To develop network schedules that are consistent with the project's objectives, the following six steps are applied to each cost account at the lowest available level of the WBS.

Step 1: Identify activities and dependencies needed to complete each WBS element. Enough activities should be identified to show exact schedule dependencies between activities and other WBS elements. It is not uncommon to have about 100 activities identified for the first year of a WBS element that will require 10 work-years per year. Typically, there is more schedule detail for the current year, and much less detail for subsequent years. Each year, schedules are updated with additional detail for the current year. This first step is most easily accomplished by:

- Ensuring that the cost account WBS is extended downward to describe all significant products, including documents, reports, hardware and software items.
- For each product, listing the steps required for its generation and drawing the process as a work flow diagram.
- Indicating the dependencies among the products, and any integration and verification steps within the work package.

Step 2: Identify and negotiate external dependencies. External dependencies are any receivables from outside of the cost account, and any deliverables that go outside of the cost account. Informal negotiations should occur to ensure that there is agreement with respect to the content, format and labeling of products that move across cost account boundaries. This step is designed to ensure that lower level schedules can be integrated.

Step 3: Estimate durations of all activities. Assumptions behind these estimates (work force, availability of facilities, etc.) should be written down for future reference.

Step 4: Enter the schedule data for the WBS element into a suitable computer program to obtain a network schedule and an estimate of the critical path for that element. (There are many commercially available software packages for this function.) This step enables the cognizant engineer, team leader, and/or systems engineer to review the schedule logic. It is not unusual at this point for some iteration of steps one to four to be required in order to obtain a satisfactory schedule. Reserve will also be added to critical path activities, often in the form of a dummy activity, to ensure that schedule commitments can be met for this WBS element.

Step 5: Integrate schedules of lower level WBS elements, using suitable software, so that all dependencies between WBS elements are correctly included in a project

network. It is important to include the impacts of holidays, weekends, etc., at this point. The critical path for the project is discovered at this step in the process.

Step 6: Review the work force level and funding profile over time, and make final adjustments to logic and durations so that work force levels and funding levels are reasonable. Adjustments to the logic and the durations of activities may be needed to conform to the schedule targets established at the project-level. This may include adding more activities to some WBS element, deleting redundant activities, increasing the work force for some activities that are on the critical path, or finding ways to do more activities in parallel, rather than in series. If necessary, the project-level targets may need to be adjusted, or the scope of the project may need to be reviewed. Again, it is good practice to have some schedule reserve, or float, as part of a risk mitigation strategy.

The product of these last steps is a feasible baseline schedule for each WBS element that is consistent with the activities of all other WBS elements, and the sum of all these schedules is consistent with both the technical scope and the schedule goals for the project. There should be enough float in this integrated master schedule so that schedule and associated cost risk are acceptable to the project and to the project's customer. Even when this is done, time estimates for many WBS elements will have been underestimated, or work on some WBS elements will not start as early as had been originally assumed due to late arrival of receivables. Consequently, replanning is almost always needed to meet the project's goals.

Reporting Techniques

Summary data about a schedule is usually described in Gantt charts, a good example of which is shown in Figure 4. Another type of output format is a table that shows the float and recent changes in float of key activities. For example, a project manager may wish to know precisely how much schedule reserve has been consumed by critical path activities, and whether reserves are being consumed or are being preserved in the latest reporting period. This table provides information on the rate of change of schedule reserve.

Good scheduling systems provide capabilities to show resource requirements over time, and to make adjustments so that the schedule is feasible with respect to resource constraints over time. Resources may include work force level, funding profiles, important facilities, etc. Figure 5 shows an example of an unleveled resource profile. The objective is to move the start dates of tasks that have float to points where

Desirable Features in Gantt Charts

The Gantt chart shown in Figure 4 illustrates the following desirable features:

- A heading that describes the WBS element, the responsible manager, the date of the baseline used, and the date that status was reported.
- A milestone section in the main body (lines 1 and 2).
- An activity section in the main body. Activity data:
  a. WBS elements (lines 3, 5, 8, 12, 16 and 20)
  b. Activities (indented from WBS elements)
  c. Current plan (shown as thick bars)
  d. Baseline plan (same as current plan, or if different, represented by thin bars under the thick bars)
  e. Status line at the appropriate date
  f. Slack for each activity (dashed lines above the current plan bars)
  g. Schedule slips from the baseline (dashed lines below the milestone on line 12)
- A note section, where the symbols in the main body can be explained.

This Gantt chart shows only 23 lines, which is a summary of the activities currently being worked for this WBS element. It is appropriate to tailor the amount of detail to those items most pertinent at the time of status reporting.

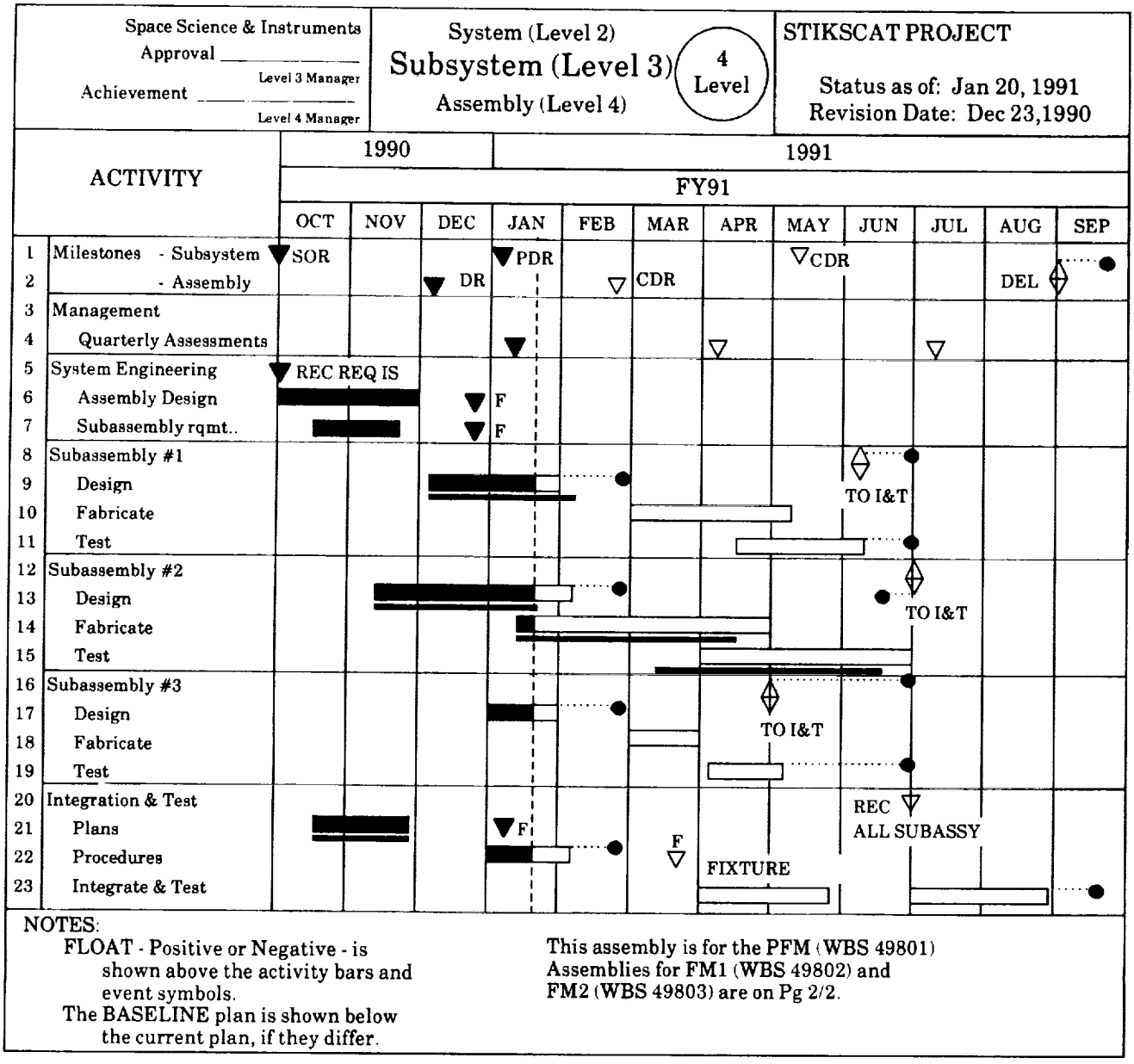


Figure 4 An Example of a Gantt Chart

the resource profile is feasible. If that is not sufficient, then the assumed task durations for resource-intensive activities should be re-examined and, accordingly, the resource levels changed.

BUDGETING AND RESOURCE PLANNING

Budgeting and resource planning involves the establishment of a reasonable project baseline budget and the capability to analyze changes to that baseline resulting from technical and/or schedule changes. The project's WBS, baseline schedule and budget should be viewed by the systems engineer as mutually dependent, reflecting the technical content, time, and cost of meeting the project's goals and objectives.

The budgeting process needs to take into account whether a fixed cost cap or cost profile exists. When no such cap or profile exists, a baseline budget is developed from

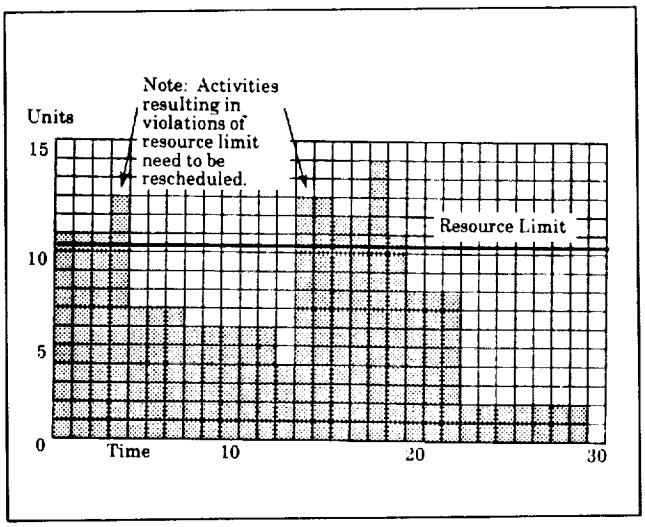


Figure 5 An Example of an Unleveled Resource Profile

the WBS and network schedule. This specifically involves combining the project's work force and other resource needs with the appropriate work force rates and other financial and programmatic factors to obtain cost element estimates. These elements of cost include:

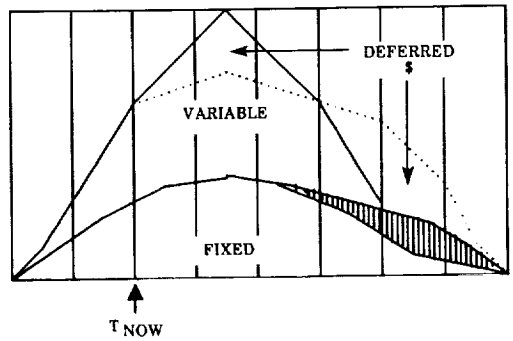
- Direct labor costs
- Overhead costs
- Other direct costs (travel, data processing, etc.)
- Subcontract costs
- Material costs
- General and administrative costs
- Cost of money (i.e., interest payments, if applicable)
- Fee (if applicable)
- Contingency

When there is a cost cap or a fixed cost profile, there are additional logic gates that must be satisfied before the systems engineer can complete the budgeting and planning process. A determination needs to be made whether the WBS and network schedule are feasible with respect to mandated cost caps and/or cost profiles. If not, the systems engineer needs to recommend the best approaches for either stretching out a project (usually at an increase in the total cost) or descoping the project's goals and objectives, requirements, design, and/or implementation approach.

Whether a cost cap or fixed cost profile exists, it is important to control costs after they have been baselined. An important aspect of cost control is project cost and schedule status reporting and assessment. Another is cost and schedule risk planning, such as developing risk avoidance and workaround strategies. At the project level, budgeting and resource planning must also ensure that an adequate level of contingency funds are included to deal with unforeseen events.

Assessing the Effect of Schedule Slippage

Certain elements of cost, called fixed costs, are mainly time related, while others, called variable costs, are mainly product related. If a project's schedule is slipped, then the fixed costs of completing it increase. The variable costs remain the same in total (excluding inflation adjustments), but are deferred downstream, as in the figure below.



To quickly assess the effect of a simple schedule slippage:

- Convert baseline budget plan from nominal (real-year) dollars to constant dollars
- Divide baseline budget plan into fixed and variable costs
- Enter schedule slip implementation
- Compute new variable costs including any work force disruption costs
- Repeat last two steps until an acceptable implementation is achieved
- Compute new fixed costs
- Sum new fixed and variable costs
- Convert from constant dollars to nominal (real-years) dollars.

RISK MANAGEMENT

Risk management comprises purposeful thought to the sources, magnitude and mitigation of risk, and actions directed toward its balanced reduction. As such, risk management is an integral part of project management, and contributes directly to the objectives of systems engineering.

> *Risk*
>
> The term risk has different meanings depending on the context. Sometimes it simply indicates the degree of variability in the outcome or result of a particular action. In the context of risk management during the systems engineering process, the term denotes a combination of both the likelihood of various outcomes and their distinct consequences. The focus, moreover, is generally on undesired or unfavorable outcomes such as the risk of a technical failure, or the risk of exceeding a cost target.

NASA policy objectives with regard to project risks are expressed in NMI 8070.4A, *Risk Management Policy*. These are to:

- Provide a disciplined and documented approach to risk management throughout the project cycle
- Support management decision making by providing integrated risk assessments (i.e., taking into account cost, schedule, performance and safety concerns)
- Communicate to NASA management the significance of assessed risk levels and the decisions made with respect to them.

There are a number of actions the systems engineer can take to effect these objectives. Principal among them is planning and completing a well-conceived *risk management program*. Such a program encompasses several related activities during the systems engineering process. The structure of these activities is shown in Figure 6.

The first is the process of identifying and characterizing the project's risks. The objective of this step is to understand what uncertainties the project faces, and which among them should be given greater attention. This is accomplished by categorizing (in a consistent manner) uncertainties by the likelihood of occurrence (e.g., high, medium, or low), and separately, according to severity of consequences. This categorization forms the basis for ranking uncertainties by their relative riskiness. Uncertainties with both high likelihood and severely adverse consequences are ranked higher than those without these characteristics. The primary methods used in this process are qualitative; hence, in systems engineering literature, this step is sometimes called qualitative risk assessment. The output of this step is a list of significant risks (by phase) to be given specific management attention.

In some projects, qualitative methods are adequate for making risk management decisions; in others, these methods are not precise enough to elucidate the magnitude of the problem, or to allocate scarce risk reduction resources. Risk analysis is the process of quantifying both the likelihood of occurrence and consequences of potential future events (or "states of nature" in some texts). The

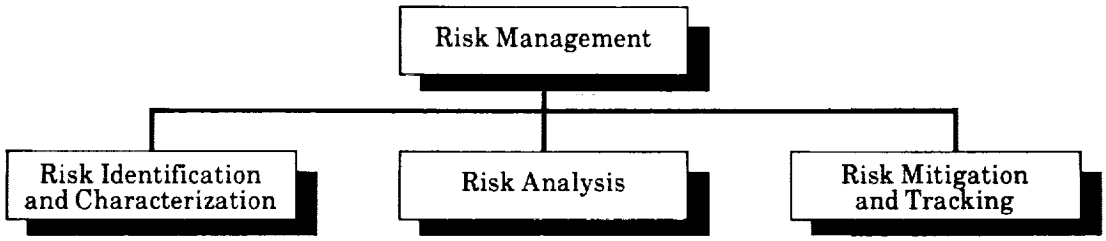


Figure 6 Risk Management Structure

systems engineer needs to decide whether risk identification and characterization are adequate, or whether the increased precision of risk analysis is needed for some uncertainties. In making that determination, the systems engineer needs to balance the (usually) higher cost of risk analysis against the value of the additional information.

Risk mitigation is the formulation, selection and execution of strategies designed to economically reduce risk. Tracking the effectivity of these strategies is also considered part of risk mitigation. Risk mitigation is often a challenge because efforts and expenditures to reduce one type of risk may increase another type. (Some have called this the systems engineering equivalent of the Heisenberg Uncertainty Principle in quantum mechanics). The ability (or necessity) to trade one type of risk for another means that the project manager and the systems engineer need to understand the system-wide effects of various strategies in order to make a rational allocation of resources.

Several techniques have been developed for each of these risk management activities. The principal ones are shown in Table 1. The systems engineer needs to choose the techniques that best fit the unique requirements of each project.

A risk management program is needed throughout the project cycle. In keeping with the doctrine of successive refinement, its focus, however, moves from the "big picture" in the early phases of the project cycle (Phases A and B) to more specific issues during product design and development (Phases C and D). During pre-operations and operations (Phases E and F), the focus changes again. A good risk management program is always forward-looking. In other words, a risk management program should address the project's ongoing risk issues and future uncertainties. As such, it is a natural part of concurrent engineering.

Risk management activities for a project should be documented in a risk management program plan. That plan, which elaborates

Risk Identification and Characterization	Risk Analysis	Risk Mitigation and Tracking
Expert interviews	Decision analysis	Watchlists/ milestones
Independent assessment (cost, schedule and technical)	Probabilistic Risk Assessment (PRA)	Contingency planning
Risk templates (e.g., DoD 4245.7-M)	Probabilistic network schedules (e.g., PERT)	Critical items/issues lists
Lessons learned files from previous projects	Probabilistic cost and effectiveness models (e.g., Monte Carlo models)	Cost/schedule control systems and Technical Performance Measure (TPM) tracking
FMECAs/ FMEAs/ Digraphs		

Table 1 Techniques of Risk Management

on the SEMP and should be updated at each phase of the project cycle, contains:

- The project's overall risk policy and objectives
- The programmatic aspects of the risk management activities (i.e., responsibilities, resources, schedules and milestones, etc.)
- A description of the tools and techniques to be used for risk identification and characterization, risk analysis, and risk mitigation
- A description of the role of risk management with respect to systems analysis, baseline change control, formal reviews, and status reporting and assessment
- Documentation requirements for each risk management product and action.

The level of risk management activities should be consistent with the project's overall risk policy established in conjunction with its NASA Headquarters program office. At present, formal guidelines for the

classification of projects with respect to overall risk policy do not exist; such guidelines exist only for NASA payloads. These are promulgated in NMI 8010.1A, *Classification of NASA Payloads, Attachment A*.

Types of Risks

There are several ways to describe the various types of risk a project manager/systems engineer faces. Traditionally, project managers and systems engineers have attempted to divide risks into three or four broad categories namely, cost, schedule, technical, and sometimes, safety (and/or hazard) risks. More recently, others have entered the lexicon, including the categories of organizational, management, acquisition, supportability, political and programmatic risks. These newer categories reflect the expanded set of concerns of project managers and systems engineers who must operate in the current NASA environment. Some of these newer categories also represent supersets of other categories. For example, the Defense Systems Management College (DSMC) Systems Engineering Management Guide wraps "funding, schedule, contract relations, and political risks" into the broader category of programmatic risks. While these terms are useful in informal discussions, there appears to be no formal taxonomy free of ambiguities. One reason, mentioned above, is that often one type of risk can be exchanged for another. A second reason is that some of these categories move together, as for example, cost risk and political risk (e.g., the risk of project cancellation).

Another way some have categorized risk is by the degree of mathematical predictability in its underlying uncertainty. The distinction has been made between an uncertainty that has a known probability distribution, with known or estimated parameters, and one in which the underlying probability distribution is either not known, or its parameters cannot be objectively quantified.

An example of the first kind of uncertainty occurs in the unpredictability of the spares upmass requirement for alternative Space Station Freedom designs. While the requirement is stochastic in any particular logistics cycle, the probability distribution can be estimated for each design from reliability theory and empirical data. Examples of the second kind of uncertainty occur in trying to predict whether a Shuttle accident will make resupply of Freedom impossible for a period of time greater than x months, or whether life on Mars exists.

Modern subjectivist (also known as Bayesian) probability theory holds that the probability of an event is the degree of belief that a person has that it will occur, given his/her state of information. As that information improves (e.g., through the acquisition of data or experience), the subjectivist's estimate of a probability should converge to that estimated as if the probability distribution were known. In the examples of the previous paragraph, the only difference, then, is the probability estimator's perceived state of information. Consequently, subjectivists find the distinction between the two kinds of uncertainty of little or no practical significance. The implication of the subjectivist's view for risk management is that, even with little or no data, the systems engineer's subjective probability estimates form a valid basis for risk decision making.

Risk Identification and Characterization Techniques

A variety of techniques is available for risk identification and characterization. The thoroughness with which this step is accomplished is an important determinant of the risk management program's success.

Expert Interviews. When properly conducted, expert interviews can be a major source of insight and information on the project's risks in the expert's area of knowledge. One key to a successful interview is in

identifying an expert who is close enough to a risk issue to understand it thoroughly, and at the same time, able (and willing) to step back and take an objective view of the probabilities and consequences. A second key to success is advanced preparation on the part of the interviewer. This means having a list of risk issues to be covered in the interview, developing a working knowledge of these issues as they apply to the project, and developing methods for capturing the information acquired during the interview.

Initial interviews may yield only qualitative information, which should be verified in follow-up rounds. Expert interviews are also used to solicit quantitative data and information for those risk issues that qualitatively rank high. These interviews are often the major source of inputs to risk analysis models built using the techniques described later.

Independent Assessment. This technique can take several forms. In one form, it can be a review of project documentation, such as statements of work, acquisition plans, verification plans, manufacturing plans and the SEMP. In another form, it can be an evaluation of the WBS for completeness and consistency with the project's schedules. In a third form, an independent assessment can be an independent cost (and/or schedule) estimate from an outside agency and/or group.

Risk Templates. This technique consists of examining and then applying a series of previously developed risk templates to a current project. Each template generally covers a particular risk issue, and then describes methods for avoiding or reducing that risk. The most widely recognized series of templates appears in DoD 4245.7M, *Transition from Development to Production . . . Solving the Risk Equation.* Many of the risks and risk responses described are based on lessons learned from DoD programs, but are general enough to be useful to NASA projects. As a general caution, risk templates cannot provide an exhaustive list of risk issues for every project, but they are a useful input to risk identification.

Lessons Learned. A review of the lessons learned files, data and reports from previous similar projects can produce insights and information for risk identification on a new project. For technical risk identification, as an example, it makes sense to examine previous projects of similar function, architecture or technological approach. The lessons learned from the *Infrared Astronomical Satellite* (IRAS) project might be useful to the *Space Infrared Telescope Facility* (SIRTF) project, even though the latter's degree of complexity is significantly greater. The key to applying this technique is in recognizing what aspects are analogous in two projects, and what data are relevant to the new project. Even if the the documented lessons learned from previous projects are not applicable at the system level, there may be valuable data applicable at the subsystem or component level.

FMECAs, FMEAs and Digraphs. Failure Modes, Effects, and Criticality Analysis (FMECA), Failure Modes and Effects Analysis (FMEA) and digraphs are specialized techniques for safety (and/or hazard) risk identification and characterization. These techniques focus on the hardware components that make up the system. According to MIL-STD-1629A, FMECA is "an ongoing procedure by which each potential failure in a system is analyzed to determine the results or effects thereof on the system, and to classify each potential failure mode according to its severity." Failures are generally classified into four severity categories:

- Category I - Catastrophic Failure (possible death or system loss)
- Category II - Critical Failure (possible major injury or system damage)

- Category III - Major Failure (possible minor injury or mission effectiveness degradation)
- Category IV - Minor Failure (requires system maintenance, but does not pose a hazard to personnel or mission effectiveness).

A complete FMECA also includes an estimate of the probability of each potential failure. These probabilities are usually based, at first, on subjective judgment or experience factors from similar kinds of hardware components, but may be refined from reliability data as the system development progresses. An FMEA is similar to an FMECA, but typically excludes the severity classification category.

Digraph analysis is an aid in determining fault tolerance, propagation and reliability in large, interconnected systems. Digraphs exhibit a network structure and resemble a schematic diagram. The digraph technique permits the integration of data from a number of individual FMECAs/FMEAs, and can be translated into fault trees, described below, if quantitative probability estimates are needed.

Risk Analysis Techniques

The tools and techniques of risk analysis rely heavily on the concept and "laws" (actually, axioms and theorems) of probability. The systems engineer needs to be familiar with these in order to appreciate the full power and limitations of these techniques. The products of risk analyses are generally quantitative probability and consequence estimates for various outcomes, more detailed understanding of the dominant risks, and improved capability for allocating risk reduction resources.

Decision Analysis. Decision analysis is one technique to help the individual decision maker deal with a complex set of uncertainties. Using the divide-and-conquer approach common to much of systems engineering, a complex uncertainty is decomposed into simpler ones, which are then treated separately. The decomposition continues until it reaches a level at which either hard information can be brought to bear, or intuition can function effectively. The decomposition can be graphically represented as a decision tree. The branch points, called nodes, in a decision tree represent either decision points or chance events. Endpoints of the tree are the potential outcomes.

In most applications of decision analysis, these outcomes are generally assigned dollar values. From the probabilities assigned at each chance node, and the dollar value of each outcome, the distribution of dollar values (i.e., consequences) can be derived for each set of decisions. Even large, complex decision trees can be represented in currently available decision analysis software. This software can also calculate a variety of risk measures.

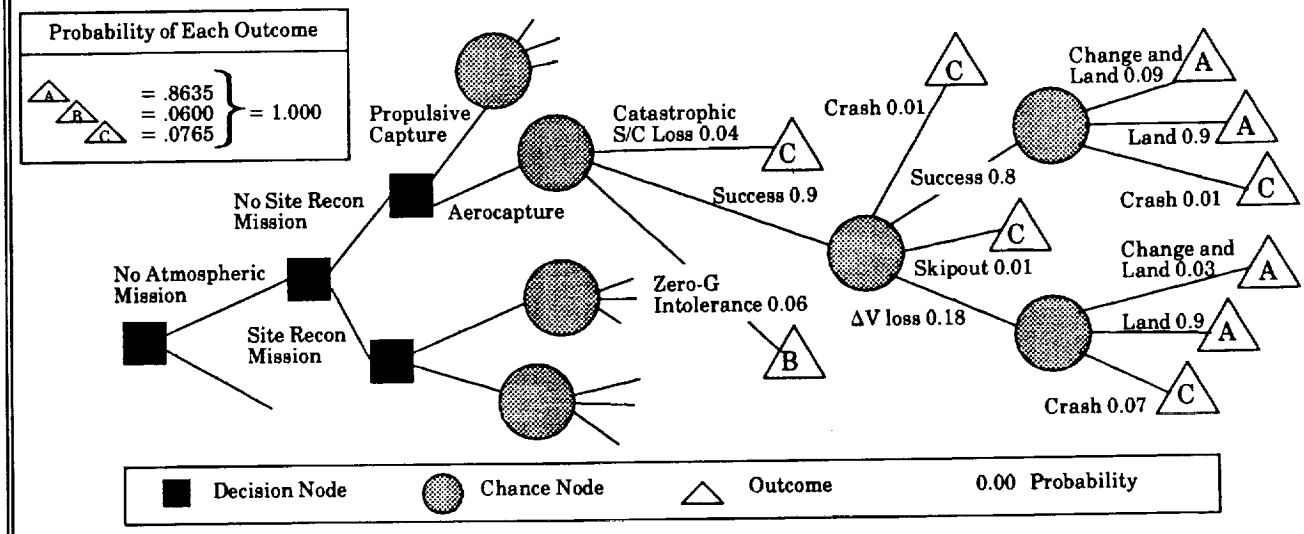
In brief, decision analysis is a technique that allows:

- A systematic enumeration of uncertainties and encoding of their probabilities and outcomes
- An explicit characterization of the decision maker's attitude toward risk, expressed in terms of his/her risk aversion
- A calculation of the value of "perfect information," thus setting a normative upper bound on information-gathering expenditures
- Sensitivity testing on probability estimates and outcome dollar values.

Probabilistic Risk Assessment (PRA). A PRA seeks to measure the risk inherent in a system's design and operation by quantifying both the likelihood of various possible accident sequences and their consequences. A typical PRA application is to determine the risk associated with a specific nuclear power plant. Within NASA, PRAs are used to demonstrate, for example, the relative

An Example of a Decision Tree for Robotic Precursor Missions to Mars

In 1990, the Lunar/Mars Exploration Program Office (LMEPO) at JSC wanted to know how robotic precursor missions might reduce the risk of a manned Mars mission. Structuring the problem as a decision tree allows the effects of different missions and chance events to be systematically and quantitatively evaluated. The portion of the decision tree shown here illustrates the calculation of the probabilities for three distinct outcomes: (A) a successful Mars landing, (B) a safe return without a landing, or (C) a disaster resulting in mission and crew loss, when no atmospheric or site reconnaissance robotic precursor missions were made and aerocapture at Mars was selected. As new information becomes available, the decision tree's data can be reviewed and updated.



Making the same calculations for every branch in the decision tree allows a determination of the best mix of robotic precursor missions as an explicit function of: (a) the contribution of each robotic precursor mission to manned mission risk reduction; (b) the cost, schedule and riskiness of each robotic mission; (c) the value of the manned mission; and (d) the science value of each robotic mission in the absence of a subsequent manned mission. Another benefit of this quantitative approach is that robotic precursors can be traded against other risk mitigation strategies in the manned mission architecture.

For more information on decision analysis, see de Neufville and Stafford, *Systems Analysis for Engineers and Managers*, 1971, and Barclay, et al., *Handbook for Decision Analysis*, 1977.

safety of launching spacecraft containing RTGs (Radioisotope Thermoelectric Generators).

The search for accident sequences is facilitated by event trees, which depict initiating events and combinations of system successes and failures, and fault trees, which depict ways in which the system failures represented in an event tree can occur. When integrated, an event tree and its associated fault tree(s) can be used to calculate the probability of each accident sequence. The structure and mathematics of these trees is similar to that for decision trees. The consequences of each accident sequence are generally measured both in terms of direct economic losses and in public health effects.

Doing a PRA is itself a major effort, requiring a number of specialized skills other than those provided by reliability engineers and human factors engineers. PRAs also require large amounts of system design data at the component level and operational procedures data. [For additional information on PRAs, refer to the *PRA Procedures Guide* (1983) by the American Nuclear Society and Institute of Electrical and Electronic Engineers (IEEE).]

Probabilistic Risk Assessment Pitfalls

Risk is generally defined in a probabilistic risk assessment (PRA) as the expected value of a consequence function — that is:

$$R = \Sigma_s P_s C_s$$

where P_s is the probability of outcome s, and C_s is the consequence of outcomes. To attach probabilities to outcomes, event trees and fault trees are developed. These techniques have been used since 1953, but by the late 1970s, they were under attack by PRA practitioners. The reasons include the following:

- Fault trees are limiting because a complete set of failures is not definable
- Common cause failures could not be captured properly. An example of a common cause failure is one where all the valves in a system have a defect so that their failures are not truly independent
- PRA results are sometimes sensitive to simple changes in event tree assumptions
- Stated criteria for accepting different kinds of risks are often inconsistent, and therefore not appropriate for allocating risk reduction resources
- Many risk-related decisions are driven by perceptions, not necessarily objective risk as defined by the above equation. Perceptions of consequences tend to grow faster than the consequences themselves — that is, several small accidents are not perceived as strongly as one large one, even if fatalities are identical
- There are difficulties in dealing with incommensurables, as for example, lives vs. dollars.

Probabilistic Network Schedules. Probabilistic network schedules, such as PERT (Program Evaluation and Review Technique), permit the duration of each activity to be treated as a random variable. By supplying PERT with the minimum, maximum and most likely duration for each activity, a probability distribution can be computed for project completion time. This can then be used to determine, for example, the chances that a project (or any set of tasks in the network) will be completed by a given date. In this probabilistic setting, however, a unique critical path may not exist. Some practitioners have also cited difficulties in obtaining meaningful input data for probabilistic network schedules.

Probabilistic Cost and Effectiveness Models. These models offer a probabilistic view of a project's cost and effectiveness outcomes. This approach explicitly recognizes that single point values for these variables do not adequately represent the risk conditions inherent in a project.

Risk Mitigation and Tracking Techniques

Risk identification and characterization and risk analysis provide a list of significant project risks that require further management attention and/or action. Because risk mitigation actions are generally not costless, the systems engineer, in making recommendations to the project manager, must balance the cost (in resources and time) of such actions against their value to the project. Four responses to a specific risk are usually available: (1) deliberately do nothing, and accept the risk, (2) share the risk with a co-participant, (3) take preventive action to avoid or reduce the risk, and (4) plan for contingent action.

The first response is to accept a specific risk consciously. Sometimes, a risk can be shared with a co-participant, that is, with a foreign partner or a contractor. In this situation, the goal is to reduce NASA's risk independent of what happens to total risk, which may go up or down. There are many ways to share risks, particularly cost risks, with contractors. These include various incentive contracts and warranties. The third and fourth responses require that additional specific planning and actions be undertaken.

Typical technical risk mitigation actions include additional (and usually costly) testing of subsystems and systems, designing in redundancy, and building a full engineering model. Typical cost risk mitigation actions include using off-the-shelf hardware and providing sufficient funding during Phases A and B. Major supportability

risk mitigation actions include providing sufficient initial spares to meet the system's availability goal and a robust resupply capability (when transportation is a significant factor). For those risks that cannot be mitigated by a design or management approach, the systems engineer should recommend the establishment of reasonable financial and schedule contingencies and technical margins.

The strategy and underlying rationale selected for a specific risk should be documented in a risk mitigation plan and its effectivity should be tracked through the project cycle, as required by NMI 8070.4A. The techniques for choosing a (preferred) risk mitigation strategy deal with the larger role of trade studies and system modeling in general. Some techniques for planning and tracking are briefly mentioned here.

Watchlists and Milestones. A *watchlist* is a compilation of specific risks, their projected consequences and early indicators of the start of the problem. The risks on the watchlist are those that were selected for management attention as a result of completed risk management activities. A typical watchlist also shows for each specific risk a triggering event or missed milestone (for example, a delay in the delivery of long lead items), the related area of impact (production schedule), and the risk mitigation strategy to be used in response. The watchlist is periodically reevaluated and items are added, modified or deleted as appropriate. Should the triggering event occur, the projected consequences should be updated and the risk mitigation strategy revised as needed.

Contingency Planning. This technique is generally used in conjunction with a watchlist. The focus in contingency planning is on developing credible hedges and work arounds, which are activated upon a triggering event. To be credible, hedges often require that additional resources be expended, which provide a return only if the triggering event occurs. In this sense, contingency planning and resources act as a form of project insurance. (The term *contingency* here should not be confused with use of the same term for project reserves.)

Critical Items/Issues Lists. A critical items/issues list (CIL) is similar to a watchlist, and has been used extensively on the Shuttle program to track items with significant system safety consequences.

C/SCS and TPM Tracking. Two very important risk tracking techniques—cost and schedule control systems (C/SCS) and Technical Performance Measure (TPM) tracking—are discussed later.

Risk Management: Summary

Uncertainty is a fact of life in systems engineering. To deal with it effectively, the risk manager needs a disciplined approach. In a project setting, a good-practice approach includes efforts to:

- Plan, document and complete a risk management program.
- Identify and characterize risks for each phase of the project. High risks, those for which the combined effects of likelihood and consequences are significant, should be given specific management attention. Reviews conducted throughout the project cycle should help to force out risk issues.
- Apply qualitative and quantitative techniques to understand the dominant risks and to improve the allocation of risk reduction resources. This may include the development of project-specific risk analysis models such as decision trees and PRAs.
- Formulate and execute a strategy to handle each risk, including establishment, where appropriate, of reasonable financial and schedule contingencies and technical margins.

- Track the effectivity of each risk mitigation strategy.

Good risk management requires a team effort—that is, managers and systems engineers at all levels of the project need to be involved. However, risk management responsibilities must be assigned to specific individuals. Successful risk management practices often evolve into institutional policy.

BASELINE MANAGEMENT

The *baseline* for a project contains all of the technical requirements and related cost and schedule requirements that are sufficiently mature to be accepted and placed under change control by the NASA project manager. The project baseline consists of two parts: the technical baseline and the business baseline. The systems engineer is responsible for managing the technical baseline and ensuring that the technical baseline is consistent with the costs and schedules in the business baseline. Typically, the project control office manages the business baseline.

Baseline management requires the formal agreement of both the buyer and the seller to proceed according to the up-to-date, documented project requirements (as they exist at that phase in the project cycle), and to change the baseline requirements only by a formal change control process. The buyer might be an external funding agency. For example, the buyer for the GOES project is NOAA and the seller is the NASA GOES project office. Baseline management must be enforced at all levels. In the next level for this same example, the NASA GOES project office is the buyer and the seller is the contractor, the Loral GOES project office.

The project-level systems engineer is responsible for ensuring the completeness and technical integrity of the technical baseline. The content of the technical baseline includes:

- Definition (or specification) of the functional and performance requirements for hardware, software and operations
- Interface definitions
- Specialty engineering requirements
- Verification plans
- Documentation trees.

Baseline management includes the following techniques:

- Baseline definition and approval
- Configuration control (and version control, if needed)
- Change control
- Traceability
- Data management
- Baseline communication.

Baseline Evolution

The project baseline evolves in discrete steps through the project life cycle. An initial baseline may be established when the top-level user requirements expressed in the *Mission Needs Statement* are placed under configuration control. At each interphase control gate, increased technical detail is added to the maturing baseline. For a typical project, there are five sequential technical baselines:

- Functional baseline at Program/Project Requirements Review (PRR, sometimes called development baseline)
- Design-to baseline at Preliminary Design Review (PDR)
- Build-to (or code-to) baseline at the Critical Design Review (CDR)
- Production (or as-built or as-coded) baseline at the System Acceptance Review (SAR)
- Operational (or as-deployed) baseline at Operational Acceptance Review (OAR).

Risk management activity must begin early and continue throughout the

decomposition process of the project cycle to prove that the core-level decisions are sound. These early detailed studies and tests must be documented and retained in the project archives, but they are not part of the technical baseline.

Configuration Management

Configuration management is the discipline of identifying and formalizing the physical and functional characteristics of a configuration item at discrete points in the product evolution for the purpose of maintaining the integrity of the product and controlling changes to the baseline. As a functional discipline, configuration management manages the documentation of the approved evolution of a product's configuration. Configuration management includes configuration or baseline identification, configuration control and configuration communication. (See Figure 7.)

Configuration management is essential to the execution of an orderly development process, to enable the modification of an existing design, and to provide for later replication of an existing design. Configuration management often provides the information needed to track the technical progress of the project.

Configuration identification of a baseline is evidenced by documentation such as requirements documents, specifications, drawings, code listings, process specifications and material specifications. Configuration documentation is not considered part of the technical baseline until approved by control gate action of the buyer.

Configuration control is the process of controlling changes to any approved baseline by formal action of a change board that is controlled by the same authority that previously approved the baseline. Typically, the change control board meets to consider change requests to the business or technical baselines of the project. The project manager is usually the board chair, and the configuration manager the secretary, who skillfully guides the process and records the official events of the process.

In a change control board forum, a number of issues should be addressed:

- What is the proposed change?
- What is the reason for the change?
- What is the design impact?
- What is the effectiveness or performance impact?
- What is the schedule impact?
- What is the project life-cycle cost impact?
- What is the impact of not making the change?
- What is the risk of making the change?
- What is the impact on operations?
- What is the impact to support equipment and services?
- What is the impact on spares requirements?
- What is the effectivity of the change?
- What documentation is affected by the change?
- Is the buyer supportive of the change?

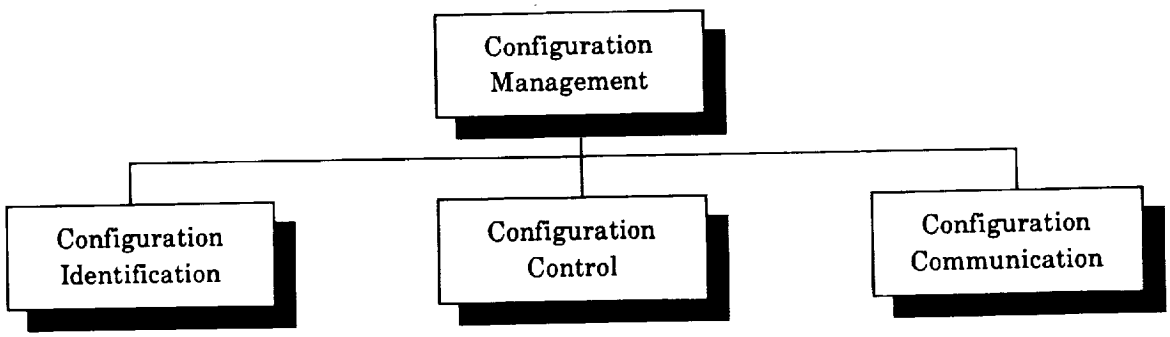


Figure 7 Configuration Management Structure

A review of this information should lead to a well-informed decision. When this information is not available to the change control board, unfounded decisions are made, often with negative consequences to the project.

Change Control Board Conduct

Objective: To review evaluations and then approve or disapprove proposed changes to the project's technical, operations or business baseline.
Participants: Project manager (chair), project-level systems engineer, managers of each affected organization, configuration manager (secretary), presenters.
Format: Presenter covers recommended change and discusses related system impact. The presentation is reviewed by the systems engineer for completeness prior to presentation.
Decision: The CCB members discuss the Change Request (CR) and formulate a decision. Project manager agrees or overrides.

Configuration control always includes the management of approved baseline documentation, so configuration control is required on a no-change project as well as a frequently changing one. Configuration management and configuration control embrace the function of data management, which ensures that only up-to-date baseline information is available to the project staff. The data management function also encompasses managing and archiving supporting analyses and trade study data, and keeping it convenient for project use.

Configuration verification is part of configuration control. It ensures that the resulting products conform to the intentions of the designers and to the standards established by preceding approved baselines. Each control gate serves to review and challenge the data presented for conformance to the previously established baseline constraints. The Physical Configuration Audit control gate verifies that the physical configuration of the product corresponds to the build-to (or code-to) documentation previously approved at the CDR. The Functional Configuration Audit control gate verifies that the acceptance test results are consistent with the test requirements previously approved at the PDR and CDR. The Formal Qualification Review control gate verifies that the as-built product is consistent with the as-built or as-coded documentation and describes the ultimate configuration of the product. This review follows all modifications needed to implement qualification-caused corrective actions.

For disciplined software development, additional configuration control methods are recommended:

- Computer Resources Working Group (CRWG)—ensures the development environment is adequate for the job
- Software Configuration Review Board—change board for software baseline changes
- Software Development Library—management controlled repository for software development documentation and tools
- Software Development Folder (SDF)—developer-controlled repository for development documentation and tools.

The configuration manager performs the following functions:

- Conceives, documents and manages the configuration management system
- Acts as secretary of the change control board (controls the change approval process)
- Controls changes to baseline documentation
- Controls release of baseline documentation
- Initiates configuration verification audits.

Configuration communication is the process of conveying to all involved parties the approved baseline progression in a timely manner. This is essential to ensure that

developers only pursue options that are compatible with the approved baseline.

Communication also keeps developers knowledgeable of the approved baseline and the necessity of approaching the change control board for approval of any deviations considered necessary to further develop the system.

The project's approach to configuration management should be documented in the project's Configuration Management Plan.

Change Control and Version Control

Once a baseline is placed under change control, any change requires the approval of the change control board. The project manager chairs the change control board, while the systems engineer or configuration manager is responsible for reviewing all material for completeness before it is presented to the board, and for ensuring that all affected organizations are represented in the change control board forum.

Change control is essential at both the contractor and NASA Center levels. Changes determined to be Class 1 to the contractor must be referred to the NASA project manager for resolution. This process is described in Figure 8. The use of a preliminary Engineering Change Proposal (ECP) to forewarn of an impending change provides the project manager with sufficient preliminary information to determine whether the contractor should spend NASA contract funds on a formal ECP. This technique is designed to save significant contract dollars.

Class 1 changes affect the approved baseline and hence the product version identification. Class 2 changes are editorial changes or internal changes not "visible" to the external interfaces.

Overly formalized systems can become so burdensome that members of the project team may try to circumvent the process. It is essential that the formality of the change process be appropriately tailored to the needs of each project. However, there must always be an effective change control process on every project.

For software projects, it is routine to use version control for both pre-release and post-release deliverable systems. It is equally important to maintain version control for hardware-only systems.

Approved changes on a development project that has only one deliverable obviously are only applicable to that one deliverable item. However, for projects that have multiple deliverables of "identical" design, changes may become effective on the second or subsequent production articles. In such a

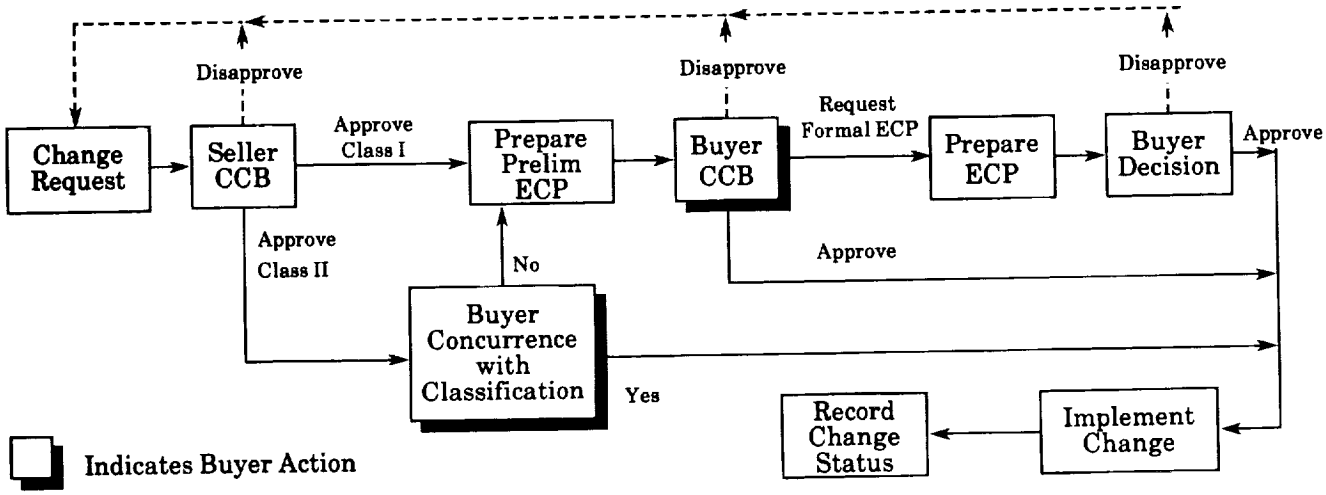


Figure 8 Contract Change Control Process

situation, the change control board must decide the effectivity of the change, and the configuration control system must maintain version control and identification of the as-built configuration for each article. Incremental implementation of changes is common in projects that have a deliberate policy of introducing product or process improvements. As an example, the original 1972 plan held that each of the Space Shuttle orbiters would be identical. In reality, each of the orbiters is different, driven primarily by the desire to achieve the original payload requirement of 65,000 pounds. Proper version control documentation has been essential to the sparing, fielding and maintenance of the operational fleet.

Data Management and Requirements Traceability

Data management is an essential and associated function to configuration management. Data management ensures that official baseline data is retained, available and controlled for all official project use. Data management is essentially the official project library and reference desk.

The data manager performs the following functions:

- Conceives, documents and manages the documentation management system
- Manages changes to baseline documentation
- Manages the release of baseline documentation
- Manages the project library.

Before the project team can produce a tangible product, engineering must produce descriptions of the system using words, icons (drawings) and numbers (i.e., symbolic information). The project team must have a common understanding of the words and icons in order to be able to go from an idea to a properly functioning system.

Since the systems engineer spends time working with information about the system rather than the system itself, there are several vital characteristics the symbolic information must have. First, the information must be *shareable*. Whether it is in electronic or paper form, the data must be readily available in the most recently approved version to all members of the team.

Second, symbolic information must be *durable*. This means that it must be recalled accurately every time and represent the most current version of the baseline. The baseline information cannot change or degrade with repeated access of the database or paper files, and cannot degrade with time. This is not a trivial requirement, poor data management practices (e.g., allowing someone to borrow the only copy of a document or drawing) can allow controlled information to become lost. Also, material must be retained for the life of the program (and possibly beyond), and a complete set of documentation for each baseline change must be retained.

Third, the symbolic information must be *traceable* upward and downward. A data base must be developed and maintained to show the parentage of any requirement. The data base must also be able to display all children derived from a given requirement. Finally, traceability must be provided to engineering reports that document trade study results and other decisions that played a key role in the flowdown of requirements.

It is the responsibility of the systems engineer to ensure the active, approved baseline is communicated to all those relying on it. This technique keeps all participants apprised as to the distinction between what is frozen under formal change control and what can still be decided without change control board approval.

REVIEWS, AUDITS AND CONTROL GATES

The intent and policy for reviews, audits and control gates should be developed during

Phase A and defined in the Project Implementation Plan. The specific implementation of these activities should be consistent with, though not limited to, the types of reviews and audits described in this section. The same tailoring applies to the timing of reviews, audits and control gates.

The purpose of a *review* is to furnish the forum and process to provide NASA management and their contractors assurance that the most satisfactory approach, plan or design has been selected, that a configuration item has been produced to meet the specified requirements, or that a configuration item is ready. Reviews (technical or management) are scheduled to communicate an approach, demonstrate an ability to meet requirements or establish status. Reviews help to develop a better understanding among task or project participants, open communication channels, alert participants and management of problems and open avenues for solutions.

Project Termination

It should be noted that project termination, while usually disappointing to project personnel, may be a proper reaction to changes in external conditions or to an improved understanding of the system's projected cost-effectiveness.

The purpose of an *audit* is to provide NASA management and its contractors a thorough examination of adherence to program or project policies, plans, requirements and specifications. Audits are the systematic examination of tangible evidence to determine adequacy, validity and effectiveness of the activity or documentation under review. An audit may examine documentation of policies and procedures as well as verify adherence to them.

The purpose of a *control gate* is to provide a scheduled event (either a review or an audit) that NASA management will use to make program or project go/no-go decisions. A control gate is a management event in the project cycle that is of sufficient importance to be identified, defined and included in the project schedule. It requires formal examination to evaluate project status and to obtain approval to proceed to the next management event according to the Project Implementation Plan.

GENERAL PRINCIPLES FOR REVIEWS

Review Boards. The convening authority, who supervises the manager of the activity being reviewed, normally appoints the review board chair. Unless there are compelling technical reasons to the contrary, the chair should not be directly associated with the project or task under review. The convening authority also names the review board members. The majority of the members should not be directly associated with the program or project under review.

Internal Reviews. During the course of a project or task, it is necessary to conduct internal reviews that present technical approaches, trade studies, analyses and problem areas to a peer group for evaluation and comment. The timing, participants and content of these reviews are normally defined by the project manager or the manager of the performing organization. Internal reviews are also held prior to participation in a formal, control gate review.

The internal reviews provide an excellent means for controlling the technical progress of the project. They also should be used to ensure that all interested parties are involved in the design/development process early on, and throughout the process. Thus, representatives from areas such as manufacturing and quality assurance should attend the internal reviews as active participants. They can then, for example, ensure that the design is producible and that quality is managed through the project cycle.

In addition, some organizations utilize a *Red Team*. This is an internal, independent, peer-level review conducted to identify any

deficiencies in requests for proposals, proposal responses, documentation or presentation material prior to its release. The project or task manager is responsible for establishing the Red Team membership and for deciding which of their recommendations are to be implemented.

Review Presentation Material. Presentations using existing documentation such as specifications, drawings, analyses and reports may be adequate. Copies of any prepared materials (such as viewgraphs) should be provided to the review board and meeting attendees. Background information and review presentation material of use to board members should be distributed to the members early enough to enable them to examine it prior to the review. For major reviews, this time may be as long as 30 calendar days.

Review Conduct. All reviews should consist of oral presentations of the applicable project requirements and the approaches, plans or designs that satisfy those requirements. These presentations normally are given by the cognizant design engineer or his/her immediate supervisor.

It is highly recommended that in addition to the review board, the review audience include project personnel (NASA and contractor) not directly associated with the design being reviewed. This is required to utilize their cross-disciplinary expertise to identify any design shortfalls or recommend design improvements. The review audience should also include non-project specialists in the area under review, and specialists in manufacturing and fabrication, testing, quality assurance, reliability and safety. Some reviews may also require the presence of both the contractor's and NASA's contracting officers.

Prior to and during the review, board members and review attendees may submit requests for action or engineering change requests (ECR) that document a concern, deficiency or recommended improvement in the presented approach, plan or design. Following the review, these are screened by the review board to consolidate them and to ensure that the chair and cognizant manager(s) understand the intent of the requests. It is the responsibility of the review board to ensure that adequate closure responses for each of the action requests are obtained.

Post Review Report. The review board chair has the responsibility to develop, where necessary, a consensus of the findings of the board, including an assessment of the risks associated with problem areas, and develop recommendations for action. The chair will submit, on a timely basis, a written report, including recommendations for action, to the convening authority with copies to the cognizant managers.

Standing Review Boards. Standing review boards are selected for projects or tasks that have a high level of activity, visibility and/or resource requirements. Selection of board members by the convening authority is generally made from senior Center technical and management staff. Supporting members or advisors may be added to the board as required by circumstances. If the review board is to function over the lifetime of a project, it is advisable to select extra board members and rotate active assignments to cover needs.

SPECIFIC TYPES OF REVIEWS

This section describes the types, purpose, timing and content of most of the reviews that may occur during the conduct of projects or tasks. Review material should be keyed to project documentation when available to minimize separate efforts.

Program/Project Requirements Review.

Purpose. The Program/Project Requirements Review (PRR) establishes the project

development (i.e., functional) baseline. It ensures that:

- The project objectives (particularly the research and/or science objectives) have been properly translated into definite and unambiguous statements of requirements.
- The impact of these requirements on the design of the major project elements and systems is sufficiently well understood that trades between requirements and constraints can be properly made.
- The management techniques, procedures, agreements and resources to be utilized by all project participants are evaluated.

Timing. At the completion of the Concept Definition Phase (Phase B) activities, just prior to issuing the Source Selection Request for Proposal.

Agenda. The appropriate items from the following review items/data checklist should be addressed:

- Status of action items from the Conceptual Design Review (CoDR)
- Project Plan
- Mission objectives
- Research objectives
- Science objectives
- Design criteria and approach
- System trade analyses
- Design analyses and trade studies
- Final system specification
- Preliminary interface specifications
- Software system requirements
- Work breakdown structure
- Preliminary manufacturing plan
- Preliminary ground operations plan
- Preliminary payload integration plan
- Preliminary flight operations plan
- Preliminary data management plan
- Configuration management plan
- Reliability requirements and plan
- Quality assurance requirements and plan
- System safety requirements and plan
- Project policy and requirements
- Management structure
- Budget constraints
- Schedule
- Risk management activities.

Preliminary Design Review. The Preliminary Design Review (PDR) is not a single review but a number of reviews starting with the system PDR, followed by reviews conducted on specific configuration items (CIs).

Purpose. The PDR establishes the design-to baseline and ensures that it meets the program, project, system, subsystem or specific CI baseline requirements. The PDR process should:

- Establish the ability of the selected design approach to meet the technical requirements.
- Establish the compatibility of the interface relationships between the specific configuration item and other interfacing items.
- Establish the integrity of the selected design approach.
- Establish the operability of the selected design.
- Assess compliance with quality assurance, reliability and system safety requirements.
- Address status, schedule and cost relationships.
- Establish the feasibility of the approach.

Timing. After design-to specifications are developed and after risk reduction analyses are available.

Agenda. The appropriate items from the following review items/data checklist should be addressed:

- Status of action items from the applicable Hardware or Software Specification Review(s)
- Final functional requirements and specifications
- Technical justification for the performance specified

- Experiment performance analysis, including an analysis of instrument accuracy requirements
- Design parameters and constraints
- Environmental design requirements
- Interface design requirements
- Requirements traceability results
- Software standards to be applied
- Design and safety codes and standards to be applied
- Results of technical feasibility modeling and testing
- Design optimization analyses
- Discussion of block diagrams
- Compliance with functional requirements and specifications
- Suitability of inherited designs and hardware
- Lists of preliminary parts, materials and processes
- Spares requirements philosophy
- Preliminary data management flow and reduction plans
- Preliminary payload integration plan
- Preliminary ground operations plan
- Preliminary flight operations plan
- Requirements and plans for support equipment, including ground support equipment (GSE)
- Preliminary reliability analyses, including single-point failure mode policy
- Preliminary system safety analyses
- Quality Assurance Plan
- Hardware and/or software verification plans
- Hardware and software development plans and schedules (including verification tests or analyses to be performed)
- Present status of item under review, including cost and technical developments
- Risk management activities.

Critical Design Review. The Critical Design Review (CDR) is not a single review but a number of reviews starting with specific CIs and ending with the system CDR.

Purpose. The CDR verifies the suitability of a CI design in meeting the specified requirements and establishes its build-to and/or code-to baseline. The CDR determines whether the design is compatible with the specified requirements, and verifies that the design conforms to the requirements established at the PDR and updated to the time of the CDR. During the CDR, the integrity of the design is verified through review of analytical and test data.

Following the CDR, the CI specifications and drawings are updated and placed under configuration control, and may be then released for fabrication and/or coding.

Timing. When the design of a CI is complete and after the completion of producibility demonstration. It should be held early enough to allow for corrective action and before total design freeze, the purchase of significant equipment or fabrication of final hardware.

Agenda. The appropriate items from the following review items/data checklist should be addressed:

- Status of PDR action items
- Design requirements and specifications
- Interface requirements and specifications
- Design approach
- Assessment of hardware and software inheritance
- Test procedures
- Producibility demonstration results
- Scale model test results
- Design trades and alternatives considered
- Reliability, maintainability and operability considerations
- Spares list
- Conformance of the design to functional and user requirements
- Conformance to environmental design requirements
- Differences between the configuration item, system and subsystem performances in relation to the performances estimated at the PDR
- Final hardware and software design verification plans

- Detailed mechanical (including electronic packaging, thermal, hydraulic and pneumatic) design
- Detailed electronic and electrical circuit design
- Detailed software design
- Interface details and agreements
- Mechanical and electronic parts stress analysis results
- Final reliability analyses, including single-point failure analyses against the reliability policy
- System safety analyses
- Electronic parts classifications and screening specifications
- Nonelectric parts, materials and processing list
- Materials and processing specifications
- Purchased devices list
- Manufacturing and fabrication plans
- Quality assurance plans and procedures
- Configuration control plans
- Qualification and acceptance test plans
- Calibration plan
- Data management flow and data reduction plan
- Support equipment and GSE requirements and plans
- Spares provisioning plan
- Ground operations plan
- Payload integration plan
- Flight operations plan
- Present status of item under review, including cost and technical developments
- Risk management activities.

Test Readiness Review. The Test Readiness Review (TRR) is not a single review but a series of reviews conducted prior to the start of verification testing of each test article, CI, subsystem and/or system.

Purpose. The TRR establishes the decision point to proceed with planned verification (qualification and/or acceptance) testing of test articles, CIs, subsystems and/or systems to acquire official sell-off verification data. The TRR assesses the adequacy of the test planning and compatibility with the verification requirements and specifications.

Timing. After completion of preliminary testing and prior to the start of official verification testing.

Agenda. The appropriate items from the following review items/data checklist should be addressed:

- Description of test article
- Test objectives
- Verification requirements and specifications
- Applicable test plans
- Applicable test procedures
- Test configuration and functional block diagrams
- Test equipment and circuitry
- Test equipment calibration
- Data to be collected, and collection and preservation methods
- Quality assurance plan
- Safety plan
- Test failure procedures
- Personnel responsibilities and qualifications
- Present status of item under review including cost and technical developments
- Risk management activities.

System Formal Qualification Review.

Purpose. The System Formal Qualification Review (SFQR) establishes the system production baseline by verifying that the system performance meets the system qualification specifications. The qualification testing demonstrates that the system meets its performance and operational requirements within the specified margins. The SFQR is the decision point for customer approval of the qualification certification of the design.

Timing. After the completion of all lower-level qualification testing.

Agenda. The appropriate items from the following review items/data checklist should be addressed:

- Status of action items from the applicable CDRs and TRRs
- Description of system tested, including all subsystems and functional block diagrams
- Qualification test objectives
- Qualification test requirements and specifications
- Description of test facilities
- Description of test configurations
- Subsystem qualification test results
- System qualification test results
- Qualification by similarity analysis
- Nonconformance reports/status
- Waivers and deviations
- Open work list
- Environmental retest following corrective action of any failures
- Strength and fracture mechanics for as-built hardware
- Software development documentation
- Summary of qualification status of all end items subjected to separate qualification tests
- Operational manuals
- Maintenance manuals
- Present status of system under review, including cost and technical developments
- Risk management activities.

Functional and Physical Configuration Audit.

Purpose. A Functional Configuration Audit (FCA) verifies that each as-built configuration item, test article, subsystem and/or system satisfies the functional and performance requirements specified in their respective design-to specifications.

A Physical Configuration Audit (PCA) verifies that each as-built test article, CI, subsystem and/or system:

- Satisfies the physical requirements (weight, center of gravity, moments of inertia, surface finish, cleanliness, etc.) specified in their respective design specifications
- Is correctly documented in as-built drawings, code listings, user manuals, etc.

Timing. Following the completion of the SFQR. Usually held in conjunction with the System Acceptance Review (SAR). For single unit projects, the FCA/PCA may be held prior to qualification testing.

Agenda. The appropriate items from the following project documentation should be addressed:

- CI, subsystem and system specifications
- Design drawings and engineering orders
- Subsystem and system schematics and block diagrams
- Design verification matrices for each configuration item, subsystem and system
- Inspection results
- Material and electronic parts certifications
- Materials process certifications
- Material Utilization List (MUL)
- Installed non-flight hardware list
- Test results
- Demonstration results
- Nonconformance reports/status
- Results of each Configuration Item Acceptance Review (CIAR)
- Results of the SFQR.

System Acceptance Review.

Purpose. The System Acceptance Review (SAR) provides the decision point to confirm that the design is ready for either integration, acceptance or replication.

Timing. Following the completion of the SFQR and prior to the Multi-Unit Procurement Phase and/or the Pre-Operations Phase (Phase E).

Agenda. The appropriate items from the following project documentation should be addressed:

- Brief description of system under review
- Verification requirements
- Results of the system FCA and PCA
- Results of the SFQR

- System verification report (qualification and operation)
- System acceptance report
- Final systems operations and maintenance methods
- System development lessons learned document
- Safety analyses status
- Present status of system under review, including cost and technical developments
- Risk management activities.

Safety Reviews. System safety is the application of engineering and management principles, criteria and techniques to optimize safety within the constraints of operational effectiveness, time and cost through all phases of the project cycle. A series of system and occupational safety reviews are held during the project cycle, many of which are held concurrently with other project reviews. Following are descriptions of these reviews and their relationship to the other project reviews.

Occupational Safety Reviews. The requirements for these reviews are not covered here. However, the systems engineer should be aware that many occupational safety requirements can impose requirements on flight and/or ground equipment, such as the shipping and handling of pressure vessels or toxic or explosive materials. Early reviews with Center occupational safety personnel should be held to identify and understand any problem areas and specify the requirements to control them.

Conceptual Design Safety Review.

Purpose. The Conceptual Design Safety Review (CoDSR) ensures that safety requirements have been included in the conceptual design and that a preliminary assessment of the potential hazards has been made. At several NASA Centers, the CoDSR is called the Phase 0 Safety Review.

Timing. At the completion of the Mission Needs and Conceptual Studies Phase (Phase A). It should be held concurrently with the Conceptual Design Review (CoDR).

Agenda. The appropriate items from the following list should be addressed:

- Purpose of the project, facility or equipment
- Design requirements
- Safety requirements
- Preliminary project safety plan
- Preliminary hazard analysis
- Safety staffing and management structure
- Safety budget
- Schedule
- Risk management activities.

Project Requirements Safety Review.

Purpose. The Project Requirements Safety Review (PRSR) establishes the project safety requirements baseline and ensures that:

- The project safety objectives have been properly translated into definite and unambiguous statements of requirements.
- The impact of these requirements on the design of the major project elements and systems is sufficiently well understood that trades between requirements and constraints can be properly made.
- The management techniques, procedures, agreements and resources to implement the safety program by all project participants are evaluated.

Timing. At the completion of the Concept Definition Phase (Phase B) activities just prior to issuing the Source Selection Request for Proposal. It should be held concurrently with the PRR.

Agenda. The appropriate subjects from the following list should be addressed:

- Purpose of the project, facility or equipment

- Status of action items from the CoDSR
- Design requirements
- Safety requirements
- Updated preliminary project safety plan
- Updated preliminary hazard analysis
- Safety staffing and management structure
- Safety budget
- Schedule
- Risk management activities.

Preliminary Design Safety Review. The Preliminary Design Safety Review (PDSR) is not a single review but a series of reviews conducted on specific configuration items, subsystems and the system.

Purpose. The PDSR ensures that the proposed CI, subsystem and/or system designs satisfy the project and Center safety requirements. At several NASA Centers, the PDSR is called the Phase I Safety Review.

Timing. At the completion of preliminary design and prior to the start of major detail design activities. It should be held concurrently with the PDRs.

Agenda. The appropriate subjects from the following list should be addressed:

- Description of design under review
- Status of safety-related action items from applicable hardware or software specification reviews
- Updated project safety plan
- Updated safety analysis reports
- Updated preliminary hazard analyses (sometimes called the Phase I Hazard Analyses)
- Preliminary Failure Modes and Effects Analysis (FMEA)
- Preliminary Critical Items List (CIL).
- List of limited-life items
- Accident or mishap investigation reports
- Waiver and deviation request dispositions
- Present status of safety activities, including cost and technical developments
- Risk management activities.

Critical Design Safety Review. The Critical Design Safety Review (CDSR) is not a single review but a series of reviews conducted on specific configuration items, subsystems and the system.

Purpose. The CDSR establishes the baseline for safety requirements, safety hazard controls and verification methods to be implemented in verifying those controls. At several NASA Centers, the CDSR is called the Phase II Safety Review.

Timing. When the design of a configuration item is essentially complete and prior to total design freeze, the purchase of significant equipment, or fabrication of final hardware. It should be held concurrently with the CDRs.

Agenda. The appropriate subjects from the following list should be addressed:

- Description of design under review
- Status of safety-related action items from applicable hardware or software PDSRs
- Final project safety plan
- Updated safety analysis reports
- Updated preliminary hazard analyses (sometimes called the Phase II Hazard Analyses)
- Final Failure Modes and Effects Analysis
- Final Critical Items List
- List of limited-life items
- Accident or mishap investigation reports
- Waiver and deviation request dispositions
- Present status of safety activities including cost and technical developments
- Risk management activities.

System Acceptance Safety Review.

Purpose. The System Acceptance Safety Review (SASR) provides the decision point to confirm that all project safety requirements have been satisfied and confirms the satisfactory completion of all hazard control verification items and open safety items. At several NASA Centers, the SASR is called the Phase III Safety Review.

Timing. Following the completion of the SFQR and prior to the Multi-Unit Procurement Phase and the Pre-Operation Phase (Phase E). It should be held concurrently with the SAR.

Agenda. The appropriate subjects from the following list should be addressed:

- Description of design under review
- Status of safety-related action items from applicable hardware or software CDRs
- Updated safety analysis reports
- Updated preliminary hazard analyses (sometimes called the Phase III Hazard Analyses)
- Accident or mishap investigation reports
- Waiver and deviation request dispositions
- Present status of safety activities, including cost and technical developments
- Risk management activities.

Launch or Operational Safety Readiness Reviews.

Purpose. These reviews ensure the flight and/or ground operational safety of the item under review by certifying that:

- A CI, subsystem or system complies with all program and/or project safety requirements.
- Approved controls for all identified safety hazards have been implemented.
- All personnel involved in the handling and/or operation of the item under review have received the required training.

Timing. Following installation and integration and prior to flight and/or start of ground operations.

Agenda. The appropriate subjects from the following list should be addressed:

- Brief description of item under review
- Safety requirements and specifications
- Safety compliance data package
- Hazard analyses/reports with supporting data
- Critical items list
- Limited-life item list
- Accident or mishap investigation reports
- Nonconformance reports/status
- Personnel training requirements
- Personnel training status
- Present status of safety activities, including cost and technical developments
- Risk management activities.

STATUS REPORTING AND ASSESSMENT

An important part of systems engineering planning is determining what is needed in time, resources and people to realize the system that meets the desired goals and objectives. Planning functions such as WBS preparation, scheduling and fiscal resource requirements planning, were discussed earlier. Project management, however, does not end with planning; project managers need visibility into the progress of those plans in order to exercise proper management control. This is the purpose of the status reporting and assessing processes. Status reporting is the process of determining where the project stands in dimensions of interest such as cost, schedule and technical performance. Assessing is the analytical process that converts the output of the reporting process into a more useful form for the project manager; namely, what are the future implications of current trends? Lastly, the manager must decide whether that future is acceptable, and what changes, if any, in current plans are needed. Planning, status reporting, and assessing are systems engineering and/or program control functions; decision making is a management one.

These processes together form the feedback loop depicted in Figure 9. This loop takes place on a continual basis throughout the project cycle.

This loop is applicable at each level of the project hierarchy. Planning data, status reporting data and assessments flow up the hierarchy with appropriate aggregation at each level; decisions cause actions to be

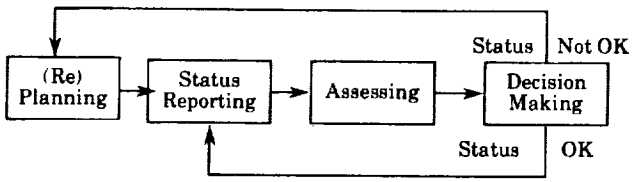


Figure 9 Planning and Status Reporting Feedback Loop

taken down the hierarchy. Managers at each level determine (consistent with policies established at the next higher level of the project hierarchy) how often, and in what form, reporting data and assessments should be made. In establishing these status reporting and assessment requirements, some principles of good practice are:

- Use an agreed-upon set of well-defined status reporting variables
- Report these core variables in a consistent format at all project levels
- Maintain historical data for both trend identification and cross-project analyses
- Encourage a logical process of rolling up status reporting variables, (e.g., use the WBS for obligations/costs status reporting and PBS for mass status reporting)
- Support assessments with quantitative risk measures
- Summarize the condition of the project by using color-coded (red, yellow, and green) alert zones for all core reporting variables.

Regular, periodic (e.g., monthly) tracking of the core status reporting variables is recommended, through some status reporting variables should be tracked more often when there is rapid change or cause for concern. Key reviews, such as PDRs and CDRs, are points at which status reporting measures and their trends should be carefully scrutinized for early warning signs of potential problems. Should there be indications that existing trends, if allowed to continue, will yield an unfavorable outcome, replanning should begin as soon as practical.

This section provides additional information on status reporting and assessment techniques for costs and schedules, technical performance, and systems engineering process metrics.

Cost and Schedule Control Measures

Status reporting and assessment on costs and schedules provides the project manager and systems engineer visibility into how well the project is tracking against its planned cost and schedule targets. From a management point of view, achieving these targets is on a par with meeting the technical performance requirements of the system. It is useful to think of cost and schedule status reporting and assessment as measuring the performance of the "system that produces the system."

NHB 9501.2B, *Procedures for Contractor Reporting of Correlated Cost and Performance Data*, provides specific requirements for cost and schedule status reporting and assessment based on a project's dollar value and period of performance. Generally, the NASA Form 533 series of reports is applicable to NASA cost-type (i.e., cost reimbursement and fixed-price incentive) contracts. However, on larger contracts (>$25M) which require Form 533P, NHB 9501.2B allows contractors to use their own reporting systems in lieu of 533P reporting. The project manager/systems engineer may choose to evaluate the completeness and quality of these reporting systems against criteria established by the project manager/systems engineer's own Center, or against the DoD's *Cost/Schedule Cost System Criteria* (C/SCSC). The latter are widely accepted by industry and government, and a variety of tools exist for their implementation.

Assessment Methods. The traditional method of cost and schedule control is by comparing baselined cost and schedule plans against their actual values. In program control terminology, a difference between actual

performance and planned costs or schedule status is called a *variance*.

Figure 10 illustrates two kinds of variances and some related concepts. A properly constructed work breakdown structure (WBS) divides the project work into discrete tasks and products. Associated with each task and product (at any level in the WBS) is a schedule and a budgeted (i.e., planned) cost. The *Budgeted Cost of Work Scheduled* ($BCWS_t$) for any set of WBS elements is the budgeted cost of all work on tasks and products in those elements scheduled to be completed by time *t*. The *Budgeted Cost of Work Performed* ($BCWP_t$) is a statistic representing actual performance. $BCWP_t$, also called *Earned Value* (EV_t), is the budgeted cost for tasks and products that have actually been produced (completed or in progress) at time *t* in the schedule for those WBS elements. The difference, $BCWP_t - BCWS_t$, is called the schedule variance at time *t*.

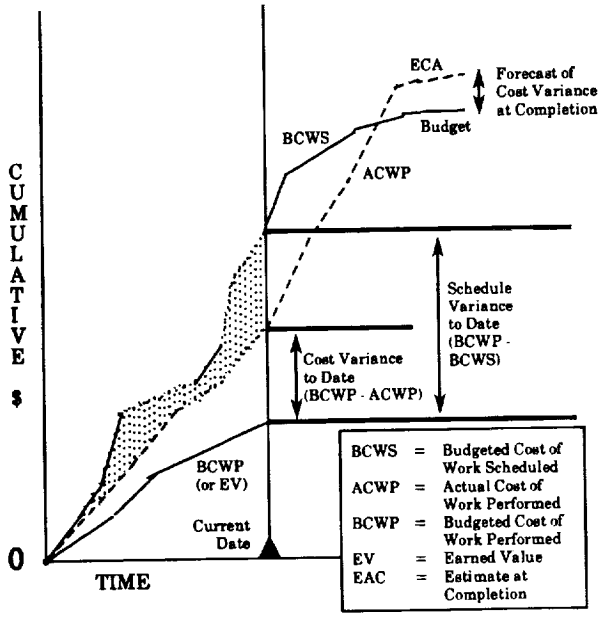


Figure 10 Cost and Schedule Variances

The *Actual Cost of Work Performed* ($ACWP_t$) is a third statistic representing the funds that have been expended up to time *t* on those WBS elements. The difference between the budgeted and actual costs, $BCWP_t - ACWP_t$, is called the cost variance at time *t*. Such variances may indicate that the cost *Estimate at Completion* (EAC_t) of the project is different from the budgeted cost. These types of variances enable a program analyst to estimate the EAC at any point in the project cycle.

If the cost and schedule baselines and the technical scope of the work are not fully integrated, then cost and schedule variances can still be calculated, but the incomplete linkage between cost data and schedule data makes it very difficult (or impossible) to estimate the current cost EAC of the project.

Control of Variances and the Role of the Systems Engineer. When negative variances are large enough to represent a significant erosion of reserves, then management attention is needed to either correct the variance, or to replan the project. It is important to establish levels of variance at which action is to be taken. These levels are generally lower when cost and schedule baselines do not support Earned Value calculations.

The first action taken to control an excessive negative variance is to have the cognizant manager or systems engineer investigate the problem, determine its cause and recommend a solution. There are a number of possible reasons why variance problems occur:

- A receivable was late or was unsatisfactory for some reason.
- A task is technically very difficult and requires more resources than originally planned.
- Unforeseeable (and unlikely to repeat) events occurred, such as illness, a labor strike, a fire or some other calamity.

Although the identification of variances is largely a program control function, there is an important systems engineering role in their control. That role arises because the correct assessment of why a negative variance is occurring greatly increases the

chances of successful control actions. This assessment often requires an understanding of the cost, schedule and technical situation that can only be provided by the systems engineer.

Computing the Estimate at Completion

EAC can be estimated at any point in the project. The appropriate formula depends upon the the reasons associated for any variances that may exist. If a variance exists due to a one-time event, such as an accident, then EAC = BUDGET + ACEP - BCWP where BUDGET is the original planned cost at completion. If a variance exists for systemic reasons, such as a general underestimate of schedule durations, or a steady redefinition of requirements, then the variance is assumed to continue to grow over time, and the equation is: EAC = BUDGET × (ACWP/BCWP).

It is also possible that EAC will grow at a greater rate than estimated by the above equation if there are a growing number of liens, action items or significant problems that will increase the difficulty of future work. Such factors could be addressed using risk management methods.

In a large project, a good EAC is the result of a variance analysis that may use a combination of these estimation methods on different parts of the WBS. A rote formula should not be used as a substitute for understanding the underlying causes of variances.

Technical Performance Measures

Status reporting and assessment of the system's technical performance measures (TPMs) complements cost and schedule control. By tracking the system's TPMs, the project manager gains visibility into whether the delivered system will actually meet its performance specifications (requirements). Beyond that, tracking TPMs ties together a number of basic systems engineering activities—that is, a TPM tracking program forges a relationship among systems analysis, functional and performance requirements definition and verification and validation activities.

- Systems analysis activities identify the key performance or technical attributes that determine system effectiveness; trade studies performed in systems analysis help quantify the system's performance requirements.
- Functional and performance requirements definition activities help identify verification and validation requirements.
- Verification and validation activities result in quantitative evaluation of TPMs.
- "Out-of-bounds" TPMs are signals to replan fiscal, schedule and people resources; sometimes new systems analysis activities need to be initiated.

Tracking TPMs can begin as soon as a baseline design has been established, which can occur as early as Phase B. A TPM tracking program should begin not later than the start of Phase C. Data to support the full set of selected TPMs may, however, not be available until later in the project cycle.

Selecting TPMs. In general, TPMs can be generic (attributes that are meaningful to each Product Breakdown Structure [PBS] element, like mass or reliability) or unique (attributes that are meaningful only to specific PBS elements). The systems engineer needs to decide which generic and unique TPMs are worth tracking at each level of the PBS. The systems engineer should track the measure of system effectiveness (when the project maintains such a measure) and the principal performance or technical attributes that determine it, as top-level TPMs. At lower levels of the PBS, TPMs worth tracking can be identified through the functional and performance requirements levied on each individual system, segment, etc.

In selecting TPMs, the systems engineer should focus on those that can be objectively measured during the project cycle. This measurement can be done directly by testing or indirectly by a combination of testing and analysis. Analyses are often the only means available to determine some high-level

TPMs such as system reliability, but the data used in such analyses should be based on demonstrated values to the maximum practical extent. These analyses can be performed using the same measurement methods or models used during trade studies. In TPM tracking, however, instead of using estimated (or desired) performance or technical attributes, the models are exercised using demonstrated values. As the project cycle proceeds through Phases C and D, the measurement of TPMs should become increasingly more accurate because of the availability of more "actual" data about the system.

Lastly, the systems engineer should select those TPMs that must fall within well-defined (quantitative) limits for reasons of system effectiveness or mission feasibility. Usually these limits represent either a firm upper or lower bound constraint. A typical example of such a TPM for a spacecraft is its injected mass, which must not exceed the capability of the selected launch vehicle. Tracking injected mass as a high-level TPM is meant to ensure that this does not happen.

Assessment Methods. The traditional method of assessing a TPM is by establishing a time-phased planned profile for it, and comparing the demonstrated value against that profile. The planned profile represents a nominal "trajectory" for that TPM taking into account a number of factors. These factors include the technological maturity of the system, the planned schedule of tests and demonstrations, and any historical experience with similar or related systems. As an example, spacecraft dry mass tends to grow during Phases C and D by as much as 25 to 30 percent. A planned profile for spacecraft dry mass may try to compensate for this growth with a lower initial value. The final value in the planned profile usually either intersects or is asymptotic to an allocated requirement (or contract specification). The planned profile method is the technical performance measurement counterpart to the Earned Value method for cost and schedule control described earlier.

Examples of High-Level TPMs for Planetary Spacecraft and Launch Vehicles

High-level technical performance measures (TPMs) for planetary spacecraft include:

- End-of-mission (EOM) dry mass
- Injected mass (includes EOM dry mass, baseline mission plus reserve propellant, other consumables and upper stage adaptor mass)
- Consumables at EOM
- Power demand (relative to supply)
- Onboard data processing memory demand
- Onboard data processing throughput time
- Onboard data bus capacity
- Total pointing error

Mass and power demands by spacecraft subsystems and science instruments may be tracked separately as well.

For launch vehicles, high-level TPMs include:

- Total vehicle mass at launch
- Payload mass (at nominal altitude or orbit)
- Payload volume
- Injection accuracy
- Launch reliability
- In-flight reliability
- For reusable vehicles, percent of value recovered
- For expendable vehicles, unit production cost at the n^{th} unit.

A closely related method of assessing a TPM relies on establishing a time-phased margin requirement for it and comparing the actual margin against that requirement. The margin is generally defined as the difference between a TPM's demonstrated value and its allocated requirement. The margin requirement may be expressed as a percent of the allocated requirement. The margin requirement generally declines through Phases C and D, reaching or approaching zero at their completion.

Depending on which method is chosen, the systems engineer's role is to propose reasonable planned profiles or margin requirements for approval by the cognizant manager. The value of either of these methods is that they allow management by exception—that is, only deviations from

planned profiles or margins below requirements signal potential future problems requiring replanning. If this occurs, then new cost, schedule and/or technical changes should be proposed. Technical changes may imply some new planned profiles. This is illustrated for a hypothetical TPM in Figure 11(a). In this example, a significant demonstrated variance (i.e., unanticipated growth) in the TPM during design and development of the system resulted in replanning at time t. The replanning took the form of an increase in the allowed final value of the TPM (the "allocation"). A new planned profile was then established to track the TPM over the remaining time of the TPM tracking program.

The margin management method of assessing is illustrated for the same example in Figure 11(b). The replanning at time t occurred when the TPM fell significantly below the margin requirement. The new higher allocation for the TPM resulted in a higher margin requirement, but it also immediately placed the margin in excess of that requirement.

Both of these methods recognize that the final value of the TPM being tracked is uncertain throughout most of Phases C and D. The margin management method attempts to deal with this implicitly by establishing a margin requirement that reduces the chances of the final value exceeding its allocation to a low number, for example, five percent or less. A third method of reporting and assessing deals with this risk explicitly. The risk management method is illustrated for the same example in Figure 11(c). The replanning at time t occurred when the probability of the final TPM value being less than the allocation fell precipitously into the red alert zone. The new higher allocation for the TPM resulted in a substantial improvement in that probability.

The risk management method requires an estimate of the probability distribution for the final TPM value. Early in the TPM tracking program, when the demonstrated

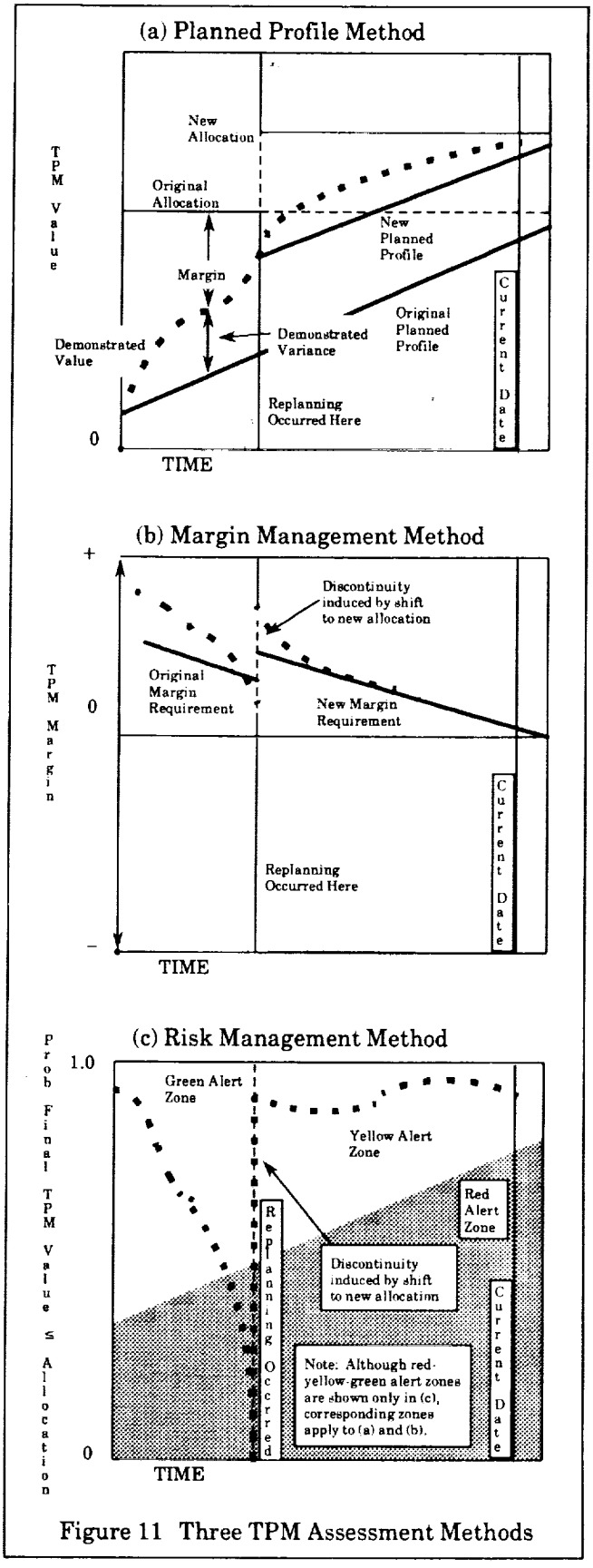
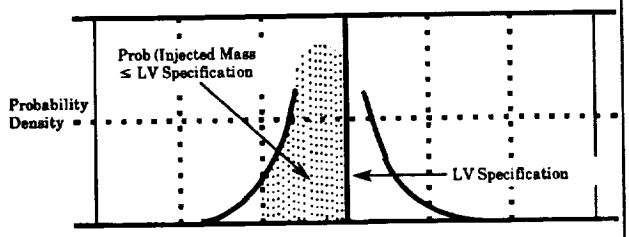


Figure 11 Three TPM Assessment Methods

An Example of the Risk Management Method for Tracking Spacecraft Mass

During Phases C and D, a spacecraft's injected mass can be considered an uncertain quantity. Estimates of each subsystem's and each instrument's mass are, however, made periodically by the design engineers. These estimates change and become more accurate as actual parts and components are built and integrated into subsystems and instruments and are integrated into spacecraft. Injected mass can also change during Phases C and D as the quantity of propellant is fine-tuned to meet the mission design requirements. At each point during development then, the spacecraft's injected mass is better represented as a probability distribution rather than as a single point.

The mechanics of obtaining a probability distribution for injected mass typically involve making estimates of three points — the lower and upper bounds and the most likely injected mass value. These three values can be combined into parameters that completely define a probability distribution like the one shown in the figure below.



The launch vehicle's "guaranteed" payload capability, designated the "LV Specification," is shown as a bold vertical line. The area under the probability curve to the left of the bold vertical line represents the probability that the spacecraft's injected mass will be less than or equal to the launch vehicle's payload capability. If injected mass is a TPM being tracked using the risk management method, this probability could be plotted in a display similar to Figure 11(c).

If this probability were nearly one, then the project manager might consider adding more objectives to the mission in order to take advantage of the "large margin" that appears to exist. In the above figure, however, the probability is significantly less than one. Here, the project manager might consider descoping the project, for example, by removing an instrument or otherwise changing mission objectives. The project manager could also solve the problem by requesting a larger launch vehicle!

value is based on indirect means of estimation, this distribution typically has a larger statistical variance than later, when it is based on measured data, e.g., a test result. When a TPM stays along its planned profile (or equivalently, when its margin remains above the corresponding margin requirement), the narrowing of the statistical distribution should allow the TPM to remain in the green alert zone (in Figure 11(c)) despite its growth. The three methods represent different ways to assess TPMs and communicate that information to management, but whichever is chosen, the pattern of success or failure should be the same for all three.

Relationship of TPM Tracking Program to the SEMP. The SEMP is the usual document for describing the project's TPM tracking program. This description should include a master list of those TPMs to be tracked and the measurement and assessment methods to be employed. If analytical methods and models are used to measure certain high-level TPMs, then these need to be identified. The reporting frequency and timing of assessments should be specified as well. In determining these, the systems engineer must balance the project's needs for accurate, timely and effective TPM tracking against the cost of the TPM tracking program. The TPM tracking program plan, which elaborates on the SEMP, should specify each TPM's allocation, time-phased planned profile or margin requirement, and alert zones, as appropriate to the selected assessment method.

Systems Engineering Process Metrics

Status reporting and assessment of systems engineering process metrics provides additional visibility into the performance of the "system that produces the system." As such, these metrics supplement the cost and schedule control measures discussed earlier.

Systems engineering process metrics try to quantify the effectivity and productivity of

the systems engineering process and organization. Within a single project, tracking these metrics allows the systems engineer to better understand the health and progress of that project. Across projects (and over time), the tracking of systems engineering process metrics allows for better estimation of the cost and time of performing systems engineering functions. It also allows the systems engineering organization to demonstrate its commitment to the TQM principle of continuous improvement.

Selecting Systems Engineering Process Metrics

Generally, systems engineering process metrics fall into three categories: those that measure the progress of the systems engineering effort, those that measure the quality of that process, and those that measure its productivity. Different levels of systems engineering management are generally interested in different metrics. For example, a project manager or lead systems engineer may focus on metrics dealing with systems engineering staffing, project risk management progress and major trade study progress. A subsystem systems engineer may focus on subsystem requirements and interface definition progress and verification procedures progress. It is useful for each systems engineer to focus on just a few process metrics. Which metrics should be tracked depends on the systems engineer's role in the total systems engineering effort. The systems engineering process metrics worth tracking also change as the project moves through the project cycle.

Collecting and maintaining data on the systems engineering process is not without cost. Status reporting and assessment of systems engineering process metrics divert time and effort from the process itself. The system engineer must balance the value of each systems engineering process metric against its collection cost. The value of these metrics arises from the insights they provide into the process that cannot be obtained from cost and schedule control measures alone. Over time, these metrics can also be a source of hard productivity data, which are invaluable in demonstrating the potential returns from investment in systems engineering tools and training.

Examples and Assessment Methods. Table 2 lists some systems engineering process metrics to be considered. That list is not intended to be exhaustive. Because some of these metrics allow for different interpretations, each NASA Center needs to define them in a common-sense way that fits its own processes. For example, each Center

Function	Systems Engineering Process Metric	Category
Requirements development and management	Requirements identified vs. completed vs. approved	S
	Requirements volatility	Q
	Trade studies planned vs. completed	S
	Requirements approved per systems engineering hour	P
Design and development	Specifications planned vs. completed	S
	Processing of ECRs/ECOs	Q
	Engineering drawings planned vs. related	S
Verification and Validation (V&V)	V&V plans identified vs. approved	S
	V&V procedures planned vs. completed	S
	Functional requirements approved vs. verified	S
	V&V plans approved per systems engineering hour	P
	V&V procedures completed per systems engineering hour	P
	Processing of trouble reports	Q
Reviews	Processing of Review Item Discrepancies (RIDs)	Q
	Processing of action items	Q

S = Progress, or schedule-related
Q = Quality-related
P = Productivity

Table 2 Systems Engineering Process Metrics

needs to determine what it meant by a *completed* versus an *approved* requirement, or whether these terms are even relevant. As part of this definition, it is important to recognize that not all requirements, for example, need be lumped together. It may be more useful to track the same metric separately for each of several different types of requirements, for example.

Quality-related metrics should serve to indicate when a part of the systems engineering process is overloaded and/or breaking down. These metrics can be defined and tracked in several different ways. For example, requirements volatility can be quantified as the number of newly identified requirements, or as the number of changes to already-approved requirements. As another example, engineering change request (ECR) processing could be tracked by comparing cumulative ECRs opened versus cumulative ECRs closed, or by plotting the age profile of of open ECRs, or by examining the number of ECRs opened last month versus the total number open. The systems engineer should apply personal judgment in picking the status reporting and assessment method.

Productivity-related metrics provide an indication of systems engineering output per unit of input. Although more sophisticated measures of input exist, the most common is the number of systems engineering hours dedicated to a particular function or activity. Because not all systems engineering hours cost the same, an appropriate weighing scheme should be developed to ensure comparability of hours across systems engineering personnel.

Displaying schedule-related metrics can be accomplished in a table or graph of planned quantities vs. actuals. With quality- and productivity-related metrics, trends are generally more important than isolated snapshots. The most useful kind of assessment method allows comparisons of the trend on a current project with that for a successful completed project of the same type. The latter provides a benchmark against which the system engineer can judge personal efforts.