GUIDE FOR MANAGING AND WRITING REQUIREMENTS

by

Ivy F. Hooks

This work prepared with contributions from NASA.

Compliance
Automation, Inc.

COMPLIANCE AUTOMATION, INC.
1221 South Main Street, Suite 204
Boerne, Texas 78006-2836

Acknowledgments: The author would like to acknowledge the contributions of those NASA and NASA contractor personnel who reviewed this guide and provided valuable feedback: Chuck Allen, Richard Armstrong, Paula Beckstrom, Cliff Hess, Pedro Martinez, John Park, Barney Roberts, and Glen VanZandt. The author would also like to thank Allison Pulcifer, without whose assistance this guide would have never been published.

Correction

This guide was written in 1994. There are several examples of requirements in the guide that have the phrases "be capable of" or "be able to". We no longer advocate the use of these phrases and recommend that your requirements will be clearer, shorter, more succinct, and verifiable if you do not use these phrases.

TABLE OF CONTENTS

SECTION 4 - WRITING REQUIREMENTS ……………………………………………….……4-1

SECTION 5 - REQUIREMENT ATTRIBUTES ……………………………………………5-1

INTRODUCTION

This guide is prepared to help engineers, scientists, managers and other personnel in the process of writing and managing requirements. Although the examples here are NASA-aerospace related, the process, problems, and solutions are applicable to all types of projects and systems.

Each program or project involves the development, design, and delivery of a system. To acquire a system, requirements must be written to define what the system is to do. Design activities determine how the system can be built or acquired to meet the requirements. The design activity also results in the definition of requirements for the subparts of the system. Subpart design activities then determine how each subpart can be built or acquired to meet its requirements and often result in he derivation of another level of subpart requirements.

Requirements may be written by a user or customer, by a project team, or by design engineers. Very often the first job a new employee is given involves requirements -- writing, reviewing, rewriting, or verifying. However, that employee has probably never had any training in requirements. Many first time project managers also have not had training or experience in the requirements process. This guide is to help you wade through a complex process and get your job done right the first time.

The requirement process is critical across the life-cycle phases of the project. Figure 1 shows a NASA life-cycle chart.

PHASE:	**A** Preliminary Analysis	**B** Definition	**C** Design	**D** Development	**E** Operations
Activities:	Conceptual studies Exploration of alternatives	Preliminary design Concept solution	Detail design System development	Final design & development Fabrication Test	Support Product improvement
Requirement Related Documents:	Program Plan Draft System Specification	Baseline System Specification	Segment Specs Element Specs	Maintain Specs ------------------->	Maintain Specs ------------------->
Reviews:		SRR PDR	CDR	SAR FRR	ORR

Figure 1. Life-Cycle

Requirement definition begins in Phase A, with refinements and a baseline occurring during Phase B . During late Phase B and in Phase C, lower level requirements, e.g., segment, element, and subsystem requirements, are derived. The major requirement definition phase, for all levels is completed by Phase C, although revisions will continue to occur throughout the life-cycle. Requirement revisions result from the identification of new needs; discoveries in the design or development process; problems encountered in verification; changes in operations; or because of outside influences, e.g., budget constraints. The writing of requirements is not a one-time event, it continues over the entire life-cycle.

Each requirement must be verified and verification may occur throughout the design, development, and operational phases. Any changes or additions to requirements will cause re-verification.

Once a system is operational, it is important to continue to maintain and control the requirements. Each new or changed requirement that is introduced during the operational phase must be tracked through design and verification and into updated operational procedures, as shown in Figure 2. Design changes that result from operational considerations must be shown to be consistent with existing requirements, or the requirements must be changed. The old, or revised, requirements must be re-verified when a design change is implemented. Thus, requirements evolve and must be maintained over the entire life cycle of the program or project.

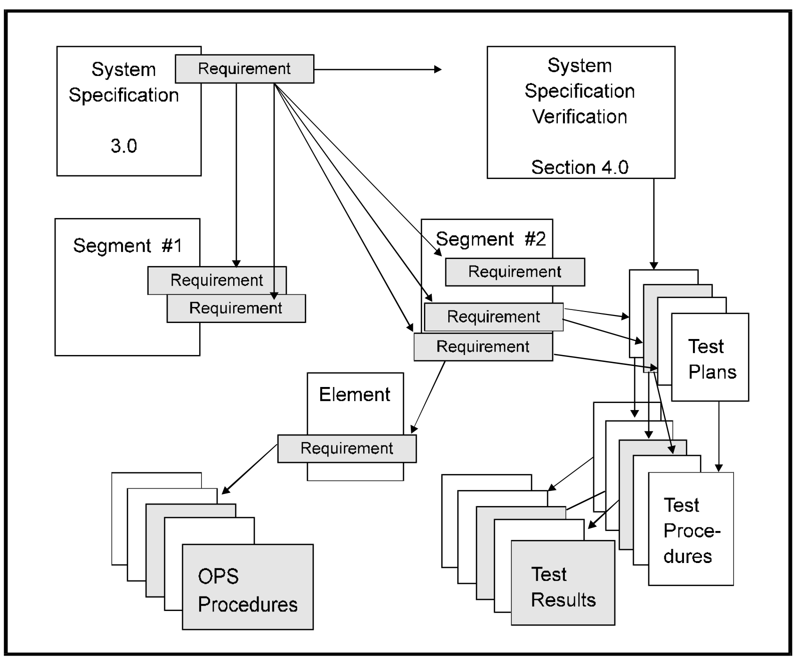

Figure 2. Change to a requirement has many impacts.

Writing and managing requirements is part of a larger process, that of program/project management. This guide will address only those aspects of program/project management that directly affect, or are affected by, requirements. Requirements are derived through engineering processes that include design and analysis, trade studies, concept development and other activities, such as prototyping. This larger engineering picture is covered only as it applies to requirement definition.

HOW THIS GUIDE IS ARRANGED

The following paragraphs summarize the content of each section of the guide. While the guide refers to the system throughout, the information is applicable to all levels -- system, segment, element, and subsystem.

SECTION 1
SCOPE

During Phase A the project scope is defined. Scope includes a definition of the need for the system, its goals, objectives, constraints, mission, and operational concepts. Scope also includes management information such as responsibility, schedules, and budgets. This effort is a prelude to writing requirements. The scope information will drive or constrain the requirements. The project scope will evolve through Phase A and be documented in a Program Plan. Phase B design studies will be constrained by the scope definition and Phase B results may drive changes to the scope and hence to the Program Plan. Section 1, Scoping a Project, provides information on this process and its documentation.

SECTION 2
MANAGING REQUIREMENTS

Requirements definition also begins in Phase A, but only after the Program Plan is documented and the scope is bounded. This section discusses managing requirements, from the development of the Program Plan through all life-cycle phases. Managing requirements includes developing and executing the processes for definition, verification, review, and maintenance of requirements and their specifications.

SECTION 3
LEVELS OF REQUIREMENTS

The system requirements, documented in a System Specification, are provided in a draft form at the beginning of Phase B. Midway through Phase B, the requirements, which are updated throughout this phase, undergo a System Requirements Review (SRR), and are baselined. The system design, which is evolving though Phase B, drives the next level of requirements. A project may have many levels, e.g., system, segment, element, subsystems, or only a few, e.g., system and subsystem. Each level results in a set of requirements applicable to the level and documented in a specification. This section defines the different levels and how they evolve.

SECTION 4
WRITING REQUIREMENTS

This section covers writing good requirements, addresses many common requirement writing problems, and gives examples.

SECTION 5
REQUIREMENT ATTRIBUTES

This section describes requirements related information, including allocation, traceability, rationale, verification, and others. It covers how to collect the information and how to document it.

SECTION 6
SPECIFICATION

This section focuses on the content of a system requirements specification and provides a sample outline. It then expands the outline with example requirements from Simplified Aid for EVA Rescue (SAFER) and the Assured Crew Return Vehicle (ACRV).

Appendix A: Related Publications
Appendix B Acronyms

HOW TO READ THIS GUIDE

Each section of the guide is fairly independent of the others. However, before anyone attempts to write or manage requirements, they need to understand about project scope. Therefore, it is recommended that all readers acquaint themselves with Section 1 before reading other sections.

This guide is applicable to all sizes of programs and projects. Small in-house projects will differ from large-contracted projects in the number of documents required. The differences are pointed out in the sections where the differences are relevant.

MANAGING REQUIREMENTS

If you are in charge of managing some aspect of requirements, then Section 2 will provide you guidance; Section 6 will provide further insight into how to set up your specification; Section 3 will help you understand your level and its relationship to other levels; Section 5 will enlighten you on other data you need to maintain in order to manage the process; and Section 4 will help you in ensuring that the requirements written are good requirements.

WRITING REQUIREMENTS

If you have the responsibility of writing some level of requirements, then Section 4 will help you to write good requirements. Section 6 will help you decide if you have all the requirements and where they belong in the specification. Section 5 will explain other data you need to record as you write a requirement.

REVIEWING REQUIREMENTS

If you are reviewing requirements, Section 4 will help you understand if the requirements are good or have problems. You may also need to read about reviewing requirements in Section 2. If you are uncertain as to the level of the requirements, then reading Section 3 will help to clarify this information. If you are uncertain about the document outline and content on this subject in Section 2 and Section 6 will help you.

VERIFICATION

While this guide does not go into detail on how to verify requirements, Section 3 addresses verification planning, Section 4 discusses problems in writing verifiable requirements, Section 5 discusses capturing verification data, and Section 6 provides the verification content of the specification.

EXAMPLES: There are a number of examples provided in this guide. A large number of them come from the Simplified Aid for EVA Rescue (SAFER) project and from the Assured Crew Return Vehicle (ACRV) project. Both of these projects are described in Section 1, Scope, which will help orient you to each project's content.

Section 1 -- SCOPING A PROJECT

Programs and projects arise in many different ways. The President may mandate a program, e.g., Apollo. The agency may identify a need, e.g., Space Shuttle. A project need may be created by a larger program, e.g., having a new space vehicle may create the need for a training system. An idea may be conceived by an engineering group for doing things a better way, e.g. Simplified Aid for EVA Rescue (SAFER). Other projects may be driven by external drivers or constraints, e.g., to reduce administrative cost a common payroll system for all NASA centers is needed.

Before a project team can write requirements, the project must be scoped. The scope definition is done in Pre-Phase A or Phase A. It culminates in document called a Program Plan, PP, that all personnel will use to guide the studies, analysis, and requirements definition of the project. A description of what must be done is provided, followed by examples from real projects.

The term Program Plan will be used throughout this guide. Organizations have various names for this document, e.g., the Navy calls it the Operational Requirements Document (ORD) and in-house projects at JSC call it a Project Management Plan (PMP).]

WHAT IS PROJECT SCOPE

This section describes the information that must be defined and documented in order to scope a project.

WHY SCOPE IS NEEDED

It is essential for each project to clearly define and document its scope so that the project can move forward in a coordinated manner, approval and funding can be obtained, and requirements written. This guide is primarily focused on the latter -- writing requirements.

All individuals who will write requirements, review requirements, or create designs will create their own scope definition if you do not give them one. Each individual's scope definition may differ significantly from that of the project management and will surely differ between the individuals. The result is requirements written for very different goals, objectives, constraints, assumptions, operational concepts, and systems. Battles will be fought, not about requirements, but about these very basic precepts. The requirements will be incomplete or conflicting, result in increased costs and schedule overruns, and not deliver what is needed.

WHAT IS SCOPE

Scope bounds the project by defining:

1. Need for the system
2. Goals, objectives, and constraints
3. Mission and operational concepts
4. Assumptions
5. External interfaces for the system
6. Major parts of the system
7. Budget
8. Schedule
9. Management authority and responsibility.

(1) The *need* must be the first thing defined and is generally a very short statement. It may be as simple as *"Place a man on the moon and return him safely, within the decade."*

All other items will evolve over time as ideas are conceived, tried, modified, or rejected. Some may be given, e.g., *the need defines the maximum schedule - 10 years.*

The next items (2 - 4) are shown in an order, but in fact, they are developed through an iterative process. Goals, objectives, and constraints may be related to both system definition, i.e., what the system will do, and project management, i.e., how it is to be done.

(2) There may be one or many *goals.* A goal states what you want to accomplish, e.g., *reasonably comfortable quarters for the crew during transport and on the lunar surface.*

Objectives generally expand upon goals. Objectives are often how you will accomplish the goal e.g., *develop crew module with space for moving about.*

Constraints define any limitations that must be placed on the system development or the project management; e.g., *provide the development of the crew module within x-years and for x-dollars.*

(3) Missions describe how the system will be used; e.g., *crew will descend to lunar surface and explore.*

Operational concepts will be defined to provide a basis for all studies and trades. There may be multiple operational concepts in consideration early in the program; e.g., *combined crew/service module versus separate crew module and service module.*

(4) During this process a number of *assumptions* will be made. These need to be documented, i.e., *three persons are needed for a lunar mission.*

(5) The systems external interfaces need to be considered, they will often result in a constraint, e.g., *use the Mission Control Center at JSC to control the flight to the moon.*

To define this information, it is necessary to conduct a number of trade studies and analyze possible options, constraints, and development risks. This may require few people and a short time or large numbers of people and a lengthy period of time. The size of the project, the number of external interfaces, and the risk involved will drive what is required.

(6) As the concepts are refined and studies made, the *parts* of the system will evolve; e.g., *two-stage launch vehicle, crew module, service module, lunar ascent/descent modules.*

(7 - 9) Budgets, schedules, and management must be defined. The first two items will either be constraints and defined by external sources, e.g., Congress, or will evolve with the design of the system. Management control must be defined so that all participants know the chain of command.

A preliminary set of this information may be available to begin Phase A if there has been a Pre-Phase A activity. Phase A will result in a documented Program Plan (PP) that contains the information to begin Phase B design work.

As design studies are conducted, changes may need to be made to the PP. The reason that changes will need to be made include:

*Inability to meet a goal or objective
*New objective from external source
*Lack of budget to accomplish all goals
*New technology opens new possibilities
*Expected new technology cannot meet schedules.

All changes need to be documented and made available to all personnel. Everyone needs to stay on the same track throughout the development of the system. To ensure this, you must document and distribute the information.

DOCUMENTING SCOPE

The purpose in documenting the scope is to ensure that everyone that signs up to do the job has signed up to do the same job. Not only must a document exist, it must be made available. This is often overlooked and many people working on a project have never seen, and do not even know, that a scope document exists. It is of little use if it is not read by everyone. Even if you are managing a very small project, you need to document its scope to ensure success.

A formal scope document, is essential to keep the project on track. It should be produced as early as possible and updated as changes warrant. The content of the document includes the list of ten items stated above and is the same whether you have a multi-billion dollar program or a small project. This guide will refer to the scope document as the Program Plan (PP).

> ***The Program Plan should be short, succint, and available.***

NASA Headquarters requires that each major program have a Program Plan (PP), defined in NMI 7120.4B. At JSC the Engineering Directorate has developed a Project Management Plan (PMP) for this purpose. Some other NASA centers have their own individual scope documents. Each DoD area, e.g., Air Force or Navy, has its own scope document. While formats of these documents may vary, the contents are essentially the same.

Developing a PP is necessary if the program is initiated at NASA Headquarters. The PMP is essential for in-house JSC Engineering Directorate projects. Though not officially required for some endeavors, you will do well to create a PP regardless of what type of project or sub-project you are managing.

For example, your responsibility may be to develop a training simulator for some portion of a large program. You will have many constraints and requirements placed on you by the program you are to support. These are necessary but not sufficient to describe your project. You must develop your own operational concept and objectives in order to meet the higher level requirements and constraints. By developing a PP for your project, you will capture this and other information necessary for you to do a thorough job.

FLOW DOWN OF INFORMATION

For contracted projects the information in the PP is flowed down to other documents as shown in Table 1. The table shows the major items from the PP and indicates that they will appear in the System Specification or Statement of Work (SOW).

Program Plan	Spec	SOW
Need	*	#
Goals, objectives, constraints	*	#
Mission/Operational Concepts	*	#
Assumptions	*	#
External Interfaces	*	#
Major Parts	*	#
Budget		*
Schedule		*
Management		*

Table 1. Flow down of information

An * indicates that the information will appear. An # indicates that it will drive information that appears in the document. For example, the operational concept will appear in the specification and it will not appear in the SOW, but will be reflected in trade studies or risk assessment tasks that the contractor is required to perform.

The information may be an exact repeat or it may be modified in the flowdown. The contractor does not receive the PP and therefore has no other access to this important information.

If you are developing an in-house project all of this information can be distributed to everyone. You will not need to repeat this information in other documents. The PP can contain the detail normally found in a Statement of Work and avoid having another

EXAMPLES

This section provides examples of Needs, Goals, Objectives, Constraints, and an Operational Concept for:

*Large program -- Space Shuttle
*Large project -- Training System
*Small project -- Simplified Aid for EVA Rescue (SAFER)

Other related information about defining the next level and interfaces is described in Section 3.

Although the following examples deal with the need, followed by goals, objectives, constraints, and finally by an operational concept, it should be apparent that the development of these items do not occur in series, but occur in parallel with many iterations.

EXAMPLE OF A LARGE PROGRAM -- SHUTTLE

The Space Shuttle Program began with a concept of Dr. Maxime Faget, then Director of Engineering and Development at the NASA Johnson Space Center. He initiated a preliminary design team on April 1, 1969.

NEED

Needs may be defined on a national scale in support of the charter of the agency or organization, such as:

***Provide a more cost effective method of putting payloads and people into low earth orbit.**

GOAL

A goal is a statement of what you must do to meet a need. To meet this need, you must consider the reasons you might want to go to low earth orbit, the different things you could do there, and how to do these things more cost effectively. The Shuttle goals focused on being able to do what a number of other systems had done in the past. The Shuttle would put man in low earth orbit (as did Gemini), it would allow on-orbit experiments and earth observations (as did Skylab), and it would carry payloads to low earth orbit (as do expendable launch vehicles). It also considered new tasks, such as deploying and retrieving a payload or carrying people and supplies to a space station.

Shuttle goals included:

***Reusable launch vehicle**
***Reusable entry vehicle**
***Vehicle to accommodate living and working in space**
***Vehicle to transport many different types of payloads**
***Ability to deploy and retrieve payloads.**

OBJECTIVES

Objectives are a further expansion of what must be done to meet the need and achieve the goals.

The Shuttle objectives included:

- **Develop a reusable engine.**
- **Develop a reusable heat shield.**
- **Develop an airplane-like space vehicle that can return and land on a runway.**
- **Determine the possible payload types and design a cargo compartment to accommodate.**
- **Determine the possible in-vehicle experiments and design a work area to accommodate.**
- **Develop a mechanism to remove and replace items in the cargo bay while in orbit.**
- **Design a system that allows crew egress and ingress to the cargo area and to perform other EVA activities.**

CONSTRAINTS

Constraints put restrictions on the project. Few constraints were placed on the initial Shuttle design. One constraint was:

- **Use the launch facilities at KSC.**

MISSION STATEMENT

The mission statement expands upon the goals and objectives. For simple systems, no mission statement is necessary. For larger systems, or for systems to perform multiple missions, the mission statement can clarify the goals.

For the Shuttle, the mission statement would be an expansion upon the goals that relate to what will be done on-orbit.

The Shuttle will carry and deploy satellites that can be boosted to other orbits. The crew and/or the ground will be able to test a satellite prior to deployment to ensure that it is operating properly. If problems are encountered that the crew or ground cannot overcome, the satellite will be returned within the Shuttle for repair on earth.

OPERATIONAL CONCEPT

The operational concept may exist before the need statement. Someone will decide it would be nice if we could do things differently -- and develop possible scenarios. Then the feasibility is examined and more ideas explored. Within NASA this occurs in pre-Phase A. Eventually enough is done that a needs statement can be written.

The operational concept, like all other parts described above, is iterative. The initial operational concept for Shuttle was this:

Two vehicles would be built, and both would be manned and winged. The first stage, a reusable booster would lift-off from the launch pad with the orbiter attached to it in a piggy-back fashion. (Imagine the Orbiter on the 747 stacked nose-up for lift-off). The booster would perform a first stage burn then release the orbiter which would burn its engines to achieve the desired orbit.

Both vehicles would have similar aerodynamic shape, both would have a crew to control them for landing, both would use the same engines, more in the booster than the orbiter, and both would use the same reentry heat shield.

After deploying the orbiter the booster would turn around and return to land on a runway at KSC to be refueled and readied for another launch.

The orbiter would burn its engines to achieve its desired orbit. The orbiter would have crew accommodation quarters, an internal experiment area, a cargo bay, and a means to access the cargo bay for EVA. It would have a remote manipulator for grappling payloads.

The crew would perform their duties for a week to ten days and would then return to earth, land on a runway, and the vehicle would be readied for its next mission.

Dr. Faget, the Shuttle architect, started the preliminary design team with all of the information stated above, and more. He had a model of what the orbiter and booster would look like, an approximate size and weight for the vehicles, and an approximate size and weight requirement for cargo.

CHANGES

As the Shuttle design progressed, a number of significant changes had to be made. To fund the project, the DoD was solicited as a customer. They added a 750-mile cross-range objective, which drove the orbiter wing configuration. They added a 60-foot payload objective, which required that the payload bay be lengthened, and shortened the area available for crew living space and for conducting experiments.

The available budget did not support a reusable orbiter and a reusable booster. The reusable booster was discarded. This drove massive changes, including removing the fuel from the orbiter and creating an external tank that violated the reusability goals. The solid rocket boosters were designed to be reusable to meet the same goal.

The use of engines for landing was discarded along with a requirement that the vehicle be able to handle a wave-off.

ASSUMPTIONS

A number of assumptions are usually made when starting a new program. Several in Shuttle were:

- **A lightweight reusable heat shield was technically feasible.**
- **A new engine could be developed that would be reusable and provide the needed thrust.**

Had either of these assumptions proven wrong, the program would have had to take a significant step backwards and start over.

EXAMPLE OF A LARGE PROJECT - TRAINING SYSTEM

This is a hypothetical case and no attempt is made to completely develop all goals, objectives, or constraints, only to provide examples.

NEED

Needs are created as different levels of a project are defined. For example, the existence of a manned space rescue vehicle creates a need for training of the crew and flight control personnel. The needs statement is:

- **Develop a training system for preparing flight crew and flight control personnel in the operations and maintenance of Vehicle X.**

GOALS

In developing the goals, you consider past experience, the cost and time associated with training, the characteristics of the vehicle, and the operations that will necessitate training.

The training system might have as its goals:

- **Capability to train the crew and flight control personnel simultaneously.**
- **Capability to train crew and flight control personnel independently.**

These statements are not contradictory, they are requiring two different scenarios to meet program needs and avoid past problems.

OBJECTIVES

To meet the goals there are a number of possibilities. These include complete real-time simulations, part-task trainers, classroom training, computer-based training, mock-ups, and possibly a number of other options.

The objectives might be:

- **Trade-off different types of training methods to reduce training time and cost.**
- **Define a system that can evolve as changes occur to the vehicle.**
- **Define a system that can be easily reconfigured for different missions.**

CONSTRAINTS

A constrain might be:
- **Use existing facility space.**

OPERATIONS CONCEPT

In this case, a great deal must be known about the vehicle, the tasks of the flight crew, and the tasks of the flight control personnel before the concept or the goals and objectives can be written. Failure to have this information will require too many assumptions and almost always leads to a more expensive system that will be revised repeatedly.

The operations concept is that Vehicle X is an emergency return vehicle. It will carry a maximum of four crew members. It is being designed to minimize crew intervention, i.e., nearly autonomous. The flight control personnel will have voice contact with the crew while they are in the vehicle. The flight control personnel will only be called to support when the vehicle is needed, which is hopefully very seldom. Two concepts will be considered.

Traditional Concept.
- A mock-up for doing ingress, egress, and component replacement.
- A flight simulator that includes the actual flight software, crew displays and controls, and dimensions of the vehicle.
- Interface to the control center for running simulations with flight control personnel.
- Computer based training for separate training of flight control personnel.

New Concept
- Develop trainer capable of meeting all objectives and independent of control center.
- Perform mock-up and flight training in a single trainer.
- Utilize latest computer based training technology to simulate vehicle and ground systems.

As the design progresses, the two concepts will be further explored and trades made to select one, or some combination, for final design and development.

ASSUMPTIONS

Assumptions for the trainer will include specific values for the following:
- Amount of time it will take to train the different groups of people.
- Frequency of refresher training
- Number and type of instructors required.

EXAMPLE OF SMALL PROJECT -- SAFER

SAFER is a small in-house project being developed at JSC. It will initially fly on a Design Test Objective (DTO) now scheduled for launch in September 1994.

NEED

Needs also exist for much smaller projects. NASA engineers felt that a contingency maneuvering unit was needed for station operations and could provide a means to solve some potential Shuttle problems as well. They had experience with the manned maneuvering unit (MMU) developed for Shuttle. They felt that the MMU was too large and expensive to meet these contingency needs. They conceived of a smaller simpler system, SAFER.

The SAFER project has this as their needs statement:

- **A capability for an EVA crew member to reach all external Shuttle locations.**
- **A capability for an adrift EVA crew member to return to Station.**

GOALS

The goal for SAFER is:

- **Provide a small simple EVA crew maneuvering system which meets both Space Shuttle and Space Station needs.**

OBJECTIVES

SAFER Objectives are:

- **Establish a common set of SAFER requirements to meet both Space Shuttle and Space Station program needs.**
- **Develop a flight demonstration SAFER.**
- **Validate SAFER system performance on an early Shuttle flight.**
- **Develop a flight SAFER for both Space Shuttle and Space Station.**

CONSTRAINTS

A SAFER constraint is:

- **Flight production project initiated after successful completion of flight demonstration project.**

OPERATIONAL CONCEPT

The SAFER operational concept is:

Space Shuttle

- Carry SAFER on every flight.
- In case of an in-flight emergency, use SAFER to get EVA crew member to the problem area (e.g., External Tank Doors or a damaged tile or windshield).

Space Station

- Each crew member wears SAFER during all EVAs.
- In case of adrift EVA crew member, use SAFER to get EVA crew member back to Station.

Flight Demonstration Mission

- Perform a scheduled EVA with 4.75 hours allotted for SAFER activities.
- Two EVA crew members flight test SAFER.
- Evaluate overall system performance and both Shuttle and Station Mission scenarios.

SUGGESTION

It is often difficult to differentiate between goals and objectives, if you are having a problem then just put the topics down as goals or objectives and do not waste time determining which is which. It will eventually become clear.

Section 2 -- MANAGING REQUIREMENTS

This section addresses the management of requirements from the beginning of the project through the life-cycle. There are six parts to this section:

- Managing the Project Beginning
- Types of Requirements
- Managing Requirements Definition
- Reviewing Requirements
- Requirements Maintenance
- In-House versus Contracted Programs

Although this section refers to projects and systems, the management steps are applicable at each level. That is, if you are responsible for the development of an element, the same steps apply as if you were responsible for the system. The major difference is the amount of design that has taken place and the constraints that will be imposed upon you because of completed work at higher levels.

MANAGING THE PROJECT BEGINNING

If you are the person in charge of the beginning of a project, or of the beginning of a part at any level, it is important that you understand the material contained in *Section 1 -- Scoping a Project.*

WHO

For large programs, the responsibility for the development of the Program Plan resides at NASA Headquarters. An individual will be responsible for ensuring that the PP is developed. The PP is essential to program authorization.

For smaller projects the PP will probably be developed by a system architect. This is the person with the idea who wants to get a project started. The project will need approval and funding, and thus the plan is essential.

This early work, which may be done in pre-Phase A or Phase A, with or without contractor support, must have an individual manager assigned to get the job done.

PROGRAM PLAN

Since the PP is essential to all that comes later, it is important that the manager assure that all inputs have been obtained, all information documented, and the PP distributed to all who will need it for the steps that follow.

If changes occur to invalidate or change any part of the PP, it is the manager's responsibility to provide the information to all parties involved and to solicit feedback from them on the impact to their effort.

REQUIREMENTS

As the requirements are defined, you want the authors to provide traceability back to the PP. You will then review the requirements to determine, if in fact, they do trace to the PP, or if there are omissions, conflicts, or gold-plating.

If you are somewhere lower in the hierarchy and trying to write requirements, then you need to look to the management of the project for the definitions that allow you to write good requirements. If the information does not appear to be documented, then you need to make assumptions and create your own set of information. Present this to the project management for concurrence before writing requirements, or you will be rewriting requirements a number of times.

TYPES OF REQUIREMENTS

After the Program Plan is complete it is time to weed out alternate concepts, perform trade studies, and define system requirements. During this period many designs may be considered, prototypes may be built, and other engineering studies conducted to resolve risks. The effort may be undertaken by the government alone or with the assistance of contractors. In either case, the government is responsible for the definition of the system and program requirements.

SYSTEM & PROGRAM REQUIREMENTS

Each project will have two types of requirements -- **system** and **program**. System requirements drive the design of the system. Program requirements define tasks. Both types will be defined in the PP, but all other documents will separate the two types.

System requirements are documented in a Requirements Specification and take the form:

- The System shall perform ...
- The System shall be capable of...
- The System shall provide ...

Program requirements are documented in a Statement of Work (SOW), and take the form:

- The Contractor shall deliver the set of documents defined in....
- The Contractor shall perform design trades for...
- The Contractor shall analyze...

[In-house projects may expand the PP to include information contained in the SOW and avoid having this document.]

It is important to keep the system and program requirements separate. The system requirements must be verified before the product (the system) can be accepted by the government. The program requirements are managerial and often subjective. Both are necessary to manage a project.

MANAGEMENT

These two types of requirements are related. A common failure is to have one group define the system requirements and another the program requirements, with no coordination and no checks to ensure compatibility. This will cause confusion and increase cost and risks.

As requirements are written, the related tasks also need to be documented. For example, if there are risks identified in certain areas, you will want the contractor to undertake a task to assess the risks and alternatives. If you have defined margins of safety that are dependent upon the design for detail definition, you will need a task to do the analysis and document the results.

Each requirement you write, whether in a specification or in a SOW, is going to cost money. Therefore it is very important that you consider the ramifications of each one before you baseline or release the documents.

The System Specification and the SOW must be responsive to the Program Plan and must be tightly integrated.

MANAGING REQUIREMENTS DEFINITION

In order to manage a project, one must manage the requirements. Costs, schedules, resources, and other management factors are requirements driven. The quality of the requirements will also drive all these factors. Poorly defined requirements will result in rework and throw away work -- driving both cost and schedule performance beyond budgets.

ESTABLISH A PROCESS

Requirements will evolve even as the PP is being developed. Actual documentation of the requirements, in any form, will probably not occur until there is a Program Manager named. Managing requirements is a primary Program Management function. The Program Manager may do the job or may appoint someone to head the effort -- the chief engineer or a system engineer may be assigned this job.

Developing system requirements will require inputs from a variety of people. Assembling the correct team is crucial to this step. A popular term these days is *Concurrent Engineering* .

Concurrent Engineering is a management approach that recognizes all phases of the life-cycle. It emphasizes obtaining early input from all life-cycle participants. The participants may include customers, developers, technical discipline specialists, operators, quality, safety, reliability, maintenance, and verification personnel.

The first step is often to assemble the team of personnel. This should be the second step -- the first step is to set up the process for managing requirements.

The process includes the definition of who is responsible for collecting requirements, the schedule, the format for presentation of proposed requirements, and the review process for a requirement.

Now you can assemble the right team and educate them:

- About the PP
- On the importance of requirements
- On the process
- On what is a good requirement

DEFINE THE SPECIFICATION OUTLINE

A great deal of time can be spent in determining the structure of the specification. A great deal of time can be wasted in repeatedly revising the structure. That is why there are standards. Your job is to develop a program or project, not deal with document structure.

If you use the system specification outline provided here (Section 6) and modify it to meet your needs, you can escape some of this problem. Using the outline will help you with contracted work, because this is the outline that a contractor expects for a system specification.

The system specification is applicable to many different kinds of systems, e.g., missiles, airplanes, management information, as well as segments, elements, and end items. If it is not applicable to your work, then you may want to use another standard. If you are developing a pure computer or software only system then you may want to use an IEEE standard or MilStd 2167A. The point is that there are standards and you do not need to reinvent the wheel.

Specification Content

The specification outline gives you some guidance on content, but you need to give your team more detailed instructions, including:

- **expanded outline**
- **descriptive lead-ins to sections**
- **design goals.**

Expanded Outline. The following are suggestions for how to expand the outline by considering the operational aspects of the system - (1) operational phases or (2) operators and users.

(1) For systems that have distinct operational phases, expand the outline for each phase. You will have very specific requirements imposed upon your system by the different phases: e.g., pre-launch, launch, on-orbit operations. Collect the requirements along these phases to make the job simpler.

The following is and example from the ACRV system specification:

Pre-launch Operations

- *The ACRV System shall be capable of integrating the flight elements with the launch vehicle*

.

Launch and Delivery Operations

- *The ACRV System shall be capable of transferring a passive ACRV between the launch vehicle and the Space Station.*

Attached Operations

- *The ACRV System shall be capable of attaching the ACRV to the Space Station.*
- *The ACRV System shall be capable of separating the ACRV from the Space Station without Space Station assistance.*

(2) For systems that have different users and operators, expand the outline for these different groups. In an administrative computer system you may have requirements related to maintenance personnel, system operators, administrative and clerical users, and management. You may have requirements from the training personnel as well.

The best break-out for your system may be very different than either of these examples, but if you examine the system from the operations perspective, you will find the approach that will work best.

Lead-In Sections. Major sections of your document need lead-in descriptions to acquaint the reader with the purpose of the section. These lead-ins should generally be no more than a paragraph in length, and should help the reader understand what is to follow. You do not want this description in front of every paragraph, only at the beginning of major sections. You need to determine what type of information you need and then give the authors instructions about preparing the data -- when it is required and a good example.

The following are a few examples of lead-in sections from **ACRV**:

3.2.6.2 Induced Environment
The induced environments to which the ACRV System will be exposed include those resulting from attachment to the Space Station and those resulting from flight in the Orbiter.

3.2.8 Storage
This section contains the requirements for storage of ACRV System hardware on the ground.

You need to determine a general outline for your requirements specification, such as the Specification Outline in Section 6, and you need to expand the outline to cover your particular system, segment, or element. You want to use this outline as a checklist for assuring that you define all of the requirements. You may not create the formal document until you have collected and reviewed the requirements.

Design Goals. In addition to requirements, your specification may contain design goals. It is important that you clearly indicate what are goals versus what are requirements. As a minimum, a goal will use the word *should* and not the word *shall*.

The ACRV program kept their design goals separate and placed them Section 6.0 of their System Specification, which is the only non-binding section of the specification. They also did not use stand-alone vague generalities, e.g., minimize cost. When they stated that they wanted to minimize cost, they provided the detail shown here as an example:

The ACRV System life will be for a 30 year period. The total life-cycle costs must be considered in the design, to prevent excessive operations cost of the life of the project. The following are intended to reduce life-cycle costs:

- *Consider increasing design margins above standard aerospace practices to increase reliability without the need for extensive testing or analysis.*
- *Minimize the need for unique GSE.*
- *Maximize the use of existing KSC facilities... Maximize the use of existing available government transportation systems.*

DEFINING SYSTEM REQUIREMENTS

During the project beginning, the system was defined in fairly broad terms. It is now necessary to write requirements for the system so that a system design can be created to meet those requirements. This is an iterative process.

Your PP may have provided multiple operations concepts. It may have cited a number of risks that must be addressed in order to proceed with design. Thus, you cannot just write requirements, but must perform design trades and risk analyzes, to narrow options and define requirements.

It is very useful to work from a checklist to define requirements so that nothing is overlooked. If the system is a new space vehicle, everyone is aware that they must write functional (what the vehicle must do) and performance (how well it must do it) requirements to describe the vehicle, but these are only a part of the requirements needed to bring the system into full operations. The system will also have requirements for such items as:

- Interfaces
- Transportability
- Facilities
- Reliability
- Maintainability
- Safety
- Environmental Conditions
- Design and Construction Standards
- Logistics
- Personnel
- Training

Using a standard outline, expanded for your system, will help you in capturing all the requirements.

COLLECTING REQUIREMENTS

An effective means of collecting requirements is to have each written on a bullet chart or standardized form. For each requirement, you will want to know its parent requirement(s) and its rationale, i.e.,

- Why needed
- Assumptions
- Design trades
- How related to expected operations

If at the top level, the parent requirement will be in the PP. In this case the parent might be a goal, objective, constraint, or a high level requirement. If at a lower level, then the parent requirement will be in the next higher specification or in a reference from that specification, e.g., a safety standard.

This bullet chart or forms process works well because individuals are not forced to create sentences and they are looking at requirements, or groups of requirements, and not being overwhelmed by an entire document. Most organizations have change request forms that can be used for the initial collection of requirements. The form should contain space for the following data:

- requirement text
- parent requirement (reference number or text) rationale
- author's name
- verification phase/method
- document section where requirement belongs
- control data, e.g., date, tracking number, etc.

The initial form may not contain the verification data, but it needs to be determined before you baseline the requirement.

VERIFICATION PLANNING

Section 4 in the system specification is used to describe the verification approach for your system. Defining this approach as you define the requirements will help to ensure that the requirements can be verified. As you write each requirement also document the phase(s) or level(s) where the requirement will be verified and method(s) that will be used. Discussions about the verification should take place before the requirement is baselined.

These discussions will help clarify the requirements. If a requirement is somewhat vague, then trying to determine how to verify it can force more details. If the requirement is very restrictive, then these discussions will expose the difficulty of the verification required, and perhaps relax the requirement.

Verification can consume a large percent of program costs. Ask yourself if your requirement will demand new facilities and equipment in order to test it. It may be possible to relax the requirement and use existing facilities and equipment. If the requirement cannot be relaxed, then the new facilities and equipment need to be described in the specification and included in all cost projections.

You need to consider the implications of verification when writing the requirement--not when the design is complete or test ready to begin.

Some verification problems occur because of how we state numeric limits in requirements. Often the numbers are stated very exactly, e.g., *The system shall operate at 2 psi.*

This requirement means that the system will be tested to prove that it can operate at exactly 2 psi. If there is any variation, you will have failed verification. That is what should happen if the number must be exactly 2 psi. But if it could be 2 psi +/- .002 psi, then you have overstated the requirement.

This overstatement may increase the testing costs -- you will need very exact equipment to measure 2 psi but may can use slightly less exact -- and hopefully less expensive -- equipment to measure the 2 psi with a variation. If you fail the test you will spend money to either get a waiver (lots of meetings and lots of paper work) or you may have to change the system to meet the requirement (redesign).

Over-specification, often unintentionally, is a major verification cost driver.

REVIEWING REQUIREMENTS

A team of people will be writing requirements and part of your job is to manage the activity. It is important to have a good review plan and process in place to do this. Many review processes consist of publishing a specification and writing RIDs. This is a costly, timely, and ineffective process. At some point you are going to do this, but it is not the first review process, nor even the second.

If, instead of creating a formal specification, you have all requirements submitted as suggested above, you handle requirement reviews in smaller sets than a full specification. Everyone does not need to attend meetings to review requirements. The following are suggestions for approaches to reviewing requirements.

INTERNAL REVIEW APPROACHES

The first approach uses a system engineer or system engineering organization to review and coordinate the requirements. This adds the technical breadth that is needed to recognize conflicts among requirements from different areas.

System Engineering

- All requirements submitted to system engineering.
- System engineering reviews and determines other information needed and makes first approve/disapprove decision.
- System engineering presents results to management (possibly by document section).
- Management makes comments for changes.
- Results of this are then released to entire team for review -- requirements still on forms or printed out with rationale, parent, and other data available to reviewers.
- Reviewers redline and return forms (or do on-line review/comment if you have automated process).
- System engineering reviews this input identifies any issues.
- Issues discussed with management.
- Meeting held to review issues with entire team.

The second approach uses a coordinator and puts more work back on the individual authors and the team. If you do not have someone to do the system engineering job, you may have to operate in this manner.

Coordinator

- All requirements submitted to coordinator.
- All requirements distributed by coordinator to entire team.
- All team members review and return comments to coordinator.
- Coordinator tracks requirements and responses for management.
- Coordinator provides feedback to authors based on responses.
- Author is responsible for getting with anyone who has question or concern and trying to resolve.
- Coordinator maintains list of issues - author must state when resolved.
- Coordinator provides all requirements plus list of all unresolved issues to management.
- Management reviews requirements and issues and their list of concerns.
- Entire team meets to discuss and resolve issues.

If you have the process automated, you can have all proposed requirements and their related data on-line. Reviewers can work from the on-line copy and provide comments on-line. Reviewers can see each other comments and preclude redundant information. When updates are made, everyone can see the results immediately. When the issues are resolved, all the information is ready to be published in the form the team and management have agreed upon.

The processes above are for handling a majority of the requirements. For these, single authors or small teams can get the work done without any large meetings. For some of the requirements, there is a need for meetings which include management and possibly the whole team, plus others. These are the controversial requirements, or ones where a value is very critical.

Large meetings where every requirement is read and discussed are generally a waste of a great deal of time. You will have this situation if you cannot provide the team with good information. The forms will be a big help -- they can see more than just the requirement. Having the information on-line where everyone can stay current is another important factor in reducing the number and size of meetings. Many people attend only because they do not think they will know what is happening otherwise.

REVIEW CHECKLISTS

Anyone reviewing requirements, their own or someone else's, need to consider some basic questions about each requirement and then about the requirements in relation to the document

Is the requirement:

- Necessary? -- If you omitted the requirement, would it make any difference?
- Stating **What** or is it stating **How** (implementation)? -- If it is implementation, then ask **Why** and get the real requirement defined.
- Verifiable? If it is not verifiable can it be fixed with word changes or is there are larger problem to be worked?
- Clear, concise, grammatically correct?
- At the correct level or leaping to a lower level?
- Responsive to its parent requirement(s)?

Does the document:

- Contain the correct requirements in each section?
- Include no conflicting requirements?
- Include all known requirements, including those with values to be determined (TBD)?

For further items to check in reviewing requirements, see Section 4 - *Writing Requirements.*

After your internal review you should have a set of requirements that you believe are about as good as they can be at this time. Before you release them, either as forms or as a document, have a technical editor review to clean up the grammar, punctuation, and spelling. It is counterproductive to get external review responses to editorial items. Either system engineering, authors, or management need to check the editorial changes -- if grammar changes are made, the entire intent can be lost. Now you are ready for an external review.

EXTERNAL REVIEW

This external review is important, and may include a review by contractors as well as your company. You should have a say in who the reviewers are to be and how you will accept comments. If there are a number of organizations involved, have each organization do a screening before submitting their comments. You do not need to be a referee for someone else's organization.

Again, you should have a process defined that covers:

- Schedule
- Participants
- Information distribution
- Instructions for review
- Format for responses
- Screening of responses by system engineering or management
- Presentation of response issues
- Meeting with participants to resolve issues

Your review instructions can recommend that all editorial comments be contained in a single response. Some people think that filling out a form for each minor editorial comment is appropriate, but it creates a lot of paper for information that does not warrant it.

Again, if you are equipped for an on-line review, you can reduce the redundant comments and curtail the paper.

The purpose of the definition and review processes is to produce a specification that is:

- Free of design -- state *WHAT* not *HOW*
- Maintains the correct level of detail -- these are system requirements, not subsystem
- Captures all the necessary requirements -- use a standard checklist
- Captures a justification for every requirement
- Ensures that all requirements are verifiable
- Allows only *NECESSARY* requirements

REQUIREMENT MAINTENANCE

Your System Specification will be released as a part of the Request for Proposal (RFP). It will not be baselined, this occurs midway into the Design Phase after more technical information is available. If you have done the review process well, then the draft specification you release will provide the basis for obtaining good proposals to develop your system, segment, or element. The following covers aspects of managing requirements for the RFP release and thereafter.

RFP MANAGEMENT

It is critical that the Specification you release be consistent with the Statement of Work (SOW) and that both documents be responsive to the Program Plan (see Figure 2-1). If you do not ensure these items, then the proposals you receive will contain inconsistencies and problems that will cost you money.

Even while the proposal process is active and you are cut off from the contractors, you need to be continuing to update and analyze your requirements. Your in-house team will conduct studies, events may occur which impact the requirements, and you may feedback, in the form of questions, from the contractors.

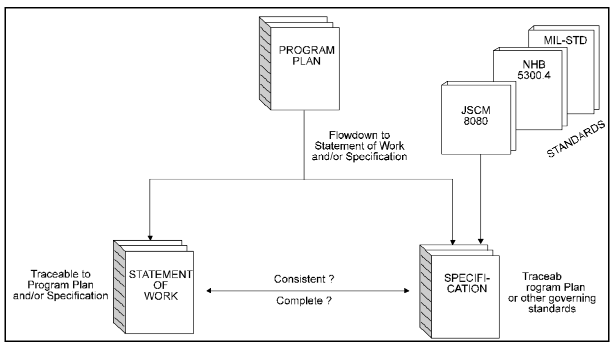

Figure 2-1 - Consistency and Completeness in RFP Process

If the proposal process is lengthy and major changes occur, you may need to work with procurement to either get the changes to the contractors, or to plan a period for an updated bid. Minor changes are not critical and can be handled at contract negotiations. However, major changes will make contract negotiations a nightmare for both sides and can result in protests.

All changes that are proposed during this period need to be assessed just as the initial requirements were assessed for need, quality, and clarity. When the contract is let and the contractor on-board, you will reassess the requirements with the contract team to make sure that there are no misunderstandings.

PHASE B MANAGEMENT

As the design evolves, the requirements will also evolve. You will need a process to manage the change to the requirements that will occur during this time period. Mid-way through the designing phase you will baseline your specification following the System Requirements Review (SRR).

Traceability. At this time, work will begin on lower level specifications at the contractor. Traceability becomes an important process to these activities. You will need a process to trace your requirements to the next level specifications, to the design, and to test plans and procedures (see Figure 2-2).

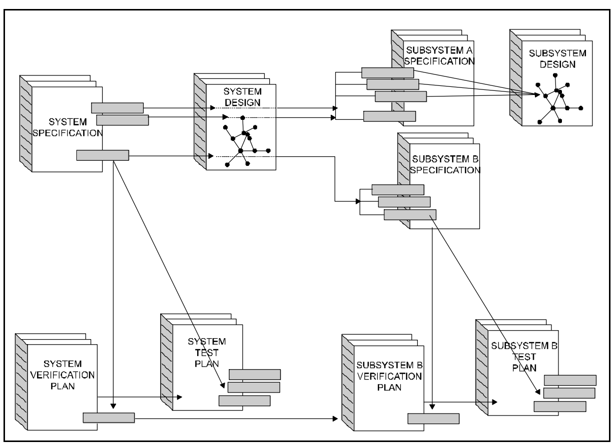

Figure 2-2 - Traceability

An important aspect of the Critical Design Review is that the contractor clearly demonstrate how the design meets the requirements. The verification planning must show how each requirement will be verified.

In the past all of this effort was manual. Large armies of people tracked the requirements and their traceability. In today's environment, this is neither practical nor affordable. Automation is essential to do this work successfully within budget constraints.

OTHER DATA

There are some other steps you can take at this time to ensure better proposals and a better design effort:

- Assign ownership to each requirement
- Prioritize all requirements
- Define the risk associated with each requirement.

Ownership. This may be the original author or another person. It needs to be the person that others will consult if there were questions, and the one who will assess the impact of a change to the requirement. Having ownership established makes it easier to find the person for consulting or change. It also makes it possible to transition to new personnel when the current owner is not available.

Priority. If you do not prioritize the requirements, the designers will define their own as they make trades. They will probably not do it to your satisfaction.

Risk. Some requirements are virtually without risk. Others may have considerable risk -- technical, budget, or schedule. By identifying the risk it is possible to keep focus on problem areas.

MANAGING CHANGE

Change is inevitable, system development is an iterative process. Although you will not have an official baseline until SRR, you must still control the document and its changes.

Again you need a process -- who is to manage, the format of proposed changes, the review and approval steps, and who can and must update the specification.

Whenever you conclude a change is necessary, make it as rapidly as feasible. The system requirements are influencing the design and the development of the next level of specifications. If you wait to change, even though you know the change is going to take place, you are allowing work to continue with incorrect information.

If you have assigned priorities to your requirements then you can handle change more easily. Those requirements with priority 1 should not be allowed to change without a lot of discussion -- else why did you make them priority 1. Those that are priority 3 were probably expected to change so you can be more lenient with their changes.

If you have assigned an owner(s) to each requirement, you know the person(s) with the knowledge and interest that need to be consulted prior to any change of the requirement.

If you maintain requirements traceability between your level to other levels, you will be able to assess the impact of the change throughout the system.

If you maintain traceability between your requirements and verification plans, you will be able to assess verification impacts resulting from proposed requirements changes.

MANAGING IN-HOUSE VERSUS CONTRACTED WORK

The steps and documents described above apply in general for both contracted and in-house projects. In-house projects need a PP, or equivalent document. They need to document their system and program requirements. The major difference is in the documentation. Since in-house participants have access to the PP, it is not necessary to repeat information in the Specification that is contained in the PP. It may simply be referenced. In-house projects may use an existing task-order system or Memorandum of Understanding (MOU) to define program requirements (tasks) as opposed to writing a separate SOW. In-house projects do not issue an RFP, but there is a similar milestone for proceeding with the design. Internally in Engineering at JSC, a Project Requirements Document (PRD), that is equivalent to the system specification, must be approved before proceeding.

Section 3 -- HOW SYSTEMS ARE ORGANIZED
LEVELS OF REQUIREMENTS

Few systems can be developed completely by an individual or even by a very small group of people. Today's systems are composed of complex hardware and software. They are driven by operational needs and the skills of the operators. Life-cycle cost must be considered from inception. Technology is so complex that experts in many areas are needed to ensure the system's success. It takes different types of management and technical skills to cover all aspects of the system design and development. The decomposition of the system into smaller and smaller pieces is tailored to involve the right skills at the right time. This decomposition will be referred to as levels. Figure 3-1 shows the levels that are used on programs. The number of levels required is dependent on project size. A large program will need all levels while a small project may have only the system and subsystem levels.

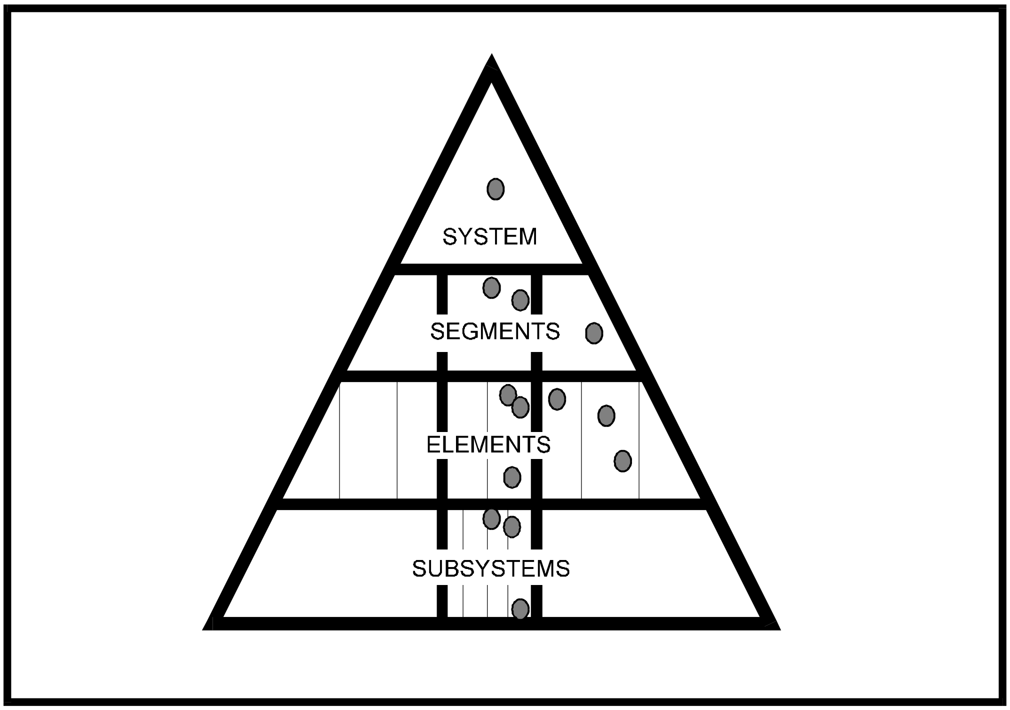

FIGURE 3-1 - Levels of a System

THE NEED FOR LEVELS

Levels are used to control the evolution of a program. A formal process provides for control as the system is decomposed to smaller units. This process is to ensure that each requirement is addressed in the design at the proper point in time. This section discusses the steps of the formal process and how a system evolves, including informal processes.

SYSTEM DEVELOPMENT STEPS

The development of a system involves the following steps:

System
- define system needs, goals, objectives, constraints, operational concept, and segments
- define system requirements
- perform system design

Segment (for each segment)
- define segment needs, goals, objectives, constraints, operational concept, and elements
- define segment requirements
- perform segment design

Element (for each element)
- define element needs, goals, objectives, constraints, operational concept, and subsystems
- define element requirements
- perform element design

Subsystems (for each subsystem)
- define subsystem requirements
- perform subsystem design

Parts
- define part requirements
- make or buy parts

Note that at each level the next level pieces are defined, e.g., at the system level the segments are defined.

At each level several steps are shown, however these steps are often performed by different organizations. The government may perform the first two steps under System, then contract for the system design. The system design contractor will do the first two steps under Segment, and then may contract separately for the design of each segment.

SYSTEM EVOLUTION

Each step shown above has program milestones for requirement reviews and design reviews. These are usually serial -- i.e., the steps flow from the first step through the last in what is generally referred to as a waterfall.

What is often confusing is that the work does not necessarily follow the waterfall. That is, subsystem requirements definition and design trades may take place before the system requirements and design are complete.

This may occur because there is a long lead item at the subsystem level that must be addressed early to meet delivery dates and to reduce program risks. It may occur because of the need to validate that the system requirements are realistic.

Low level requirements definition and design studies are necessary ahead of their logical steps for the circumstances described above. This work may be performed by the government or by a contractor. The work may involve the creation of prototypes.

The results of these studies will influence the overall system requirements and design and can change the higher level requirements or even goals and objectives.

The formal steps to define lower level requirements and design will still take place at their appointed time in the sequence. The purpose of the formal serial process is to ensure that all lower level requirements are traceable to, and consistent with, higher level requirements and to ensure that all requirements are met.

NUMBER OF LEVELS AND NUMBER OF DOCUMENTS

This section deals with determining how many levels you may need and then how many specification documents you may need. Another level does not necessarily mean another specification document.

At each level you will define the following level, e.g., the system level may define its next level as segments or as subsystems. You will also define the number and type of pieces at the next level; e.g. the system may define three segments - ground, flight, and operations. This section deals with how you define segments, or elements, or subsystems.

DEFINING LEVELS

How much is enough must be carefully considered at the beginning of the project and as you move to each subsequent level. The use of levels should assist the management of the program, not burden it.

At each level management will decide how many pieces will comprise the next level. This is not something you can decide on the day you start a project, this is part of the evolution of the project.

Consistency in numbers of pieces at each level is not required. One segment may have three elements and another six elements. One segment may have no element level, but drop directly to subsystems. The way you decide on the pieces, for example the segments, is ***very carefully.***

This decision is one of the most crucial you will make. It will help you to manage your project or drive you to distraction. Two things should drive the definition of pieces -- the interfaces between the pieces and the management and technical skills required by each piece. Both of these topics are covered later in this section. To effectively design and develop a system, you need to establish the levels and define the requirements and perform the design in the formal sequence stated above, but using only as many levels as your program requires.

DEFINING SPECIFICATIONS

There is a tendency to want to define the program document tree on day-one. The document tree needs to evolve with the project. Thus, initially your program document tree will contain only a system requirements specification and the specifications for the next level, e.g., segment.

The system specification contains all system requirements, describes the segment divisions, and contains some segment requirements.

Until you get into segment design you should not even consider what the next level of the document tree will look like, e.g., the elements -- it will fall out from the design. Each segment manager will determine his/her elements. The number of specifications you require is dependent on the management and control of each level.

Having a level does not demand a separate document.

You need separate requirements specifications when you are handing-off the management and control of a part to someone else. If there is no hand-off of management and control, then it is your option to create separate documents or not. This does not abolish the need for the different level requirements, it only affects the document in which they will reside.

If your project is small and internal then you could conceivably create only one document, a system specification. Your subsystem details could all be placed in this document. Just because you have this next level does not require that you write a separate document.

CONSIDERATION AT EACH LEVEL

At each level, management will determine the next level and its composition. If your project is large and complex, then as the system program manager, you may determine that the next level must be elements. You will probably have decided this based on the work to be done and who can do it. Thus your elements will be broken along management lines. You must also consider the interfaces between the elements before finalizing your decision.

MANAGEMENT

Consider the management and technical implications when defining each level.

There should be a good reason for defining the pieces and care should be taken in this process. The system should be divided in a manner that puts full management responsibility and control over each segment (or element, or subsystem) into a single organization.

If the segments are so convoluted that every decision has to go back to the system level for a decision you will be spending too much time and money getting the job done.

If we consider a lower level of breakout, like the subsystems, a critical area is technical skills. For example, a space vehicle will include structure, propulsion, power, navigation and control.

There are disciplines for each of these functional areas that contain the experts needed to design and build the subsystems. If the vehicle is comprised mainly of propulsion, with only enough structure to hold the thing together, you might have decide to have a propulsion subsystem that includes the structure and not have a separate structures subsystem. The propulsion subsystem is then responsible for not only tanks and thrusters but also for the structure that holds it all together.

If structure is large and complex and interfaces with many parts beside the propulsion subsystem, then a separate structures subsystem is in order. The structures and the propulsion subsystems will need to define their interfaces if they are managed by different organizational elements.

INTERFACES

Minimize the interfaces and keep them simple.

Interfaces are a major cost driver of all systems. To control costs you must minimize the interfaces and keep them simple. This statement would seem to be obvious, but it is apparently not considered on many projects until it is too late.

At one time interfaces were the connection of pieces of hardware - structure, wires, etc. These interfaces, while often complex, were simple compared to the current situation where software interfaces must be considered. Software at any level, can interface with everything else at that level.

Software does not have bounds in the sense that hardware does. The software may send commands and receive data from every other subsystem. It will probably have more interfaces than all other subsystems combined.

This is a relatively new phenomena. In the past airplanes were developed where less than 15% of the cost (read also complexity and interfaces) was software. Today, software can be 85% of the cost. Thus, a software, or avionics plus software, subsystem is in order.

If the software crosses elements then it may need to be an element. If it crosses segments, then it needs to be a segment.

DOCUMENTING INTERFACES

Interface requirements have a life of their own and often a document of their own. There are two types of interfaces, external and internal.

External Interfaces. The interface is between your system and another system. The other system is outside of your control.

Example: You are responsible for a payload system that is to be launched aboard an expendable launch vehicle. You must define the requirements imposed on your system, the payload, by the launch vehicle.

The launch vehicle's Interface Control Document (ICD) will define the as-built interface. Your system specification will have requirements as follows:

The payload system shall provide a structural interface to the launch vehicle in accordance with ICD xxxx.

The payload system shall prove a data interface to the launch vehicle in accordance with ICD xxxx.

The number and type of requirements will depend upon the amount of interface you and the launch vehicle require. If you have no requirement to provide data through the launch vehicle, and the launch vehicle does not require such a data transfer, then you will not need the second requirement.

Internal Interfaces. These are interfaces between the pieces that you have defined. In the system specification you must define the interface requirements for the segments. These requirements are high level and state what must interface.

Example: Your system has responsibility for developing and delivering a payload to geosyncronous orbit. Your segments include the payload segment and a transfer vehicle segment.

If both of these segments require development, then your requirements might include:

The payload segment and the transfer vehicle segment shall have a structural interface that enables attachment, launch checkout, launch, and separation as defined in IRD xxxx.

The payload segment and the transfer vehicle segment shall have a data interface for sending data and receiving commands as defined in IRD xxxx.

You will need to create an interface requirements document (IRD) to document the requirements for the interface. Your system specification will reference this IRD. As the design of the segments evolves and agreements are reached on the interface, then an ICD can be developed and referenced in lower level specifications.

If, you are using an off-the-shelf transfer vehicle, it will have an existing ICD and will require that the payload segment meet its ICD.

For small programs, it is not necessary to create a separate document for the ICD. It can be an appendix to the specification.

EXAMPLES OF LEVELS

The following two examples are at extremes in defining levels. The first is for a large contracted program, using the Shuttle as an example. The second is for a small in-house project, using SAFER as an example.

LARGE CONTRACTED PROGRAM SHUTTLE

The Space Shuttle Program system requirements are documented in NSTS 07700 Vol. X -- Space Shuttle Flight and Ground System Specification.

The system is comprised of a number of segments to accomplish the following:

- Train personnel
- Produce system elements
- Provide support facilities
- Test and accept system elements
- Transport system elements to using site
- Perform turnaround and maintenance operations
- Perform system engineering and facility maintenance
- Provide mission operations support
- Perform assembly and launch
- Perform mission operations

More items are covered in Volume X.
Parts and Interfaces. All of these functions result in segments. Interfaces between all these segments had to be addressed and documented in Space Shuttle program ICDs.

The vehicle segment is responsible for producing the system elements -- Orbiter, main engines, external tank, and solid rocket boosters.

The Shuttle vehicle element interfaces are considerably more complicated than those of Apollo. The Apollo launch vehicle had structural and wiring interfaces to the other elements.

Contrast this with the Shuttle where both the Orbiter main engines and the solid rocket booster are active during ascent and are interconnected to the other elements. The tank and Orbiter are interfaced for transferring propellant. The booster is structurally interfaced to the tank and receives commands for thrust vectoring and separation from the Orbiter. The main engines are structurally integrated into the Orbiter and, although they have their own computers, must send data to and receive commands from the Orbiter.

This complexity necessitated a separate segment for avionics and software. A separate Orbiter Avionics Office was established to manage the complex interface and all requirements for avionics and software, regardless of what segment implemented the hardware or software.

Whatever next level you are defining -- be it segments, elements, or subsystem, you need to consider the interfaces between them. For example, lay out your proposed definition of subsystems, then define for each subsystem its interface with every other subsystem. If this results in complex interfaces then redefine the subsystems to simplify the interfaces.

Management of Parts. In a typical contracted effort, the government creates and controls the system specification and a contractor performs the system design and generates the subsequent segment specifications for government approval. The contractor issues these specifications to other contractors or to in-house teams to do the segment designs. The government has its technical experts review the system design and segment specifications.

On large projects there is generally a government organization responsible for each segment. During the Shuttle development, the system organization resided at JSC and it included the management of the vehicle segment, with Rockwell International as the vehicle integration contractor. Each vehicle element was managed by a government organization (e.g., Orbiter Project Office) and a prime contractor.

Each element contractor is responsible for ensuring that the subsystem specifications are created for its element. Depending upon the government organization involved, the government may send in subsystem experts to assist in reviews during this process. NASA JSC has typically had subsystem managers for large projects.

SMALL IN-HOUSE PROGRAM -- SAFER

On a small in-house program the levels of specifications differs significantly from a large project. The SAFER Project has five specifications:

- PRD
- Flight Element
- Mission Support Element
- Ground Support Element
- Software Subsystem

The system specification (PRD) is controlled by the Orbiter and GFE Projects Office and the Director of Engineering. The element level (also called an end-item) consists of three parts -- the flight element, the mission support element, and the ground element. All of these elements are controlled by a single organization (Project Office within a Division).

Although controlled by a single organization, the SAFER elements require different functional and technical skills to define their requirements and perform the design. The different specifications reflect those different skills.

The Flight Element has a number of subsystems, but only one subsystem specification. Software, which has many more requirements than the other subsystems and interfaces to almost all other subsystems, has its own specification.

All other subsystem requirements are contained in the element specification. Since the work is being done in-house, controlled by a single organization, and has a fairly small number of requirements, it is not necessary to create separate subsystem requirements documents.

Each subsystem will have a design specification that includes details of the design and drawings of all parts. While there is a one-to-one relationship between requirements and design there need not be a one-to-one relationship of documents -- requirements specifications to design specifications.

THE LEVEL TRAPS

There are several level traps to avoid when writing requirements. The first is the implementation trap -- the result of thinking about the design and failing to step back and think about the need. The second is the subject trap where the wrong subject is used in writing a requirement sentence and implies lower level design. These are described below.

IMPLEMENTATION TRAP

If you have been doing design studies at a low level, you may begin to document these results as high level requirements -- this is a level trap. You will be defining implementation instead of requirements.

An example of this occurred during the definition of the ACRV requirements. An individual submitted a requirement like this:

The ACRV System shall enter when sea state is at TBD conditions.

The ACRV had no requirement for a water landing -- that was a design option. The individual had been working with that design option and, from previous Apollo experience, known that crew rescue was possible only in certain sea states.

When asked **WHY** the requirement was needed, the individual stated that the crew could not be left in the module for a lengthy period of time, thus the landing needed to be where and when sea states could accommodate crew rescue. He had a valid requirement -- but not the one he had written. Whether the ACRV landed on water or land, removing the crew within a limited time period was essential. Thus the real requirement was: *The ACRV System shall provide for crew removal within TBD time of landing.*

The question **WHY** will resolve most implementation requirement errors. Always ask WHY a requirement is needed to insure that you have not fallen into the implementation trap.

SUBJECT TRAP

There is also an indirect way to fall into a level trap. In a system specification you are writing requirements on the system. As an example, you may want to require that the system provide control. A set of requirements might be written that read as follows:

- *The guidance and control subsystem shall provide control in six degrees of freedom.*
- *The guidance and control subsystem shall control attitude to 2 +/- 0.2 degrees.*
- *The guidance and control subsystem shall control rates to 0.5 +/- 0.05 degrees/ second.*

The trap encountered is one of defining a guidance and control subsystem. Controlling attitude and rate is a system problem, it requires not only a guidance and control subsystem but also a propulsion subsystem to achieve these rates. What subsystems will be required to accomplish the requirements is part of the design process. The requirements should be written from the system perspective, as follows:

- *The system shall provide six degrees of freedom control.*
- *The system shall control attitude to 2 +/- 0.2 degrees.*
- *The system shall control rates to 0.5 +/- 0.05 degrees/second.*

The author of the original requirements was not trying to define the lower level breakout. He probably comes from a control background and sees the system from that perspective and hence writes requirements that way. The flow down of requirements, to all affected segments, elements, and subsystems, will be badly affected if these requirements are not written correctly.

SUMMARY

- **Evolve level definition as the project evolves.**
- **Minimize interfaces and keep them simple.**
- **Consider the management organization and technical capabilities when defining levels and parts.**

Section 4. WRITING REQUIREMENTS

This section will address what makes a good requirement. It will cover some of the most common problems that are encountered in writing requirements and then describe how to avoid them. It also includes examples of problem requirements and how to correct them.

GOOD REQUIREMENTS

A good requirement states something that is necessary, verifiable, and attainable. Even if it is verifiable and attainable, and eloquently written, if it is not necessary, it is not a good requirement. To be verifiable, the requirement must state something that can be verified by examination, analysis, test, or demonstration. Statements that are subjective, or that contain subjective words, such as "easy", are not verifiable. If a requirement is not attainable, there is little point in writing it. A good requirement should be clearly stated.

Need. If there is a doubt about the necessity of a requirement, then ask: What is the worst thing that could happen if this requirement were not included? If you do not find an answer of any consequence, then you probably do not need the requirement.

Verification. As you write a requirement, determine how you will verify it. Determine the criteria for acceptance. This step will help insure that the requirement is verifiable.

Attainable. To be attainable, the requirement must be technically feasible and fit within budget, schedule, and other constraints. If you are uncertain about whether a requirement is technically feasible, then you will need to conduct the research or studies to determine its feasibility. If still uncertain, then you may need to state what you want as a goal, not as a requirement. Even if a requirement is technically feasible, it may not be attainable due to budget, schedule, or other, e.g., weight, constraints. There is no point in writing a requirement for something you cannot afford -- be reasonable.

Clarity. Each requirement should express a single thought, be concise, and simple. It is important that the requirement not be misunderstood. Simple sentences will most often suffice for a good requirement.

COMMON PROBLEMS

The following lists the most common problems in writing requirements, each of these are discussed in detail below:Making bad assumptions

- **Writing implementation (HOW) instead of requirements (WHAT)**
- **Describing operations instead of writing requirements**
- **Using incorrect terms**
- **Using incorrect sentence structure or bad grammar**
- **Writing unverifiable requirements**
- **Missing requirements**
- **Over-specifying**

Another problem in writing requirements is a result of having poor document structure and content control, this subject is discussed in Section 3, Management.

BAD ASSUMPTIONS

Bad assumptions occur either because requirement authors do not have access to sufficient information or the information does not exist. You can eliminate the first problem by creating a Program Plan (PP) and making it available to all authors. You can create and maintain a list of other relevant documents and make these easily accessible to each author. If you have automated the process, you can offer documents on-line and you can filter the information within the documents so that individual authors can get copies of only the data that they need.

In the second case where information does not exist, the requirement author should document all assumptions with the requirement. When the requirement is reviewed, the assumptions can also be reviewed and problems quickly identified. It is also useful to document the assumptions even if the authors were provided the correct information. You cannot ensure that all authors have read all the information or interpreted it correctly. If they document their assumptions, you will avoid surprises later.

IMPLEMENTATION

An Air Force RFP was released for the development of a requirements management tool. The first requirement was to "provide a data base". The statement is one of implementation and not of need, and it is common to find such statements in requirement specifications. Specifications should state **WHAT** is needed, not **HOW** it is to be provided. Yet this is a common mistake made by requirement writers. Most authors do not intend to state implementation, they simply do not know how to state the need correctly.

To avoid stating implementation, ask yourself **WHY** you need the requirement. In the example cited, it can be seen that by asking **WHY**, the author can then define all of the needs that the system must meet and will then state the real requirements, e.g.:

- provide the capability for traceability between requirements
- provide the capability to add attributes to requirements
- provide the ability to sort requirements.

These requirements state WHAT is needed, not HOW to accomplish it. Each of the above listed requirements might result in a data base type of system, but the requirement for the data base was not needed.

There are two major dangers in stating implementation. The one most often cited is that of forcing a design when not intended. If all the needs can be met without a data base, then why state the need for a data base. If they cannot be met another, or better, way, then a data base will be the solution -- whether or not there was a requirement for a data base.

The second danger is more subtle and potentially much more detrimental. By stating implementation, the author may be lulled into believing that all requirements are covered. In fact, very important requirements may be missing, and the provider can deliver what was asked for and still not deliver what is wanted. Providing a data base will not be sufficient for someone needing a requirements management tool. It is the capabilities of the tool that need to be stated as requirements.

At each level of requirements development the problem of needs versus implementation will occur. At the system level the requirements must state WHAT is needed. The system designer will determine HOW this can be accomplished and then must define WHAT is needed at the subsystem level. The subsystem designer will determine HOW the need can be met and then must define WHAT is needed at the component level.

To ensure that you have not stated implementation, ask yourself WHY you need the requirement. If this does not take you back to a real need statement, then you are probably stating a need and not implementation.

Example: See *The Implementation Trap*, page 3-8.

OPERATIONS VS REQUIREMENTS

This problem is somewhat similar to the implementation problem. Since SAFER hit it several times, their examples are provided:

The first example is of a requirement that was in the environment section of the specification. Another requirement, in the physical section, stated the need for the SAFER FTA to be stored in the Airlock Stowage Bag. What is stated is a description of the operations, not a requirement about the environment.

The SAFER FTA shall be stowed in the Orbiter Airlock Stowage Bag for launch landing, and on-orbit stowage.

The requirement is:
The SAFER FTA shall be designed for the stowage environment of the Airlock Storage Bag for launch, entry, landing, and on-orbit, as defined in TBD.

The next requirement again describes the operations and is confusing.

The SAFER FTA shall be operated by an EVA Crewmember wearing EMU sizes medium through extra large without limiting suit mobility.

The statement was rewritten and resulted in a requirement and a design goal. The design goal is needed because no quantifiable requirement can be written regarding suit mobility.

The SAFER FTA shall be designed for use with EMU sizes medium through extra large.

The SAFER FTA should not limit EVA crewmember mobility.

The danger in stating operations, instead of a requirement is (1) the intent may be misunderstood and (2) determining how to verify can be a problem.

USE OF TERMS

In a specification, there are terms to be avoided and terms that must be used in a very specific manner.

Authors need to understand the use of **shall, will,** and **should:**

- Requirements use shall
- Statements of fact use will
- Goals use should.

These are standard usage of these terms in government agencies and in industry. You will confuse everyone if you deviate from them. The requirements must be verified.

Terms such as *are, is, was, and must* do not belong in a requirement. They may be used in a descriptive section or in the lead-in to a requirements section of the specification.

There are a number of terms to be avoided in writing requirements, because they confuse the issue and can cost you money, e.g.

- Support
- But not limited to
- Etc.
- And/Or

The word **support** is often used incorrectly in requirements. *Support* is a proper term if you want a *structure to support 50 pounds weight*. It is incorrect if you are stating that the *system will support certain activities.*

WRONG: The system shall support the training coordinator in defining training scenarios.
RIGHT: The system shall provide input screens for defining training scenarios. The system shall provide automated training scenario processes.

The terms **but not limited to** and **Etc.** are put in place because the person writing the requirements suspects that more may be needed than is currently listed. Using these terms will not accomplish what the author wants and can backfire.

The reason the terms are used is to cover the unknown. The contractor will not increase the cost in the proposal because you added these terms. The only way to get the work added is to place an analysis task in the SOW to determine if more items need to be added to the list. In the SOW you can control what effort the contractor will expend to address these unknowns. If more items are found, you may have to increase the scope of the contract to cover the additions.

If you have these terms in your requirements specification, the contractor may use them as an excuse for doing unnecessary work for which you must pay. You cannot win by using the terms in the specification.

The term **and/or** is not appropriate in a specification. If you use *and/or* and the contractor does the or he has met the terms of the contract. Either you want *item 1 and item 2* or you will be satisfied with *item 1 or item 2.* Again, if you use the term *or*, then the contractor has met the terms of the contract if he does either item.

REQUIREMENT STRUCTURE/GRAMMAR

Requirements should be easy to read and understand. The requirements in a system specification are either for the system or its next level, e.g. segment. Each requirement can usually be written in the format:

- The System shall provide........
- The System shall be capable of ....
- The System shall weigh .....
- The Segment #1 shall provide ....
- The Segment #2 shall interface with ...

Note: The name of your system and the name of each segment appears in these locations. If you have a complex name, please use the acronym, or your document will have many unneeded pages just because you have typed out a long name many times.

Each of these beginnings is followed by **WHAT** the System or Segment shall do. Each should generally be followed by a single predicate, not by a list. There are situations where a list is appropriate, but lists are over-used. Since each item in the list must be verified, unless all items will be verified by the same method and at the same time, it is generally not appropriate to put items in a list.

Requirement statements should not be complicated by explanations of operations, design, or other related information. This non-requirement information should be provided in an introduction to a set of requirements or in rationale.

You can accomplish two things by rigorously sticking to this format. First, you avoid the Subject Trap (see page 3-8). Second, you will avoid bad grammar that creeps into requirements when authors get creative in their writing.

Bad Grammar. If you use bad grammar you risk that the reader will misinterpret what is stated. If you use the requirements structure suggested above, you will eliminate the bad grammar problems that occur when authors try to write complex sentences and use too many clauses.

Another solution is to write requirements as bullet charts. When the content is agreed upon a good writer can convert the information into a sentence for the specification.

Authors will also try to put all that they know in a single sentence. This results in a long complex sentence that probably contains more than one requirement. Bullet charts or one good editor can alleviate this problem.

Example:

The ACRV System shall provide special medical life-support accommodations for one ill or injured crew member consisting of medical life-support and monitoring equipment and the capability of limiting impact accelerations on that crew member to be not greater than.... for a total impulse not to exceed ....

The requirement above contains a number of requirements, it needs to be broken into at least four requirements and it could use a lead-in such as:

The ACRV will be used as an ambulance for an ill or injured crew member. Only one crew member will accommodated at a time. The following define the unique requirements for this capability.

- *..provide medical life-support accommodations for one crew member*
- *..provide monitoring equipment for one crew member*
- *..limit impact accelerations to the ill or injured crew member to no greater than...*
- *..limit total impulse to the ill or injured crew member to ...*

UNVERIFIABLE

Every requirement must be verified.

Because every requirement must be verified, it is important to address verification when writing the requirements. Requirements may be unverifiable for a number of reasons. The following discusses the most common reason -- use of ambiguous terms.

Ambiguous Term. A major cause of unverifiable requirements is the use of ambiguous terms. The terms are ambiguous because they are subjective -- they mean something different to everyone who reads them. This can be avoided by giving people words to avoid. The following lists ambiguous words that we have encountered:

- **Minimize**
- **Maximize**
- **Rapid**
- **User-friendly**
- **Easy**
- **Sufficient**
- **Adequate**
- **Quick**

The words **maximize** and **minimize** cannot be verified, you cannot ever tell if you got there.

The words minimum and maximum may be used if the context in which they are used can be verified.

What is **user-friendly** and what is **rapid?** These may mean one thing to the user or customer and something entirely different to a designer. When you first begin writing your requirements, this may be what you are thinking, but you must write the requirements in terms that can be verified. If you must use an ambiguous term in first draft documents, put asterisks on either side of the term to remind yourself that you are going to have to put something concrete in the requirement before you baseline the document.

There may be cases where you cannot define, at your level, exactly what is needed. If this is the case, then you should probably be writing a design goal, not a requirement. You can do this by clearly indicating that your statement is a goal, not a requirement. Use of the word *should*, instead of the word *shall*, will denote a design goal.

MISSING

Missing items can be avoided by using a standard outline for your specification, such as that shown in Section 6 of this guide, and expanding the outline for your program.

Many requirements are missed because the team writing the requirements is focused on only one part of the system. If the project is to develop a payload, the writers will focus on the payload's functional and performance requirements and perhaps skip other important, but less obvious, requirements. The following is a checklist of requirement drivers you need to consider:

- *Functional*
- *Performance*
- *Interface*
- *Environment*
- *Facility*
- *Transportation*
- *Training*
- *Personnel*
- *Reliability*
- *Maintainability*
- *Operability*
- *Safety.*

You will need to develop detailed outlines for your specification for the functional and performance requirements, and in perhaps other areas. This subject is discussed, with examples, in Section 2, Management.

You may also have a number of requirements that you must include by reference. In particular, those standards that define quality in different disciplines (materials and processes) or for different projects. If you fly aboard the Shuttle, they have a set of guidelines that you must impose on your system.

Detailed requirements analysis is necessary to assure that all requirements are covered. There are a number of approaches to performing requirements analysis and a number of tools for doing this work. Detailed requirements analysis is beyond the scope of this guide.

Another tool to help you ensure that you have all the requirements is that of Allocation. This is discussed in Section 5, Requirements Attributes.

OVER-SPECIFYING

The DoD has stated that over-specification is the primary cause of cost overruns on their programs. Over-specifying is most often from stating something that is unnecessary or from stating overly stringent requirements.

Unnecessary Items. Unnecessary requirements creep into a specification in a number of ways. The only cure is careful management review and control.

People asked to write requirements will write down everything they can think of. If you do not carefully review each requirement and why it is needed before baselining the specification, the result will be a number of unneeded requirements.

Example: The Space Station Training Facility (SSTF) had a requirement for a high- fidelity star field. The author knew that a new high-fidelity star field was being developed for the Shuttle Mission Simulator (SMS) and assumed they might as well put the same thing in the SSTF. The crew needs a background to view outside the Space Station, but there is no need for a high-fidelity star field, since they do not use the stars for navigation.

The requirement needs to be written for a visual background for crew orientation. The design process will determine if using the SMS star field is a cost effective solution or if something simpler is adequate and more cost effective.

Example: A number of SSTF requirements were deleted when their authors were queried as to the need. The need they stated was that "it would be nice to have". Most programs do not have budgets for *nice to have* items.

Unnecessary requirements can also appear after baseline if you let down your review and control process. In ACRV a number of requirements were added after the initial baseline that were not needed. One such instance occurred because of an error in the baseline document.

Example: The baseline document had two requirements:

The ACRV System shall be capable of operating over a planned operational life of thirty (30) years.

The Flight Segment shall provide an operational life of 30-years for the flight elements.

The second requirement, for the Flight Segment, was not required, the System requirement was adequate. The action that should have been taken was to delete the Flight Segment requirement. Instead, two more requirements were added to require a 30- year operational life of the other two segments.

At least one other requirement was added to ACRV that was a duplicate of an existing requirement. The wording of the two differed only slightly and their rationale was the same. It requires careful attention to detail to avoid this type of problem.

Overly Stringent. Most requirements that are too stringent are that way accidentally, not intentionally. A common cause is when an author writes down a number but does not consider the tolerances that are allowable.

Thus, you should not state that something must be a certain size, e.g., 100 sq.ft., if it could just as easily be 100 +/- 10 ft. You do not need to ask that something deliver a payload to exactly 200 n.m. if greater than or equal to 200 n.m. is acceptable.

Some of the major horror stories of the aerospace industry deal with overly-stringent requirements. One contractor was severely criticized for charging $25,000 per coffee pot in airplanes built for the government. But the requirements for the coffee pot were so stringent, that the plane could have crashed and yet no coffee could spill. It cost a great deal to develop the coffee pot and to verify that it met its requirements. Each copy had to be built to a stringent design.

The solution to this problem is to discuss the tolerances allowable for any value and then to write the requirement to take into consideration those tolerances. Each requirement's cost should also be considered.

Section 5 -- REQUIREMENT ATTRIBUTES

This section deals with information related to requirements. Much effort goes into defining requirements and many pieces of data are discussed, exchanged, and evaluated. This data includes:

- **Allocation** - What next level area, e.g., subsystem, is responsible for responding to this requirement.
- **Traceability** - Links between requirements and links between requirements and information in other documents.
- **Rationale** - What was assumed, why needed, how related to operations, and how driven by design.
- **Verification** - What phase and method will be used to prove the requirement.
- **Other** - Who is responsible for this requirement (owner), priority, risk, etc.

This data needs to be captured and maintained along with the requirements to manage the project over its life-cycle and to ensure completeness of the requirements. Some of this data that is mandatory on many programs, e.g., traceability and verification. Other information has proven to be valuable, e.g., rationale and priority.

The project team can provide and control the data or leave it to each individual involved to make assumptions about the data. Clearly the latter can be costly since it will create extra work for each individual. The latter also creates considerable risk, because everyone will not make the same assumptions. Bad assumptions are major causes of bad requirements. The cost of correcting a bad assumption is small, if found during requirements definition, but it can be enormous if not found until later in the life-cycle.

ALLOCATION

Allocation refers to pointing each requirement in a specification to the next level specifications that should respond to that requirement. Requirements that appear in Sections 3.1 through 3.6 of the specification are for the entire system. Some of these are applicable to all specifications at the next level, e.g., segment, and some are applicable only to a subset of the segments. Each requirement needs to be allocated to the segment(s) of the system to which it applies.

RESOURCE ALLOCATION

Almost everyone understands allocation of resources. If the system is required to produce 75 kw of power, then this number must be allocated to the segments in terms of how much each can use. But each requirement, whether or not a resource, needs to be allocated.

OTHER ALLOCATIONS

If the authors of the system specification allocate each requirement to the next level a number of benefits can accrue.

(1) You will lessen the chances that a requirement will slip through the cracks. This occurs when two segments each think the other has responsibility for the requirement, so neither acts upon it. If you a have allocated it to each of them, they will each respond accordingly.

(2) You will lessen the chance of unnecessary work being performed. If the requirement is to be performed by segment A and B, but segment C thinks it is in his area, he will spend money before you can recognize the mistake. If you have allocated the requirement to A and B, but not to C, this problem should not occur.

(3) You will write cleaner requirements. As you allocate, you can assess the requirements to determine if you have written the same requirement multiple times for each segment, when you could have written a single requirement and allocated it to the appropriate segments. You will find that you have written a segment specific requirement, when what you really needed was a system requirement because multiple segments are affected.

(4) You will identify interface requirements. When a requirement is allocated to multiple segments there may be a need for interface definition. Allocation will help expose this.

HOW TO ALLOCATE REQUIREMENTS

As you write the requirements, allocate each to the next level. If you are writing system requirements, and the next level is segments, then each system requirement will be allocated to one or more segments.

Print out just those requirements you have allocated to one segment, e.g., the flight segment, and review them for completeness. Taking this narrower view will help you determine if you have all the requirements.

Then review all other allocations, e.g., for each requirement in your flight segment list, review its other allocations. You may discover that you are missing allocations to other segments, or you have made unnecessary allocations. If you are missing an allocation it may be because you have written a specific requirement for the other segment. You will have to use your best judgment to determine if you need two requirements, or if you can write one requirement and allocate it to both segments.

If you have your requirements in a word processor, allocation can be difficult to do, because to do it well, you need to be able to sort the requirements by their allocation. If you use a requirements management tool that provides allocation, or some other method of sorting, this job will be a lot easier.

TRACEABILITY

One important aspect of managing requirements is that of traceability. Traceability refers to relationships between requirements and between requirements and other data such as design, verification, and procedures.

All requirements in the system specification should be traceable to the Program Plan -- needs, goals, objectives, constraints, or operations concepts. All requirements beneath the system level must be derived from the next higher level specification. If a requirement is defined at a lower level that cannot trace its parentage back to a higher level then, either the requirement is not needed or a higher level requirement is missing. The DoD requires that a requirements traceability matrix be prepared at each level in the process to show the parent-child relationship of the requirements. Traceability from the Program Plan to the System Specification is an internal DoD task. Traceability from the System Specification to lower level specifications is usually a contractor responsibility.

All requirements will impact the design and must be verified. Many requirements will drive operational procedures. Traceability between requirements and design, requirements and verification plans, and requirements and procedures can provide invaluable support to project management and especially to change impact assessments.

PROJECT MANAGEMENT

It is essential to show that each requirement has been flowed down to the next level until an end product is reached. Otherwise, requirements may be ignored until very late in the design or development process. Even worse, some may not be discovered until verification is nearly complete.

Those in charge of a requirements specification need to ensure that the next level of requirements are complete, consistent, and do not result in gold plating. To analyze the next level of requirement it is essential to have traceability between your parent requirement and the child requirements that are attributed to you.

If you are the author of system requirements, you may have child requirements in each next level document. Pouring through these documents will be time consuming. If you are diligent, you may find those child requirements that you believe are derived from your parent requirements. You may be able to determine if these child requirements are complete and consistent. You will never know if other child requirements have been written and attributed to your parent requirements. You will find it difficult to locate the gold-plating that may have occurred at the next level.

Establishing and maintaining traceability between levels of requirements is essential to project management. Each parent level requirement must be traced to its next level child requirements. When these two levels of requirements are resolved, then the process is repeated, but with the former child requirement now serving as a parent to the next level child requirement. This process continues until an end product is reached.

CHANGE MANAGEMENT

Traceability between requirements and between requirements and other documents is essential for processing change requests. It is important to look at the parent requirement(s) of the requirement being changed to ensure that a higher level change request is not also needed. If design work is complete, then the design must be assessed to determine the impact of the changing requirement. If the next level of requirement has been written, then all child requirements must be assessed to determine the impact of the change.

If the change occurs after testing has begun or after operational procedures are written, then the impact on test plans, test results, and procedures must be added to the change impact assessment. Unless there is traceability between the changing requirement and other related requirements and other related data, it is impossible to determine the full impact of the change.

Lack of traceability can result in the approval of a change with an acceptable impact, only to discover, perhaps too late, that an unacceptable impact will in fact be encountered.

HOW TO MAINTAIN TRACEABILITY

A number of requirements management tools make the process of maintaining traceability a straight forward matter. Without such tools, you will have to keep track in some type of data base. This is extremely difficult, since you must track every change to all specifications and other documents.

For a single level of traceability, you can use a method used by many contractors. As each requirement is written, it is placed on a data sheet that includes the parent(s) that drove the requirement. This enables a review of all child requirements, as they are written, to ensure that they are complete, consistent, and do not include gold-plating. It is a difficult process to return to these sheets when you are doing change management.

RATIONALE

Many project problems can be avoided if each author documents the rationale for each requirement as it is written. Rationale describes some or all of the following related information:

- Assumptions
- Why needed
- How related to expected operations
- Design decisions

Some managers erroneously believe that this process is too time consuming and should be skipped. The amount of effort required to do this initially, when the requirement is written, is minuscule compared to the time that will be wasted later when this data is needed and is not available.
No requirement should ever be put into a specification until its rationale is well understood. Most organizations capture and understand rationale for each change, but they fail to follow the same process with the initial baseline of the specification.

There are enormous benefits to documenting rationale with each requirement -- for the initial baseline and for all changes. The following paragraphs will explain the types and benefits of rationale , and how to capture rationale.

TYPES OF RATIONALE

There are at least four types of information that may be contained in rationale. A requirement may have one or more of these as its rationale.

Assumption. Any assumption made when writing the requirements should be documented; e.g.,

- Assumed no more than ten launches per year.
- Assumed use of existing launch facilities.
- Assumed technology mature for light weight materials.

Why Needed. This may be obvious from the requirement and its parent requirements. If not obvious, it needs to be explained, e.g.:

- The crew needs the data available during EVA periods.
- Only one hand controller will be carried. It will be a right hand configuration. All crew will be trained for this configuration.
- Need two colors of lights since display is in small area and crew cannot discriminate between two red lights.

Having the information available will help preclude someone changing the requirement for the wrong reason at a later time.

Operations. Many requirements are driven by expected operations. Information about operations needs to be maintained with the requirement for change impact assessments, e.g.:

- Two crew members will be available to operate the device.
- Operations will be done only in daylight.
- Will use the same tool set as used for XYZ system.

Operational changes may make it impossible to have two crew members to operate the device, or make it essential that operations also be done in artificial light. When these changes occur, they will probably impact the requirements. If no history is maintained about the relationship between the requirements and the operations, no one will recognize the need to alter the requirements. Problems may remain hidden until operations begin.

Design Drivers. A design for the system drives the requirements at the next level, e.g., segment. This is the place to record what was done so that when changes are proposed or the system needs to be upgraded, the history of the current configuration is available. These may be more lengthy than other rationale. Some brief examples are:

- Trades conducted for fuel cells, solar panels, and batteries. Selected batteries because of the length of time used and minimum cost.
- Considered hardware, software, and combined hardware/software. Determined that all could be done in software, faster, cheaper, and with no weight impact.
- Flight dynamic trades conducted to minimize amount of heating during entry while staying within the control parameters dictated by flight control organization.

When someone proposes a change to a requirement where rationale is documented, it will be obvious how it was initially derived. This provides a base to determine if additional work must be done before the change can be made. Rationale is known when the requirement is written. It will be quickly lost if not immediately documented.

BENEFITS OF DOCUMENTING RATIONALE

The benefits that result from documenting rationale as requirements are written include:

- Identify bad assumptions
- Weed out unnecessary requirements
- Avoid implementation where not intended
- Maintain corporate history
- Support the change process
- Improve communications within your team and with your contractors.

Assumptions. Bad assumptions are a major cause of bad requirements. If you do not know the assumptions the requirement author made, you may accept a requirement that is totally incorrect. It is relatively easy to identify a bad assumption and correct a requirement before other work is done. If you do not do this, the problem will surface eventually and the later it surfaces the more you will pay to undo work and back-up to rewriting the requirement. Studies for software show that the cost grows almost exponentially. If you do not find the problem until you are operational, the cost can be 1000 times what it would have been if the assumption were caught during the requirements definition phase.

Need. If an author must state why a requirement is needed, you may save yourself the trouble of ever seeing the requirement -- he/she may talk him/herself out of the requirement. Even if they don't, when you see the reason, you may find that the requirement is not needed, or is not needed at the current level.

Implementation. When people state why a requirement is needed you can identify those that state implementation. The author probably did not mean to define implementation, but did not know how to write the requirement.

By stating the need, the author will reveal the real underlying requirement.

Corporate History. Even short term projects will have turn-over in personnel, and long term projects will see some memory loss in those that remain. By documenting the rationale, you preserve information you will need later to jog memories or to educate new personnel.

Change. If it is not clearly understood how a requirement came into being it is sometimes very difficult to determine if a proposed change is suitable. This problem has two sides. One is that the change will be made without consideration for some very important point. The flip side is that great battles will be fought about a change for a very minor point.

The former happens because of lack of data and "ignorance is bliss". The latter happens because of lack of information and fear of making a mistake. Either can be costly and the former may be catastrophic.

Communication. Documenting rationale improves communication. Even on a small team a great deal of understanding is assumed that may not really exist. When team members see the rationale of other requirement authors they have more data and can identify where communications have failed. Providing your rationale to the contractors bidding on your work gives them the knowledge to make a better response to your needs.

HOW TO CAPTURE RATIONALE

The easiest way to capture rationale is as each requirement is written. Each requirement should be accompanied by its rationale before it is considered for inclusion in the baseline specification or for any change to a specification.

If you have a tool to capture rationale and maintain it over the life of the program, you will be ahead of the game. Otherwise, you are going to need to keep all of the requirement sheets and make them available for review. The problem with this approach is that it is difficult to maintain the files and hard to look things up. This makes it easy for people to never review the rationale when changes are proposed, and negates one of the benefits of collecting the information

EXAMPLES OF RATIONALE

Rationale does not need to be a thesis. It can be very short and concise and still provide a wealth of data. The following are some examples of rationale for ACRV and SAFER requirements.

Requirement	Rationale
The Flight Segment shall be compatible with variations in Space Station atmospheric pressure as specified in SS/ACRV IRD TBD.	Equipment in the pressurized ACRV environment shall be able to return to normal operation after the ACRV has been depressurized to a certain extent and then repressurized, e.g., as a result of Space Station maintenance which requires the partial depressurization of the module where ACRV is located.
The Flight Segment elements shall provide for the activation of the ACRV independent of the Space Station.	If evacuation is a result of a Space Station emergency, the Station may not be able to assist in activation. The ACRV must be self-sufficient and able to function independently from the Space Station
The SAFER FTA shall provide six degrees of freedom manual maneuvering control.	Based on Skylab experiment M509 and MMU experience, 6-DOF pilot control is essential.
The SAFER FTA shall be certified for exposure to two Shuttle launches and landings.	Want to be able to launch it and if not used on that flight, be able to relaunch it on a second flight and use it. This is to handle the case where, for some reason, the SAFER experiment could not be done on the initial flight.
The SAFER FTA shall meet specified performance characteristics without service or maintenance for a minimum period of 60 days.	At the SAFER EARB, Oct. 91, the requirement for 60 days was identified to allow for Orbiter roll-back and launch without having to remove SAFER for recharge.

VERIFICATION

Every requirement in the specification ***MUST BE VERIFIED.*** The verification plan and approach are defined in the specification. Each requirement should be tagged with its phase/level and method of verification as it is written.

Like other attribute data, verification should be documented when the requirement is written. It is important to assess verification early. First, you want to ensure that each requirement is verifiable. Determining the verification phase and method for the requirement will help focus the required attention on the requirement to ensure that it can be verified. Second, verification will probably be a major cost element of your project. If you do not consider your verification approach early in the project, you may encounter significant cost impact later.

VERIFICATION APPROACH

Before anyone can label their requirements with a phase and method of verification, they need to know the program's approach to verification. The standard specification requires that you document your approach in Section 4.0. Since verification is a major cost element of any program, it needs to be considered and issues relating to it resolved early in the program.

There are two major parts of describing a verification approach -- **when** the verification will occur and **how** it will be done. The **how** is more restrictive, and consists of four methods:

- Examination
- Test
- Demonstration
- Analysis

The when is a bit more complicated because there are multiple ways to describe it. The DoD uses levels, e.g., system, segment, subsystem to define when an item will be verified. NASA has typically used phases, e.g., development, operations, acceptance, certification. Note that these phases are not identical to the life-cycle phases, but include other major milestones such as acceptance or certification.

In your specification, you will describe what phases/levels and methods are to be used. You will also describe your approach to system test and analysis. As each author writes a requirement, the phases and methods to be used to verify the requirement will be documented. This will identify problems with your approach and enable you to correct them and improve the accuracy of your program cost estimates.

Verification is defined by phase/method, e.g., development/analysis or certification/test. A requirement may have a single or multiple verification requirements.

HOW TO CAPTURE VERIFICATION DATA

Someone is going to capture verification data. This is not optional. Since all requirements must be verified, the quality organization will ensure that this work is done.

If you have a requirements management tool that allows you to tag each requirement with verification data, you will make the job simpler. If you do not, then someone -- you or the quality people -- will be building and tracking matrices of requirements versus verification phases and methods, throughout the life of your program.

What you do not need is a repeat of the requirements. This has been done on several programs -- each requirement is rewritten as a verification statement. What people have done is use a word processor to change the requirement statement from:

The system shall do something.
to:
Verify that the system does something.

Not only is this unnecessary, it is a great waste of money. This new revised copy of the requirements only means you have another document to maintain. Otherwise, it is worthless.

OTHER DATA

You have probably already decided that there is too much data to keep up with, but others have kept the following types of information and improved their management of the requirements by doing so. Some types that are kept include:

- Priority
- Owner
- Source
- Risk
- Work Breakdown Structure (WBS)
- Product Breakdown Structure (PBS)
- Implementation Phase
- Status

PRIORITY

Priority is used to guide the developer. Not all requirements are created equal. By the time you assign priority, you should have weeded out the unnecessary or nice to have requirements. Priority means assigning a number (1, 2, 3) to each requirement that places emphasis on those requirements for which there is no design flexibility (1), and shows the relative flexibility of other requirements (2) or (3).

Putting a low priority on a requirement does not mean that it can be dropped. It means that when making trade studies higher priority requirements cannot be played against lower priority requirements. A higher priority requirement is more rigid and should be held, while lower priority requirements may be relaxed. If you give the developer this information, he can make good trades. If you do not, he will make trades that you will reject and rework will be required. Assigning priority improves communications and can eliminate a number of meetings and repeated discussions.

OWNER

One or more persons needs to be designated as the requirement owner. The owner is someone who understands the need for the requirement and its technical implications. By assigning an owner, you have someone to consult whenever a change is proposed. You have a list of requirements for an owner that can be delegated to someone else should the current owner no longer be available. If this delegation is done before the original owner is unavailable, then he can pass his knowledge base to the person who will assume ownership.

If people know, when they submit a requirement, that they are going to be tagged as the owner, they will take a more responsible view of the requirement. If they can submit it anonymously, they will be less responsible.

SOURCE

This data may not be pertinent for every requirement. For some, this is an important piece of information. If the requirement exists because someone in the Executive Branch or in Congress mandated it, it is best to ensure that this is not forgotten. This information can be placed in Rationale.

RISK

Not all requirements have the same risk. Those that are high in risk need to be tagged and carefully tracked to ensure that they are not going to bring the program to a halt. By tagging the risk level of each requirement, you can sort the requirements and put emphasis where it is required.

WBS and PBS

Some DoD programs require a correlation matrix between WBS or PBS and the requirements in order to track cost and cost changes as requirements change. In some manner you will be tracking this information. If you can tie the WBS or PBS directly to the requirements, you may simplify the job.

IMPLEMENTATION PHASE

If you must deliver a system over time, then each requirement needs to be tagged for its implementation phase. If you can do this external to you specification, it will make your job easier. Arranging a specification to reflect phased implementation confuses the format of the specification, sometimes beyond comprehension.

STATUS

All requirements are not at the same level of maturity until late in the design process. By tagging requirements with a status level, you can help management and developers. Management needs to be aware of what is still flexible and likely to change in order to keep analysis schedules under control and to achieve firm numbers by a need date. Developers need the same information in order to do parametric studies for those items that are still likely to change.

HOW TO CAPTURE OTHER DATA

You can capture this data initially by having a form with fields to enter the data when people submit their requirements. Keeping the data current requires some automated tool or a host of people with clipboards.

Section 6 -- SPECIFICATIONS

This section contains two parts. Part 1 is an annotated outline for a system specification. Part 2 contains examples for Section 3 of the specification.

Part 1 -- OUTLINE FOR SYSTEM SPECIFICATION

This outline is applicable for a system, segment, element, or end item. The outline is based on Mil- Std-490A. It includes some modifications, proposed for Mil-Std-490B, that are similar to modifications in the Marshall Space Flight Center (MSFC) standard, MM8040, and a few other changes that are intended to remove some duplication.

This outline is only a guide and should be tailored for the system being defined.

Section 1 SCOPE

The paragraph shall briefly summarize the purpose and coverage of the document in a single paragraph. A brief overview of the intended application may be included if suitable.

Section 2 APPLICABLE DOCUMENTS

This section shall list all and only those documents identified and referred to in Sections 3, 4, and 5 of the specification. Exact issue, revision letter, and date of issue are stated for each reference (and not repeated in the body of the document). Note that the references are applicable only to the extent specifically indicated in the location where they are referenced.

Section 3 REQUIREMENTS

This section shall define the minimum requirements that a system must meet to be acceptable. The essential requirements and constraints that apply to performance, design, interoperability, reliability, user personnel skill levels, etc., of the system covered by the specification shall be stated in this section. Only requirements that are necessary, achievable, and verifiable shall be included.

3.1 System Definition This paragraph shall briefly state the purpose of the system to which this specification applies. A complete definition of the system shall be provided in order to identify major factors that affect the system design. A brief description shall be provided to identify the major physical parts and functional areas.

3.1.1 Missions. This subparagraph shall describe the missions of the system to the extent that such missions affect performance and design.

3.1.2 Operations Concepts. This subparagraph shall describe operational concepts as they affect performance and design. Each concept shall be included in an appropriate subparagraph. Typical concepts include: launch vehicle, launch site, rendezvous, recovery, servicing, training, maintenance, logistics, etc.

3.1.3 System Diagram. This subparagraph shall provide a diagram of the system.

3.1.4 Interface Definition. This subparagraph shall identify all functional and physical interfaces between (a) this system and other systems, and (b) the major components within this system. For (a) do not identify the design implementation (voltages, data rates, etc.) of these interfaces. This is accomplished by Interface Control Documents (ICDs), which are referenced here and shown on the specification tree. For (b) the detail interface requirements may be defined here or in a separate Interface Requirements Document (IRD).

(a) External Interfaces. Every system is a subset of a larger system. A payload on the Shuttle is a system, but it must interface with the Space Transportation System -- KSC for launch checkout, the Orbiter for launch and operations, and the Mission Control Center and the Shuttle Mission Simulator for preflight and flight operations. External interfaces create constraints upon the design of your system and must be clearly defined as requirements for the design of your system.

(b) Internal Interfaces. How you define the subparts of your system will affect the internal interface requirements. This is a critical and generally very difficult step. If you break your system into parts without considering the interfaces you may create complex interfaces. Interfaces must be taken into consideration when determining the subparts.

3.1.5 Government Furnished Property. This shall identify all hardware, software, and facilities provided by the Government for the program.

3.1.6 Organization and Management Relationships. This paragraph shall specify organizational responsibility for preparation, maintenance and control of this specification. Describe relationships between this specification, related higher and lower and interface specifications and the organizations and the organizations responsible.

3.2 Characteristics. This section shall define the required performance characteristics, physical characteristics, and requirements for reliability, maintainability, and environmental conditions.

3.2.1 Performance Characteristics. This subparagraph shall include general and detail requirements, under appropriate subheadings, for those performance characteristics that the system is expected to meet. (Sub-headings may be operating modes.)

3.2.2 Physical Characteristics. This subparagraph shall set forth requirements, such as weight or dimensional limits, which establish boundary conditions necessary to assure physical compatibility and which are not defined by interface requirements, design and construction requirements, or referenced drawings.

3.2.3 Reliability. This subparagraph defines the total system reliability value and the segment allocations. (Mil-Std-490B requires that these be numeric. NASA programs have not used reliability numbers in many instances, instead citing levels of redundancy or fail-safe type requirements.)

3.2.4 Maintainability. This subparagraph shall state the maintainability criteria imposed on the system, including time (e.g., Mean-Time-To-Repair), rate (e.g., man-hours per specific maintenance action or frequency of preventive maintenance), and complexity (e.g., number of people and skill levels).

3.2.5 Availability. This subparagraph shall specify the degree to which the system shall be in an operable and committable state at the start of the mission(s), where the mission(s) is called for at an unknown (random) point in time. [NOTE: If quantitative requirements for both reliability and maintainability are specified, this requirement is "Not Applicable".]

3.2.6 Environmental Conditions. This subparagraph shall specify both natural and induced environments the system is expected to experience in shipment, storage, service, and use. Where applicable, it shall specify whether the system will be required to withstand, or be protected against, specified environmental conditions. In addition, a description shall be provided as to the electromagnetic environment in which the system must operate effectively, the environment which it generates, and the external environments in which the item must survive.

Standards for many conditions exist and can be referenced depending upon the specific operations of the system.

(a) Natural environmental conditions. These include conditions that the system may encounter at any point in its life-cycle, e.g., wind loading, precipitation, ranges in temperature, humidity, atmospheric pressure, wind shear, vertical gust velocities, turbulence, energy input from solar radiation, particle mass and energy spectrums, etc.

(b) Induced environmental conditions. These include conditions that the system may encounter that are induced by (a) another system or (b) by the system itself.

For example, if the system is to fly on-board the Orbiter and be located in the crew cabin, then it must be able to accommodate the induced environment in the crew cabin. A requirement might read: *The system shall be designed to operate in the environment of the Shuttle Orbiter crew cabin, specified in Ref. Doc.xxx.*

The system itself will induce environmental conditions and these need to be controlled. A requirement might read: *The system shall control noise and vibration levels to those levels of tolerance to associated personnel, structure, equipment, and facilities specified in Ref. Doc.zzz.*

3.2.7 Transportability. This paragraph shall state any unique transportability requirements on the system.

If nothing is stated here it is assumed that commercial or government carriers can transport the system. The Space Shuttle had a number of unique transportation requirements. The booster segments had to be moved from the manufacturing site to the launch site. This requirement dictated segment length since transportation had to be on rail and rail cars have a limit to the length object they can carry.

3.3 Design and Construction Standards. This paragraph and subparagraphs shall specify minimum or essential requirements that are not adequately defined by performance characteristics, interface requirements, or referenced documents. The subparagraphs shall include appropriate design standards, requirements governing the use or selection of materials, parts and processes, interchangeability requirements, safety requirements and the like.

To the maximum extent possible, these requirements shall be specified by reference to established standards, e.g., military, NASA, OSHA. This paragraph specifies criteria for the selection and imposition of standards. Selection of the correct standards is dependent on system criticality and its use. This list should not be merely copied from another specification. Including unnecessary standards will increase the cost of the project.

3.3.1 Materials, processes and parts. This subparagraph shall contain requirements for materials and processes.

3.3.2 Nameplates and product marking. This subparagraph shall specify all requirements pertaining to nameplates or markings, referencing applicable specifications, drawings, or standards.

3.3.3 Producibility. This subparagraph shall establish requirements for the choice of fabrication techniques, design parameters, and tolerances which enable the product to be fabricated, assembled, inspected and tested economically and with repeatable quality. (This is new to 490B and needs some good examples.)

3.3.4 Interchangeability. This subparagraph shall specify the requirements for the level of assembly at which components shall be interchangeable or replaceable. Entries in this paragraph are for the purpose of establishing conditions of design, and are not to define the conditions of interchangeability that are required by the assignment of a part number.

3.3.5 Safety. This subparagraph shall specify requirements to preclude or limit hazard to personnel, equipment, or both. To the extent practical, these requirements shall be imposed by citing established and recognized standards.

Safety requirements include those that are basic to the design of the system, with respect to equipment characteristics, methods of operation, and environmental influences; and which prevent personnel injury and equipment degradation without degrading operational capability. These requirements include such things as use of redundancy, restricting the use of dangerous materials where possible; classifying explosives for purpose of shipping, handling, and destroying; abort/escape provisions from enclosures; gas detection and warning devices; grounding of electrical system; cleanliness and decontamination; explosion proofing, etc.

3.3.6 Human factors engineering. This subparagraph shall specify human factors engineering requirements for the system and shall include applicable documents by reference, e.g., NAS-3000. This paragraph should also specify any special or unique requirements, e.g., constraints on allocation of functions to personnel, and communications and personnel/equipment interactions. Included should be those specified areas, stations, or equipment that require concentrated human engineering attention due to the sensitivity of the operation or criticality of the task, i.e., those areas where the effects of human error would be particularly serious.

3.3.7 System security. This subparagraph shall specify security requirements that are basic to the design of the system with respect to the operational environment of the system. This subparagraph shall also specify those security requirements necessary to prevent access to the internal operating areas of the system and compromise of sensitive information or materials.

3.4 Computer resource requirements. This paragraph and subparagraphs are newly proposed in 490B and until they are approved will not be included in this guide.

3.5 Logistics. This paragraph shall specify logistic considerations and conditions that apply to the system.

3.5.1 Maintenance. This subparagraph shall include consideration of such factors as: (a) use of multipurpose test equipment; (b) repair versus replacement criteria; (c) levels of maintenance; (d) maintenance and repair cycles; and (e) accessibility.

3.5.2 Facilities and facility equipment. This subparagraph shall specify the constraints imposed on the system by the existing facilities and facility equipment.

Manufacturing and testing will drive facility needs, from size to cleanliness to equipment. You need to consider this as you define system requirements. Cost constraints may dictate the use of existing facilities that will drive requirements on the system in order to be able to use these facilities.

3.6 Personnel and Training. Where applicable, this paragraph shall specify requirements imposed by, or limited by , personnel or training considerations. Personnel considerations shall include the numbers and skills of personnel that will be allocated to the operation, maintenance, and control of the system. Training considerations shall include existing facilities, equipment, special/emergency procedures, hazardous tasks, and training simulators, as well as the need for additional facilities, equipment, and mission simulators.

3.6.1 Personnel. This subparagraph shall specify personnel requirements which must be integrated into system design. These requirements shall be stated in terms of numbers plus tolerance and shall be the basis for contractor design and development decisions. Requirements stated in this subparagraph shall be the basis for determination of system personnel training, training equipment, and training facility requirements. Personnel requirements shall include:

a. Numbers and skills of support personnel for each operational deployment mode and the intended duty cycle, both normal and emergency.
b. Skills and numbers of personnel that shall be allocated to the operation, maintenance, and control of the system.

3.6.2 Training. This subparagraph shall include the following training requirements, as applicable:

a. Restrictions on the type of training to be used for the system.
b. Constraints specifying the use of available Government training facilities and equipment for training purposes, where feasible.
c. Training devices to be developed, characteristics of the training devices, and training and skills to be developed through the use of training devices.
d. Limitations on the length of training time and on training locations.

3.7 Functional Area Characteristics.
(MilStd definition) This paragraph contains the allocation of system requirements from 3.2 to the segments defined in 3.1, and provides the basis for Section 3.2 of the segment specification. In a preliminary document this section may not exist if the allocation has not been performed. Where all segment specifications exist, they may be simply referenced here.

(Interpretation of MilStd). This description might imply that every requirement in Section 3.1 that is allocated to a segment should be repeated in this section. DO NOT repeat requirements. If a resource is allocated in Section 3.1, then the distribution of that resource to each segment should be made in 3.7.

Other information that should be contained in this section includes the requirements that are system driven but are segment unique. This most often will be encountered in large systems that comprise many elements, e.g., a system that contains a space vehicle plus and ground and operations segments. The space vehicle will have many requirements that do not relate to the other segments and stating them as system requirements in Section 3.1 will be confusing. Locate them in this section instead.

3.7.1 Segment A Characteristics. This contains the allocated and unique Segment A performance requirements (3.7.1.1), physical characteristics (3.7.1.2), and interface characteristics (3.7.1.3).

3.7.2 Segment B Characteristics. (As above)

3.8 Precedence. This paragraph shall specify the order of precedence of requirements, such as the specification over drawings, functional requirements over physical requirements, adherence to specified processes over other requirements, etc.

Section 4 VERIFICATION

Section 4 shall include all verification inspections necessary to ensure that the system complies with the requirements specified in Sections 3 and 5 of the specifications. Methods utilized to accomplish inspection include analysis, demonstration, examination, and test.

4.1 Contractor Responsibility. This paragraph shall clearly state the contractor's responsibility for inspection, compliance, and product quality.

4.2. Verification inspections. This section shall list all analyses, demonstrations, examinations, and test requited to verify that all requirements of Sections 3 and 5 have been achieved in the system. This paragraph shall also reference a Requirements/Verification Cross Reference matrix which cross references performance, design, and packaging requirements stated in Sections 3 and 5 with the appropriate inspection requirements stated in Section 4.

4.2.X Detailed Inspection Element X. A separate subparagraph shall be included for each detail element of the inspection to be conducted on the system.

4.2.X.1 Inspection conditions. When applicable, this subparagraph shall specify the environmental conditions under which the inspections are to be performed.

4.2.X.2 Methods of inspection. This subparagraph shall describe, in detail, the inspection to be used on the item. The description shall include the inspection method, location, and number of inspections, inspection routine, and criteria for determining conformance. Inspection methods appearing in standards and in other appropriate standardization documents shall be included only by reference.

4.2.X.3 Inspection equipment. When applicable, this subparagraph shall include requirements relating to the adequacy of the inspection equipment. Where specific inspection equipment is critical to accurate performance of the specified inspections, it shall be identified in this section. Statements regarding the contractor's responsibility for maintenance and calibration of inspection equipment shall also be included.

4.3 Special tests and examinations. When applicable, this subparagraph shall specify any special test, examinations or associated actions required as a part of the inspection of the system.

The following were included in 490A, but are not in 490B. These will probably be part of section 4.2 when it is expanded for each Element X, and are included here for information. Note that each of these defines "tests" but other methods could, and may have to be, used.

Lowest Tier Quality Assurance Provisions. This defines parts and materials testing.

Engineering Design Verification. This defines any unique testing using engineering models, prototypes, or mock-ups.

Qualification Tests. This defines tests conducted to prove the design of equipment and components. Items are stressed above highest levels predicted for operational use. Usually conducted on the first article.

Acceptance Tests. This defines tests conducted on each article to prove workmanship. Items are stressed at or above normal operating environment.

Service Life Verification. This defines life testing to be conducted, particularly on limited-life items.

Pre-Deployment Tests. This defines System tests to be conducted prior to operational use.

4.2.7 Operational Tests. This defines tests to be conducted after system is operational to provide final verification of system capability to meet requirements.

Section 5 PREPARATION FOR DELIVERY

May reference packaging standards. This is not applicable to a system specification.

Section 6 NOTES

This is the only section which is not contractually binding. it may contain definitions, abbreviations, acronyms, and other information useful in promoting an understanding of the system or its operation. Material in Section 6 must not reference material in other contractual binding sections. Items to be included are:

a. Glossary of terms
b. List of acronyms

APPENDICES

This contains tailoring, environmental data, evaluation methodology, or other data which has too much detail for incorporation in the main body of the document. Appendixes are contractually binding.

Part 2 -- EXAMPLES OF REQUIREMENTS IN A SYSTEM SPECIFICATION

Two examples will be used to demonstrate what belongs in the requirement sections of the specification -- the Assured Crew Return Vehicle (ACRV) and the Simplified Aid for EVA Rescue (SAFER). The text provided here is from the initial release of the ACRV System Performance Requirements Document (SPRD), November 9, 1988. The SAFER text is a composite of their PMP and PRD documents. Since SAFER is an in-house project, much of the description data (Section 3.1) was put in the PMP and not repeated in the PRD, as would be required for a contracted effort.

These examples are necessarily incomplete, since providing the complete documents would make this document far too large. Neither project completed sections 3.3, 3.5, or 3.6 so there are a number of blank subsections in these areas. SAFER wrote their PRD such that most of the System level requirements were for the Flight Element, as opposed to their entire system. They put requirements for the other elements, Ground and Operations, in Section 7.1 and 7.2. Had they taken a classical approach, there would have been very few requirements in Sections 3.2 thru 3.6 and 3.7.1 would have been the bulk of the specification.

The original ACRV SPRD is a classic example of a system specification and copies are available upon request. The ACRV has modified the SPRD over time. In some instances they have violated some of the initial groundrules to which the original document was developed.

The SAFER PMP is a very good example of a PMP for an in-house effort. Since this PMP is for a prototype, it is not as detailed as may be required for production units.

Specification Section	**ACRV**	**SAFER**
3.1 **System Definition**	Stated that this section was descriptive and not binding on the design.	• *This section defines and describes the three SAFER system elements as shown in Figure 1.*
3.1.1 Missions	Described three missions, e.g., ? *Mission 1. Provide the crew with the ability to safely return to earth from the Space Station in the event NSTS flights are interrupted for a time that exceeds Space Station ability for crew support and/or safe operations.*	*Flight Test Project. The SAFER FTP represents the effort and requirements to develop a SAFER Flight Test Article for a near-term Space Shuttle Flight demonstration to satisfy DTO-661. FTP objectives, philosophy, overall approach, and technical baseline are stated in JSC 25563.* (JSC 25563 is the PMP - goals, objectives, etc., are in the PMP and were provided in Section 2 -- Example of Small Project -- Safer.)
3.1.2 Operations Concepts	Broke into pre-mission, mission, and post-mission concepts, e.g.: *Pre-Mission* ? *Mission planning for ACRV operations will be accomplished prior to the initial launch.......* ? *The ACRV System flight elements will be delivered to a facility at the launch site where the final preparations will be made prior to integration with the launch vehicle...* ? *The ACRV will remain attached in a quiescent state until the need for its use arises.....* *Mission Concepts* -- (this paragraph contained a general statement and then a concept for each of the three missions), e.g., *Unavailability of NSTS* ? *A preplanned evacuation will be required as a result of the NSTS ...* ? *The decision to use the ACRV will be made several weeks in advance...* ? *Final preparation of the ACRVs will include activation, loading, ingress, and checkout...* • *Post Mission Concepts*	(The following SAFER data was contained in the PMP.) *The SAFER operational concept is:* ***Space Shuttle*** ? *Carry SAFER on every flight.* ? *In case of an in-flight emergency, use SAFER to get EVA crew member to the problem area (e.g., External Tank Doors or a damaged tile or windshield).* ***Space Station*** ? *Each crew member wears SAFER during all EVAs.* ? *In case of adrift EVA crew member, use SAFER to get EVA crew member back to Station.* ***Flight Demonstration Mission*** ? *Perform a scheduled EVA with 4.75 hours allotted for SAFER activities.* ? *Two EVA crew members flight test SAFER.* ? *Evaluate overall system performance and both Shuttle and Station Mission scenarios.*

Specification Section	ACRV	SAFER
3.1.3 **System Description.**	Had several subsections. Described the unique aspects of the ACRV System, e.g., ? *Must support Freedom over its indefinite service life of not less than thirty years.* ? *Will be in a quiescent state a majority of its service life.* Described the Operational Phases, e.g., ? *Pre launch Operations -- Covers the period from delivery of the hardware to the launch processing area....* ? *Launch and Delivery Operations -- Covers the period from lift-off to the handover of physical control of the ACRV.....* ? *Attached Operations. -- Covers operations while the ACRV is physically docked with the Space Station and covers separation and berthing operations.* Described the Segments, e.g., ? *Flight Segment -- This segment consists of the ACRV flight vehicle and any other flight hardware or software required to integrate the ACRV with the launch vehicle and with the Space Station.* ? *Ground Segment* ? *Mission Support Segment*	? *The SAFER Flight Test Article is a small, self-contained, one-person, free-flyer that provides the EVA crewmember with adequate propellant and control capability to maneuver near the Space Shuttle Orbiter. It fits around the Extravehicular Mobility Unit life support system as shown in Figure 3 without limiting suit mobility. It provides six degrees-of-freedom control through a single hand controller that is attached to the space suit Display and Control Module. A minimum delta velocity of 10 ft/sec is provided with the initial charge and it can be recharged during and EVA in the Orbiter cargo bay. The unit folds for storage in the Orbiter Airlock Stowage Bag during launch and landing and when not in use of orbit as shown in Figure 4.*
3.1.4 Interface Definition	? *The ACRV system flight elements shall be capable of interfacing with the NSTS in accordance with ICD-2-19001.*	? *The SAFER FTA shall attach to the EMU PLSS interface points in accordance with ICD-HSD-4-0013-OC-0.*
3.1.5 Government Furnished Property	? *GFP will be identified during Phase B studies and will be described in the Segment Specifications.*	? *The SAFER FTA shall use the following existing equipment: MMU pressure regulator, pressure gage, toggle valve, and quick disconnect; and Shuttle qualified alkaline dry cell batteries.*

Specification Section	ACRV	SAFER
3.1.6 Organization and Management Relationships	? *The ACRV Office at the Johnson Space Center is responsible for the preparation, maintenance, and control of this document. This document is subservient to... The hierarchy of the requirements specifications for the ACRV Program is shown in.* ? *The ACRV SPRD specifies that there will be thr**e** segments ant that each segment's requirements will be defined in a Segment performance Requirements Specification. The segment documents will be developed by the Phase B contractors and will be approved and controlled by the ACRV Office. Each Phase B contractor will, with ACRV Office approval, define the elements of each segment.....*	The SAFER data is in the PMP and summarized here: ? The PMP defined in detail the list of offices that have responsibility for SAFER and for each listed detail tasks. Tasks included who would chair review boards and who would attend; who would provide specific hardware; who was responsible for each document; and a list of deliverables for each organization.
3.2 Characteristics		
3.2.1 Performance Characteristics	? *This section specifies the minimum performance requirements to accomplish he mission defined in Section 3.1.2 consistent with the operational concepts defined in Section 3.1.3. ACRV System general performance requirements are presented first, followed by requirements that apply to a particular operational phase.*	No lead paragraph.
3.2.1.1 General Performance	? *The ACRV System shall be capable of emergency activation and separation.* ? *The ACRV shall be capable of performing a medical evacuation of an injured or ill crewmember to earth.* ? *The ACRV shall be capable of landingsite selection.*	? *The SAFER FTA shall provide a free-flyer, self-maneuvering capability for an EVA crewmember* ? *The SAFER FTA shall provide six degree-of-freedom manual maneuvering control.* ? *The SAFER FTA shall provide crewmember-selectable, three degree-of-freedom Automatic Attitude Hold (AAH).*

Specification Section	ACRV	SAFER
3.2.1.2 Operational Phase Performance	*Pre-launch Operations* ? *The ACRV System shall be capable of integrating the flight elements with the launch vehicle.* *Launch and Delivery Operations* ? *The ACRV System shall be capable of transferring a passive ACRV between the launch vehicle and the Space Station.* *Attached Operations* ? *The ACRV System shall be capable of attaching the ACRV to the Space Station.* ? *The ACRV System shall be capable ofseparating the ACRV from the Space Station without Space Station assistance.*	No similar section required for SAFER.
3.2.2 Physical Characteristics	? *Each copy of the ACRV flight elements shall be identical in form, fit, and function.* ? *The ACRV System shall provide free volume and openings for emergency ingress in accordance with NASA-STD-3000.*	? *The SAFER FTA shall fold for stowage within the volume of the Airlock Stowage Bag.* ? *The SAFER FTA shall not exceed 110 lbm fully charged.* ? *The SAFER FTA shall be capable of recharge using the Orbiter Gaseous Nitrogen System MMU Recharge Interface.*
3.2.3 Reliability	? *No single failure, except failures of primary structure and pressure vessels in rupture mode, shall result in the inability of the ACRV System to safely accomplish its mission.*	? *The SAFER shall be a simplified system without functional redundancy. The design shall allow the SAFER crewmember to react to possible failures by shutting down the system through the power switch and the manual propellant isolation valve.*
3.2.4 Maintainability	? *The ACRV System shall be capable of on-orbit servicing and maintenance of the ACRV to satisfy the operational availability requirements.* ? *The ACRV System shall be capable of returning its flight elements and ORU's in the Orbiter.*	• *The SAFER FTA shall meet specified performance characteristics without service or maintenance for a minimum period of 60 days.*

Specification Section	ACRV	SAFER
3.2.5 Availability	? *The ACRV system shall be capable of an operational availability of TBD while attached to the Space Station.* ? *The ACRV System shall be capable of returning to a fully operational state within TBD time after use.*	None
3.2.6 Environmental conditions	***a. Natural environmental conditions*** ? *The ACRV System flight elements shall be capableof on-orbit operations in the natural environment specified in JSC 30425.* ***b. Induced Environmental restraints*** ? *The ACRV and OSE shall be capable of withstanding the induced environment as defined in JSC 30230, while attached to the Space Station.*	? *The SAFER FTA shall be designed to operate in a thermal environment consistent with a beta angle equal to +/-52 deg.* ? *The SAFER FTA shall be capable of being donned, checked out, and activated in the Orbiter Airlock environment.*
3.2.7 Transportability	? *Each ACRV System flight element shall, where practical, be capable of packaging handing and transportation without disassembly.* ? *The ACRV System shall be capable of being transported by existing commercial or Government vehicles.*	None
3.2.8 Reuse/ Refurbishment	? *The ACRV System flight element refurbishment shall be TBD*	? *The SAFER FTA shall be certified for exposure to a maximum of two Shuttle launches and landings.*
3.3 Design and Construction Standard	ACRV left the specifics TBD at initial issue with few exceptions.	? *The JSC design and procedural standards, JSCM 8080, have been screened with respect to the criticality and intended usage of the SAFER FTP, and those that are applicable are identified in Table II.*
3.3.1 Materials, processes and parts		
3.3.2 Electromagnectic radiation		

Specification Section	ACRV	SAFER
3.3.3 Nameplates and product marking		
3.3.5 Interchangeability		
3.3.6 Safety	? *The ACRV and OSE shall be compliant with the safety requirements of the Space Station , applicable to attached pressurized payloads, as specified in SS/ACRV IRD TBD.* ? *The ACRV System shall be compliant with launch site safety, as specified in KHB 1700.7.*	? *Safety, Reliability, and Quality Assurance provision shall be in accordance with NSTS 21096 and JSC 17841.*
3.3.7 Human performance/human engineering.		
3.3.8 System Security	None	None
3.4		
3.5 Logistics		
3.5.1 Maintenance	? *The ACRV System shall be capable of maintaining hardware in storage to meet the operational availability requirements specified herein.* ? *The ACRV System shall be capable of on-orbit maintenance in accordance with...* ? *The ACRV System shall be capable of using the Space Station resupply system for on-orbit resupply of its parts and consumables.*	

Specification Section	ACRV	SAFER
3.5.2 Facilities		
3.6 Personnel and Training		
3.6.1 Personnel		
3.6.2 Training		
3.7 Functional Area Characteristics	3.7.1 Flight Segment ? *The Flight Segment shall provide the necessary hardware to integrate the CERV into the Orbiter.* ? *The Flight Segment shall provide for the on-orbit reinstallation of the CERV into the Orbiter for return.* ? *The Flight Segment shall provide the hardware to attach and interface the CERV to the Space Station.* 3.7.2 Ground Segment ? *The Ground Segment shall provide the hardware/software necessary for the rescue of the crew and the recovery of the CERV.* ? *The Ground Segment shall provide for the removal of a passive crew from the CERV.* ? *The Ground Segment shall provide for the transportion of an incapacitated crew (during a medical evacuation of the Space Station) to a definitive medical care facility within 2 hours of agress from the CERV.* 3.7.3 Mission Support Segment ? *The Mission Support Segment shall support the medical evacuation of an ill-injured crewmember to earth in a time period not to exceed 3 hours from crew ingress to landing.*	3.7 Characteristics of Other SAFER Elements 3.7.1 Ground Support Equipment. *The SAFER system shall include the GSE necessary for testing, data analysis, handling, and shiping, as indicated in Figures 1 and 2. Specific performance requirements are established in the MSE Prime Item Specification* 3.7.2 Mission Support Equipment. *The SAFER system shall include the MSE necessary for developing flight timelines, procedures, and consumable requirements and for flight crew training as indicated in Figures 1 and 2. Speicifc performance requirements are established in the MSE Prime Item Specification.* 3.7.2.1 Flight Simlulation. A flight simulation capability shall be provided for software verification and crew training. (More detail requirements for the simulation follow in the SAFER PRD.)

APPENDIX A
RELATED PUBLICATION

Blanchard, Benjamin S., System Engineering Management (New York: Wiley-Interscience, 1991)

Davis, Alan M., Software Requirements Analysis and Specification (New Jersey: Prentice Hall, 1990)

Grady, Jeffery, System Requirements Analysis (McGraw Hill, 1993)

Lacy, James A., Systems Engineering Management -- Achieving Total Quality (McGraw Hill 1992.)

Rechtin, Eberhardt, Systems Architecting: Creating and Building Complex Systems (New Jersey: Prentice Hall, 1991)

Alaksen, Erik W.; Harwell, Richard; Hooks, Ivy; Mengot, Roy; Ptack, Ken, "What is a Requirement?", Arlington, VA, 1993, National Council on Systems Engineering.

Hooks, Ivy, "Space and Life Sciences White Paper Describing the Essential Components and Processes Involved in Developing Life Sciences Requirements", 1991.

Hooks, Ivy; Stone, Dennis, "Requirements Management: A Case Study -- NASA's Assured Crew Return Vehicle", Seattle, Washington, 1992, Proceedings of the 2nd Annual International Symposium of the National Council of Systems Engineering.

Hooks, Ivy, "Why Johnny Can't Write Requirements", Huntsville, Alabama, 1990, AIAA Space Programs and Technology Conference.

Hooks, Ivy, "SAFER: A Case Study", April 1994.

Hooks, Ivy, "Don't Start the Building without the BluePrint: Defining the Scope of a Project", pmnetwork, April 1994.

APPENDIX B
DEFINITION OF ACRONYMS

ACRV	Assured Crew Return Vehicle
CDR	Critical Design Review
CM	Configuration Management
DoD	Department of Defense
DOF	Degrees of Freedom
DTO	Design Test Objective
EMU	Extravehicular Mobility Unit
EVA	Extravehicular Activity
FRR	Flight Readiness Review
FTA	Flight Test Article
GFE	Government Furnished Equipment
GSE	Ground Support Equipment
ICD	Interface Control Document
IEEE	Institute of Electrical and Electronics Engineers
IRD	Interface Requirements Document
JSC	Johnson Space Center
KSC	Kennedy Space Center
KW	Kilowatts
MIL. STD.	Military Standard
MMU	Manned Maneuvering Unit
MSFC	Marshall Space Flight Center
NASA	National Aeronautics and Space Administration
N.MI.	Nautical Miles
NSTS	National Space Transportation System
ORR	Operational Readiness Review
PBS	Product Breakdown Structure
PDR	Preliminary Design Review
PMP	Project Management Plan
PP	Program Plan
PRD	Project Requirements Document
PSI	Pounds per square inch
RFP	Request for Proposal
RID	Review Item Disposition
SAFER	Simplified Aid for EVA Rescue
SAR	System Acceptance Review / Safety Analysis Report
SMS	Shuttle Mission Simulator
SOW	Statement of Work
SPRD	System Performance Requirements Document
SRR	System Requirements Review
SS	Space Station
SSTF	Space Station Training Facility
TBD	To Be Determined
WBS	Work Breakdown Structur

ABOUT THE COMPANY

The **Compliance Automation, Inc. (CAI)** team of recognized experts takes pride in implementing quality requirements management processes for its clients. The Company provides strong technical expertise in all areas of requirements development, analysis, management, and control.

CAI provides training and education in the requirements definition and management process. *Requirements Definition and Management* is an entertaining one-day seminar that explains the importance of good requirements, life-cycle relationships, and how to do a better job of writing, reviewing, and managing requirements. Over 4000 people from NASA, Navy, Air Force, NOAA, major aerospace companies, and commercial companies have participated in this seminar since January 1992, their comments include:

- *This has been one of the most informative useful classes I have taken as part of my training at NASA.*
- *Learned a lot.*
- *Great class!*
- *Easy to follow and interesting dimension.*
- *Good presentation - personal experience to back up agenda.*

In response to *Was the course well suited to your needs?:*

- *Yes, I wish I could have had this course before writing my first PRD.*
- *Definitely, This short course and a longer more detailed course are needed and appreciated.*
- *Very much so. Excellent picture of problems and process of requirements.*
- *Very much so. It is a course that all middle management must be forced to take!*
- *Yes, this is what I need to do my job effectively.*
- *Yes, I've needed info of this type for a long time.*
- *Yes, this class was very useful.*
- *Yes, this course fit the level of new and inexperienced engineers, just the level our people needed.*

In response to *Whom else could benefit from this class?:*

- *Top management*
- *The customer*
- *Program managers*
- *Anyone involved anyway with the requirements processes*
- *This course should be required for most new NASA employees*
- *Everyone should be required to take this class*
- *Procurement personnel.*

Based on customer demand other courses have been derived from the original. Two and three-day requirements classes are tailored to customer needs. A one-half day class Management and Requirements is provided especially for managers. This class focuses on what managers need to know to determine if they have requirements problems, and what they can do to fix the problems.

In addition to requirements consulting and training, CAI has developed and markets the VITAL LINK products for automating the requirements management process. VITAL LINK provides a life- cycle approach to capturing requirements, maintaining requirements documents, and creating traceability between requirements and between requirements and other information, such as test plans. It enables the user to capture related information such as verification phase and method, rationale, allocations, priority, ownership, etc. It provides a powerful on-line search capability and extensive report writing.

Visit our web site at:
http://www.complianceautomation.com

1221 South Main Street, Suite 204
Boerne, TX 78006
(830) 249-0308
FAX (830) 249-0309
email: sales@complianceautomation.com

About the Author. Ivy Hooks has a twenty-year NASA career plus over ten years in private industry. She currently serves as president of Compliance Automation, Inc. where in addition to management duties, she provides consulting and training services in requirements engineering and management to both government and industry.

Ms. Hooks began her NASA career as an engineer following receipt of a Bachelor of Science degree in Mathematics from the University of Houston. She also holds a Master of Science degree in Mathematics. Her career at NASA spanned the Apollo and Shuttle programs. She developed models for lunar lighting for training the crew to land on the moon, performed analysis in support of potential Apollo pad aborts, and analyzed a number of Apollo problems related to plume impingement.

In 1969 she became a member of the initial Shuttle design team and subsequently became the Separation Integration Manager for Shuttle. In this role, she was responsible for separation of the solid rocket boosters and the external tank, as well as the separation of the orbiter and Boeing 747 during the Approach and Landing Test. This responsibility included requirements definition -- both hardware and software, system design, and verification. It encompassed aerodynamic data acquisition and analysis, including plume effects; modeling of tank propellant slosh; development of complex simulations; tests of solid rocket plume impingement effects on orbiter and tank materials; and extensive trade studies and analysis.

Ms. Hooks was the Special Assistant to the Director of Johnson Space Center during the selection of the first Shuttle astronauts. Following this assignment, she assumed the role of Verification Manager for the Shuttle Flight Software and was later the Project Manager for Flight Software. In 1981 she spent a year as the Executive Assistant for the Director of Engineering and followed this with an assignment as the Manager of Data Systems for the Shuttle Program Office.

Her last two years at NASA were spent as the Branch Chief for Data Management and as the Branch Chief for Flight Software Development. Following her NASA career she joined Barrios Technology, Inc., a small-disadvantaged business supporting NASA.

In 1985, she joined Bruce G. Jackson & Associates (BGJ&A) as a consultant providing services to major aerospace companies. She also founded a small software company -- Space Support Services (S3), which developed PCDCOM, a telemetry decommutation tool.

In 1987, BGJ&A won a Small Business Innovative Research contract with NASA to develop an automated requirements traceability tool. That effort resulted in Document Director – a requirements management tool. The Windows version of the toll was released in 1995 and is know as Vital Link. During the early sales of this product, she identified a lack of education in the area of writing and managing requirements among both government and contractor personnel. This led to the development of the Requirements Definition and Management course that she has presented to government and industry since 1990. Most of her time is now devoted to developing and providing requirements training and consulting.

Ms. Hooks is also a noted public speaker. During her NASA career she addressed many different organizations and institutions and in 1983 was awarded the NASA Outstanding Speaker Award. She has also been speaker at a number of prestigious events around the world, including IBM 100% clubs and Golden Circle events from Bangkok to Boca Raton.

NASA has acknowledged her contributions with numerous awards including the Exceptional Service Medal. She also has received the Washington D.C. Jaycee's Arthur S. Flemming Award and the Federation of Houston Professional Women's Outstanding Woman Award. She is a Fellow in the Society of Women Engineers and a charter member of the National Council on Systems Engineering.