

A Scenario-Based Process for Requirements Development: Application to Mission Operations Systems

Dr. Duane L. Bindschadler¹ and Carole A. Boyles²
Jet Propulsion Laboratory, Pasadena, CA, 91109

The notion of using operational scenarios as part of requirements development during mission formulation (Phases A & B) is widely accepted as good system engineering practice. In the context of developing a Mission Operations System (MOS), there are numerous practical challenges to translating that notion into the cost-effective development of a useful set of requirements. These challenges can include such issues as a lack of Project-level focus on operations issues, insufficient or improper flowdown of requirements, flowdown of immature or poor-quality requirements from Project level, and MOS resource constraints (personnel expertise and/or dollars). System engineering theory must be translated into a practice that provides enough structure and standards to serve as guidance, but that retains sufficient flexibility to be tailored to the needs and constraints of a particular MOS or Project. We describe a detailed, scenario-based process for requirements development. Identifying a set of attributes for high quality requirements, we show how the portions of the process address many of those attributes. We also find that the basic process steps are robust, and can be effective even in challenging Project environments.

Nomenclature

- Level 2* = Refers to the project level of the engineering hierarchy - the Project System.
- Level 3* = Refers to the system level of the engineering hierarchy. Spacecraft, Mission Operations, Launch Vehicle, and Payload Systems are examples of Level 3 entities.
- MOS* = Mission Operations System. Also referred to as Ground System or Ground Segment.
- PSE* = Project System Engineering organization.

I. Introduction

THE need to develop "good" requirements and the notion of using operational scenarios as part of that development during mission formulation (Phases A & B) are widely accepted as good system engineering practice for space missions. In the context of developing a Mission Operations System (MOS), there are numerous practical challenges to translating that notion into the cost-effective development of a useful set of requirements. These challenges can include such issues as a lack of Project-level focus on operations issues, insufficient or improper allocation of requirements, flowdown of immature or poor-quality requirements from Project level, "missing" requirements at the Project level, and MOS resource constraints (personnel expertise and/or dollars). System engineering theory must be translated into a practice that provides enough structure and standards to serve as guidance, but that retains sufficient flexibility to be tailored to the needs and constraints of a particular MOS or Project..

This paper describes a process for requirements development that is sufficiently flexible to serve a variety of environments and constraints. It owes its general architecture to the approach described by Hooks and Farry [1], and which has been applied in NASA and numerous aerospace corporations. To that general approach, we add detail and structure designed to specifically support MOS development. It may be that many of those details and that structure will also be of more general use.

Requirements definition is a difficult process. At its heart is the attempt to capture the needs of a relatively small number of customers and other stakeholders and communicate those needs in a meaningful way to the engineers and managers ultimately responsible for delivering a product. The success or failure of that product primarily rests in the

¹ Asst. Program Manager for Operations, MGSS, M/S 264-235, 4800 Oak Grove Dr.

² Asst. Program Manager for Planning & Development, MGSS, M/S 264-235, 4800 Oak Grove Dr.

perceptions of the customer and stakeholders. Requirements thus represent one means for clearly and effectively recording the agreement between customers and Project as to what constitutes success. For a Mission Operations System, it may be difficult to establish and maintain clear communications with customer and stakeholders, much less effectively capture their requirements on the MOS.

This difficulty arises in the situation (common in NASA robotic missions) where the MOS is developed as one component of a flight project. During most of formulation and implementation, the flight hardware and software and the launch system occupy the vast majority of the attention of Project management and project-level system engineering. The Project Scientist and Science Team, and the NASA sponsor (usually represented by a Program Executive or Manager) are the key customers. Their attention also tends to focus on the flight system, in part due to the Project's focus. In addition, the hierarchical development of requirements distances the MOS from the customer. In effect, the "customers" for the MOS *requirements* are the Project System Engineer (PSE) and team, and the Project Manager. But by the time flight operations begin, these individuals have often moved off onto a new development effort. The customers for the MOS (the Project Scientist and Science Team) now look to the MOS organization to fulfill their needs.

By the time a Project turns its attention to the MOS, the system is commonly engaged in verification and validation activities, with some implementation work remaining to be completed. It is well known that fixing problems arising from missing or poor-quality requirements at such late dates involves significantly more cost than in earlier phases [1]. In practice, Projects spend more money, work their people harder, and take more risk than is necessary because of inadequate requirements development.

Other issues arise even if a Project's organization applies appropriate attention and resources to an MOS when a Project's system-level (Level 2) requirements do meet all the attributes of well-written requirements. It is not uncommon for Level 2 requirements to be incomplete, lack rationale, dictate implementation, be un-verifiable, or improperly allocated. If a Project adopts such requirements and proceeds toward implementation, is there anything that can be done at Level 3 for the MOS? Another way to pose this question is to ask if a hierarchical compartmentalization of the requirements process is possible. That is, can a Level N+1 system element develop a set of high quality requirements even if the Level N requirements are incomplete or lacking some quality attributes?

II. Requirements Process Description

In laying out a framework for requirements development, we wish to ensure that it will work properly both in cases where a complete set of quality requirements are allocated to the MOS, and those where completeness or quality are less than they should be. The requirements process should therefore work well in the ideal case, while also being robust to somewhat more realistic cases.

We set forward the following goals for the requirements process:

- (1) Production of requirements that (once allocated) properly represent the complete scope of work needed for the subsystems of the MOS. Such requirements are useful as a basis for both cost estimation and design.
- (2) Clear communication of the rationale for requirements, such that informed trades between cost, scope, schedule, and risk can be made within the MOS and MOS subsystems.
- (3) Clear communication upward (to Project management and system engineering) as to the basis for and context within which MOS has accepted allocated requirements. This basis and context should be available for review in later phases of the mission. This has utility both for verification activities and for effectively and clearly communicating rationale to MOS-internal and -external stakeholders).
- (4) Production of requirements that possess a complete set of quality attributes, including clarity, rationale, verifiability, etc.,

We have designed a requirements definition process with the above goals in mind. In the following discussion, we describe the process steps and rationale, and then how the process supports the above-noted goals. Development of operational scenarios is a significant aspect of this process. The basic process steps are diagrammed in Figure 1. These consist of

- Review and assessment of allocated requirements
- Resolution of issues with allocated requirements (if any).
- Documentation of basis for acceptance of allocations.
- Development of operational scenarios (see also [2]).
- Creation of draft requirements
- Identification of scenario - requirements relationships

- Validation of requirements
- Baselining of requirements

It is worth noting that we do not intend a cookbook approach. Requirements definition is inherently iterative and will require judgment as to when requirements are “good enough.” Individual steps can be scaled up or down in formality, and may even need to be re-ordered (see Section E below). MOS engineers and managers should consider this a rough map of unfamiliar territory, not the turn-by-turn directions from a GPS unit.

A. Response to requirements allocations

The first part of the process involves analysis of requirements levied upon the MOS. Conventional practice calls for these to be Level 2 (Project System level, e.g., "The Project shall...") requirements that are allocated wholly or partly to the MOS. This represents the standard case and context for our discussion of the requirements development process. We have also observed cases in which the Project System Engineer's requirements working group created Level 3 requirements ("The Mission Operations System shall...") and levied these upon the MOS. We would recommend against such an approach. After discussing the more-standard case, we also discuss how a scenario-based process can be adapted to provide robust requirements even if the inputs to this process are less than optimal.

We begin with review and assessment of the allocated requirements. The review focuses on two questions:

- (1) Is the requirement understood by the MOS team?
- (2) is the requirement implementable within the understood scope of the MOS?

If so, a basis for acceptance is documented. If not, a similar rationale is prepared to indicate reason for MOS non-acceptance.

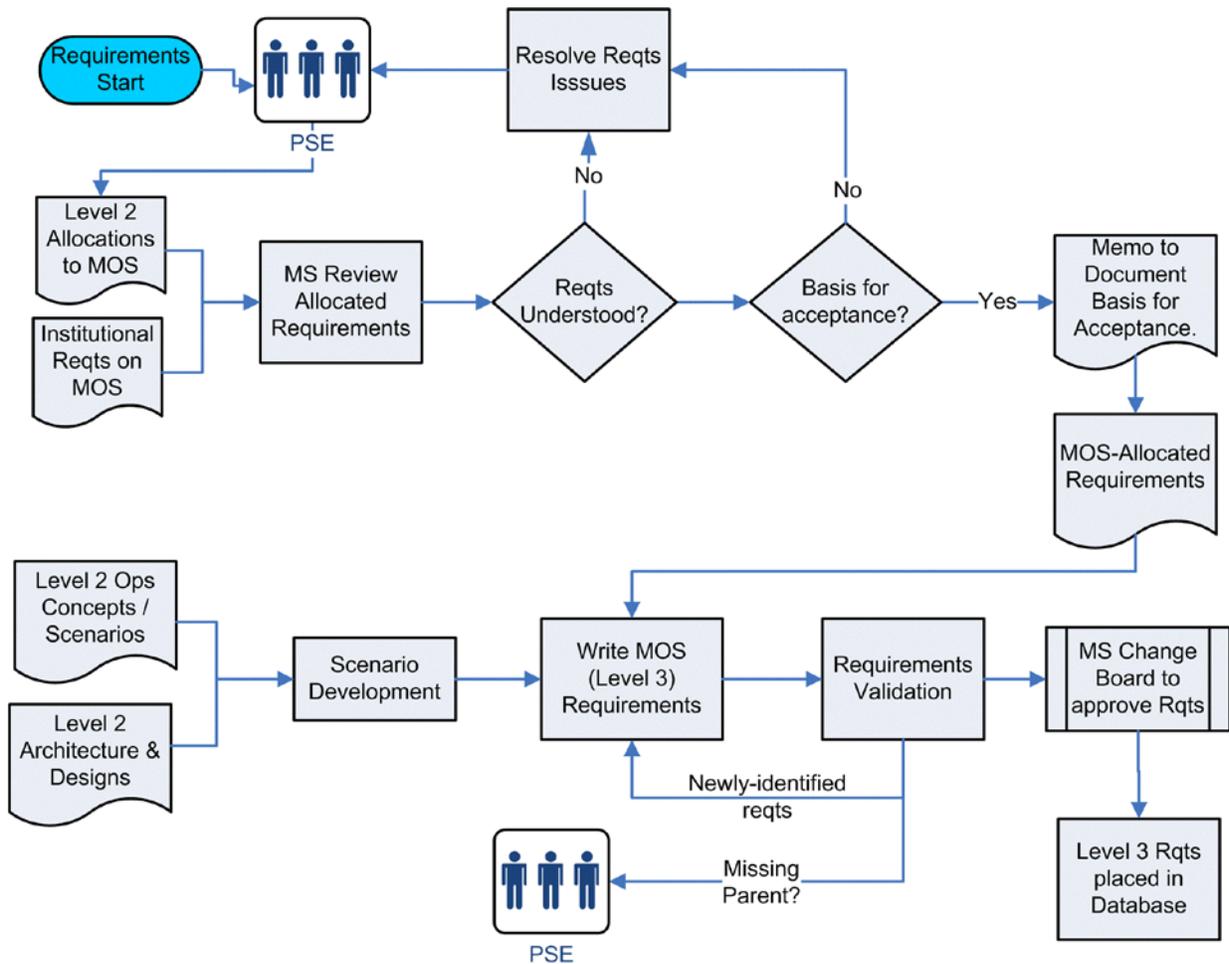


Figure 1. Flowchart of requirements development process.

The review is followed by discussion with the levying organization (PSE) to resolve issues, revise requirements if needed, negotiate any non-accepted requirements. Once issues are resolved, a memorandum is written to document the closure of issues, agreements made, and basis for acceptance of requirements by the MOS.

This portion of the process can be tailored, dependent upon the scope of the overall Project and MOS, the expertise levels of personnel, or the organizational interfaces involved. For example, a small technology demonstration mission done in-house may have minimal requirements on an MOS. Discussions between two or three key individuals may be sufficient to resolve any issues, and a relatively informal memorandum may be sufficient. A large project in which the MOS is contracted out or is done by a distinct organization will require more discussion to ensure clear communication between a larger number of individuals, and is likely to require more formal recording-keeping.

B. Development of Operational Scenarios

In parallel with (or even preceding) discussion of allocated requirements, the development of operational scenarios can begin. Here we describe scenario development in relationship to requirements development, although scenarios have multiple areas of applicability [3,2]. We first discuss a set of required characteristics for a scenario and the range of scenarios to be considered, then describe the process outlined in Figure 2 for developing scenarios.

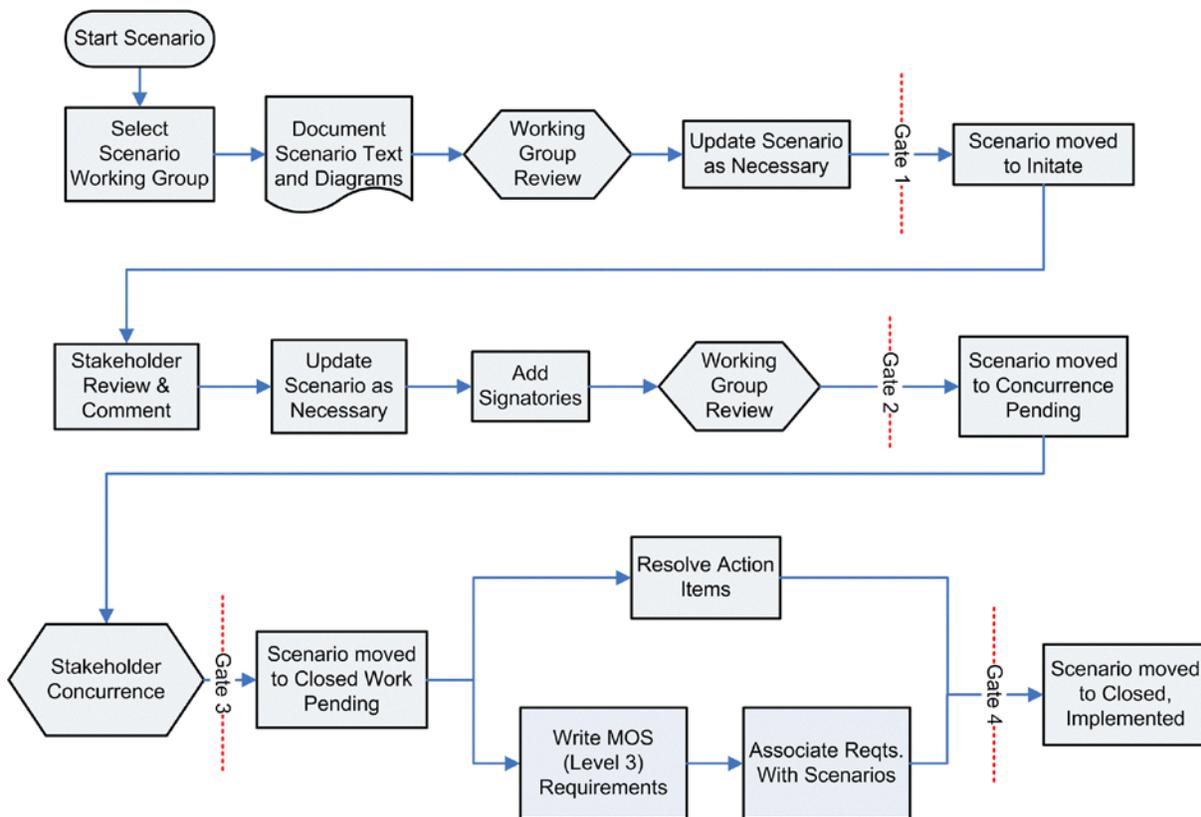


Figure 2. Scenario development process.

1. Scenario characteristics

Definitions of an operational concept and the utility and characteristics of scenarios are discussed throughout the literature [1, 4, 5, 6]. For our purposes, a set of characteristics were defined to allow the scenario information to be placed in a web-based tool and ingested into a database [2]. This approach can be considered transitional between traditional paper-based documentation and database-intensive approaches to information capture that will be needed to support model-based approaches to requirements and design.

All scenarios are considered as a set of textual blocks or units. Each scenario is characterized by a unique number, and has a number of attributes that support knowledge and control of the scenario's progress through the process, as detailed in [2]. The content of a scenario includes the following items:

- Title
- Start and End conditions for the scenario
- Flight System start and end conditions (if relevant)
- Purpose of the activity described in the scenario
- Introduction / Overview
- Assumptions and Constraints
- Interfaces used in the scenario
- Description

This was considered the minimum set for the purposes of MOS scenario definition.

2. *Scenario Scope*

Before scenario development can proceed, a set of scenarios must be defined. In cases where Project-level scenarios or an Operations Concept document already exists, the MOS scenarios must be responsive to the higher-level scenarios. If no such higher-level information exists, the MOS must consider the proper scope for its scenarios. In some cases, the task of writing a Project-level Operations Concept document is delegated to the MOS.

A hierarchical approach is useful. At the Project level, scenarios consider the interactions between the system-level elements (e.g., MOS, Flight System). At the MOS level, scenarios treat the interactions between various subsystem elements within the MOS. They may also elaborate details of the external interfaces (e.g., between MOS and Flight System). In practice, scenarios must also transgress such boundaries to the extent that the overall context and sequence of events is understandable. There are additional benefits to this transgression that are discussed below.

It is also useful to consider mission phase. As noted by [1] operational concepts can be useful for driving out requirements from all phases of a product's lifecycle. Some scenarios generally apply across multiple mission phases. Others have unique aspects within a particular phase. The scope of the overall Project and the MOS, the resources available to the MOS team, and the complexity and risk inherent in any particular mission phase or MOS activity must be considered in limiting the number and scope of scenarios to fit within budget and schedule constraints.

3. *Scenario Development Process*

The first step in developing scenarios is the selection of a working group responsible for producing scenarios. Members of the group are individually responsible for specific scenarios ("scenario owners"). One individual also functions as the Process Owner for scenario development and is responsible for ensuring that scenarios and any work associated with them are completed. Working group members should be familiar with the project-level documentation (as available), especially Level 2 Operations Concepts or scenarios, Level 2 requirements allocated to the MOS, and Level 2 architectural or design information.

As outlined in Figure 2, scenario development proceeds through a series of steps involving understanding and documenting the scenario and review and approval to proceed with further elaboration. Each review and approval step represents a "gate" at which the scenario must satisfy certain criteria in order to proceed. The criteria used for each gate are listed in Table 1.

Initial steps involve collecting information and distilling it into a written description and accompanying diagram(s). Both representations are important for communication and assessing completeness of a scenario. Once an initial draft is completed, the Scenario Working Group reviews it, updates are made as necessary, and the status of the scenario is moved from "Draft" to "Initiate." The change in status triggers notification of interested stakeholders (see [2] for details).

At this point, the scenario is available for wider review. In particular, project or flight system engineers may review scenario description and diagrams for completeness and correctness. For example, consider a scenario involving development and execution of a trajectory correction maneuver (TCM). Stakeholders may include propulsion, attitude control, and flight software subsystem engineers, as well as lead engineers for the Spacecraft, Navigation, and the MOS (with the latter representing the MOS development process for a TCM). Judgment must be applied at this point as to

Review	Criteria
Gate 1	Draft text and activity diagram.
Gate 2	Text and diagram complete. Comments addressed Actions documented
Gate 3	Key stakeholders concur (sign off).
Gate 4	Actions items resolved Inconsistencies with reqts. resolved.

Table 1. Review gate criteria. *These criteria are applied at the various review gates shown in Fig. 2.*

the level of detail that is useful to include in a particular scenario. In some cases, it may be desirable to break out a specific portion as a new, lower-level scenario to be worked in parallel, or to be deferred to subsystem-level requirements definition. The scenario tool [2] supports the concept of hierarchical scenario levels and allows (for example) the capture of a draft, lower-level scenario for later elaboration.

As any comments and changes are being incorporated, the scenario or working group lead will identify key stakeholders, from whom concurrence is desired. From the example TCM scenario above, one might identify the Spacecraft System Engineer, a NAV expert, and the MOSE as the signatories. Once the scenario is updated, the Working Group convenes a second review. Upon completion of the review, the scenario is ready to be moved to a Concurrence Pending state.

The purpose of the Stakeholder Concurrence step (Fig. 2) is to ensure that the scenario represents the overall Project Team's understanding of a given activity. Thus, both technical correctness/completeness and alignment of diverse parts of the Project organization are ensured. This latter aspect is particularly valuable to the MOS, helping to validate (or correct) assumptions about Flight System operation that drive the development of the MOS. Stakeholder concurrence is captured via electronic signature [2].

At this point, a scenario is sufficiently well defined to support requirements development (Fig. 1). This does not mean that scenario-related work is necessarily closed out. During scenario development, issues may arise or trade studies may be identified. These are captured as Action Items within the scenario tool [2] and process, and are closed out before the scenario is considered to be closed. In addition, scenarios are used during requirements validation and undergo some updates at that time. We discuss the step of associating requirements with scenarios below in the context of requirements validation. Once these steps are complete, the scenario Process Owner will be responsible for moving the scenario to a "Closed, Implemented" state.

C. Writing Requirements

Once the allocation of requirements from the Project to the MOS is established, and an appropriate set of operational scenarios developed, individual requirements on the MOS can be written. Good practice includes developing individual requirements that possess the following attributes [1]:

- 1) Reflect a need (as opposed to a desire or want).
- 2) Are verifiable
- 3) Are attainable
- 4) Are clear
- 5) Are complete

and avoid the following characteristics

6. Are based on flawed or incorrect assumptions
7. Are directing implementation (rather than stating a need).
8. Are describing operations
9. Are ambiguous, difficult to understand, or poorly written
10. Are overly specific.

We address these items by three different aspects of our overall requirements process. First, we establish a set of attributes that accompany the text of each requirement. Each entry in the requirements database must include these attributes. Second, as described above, we develop operational scenarios prior to writing requirements. These scenarios function as a shared context for requirements writers and the rest of the MOS (and possibly other elements of the Project as well). Third, having developed operational scenarios, we also take the additional step of associating each requirement with the appropriate portion of a scenario.

The attributes established for each requirement are implemented as part of the schema for our requirements database. Modern requirements tools are built as specialized front-ends for a database. Definition of the schema for requirements is an important aspect of the requirements process. Because there is overhead involved in the use of such tools, we have taken the approach of using a spreadsheet template to develop the requirements text and attributes, and then placing requirements into the more-formal tool after they have been reviewed for quality. The initial template requires the following information for each requirement (in addition to the requirement statement itself):

- Parent requirement
- Mission Phase (indicating during which phase the requirement must be met)
- Affected MOS subsystems (this represents a tentative allocation).

- Verification method to be used (e.g., test, analysis, demonstration, inspection)
- Associated Scenario(s)
- Rationale (why is this requirement needed)
- Basis for acceptance

Taking the higher-level requirements allocated to the MOS (Figure 1) as the starting point, engineers develop the requirement statement and the additional information noted above. Involving personnel who developed scenarios will help in ensuring the proper context is considered.

The initial reaction of both engineers and management may be that documenting all of this information in association with each requirement seems like a good deal of work and "is it really necessary?" The authors' experiences on a large astrophysics mission indicate that the wasted effort that results from insufficient rigor in defining requirements can easily outweigh the effort suggested here.

Documenting more than just the requirement text should reduce the number of ill-considered requirements. Consideration of each requirement item means that different aspects of the requirement are considered as it is being written. Moreover, any unattainable or unverifiable requirement that is flowed downward into multiple lower level elements can create significant unnecessary work during the later stages of development, introducing unnecessary risk and cost when most flight projects are under the most stress.

D. Validation of Requirements

Once requirements are written, validation can begin. We consider requirements validation in the same manner as [1]; that is, as the step of ensuring a requirement properly communicates the need it is intended to represent. Each requirement and its attributes are thus reviewed. Because we have included capture of the attributes noted above for each requirement, some aspects of validation should already be accomplished.

In addition, each requirement is directly associated with the appropriate portion of its associated scenario(s) (further described in [2]). Each requirement will be viewed in the context of one or more applicable scenarios. This creates opportunities to judge the consistency between specific portions of a scenario and a requirement. It can also be used to assess completeness of requirements -- for example, a lack of any requirements associated with a particular portion of a scenario may mean that requirements have been missed at both the MOS and Project levels. Finally, the juxtaposition of requirement and scenario text helps to identify whether requirements satisfy some of the quality attributes listed above, particularly numbers 6-10.

Table 2 gives an overview of how the quality attributes for requirements are addressed within various portions of the requirements process. In particular, we note that scenario development, and the direct association of requirements with particular parts of a given scenario significantly improves the number of quality attributes addressed by the process.

As requirements are validated, the expectation is that inconsistencies will be identified between the higher-level allocations, the MOS-level requirements, and the operational scenarios. These are addressed by updating one or more of these related items. For example, scenarios may help to identify missing requirements. The MOS may then derive the necessary requirement and suggest to the Project System an appropriate parent (Figure 1). In other cases, an incorrect assumption embedded in a scenario may be exposed in light of the associated requirements, and an action item will be generated to alter the scenario (Figure 2). The value of identifying and correcting such issues during requirements development, rather than in later phases of a project is well-documented across multiple industries. Once requirements are validated, the process calls for them to be approved via a formal change process. Only then are they placed into a

	Addressed by Requirement Schema	Addressed by Scenario Development	Addressed by Scenario / Req Linkage
Reflects a need	Yes	Partially	Partially
Verifiable	Yes		
Attainable	Yes	Partially	
Clear			
Complete		Partially	Yes
Based on good assumptions		Partially	Yes
Not implementation			Partially
Doesn't describe operations		Partially	Yes
Unambiguous / well-written			Partially
Not overly specific.			Partially

Table 2. Assessment of methods for addressing requirements quality.

formal requirements management tool.

E. Non-ideal Project Environments

Not all Projects will provide to the MOS an allocation of high-quality requirements and a set of useful high-level operational concepts or scenarios. Here we consider an example of a realistic project environment and how the basic process steps described above can be adapted to yield a useful requirements set.

In our experience, the requirements allocated to the MOS have had a number of issues related to completeness and rationale. We have also noted situations in which Level 2 requirements were used as inputs by Project System (Level 2) engineers and used to write Level 3 requirements on the MOS. These requirements were given as the allocation to MOS. In some cases, "allocation" to the Level 3 MOS was accomplished by replacement of "The Project System shall..." with "The Mission Operations System shall..." without any other changes to the requirement text. Insistence that the levied requirements are inappropriate or of insufficient quality may not be an effective remedy for such problems.

Consider the situation in which Level 3 requirements were developed without the benefit of appropriate scenarios or via the less than optimal allocation process as described above. When faced with such a situation, we found the same basic steps in the requirements and scenario process to be useful, even if re-ordered. The same initial steps to assess and respond to the allocated requirements were taken. We then proceeded with development of operational scenarios. Once the scenarios were in a draft state (through Gate 1), the existing requirements could then be associated with the scenarios. In our nominal process (Figure 2), no requirements would have been written at this point. Juxtaposition of the (possibly immature) requirement text with a specific portion of a scenario as a test of many of the quality criteria for requirements (Table 2) was quite effective in making apparent issues such as incorrect assumptions, or requirements that described implementation or execution of operations. In addition, lack of completeness will become apparent when important aspects of scenarios are found to have no existing requirements associated with them.

III. Conclusions

The importance of good requirements development to a successful implementation of a Mission Operations System is often inconsistent with the emphasis it is given in early phases of a Project's lifecycle. A structured process, based on well-understood fundamental principles for requirements development, and aided by relatively simple software tools [2] is shown to address many of the quality attributes needed to produce a good requirements set. Operational concepts or scenarios are useful in creating a shared context in which to develop requirements. The direct juxtaposition of requirements text with scenarios represents a powerful check on requirements quality. We also find that by detailing the steps involved in a scenario-based requirements development and their proper sequence, we are able to adapt that sequence to improve requirements development even in less than optimal Project environments.

Acknowledgments

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

¹Hooks, I.F, and Farry, K.A., *Customer Centered Products: Creating Successful Products Through Smart Requirements Management*, 1st ed., AMACOM/American Management Association, New York, 2001.

²Boyles, C.A., and Bindschadler, D.L., "Web Based Tool for Mission Operations Scenarios," *Proceedings of the SpaceOps 2008 Conference*, Heidelberg, Germany, 2008, (in press).

³Alspaugh, T.A., Anton, A.I., Barnes, T., Mott, B.W., "An integrated scenario management strategy," *Proceedings, IEEE International Symposium on Requirements Engineering*, Limerick, Ireland, 1999, p.142.

⁴Fairley, R.E., Thayer, R.H., Bjorke, P., "The concept of operations: the bridge from operational requirements to technical specifications," *Proceedings of the First International Conference on Requirements Engineering*, Colorado Springs, CO, 1994. p. 40-47.

⁵Fairley, R.E., and Thayer, R.H., "The Concept of Operations: the bridge from operational requirements to technical specifications," *Annals of Software Engineering*, Vol. 3, 1997, pp. 417-432.

⁶Weidenhaupt, K., Pohl, K., Jarke, M., Haumer, P., "Scenarios in system development: current practice," *IEEE Software*, Vol.15, No, 2, pp. 34-35.