Abstract

Documentation has been called the lifeline of computer systems. This is especially true for the Mission Operations Division (MOD) at the NASA Goddard Space Flight Center, where the high visibility of spacecraft missions requires quick and accurate resolution of computer problems. To develop documentation to support this environment, the MOD has devised a documentation structure and tailoring scheme that is flexible but not flimsy, ordered but not rigid, and focused without being limited. This paper describes the essential elements of this structure.

Keywords: Documentation Standards, System Life-Cycle

1 Introduction

Everyone agrees that computer systems must be documented. Why is it, then, that the documentation is often poorly written or not done at all? Jokes about programmers using English as their second language (after Fortran, etc.) are unfair. More often, budget and schedule constraints on the project delegate documentation to the end, where it disappears. The substantial costs to project development and operations & maintenance because of this practice have been reported for some time [6]. Others have noted that over-documentation is just as bad, that "the very voluminosity introduces a new kind of incomprehensibility" [4, p. 134]. What kind of documentation should be written, how should it be written, and when should it be written?

For spacecraft mission operations, we have found answers to these questions. To understand how our documentation structure might fit into another environment, you should first understand how the structure fits into mission operations.

2 The Mission Operations Environment

The Mission Operations Division (MOD) at the National Aeronautics and Space Administration (NASA) Goddard Space Flight Center (GSFC) is responsible for the development, maintenance, and operation of institutional and dedicated systems to support spacecraft missions. These systems include payload operations control centers, flight software, command management facilities, and spacecraft simulation systems. The MOD uses several life-cycle strategies to fulfill its mission, including the waterfall model, the rapid prototype model, and the evolutionary development model [2]. The diversity of these models requires a documentation structure that is flexible but not flimsy, ordered but not rigid, and focused without being limited. The structure proposed for the MOD, and described
in this paper, meets these criteria.

3 MOD Documentation Organization

In many organizations, all systems are developed using a standard set of documents that have been helpful in the past. The more successful the projects have been, the more ingrained the documentation set becomes. Large projects have fat documents; small projects have thin ones. The standard set of documents is universal for all.

The problem, of course, is that not all projects can or should have the same set of documentation. The extent of formal documentation required for large projects (such as the Space Station) can not be compared with that needed to upgrade a mobile simulations system. This need for more flexibility in systems management has been recognized in a recent U.S. Congressional report [5] and throughout the software industry [1].

The Mission Operations and Data Systems Directorate (M0&DSD), of which the MOD is a part, has set guidelines for life-cycle documentation content and delivery schedules [7] [8]. Using these guidelines as a basis, the MOD created a documentation organization that fit with highly structured life-cycles (such as the waterfall model) as well as with more flexible life-cycles (such as evolutionary and prototype models). A significant requirement in constructing this organization was that it be compliant with the system life-cycle model defined in the MOD Configuration Management Plan [9]. This life-cycle model, shown in Figure 1, incorporates the phase/review/baseline concepts [3] now used as an industry standard.

Note that the life-cycle phases each have a corresponding review. Because the review is the focal point for completion of a phase, we have classified documents for the review rather than for the phase.

MOD documents are arranged in a simple two-dimensional organization in which functional areas are one dimension and life-cycle reviews are the other dimension. The documents are ordered inside this row/column structure (Figure 2). The functional areas chosen are

- a. management,
- b. engineering & development,
- c. test/transition to operations, and
- d. operations & maintenance.

The life-cycle reviews, from Figure 1, are

- a. project approval review,
- b. system requirements review,
- c. system design review,
- d. subsystem requirements review,
- e. preliminary design review,
- f. critical design review,
- g. test readiness review,
- h. subsystem test review,
- i. system test review,
- j. configuration audit,
- k. operational readiness review, and
- l. operations and maintenance.

The functional areas were chosen to be separate entities involving different personnel, although the distinctions become blurred for smaller projects. In concept there can be four independent tasks proceeding in parallel for a
The management task should be first, with the Project Management Plan activities well understood before other aspects of the project are begun. The project manager starts the engineering/development task by assigning technical staff to develop requirements and design the system. Management also begins the test/transition to operations task by assigning test teams to evaluate each product. The operations and maintenance task, often an ongoing enterprise, accepts and installs completed products. The MOD documentation organization shown in Figure 2 was developed to support this natural flow of system development.

Complete descriptions of the functional areas may be found in [lo]. The careful reader will notice that this reference is for "documentation guidelines" rather than "standards". We have found a good deal of confusion exists over these terms. Must all documents defined in a "standard" be written? If a document is described in a "standard", must it always follow the defined structure? In our case, the answers to both these questions is no. From the configuration management and publications perspectives, however, certain standards must exist. For instance, the document identifier must be unique and must follow a standard. Signature pages for documents under configuration control must contain certain signatures. Procedures to change a document must be standardized. The "look and feel" of documents, including the cover and type of tables of contents, should be consistent throughout the Division. These are standards that can be accepted for all documents.

There are documents with similarities (such as Acceptance Test Plan vs. System Test Plan vs. Subsystem Test Plan) for which writing assistance, in the form of guidelines, is needed. The guidelines suggest the purpose and scope of each document, along with a proposed table of contents. Project managers can decide to combine documents or write separate documents, as appropriate. The guidelines act as a checklist of topics to be covered. However, the decision of which set of documents to develop is not left to the project manager alone.

The mechanism used to tailor the documentation set is the Project Management Plan (PMP), presented during the System Requirements Review. This document, the only one required for all projects, describes the project life-cycle phases and the extent, formality, and schedule for documents to be written for each phase of the project. In the PMP the project manager can specify a subset of reviews, baselines, and documentation appropriate for the project. Upon acceptance by the project governing authority, the PMP will define the functional life-cycle model for the project. Care must be taken in preparing the PMP because project costs and schedules will be based on the work specified in it. A sample PMP table of contents is shown in Figure 3.

The PMP also defines which documents will be controlled by the MOD Configuration Control Board (CCB) and which will be uncontrolled. Controlled documents, always significant to project development, require CCB approval before they can be changed [3] [9]. Examples are requirements and interface documents. A project manager should require CCB control over any document that should have a wide review or that impacts more than one line management
organization. Uncontrolled documents, such as training material and test reports, do not face the extensive review process given the controlled documents.

4 Summary

System life-cycle management for spacecraft mission operations is too diverse for a single standard documentation set. Four functional areas for documentation have been described: management, engineering & development; test/transition to operations; and operations & maintenance. Documents for these functional areas are further delineated by life-cycle review. A method, using the Project Management Plan, was described to tailor the complete documentation set into a subset for specific projects. The MOD has successfully used the methods described in this paper to manage project development using the waterfall, rapid prototype, and evolutionary life-cycle models. However, the techniques of the present may not apply to extremely complex missions of the future (such as the Earth Observing System) that include cross-linked databases serving scientists and engineers around the world. Research in life-cycle management [1] will continue to be monitored for applicability to MOD projects.

PROJECT MANAGEMENT PLAN

1.0 Introduction
2.0 Related Documentation
3.0 Management Approach
3.1 Overview
3.2 Organization
3.3 Work Breakdown Structure
3.4 Master Schedule
3.5 Cost Account Schedules
3.6 Resources
3.7 Reviews and Reports
3.8 Deliverables
4.0 Equipment Acquisition Planning
4.1 Procurement Activities
4.2 Equipment Evaluation Criteria
4.3 Procurement Risks
4.4 Applicable Standards
5.0 Development Planning
5.1 Engineering and Integration
5.2 Configuration Management
5.3 Quality Assurance
5.4 Software Development Plan
5.5 Hardware Development Plan
5.6 System Test Plan
5.7 Risk Management
5.8 Training
5.9 Delivery and Operational Transition
6.0 Sustaining Engineering and Operations
6.1 Sustaining Engineering and Operations Process
6.2 Configuration Management
6.3 Quality Assurance
6.4 Operator Training
6.5 Logistics
7.0 Abbreviations and Acronyms
8.0 Glossary
9.0 Appendices

Figure 3 - Project Management Plan
Sample Table of Contents
References


