RECEIVED
MAR 21 1990
T. A. ALMAGUER

GALILEO AACS LESSONS LEARNED

EDITED BY
J. CHODAS

MARCH 1990

D-66273

GALILEO

AACS

LESSONS

LEARNED

EDITED BY

J. CHODAS

MARCH 1990

GALILEO AACS LESSONS LEARNED

Introduction

The Galileo Attitude and Articulation Control Subsystem (AACS) Lessons Learned were originally compiled in late 1986 in order to capture the experiences of the development period for the 1986 launch opportunity. Brainstorming sessions were held and various individuals wrote up the lessons. Some lessons have been added since that time period to reflect the VEEGA development. It is hoped that this document will prove to be useful for future projects in the Guidance and Control Section.

The document is organized into six sections :

A - Management
B - Subsystem Engineering
C - Hardware Design & Fabrication
D - Contracting
E - Analysis/Software
F - Subsystem Testing

Highlights of each section are noted below.

Management

The Management section emphasizes schedule, cost and workforce issues. Good task planning with high and low level schedules is needed to detect problems early and to be able to react quickly. Detailed cost estimates are required to understand and track the task, and solid justification is needed to avoid having costs cut back unfairly by the Project.

Good communication is always a necessity. Staffing must be well structured and adequate for the job, and the proper authority must be delegated appropriately by upper management to allow the task to progress effectively.

Subsystem Engineering

The Subsystem Engineering section emphasizes the subsystem engineering role and how it interacts with other areas in the section. The Subsystem Cognizant Engineer (Cog E) role must be

defined clearly. This person is the interface between Systems and the rest of AACS. The Subsystem Cog E needs to be supported by a subsystem engineering staff that is large enough to get the required tasks and reviews done in a timely manner. Administrative support is very effective in offloading the routine paperwork tasks from the subsystem engineering staff.

The Spacecraft Design Team meetings should be attended by the Subsystem Cog E or his delegate, and relevant information should be passed on to the appropriate subsystem personnel, perhaps during periodic subsystem design team meetings.

Hardware Design & Fabrication

The Hardware Design & Fabrication section emphasizes hardware design, development and fabrication issues. There is no such thing as off-the-shelf hardware. Modifications are always required and are rarely cost effective, especially on inherited hardware. Breadboard and engineering model units are essential to verify designs, aid in troubleshooting, support subsystem and system testing, etc. It is important to allocate enough schedule time to factor changes found in these prototype developments into the flight hardware build.

The JPL in-house hardware fabrication capability was used extensively during GLL AACS hardware development for both the 1986 and 1989 launch opportunities, and was often essential in order to meet the overall spacecraft schedule. The structured hardware development process that was used for the VEEGA hardware builds proved to be very successful, especially in view of the tight schedule constraints.

Contracting

The Contracting section emphasizes contract preparation and implementation issues. Sound system and subsystem designs are needed before effective contracts for subsystem elements can be developed. A complete and detailed Request For Proposal helps to minimize changes during the course of the contract.

It is important that the Contract Technical Manager use an accounting system to evaluate contractor performance against schedules and costs. It was very effective to have engineering and

quality assurance representatives resident at the contractor site to maintain good and efficient communication. Timely support from Division 35 is also essential to minimize material and processes problems.

Analysis/Software

The Analysis/Software section emphasizes requirements and software development and test. Requirements need to be clearly stated and testable. In particular, fault protection requirements should be traceable to the subsystem Failure Modes and Effects Analysis and should be developed early enough to be able to influence the hardware designs. Algorithms need to be developed early and integrated together in the analysis area to uncover problems before they are delivered to the flight software or test areas. During this phase, computer simulation model development should be coordinated within the subsystem so that effort is not duplicated.

The interface between the algorithm developers and the flight software designers needs a lot of attention to minimize miscommunication and misunderstandings. Good algorithm documentation is essential and aids in this area. Flight software prototyping helps to estimate memory usage and timing. To be really effective, the flight software acceptance testing task needs to become part of the flight software life cycle. Finally, all areas should participate in subsystem testing in order to better understand how the subsystem functions as a whole.

Subsystem Testing

The Subsystem Testing section emphasizes test planning, execution and documentation as well as operations planning. Subsystem testing is required in all phases of hardware and software development. Success criteria for the tests should not be derived from test results but rather from test requirements. Test planning may show a need for several test beds with varying levels of complexity. It is essential that each test bed be certified to minimize test bed-related problems.

Subsystem testing requires participation from several areas (analysis, flight software, hardware). The test personnel can be more effective if they have experienced both system and subsystem level testing. The test documentation needs to be thorough in order

to be able to troubleshoot problems effectively. The sooner the test results are reviewed, the easier it is to diagnose and fix problems.

Operations planning needs to be staffed early and at a sufficient level to allow spacecraft operations and strategies to be coordinated with development products. Consolidating the operations support in one group provides closer interfaces and helps communication.

GALILEO AACS LESSONS LEARNED

SECTION A - MANAGEMENT

TABLE OF CONTENTS

NUMBER	TITLE
A-001	AACS SPLIT BETWEEN PROJECT OFFICES
A-002	RESOURCE ESTIMATES
A-003	INADEQUATE TECHNICAL MANAGER AUTHORITY
A-004	INADEQUATE STAFFING
A-005	COG E RESPONSIBILITIES
A-006	EARLY WARNING OF PROBLEMS
A-007	INTERNAL DELIVERY SCHEDULES
A-008	MONTHLY MANAGEMENT REVIEWS
A-009	FLOW-DOWN OF INFORMATION
A-010	SCHEDULE STANDARDIZATION
A-011	PERSONNEL PROBLEMS
A-012	LESSONS LEARNED MATERIAL

GALILEO AACS LESSONS LEARNED

DISCIPLINE MANAGEMENT DATE 10/21/86

AUTHOR R. G. HAGA

PROBLEM AACS SPLIT BETWEEN PROJECT OFFICES

A communication problem existed due to dividing the AACS between the MO&E and Orbiter offices. Each office had different priorities, reporting requirements, etc.

LESSON LEARNED

It's difficult to serve two masters. If possible, the AACS hardware and software (and supporting analysis) should be under one project office.

RECOMMENDATION

Reevaluate the AACS split between the various project offices, especially the MO&E and the Orbiter/Spacecraft offices. Inform new project managers of this desirable organization before the work breakdown structures are finalized.

GALILEO AACS LESSONS LEARNED

DISCIPLINE MANAGEMENT DATE 10/25/86

AUTHOR D. G. CARPENTER

PROBLEM RESOURCE ESTIMATES

Accurate estimates of workforce, services, and procurement costs must be made at the start of a project/task.

LESSON LEARNED

Pressures always exist to produce minimum cost estimates, based on the concept of a "success-oriented schedule." Success-oriented schedules are never achieved in this business, so cost estimates are never met. In order to sell more accurate cost estimates, it is necessary to back up the estimate with solid historical data. There will always be changes that require additional resources during the course of a program but it is very important to establish and document a solid baseline estimate. It is always easier to sell a cost estimate if it is backed up by well documented historical data.

RECOMMENDATION

Develop a solid historical cost file for each element of the subsystem. Both plan and actuals as well as changes (scope and overrun) should be included. This file should be set up at the start of all major programs, and maintained through completion. A subsystem project administrator should be assigned to each major project within the section to handle this task.
Develop a check list of all tasks, subtasks, services, procurements, etc. to aid in the generation of a cost estimate. A check list should be made for each element in the subsystem, and revised as experience dictates.
Develop a cost model of the subsystem as well as the individual elements within the subsystem. Use this cost model to verify cost estimates, and as an aid in selling the estimate to Project.
It is good practice to bury a 5-10% contingency in all cost estimates. (Probably shouldn't publish this.)
Provide proper training for individuals responsible for providing cost estimates. Consider a professional development course.
Utilize SRM simulation computer software to iterate the SRM estimate.

GALILEO AACS LESSONS LEARNED

DISCIPLINE CONTRACT MANAGEMENT DATE 7/20/88

AUTHOR G. M. BURDICK

PROBLEM INADEQUATE TECHNICAL MANAGER AUTHORITY

Technical managers for major assembly contracts (SAS/SBA, SS, AACSE) were given the responsibility for the activities without the corresponding authority. This was evidenced by the fact that Project, Division and Section management routinely attended many of the monthly management reviews and gave technical direction to the contractor. The technical manager's role, in many cases, was reduced to implementing the decisions of multiple layers of management. By diluting the technical manager's authority, his effectiveness in daily interactions with the contractor was reduced.

LESSON LEARNED

It is more effective and efficient to delegate authority along with the responsibility for any task, especially contracted items.

RECOMMENDATION

Minimize the management overhead involved in contracted activities by delegating the authority for decision-making and technical direction to the responsible technical manager. Limit, as much as possible, the number of managers attending monthly management reviews and depend on reports from the responsible manager.

GALILEO AACS LESSONS LEARNED

DISCIPLINE MANAGEMENT DATE 11/02/86

AUTHOR G. M. BURDICK

PROBLEM INADEQUATE STAFFING

Inadequate staffing was tolerated partly because it was felt that adding staff was inefficient or not fully justified by the budget. This led to chronic underachievement in several areas (subsystem engineering, test, software, and electronics). And surprisingly, due to attrition, the budget was underrun frequently in these areas. The cost of schedule delays due to underachievement cannot be estimated.

LESSON LEARNED

It is almost always beneficial to add needed staff whenever a shortfall is identified. A slight overstaffing is justified to gain schedule leverage and to account for attrition.

RECOMMENDATION

Adequate staffing should be planned based on similar experience. When a shortage of workforce is identified, staff should be added regardless of what stage the program is in. A slight overstaffing is justified to gain schedule leverage and to account for attrition.

GALILEO AACS LESSONS LEARNED

DISCIPLINE MANAGEMENT DATE 10/29/86

AUTHOR W. J. GOULET

PROBLEM COG E RESPONSIBILITIES

The Flight Software Engineering Group did not take advantage of the positions of Cog E and group leader with respect to the division of responsibility.

LESSON LEARNED

The AACS flight software was basically developed with the Cog E doing the job of both Cog E and technical group leader. The Cog E should only worry about technical issues and not be consumed with the day-to-day administrative activities. In this instance, the effectiveness of the Cog E was, at times, severely diluted. This applies to hardware tasks also, where assignment of the Cog E and Task Manager roles to two individuals works well.

RECOMMENDATION

At the start of a project, a work breakdown structure must be developed. Depending upon the scope of the development task, one or more Cog E's should be appointed to direct the technical activities. If the development staff is large and/or the group is supporting multiple projects, an appropriate number of technical group leaders must be named. The TGL will get involved to some extent in technical matters but must be primarily responsible for providing the resources necessary to accomplish the Cog E's task.

GALILEO AACS LESSONS LEARNED

DISCIPLINE MANAGEMENT DATE 11/02/86

AUTHOR G. M. BURDICK

PROBLEM EARLY WARNING OF PROBLEMS

Late identification of problems which affect schedule or cost may not be handled effectively due to inadequately allocated resources. For Galileo AACS, the chronic staffing shortfall in subsystem engineering, test, and software was, in part, the result of not being able to identify early enough the problems which resulted. In fact higher runout costs may have resulted from the lack of applied resources. Specific problems which resulted were inadequate subsystem deliveries (most notably flight software) which affected other elements of the project. The C2.1 ground software delivery schedule commitment was an example of early warning where realism in our commitment led to proper integration into overall project plans.

LESSON LEARNED

It is very important to identify problems which could affect cost and schedule at the earliest possible time so that appropriate action can be taken.

RECOMMENDATION

Tasks should be planned in sufficient detail to allow for the early identification of problems which could affect schedule and cost. These plans should involve the participation of all who are involved, especially the person or persons carrying out the work. Once established, the performance against the established plans should be monitored. Managers need to be less tolerant of schedule slips that have the potential of downstream impact. Workarounds should be identified and implemented early in the process. As changes in scope are accepted, the plans should be updated accordingly. And finally, when problems requiring management attention are identified, the next level of management should be informed as quickly as possible so that appropriate action can be taken.

GALILEO AACS LESSONS LEARNED

DISCIPLINE MANAGEMENT DATE 10/21/86

AUTHOR R. G. HAGA

PROBLEM INTERNAL DELIVERY SCHEDULES

Late hardware and software deliveries to subsystem testing result in compressed and incomplete testing with problems passed on to SAF.

LESSON LEARNED

Time cannot be made up in subsystem testing. The quality of testing will suffer.

RECOMMENDATION

Strictly enforce the meeting of internal delivery schedules to ensure adequate time for proper subsystem integration testing. This means making up time before delivery!

GALILEO AACS LESSONS LEARNED

DISCIPLINE MANAGEMENT DATE 10/27/86

AUTHOR G. K. MAN

PROBLEM MONTHLY MANAGEMENT REVIEWS

Monthly status meetings for the MO&E Office were useful, but Orbiter Office representation was not always present. This impaired the effectiveness of the meetings.

LESSON LEARNED

Status meetings forced regular review of progress and plans, promoted communication with the project and provided a forum for communication within the subsystem. The meetings could have been more effective if hardware issues were addressed at the same time.

In the second development phase after the Challenger disaster, Monthly Management Reviews were expanded to include all AACS activities (including H/W) and representatives of both Project offices were always present.

RECOMMENDATION

Monthly Management Reviews (MMR's) should be mandatory.

Review board should contain both MO&E and Orbiter office personnel.

Note: This problem would automatically be solved if the full AACS (H/W and S/W) were managed under one project office. (See A-004)

GALILEO AACS LESSONS LEARNED

DISCIPLINE MANAGEMENT DATE 11/02/86

AUTHOR G. M. BURDICK

PROBLEM FLOW-DOWN OF INFORMATION

When people working on a project are uninformed about long and short term status and plans, rumors may lead to anxiety about what is really going on. This is especially true when disruptions occur (e.g., launch delays) or when planned transitions occur (e.g., development phase to operations phase).

LESSON LEARNED

It is very important to communicate status and plans to all working levels within a project or task. By doing so, the importance of everyone's work is better understood by all.

RECOMMENDATION

Free and open communications between management and all working levels within a project or task should be encouraged. When disruptions occur, management should assess the impact as soon as possible and communicate the results widely, preferably through section or group meetings, with the section manager or supervisor presiding. When planned transitions are to be made, those affected should be informed as soon as practical.

GALILEO AACS LESSONS LEARNED

DISCIPLINE MANAGEMENT DATE 10/29/86

AUTHOR W. J. GOULET

PROBLEM SCHEDULE STANDARDIZATION

All of the AACS subsystem components did a pretty good job of scheduling and exposing their activities to both the technical and section management. However, each component did their own thing with respect to format and level of detail. This made the integration of activities and milestones very difficult at times.

LESSON LEARNED

The lack of standardized schedules among the AACS subsystem components made the job of planning and reporting laborious, and sometimes inconsistent and in error. Also, the use of PC-based planning tools, as they evolved, was not integrated into the system well enough.

RECOMMENDATION

The Section should provide guidelines for the development of schedules. These guidelines should take advantage of the schedule architecture already in place at JPL, i.e., Level 1 (overview/summary) through Level 6 (detailed implementation). The frequency of update and status reporting should also be standardized. Additionally, the Section should encourage the use of automated project planning tools.

GALILEO AACS LESSONS LEARNED

DISCIPLINE MANAGEMENT DATE 10/30/86

AUTHOR D. G. CARPENTER

PROBLEM PERSONNEL PROBLEMS

Uncooperative or unproductive personnel can do significant harm to a project. Personality differences/conflicts decrease team effectiveness.

LESSON LEARNED

Personnel who are uncooperative and/or unproductive not only affect the job that they are directly working on, but disrupt the team effort necessary to successfully complete the larger task. Significant management effort can be expended trying to "solve" the problem by patching the organization or by making minor personnel reassignments when more drastic measures are really called for. Even if the problem individual is the star player, it is better to reassign that person to another job than to try to patch the organization. NO ONE IS INDISPENSABLE! This type of problem does not solve itself, it festers and gets worse. Letting problems go could cause the loss of good team players.

RECOMMENDATION

Once a problem is identified to be caused by a worker's uncooperative or unproductive actions, move quickly and positively to take corrective action. The longer you wait to take action, the more damage is done to the team. Do not let the fact that the problem individual is the key player delay the corrective action. Problems don't go away, they only get worse.

Only use people who can interface well to be the interface personnel. Peer reviews and design walkthroughs are effective ways to bring someone with a big ego down to a more humble level.

GALILEO AACS LESSONS LEARNED

DISCIPLINE MANAGEMENT DATE 10/28/86

AUTHOR G. K. MAN

PROBLEM LESSONS LEARNED MATERIAL

Some of the lessons learned on the project have already been forgotten because we waited too long to gather this information!

LESSON LEARNED

We should not have waited so long to collect lessons learned material. Some of the material that could have been collected during the early phase of the project could have been used to change practices during the latter half of the project.

RECOMMENDATION

Lessons learned material should be collected periodically, say on a yearly basis. Towards the end, the material can be reviewed and added to as necessary. This will be an easier process if it is done routinely.

GALILEO AACS LESSONS LEARNED

SECTION B - SUBSYSTEM ENGINEERING

TABLE OF CONTENTS

NUMBER	TITLE
B-001	DEFINITION OF THE SUBSYSTEM ENGINEERING TASK
B-002	SUBSYSTEM ELEMENTS INTERFACE
B-003	SUBSYSTEM ENGINEERING STAFF
B-004	LACK OF TECHNICAL/ADMINISTRATIVE SUPPORT
B-005	SPACECRAFT DESIGN TEAM REPRESENTATIVE
B-006	INFORMATION FLOW FROM THE SPACECRAFT DESIGN TEAM
B-007	SUBSYSTEM DESIGN TEAM MEETINGS
B-008	MEETINGS
B-009	REVIEW OF ECR'S
B-010	INTERNAL SUBSYSTEM HARDWARE INTERFACES
B-011	TOP DOWN DESIGN
B-012	EVOLUTION OF CONFIGURATION CONTROL
B-013	DESIGN WORKBOOK

GALILEO AACS LESSONS LEARNED

DISCIPLINE SUBSYSTEM ENGINEERING DATE 10/30/86

AUTHOR W. O. KEKSZ

PROBLEM DEFINITION OF THE SUBSYSTEM ENGINEERING TASK

The subsystem engineering tasks and relationship with the component engineering and system engineering tasks were never well defined for GLL AACS.

LESSON LEARNED

The role of subsystem engineering, and that of the S/S Cog E, must be well defined. There must be a clear statement of the responsibilities and authority, as appropriate. This must be made clear across the subsystem to allow a good working relationship with both component and system engineers. System engineers must not be allowed to bypass the subsystem engineers and communicate directly with the component Cog E's.

RECOMMENDATION

An institutional definition of the role of subsystem engineering should be drawn up, and specific provisions defined by the section or project for each project. This needs to be worked with the tasks defined for component and system engineers to ensure that no areas of responsibility are unstaffed or being worked independently. Efficient communication must be maintained, using the subsystem engineer as the interface.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SUBSYSTEM DESIGN DATE 10/28/86

AUTHOR G. K. MAN

PROBLEM SUBSYSTEM ELEMENTS INTERFACE

Frequently it was not clear who was the point of contact for a particular work package or activity.

LESSON LEARNED

Lack of organizational clarity leads to confusion of roles and responsibilities.

RECOMMENDATION

An organization chart is needed for all interface personnel (hardware and software). Points of contact should be established formally so people are more responsive. This organization chart should be updated whenever there are personnel changes, and this chart should be distributed to all team members.

The subsystem engineer should serve as the interface point, as a mediator and to help finalize design.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SUBSYSTEM DESIGN DATE 10/30/86

AUTHOR D. G. CARPENTER

PROBLEM SUBSYSTEM ENGINEERING STAFF

Subsystem engineering is the glue that ties the subsystem together. They define and interpret requirements, develop interfaces, and act as the technical interface between the subsystem and the system design. Without adequate staffing from the beginning of a project, numerous tasks and checks and balances will not be performed that will result in mistakes and problems throughout the system and subsystem designs during the course of the project.

LESSON LEARNED

A strong subsystem engineer must be selected with the authority to develop the subsystem design. The subsystem engineer must be backed up with an adequate team of engineers to handle the numerous and diverse tasks assigned to him. In determining the proper number of engineers to be assigned to S/S engineering, take into consideration the project system organization and the number of design teams that will have to be supported as well as the type and volume of subsystem and system documentation that must be generated.

RECOMMENDATION

Develop a detailed list of tasks that are handled by subsystem engineering, including design tasks, documentation, team support, design review support, and control of ECR's, Waiver's, PFR's, etc. Develop this in the form of an all inclusive check list that can be used to estimate the staff needed to properly handle the appropriate project task.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SUBSYSTEM ENGINEERING DATE 10/30/86

AUTHOR W. O. KEKSZ

PROBLEM LACK OF TECHNICAL/ADMINISTRATIVE SUPPORT

For most of the GLL life cycle there was no attempt to provide administrative support for scheduling, documentation, and action item tracking.

LESSON LEARNED

An inordinate amount of time was spent by engineers (i.e., the S/S Cog E) running ECR's, etc. around to get signatures. Administrative help needs to come on line to keep track of ECR's, etc. and run them through the system. Help is also needed to work documentation (e.g., design workbook), and to track schedule items and action item responses.

RECOMMENDATION

Assign sufficient administrative help to the subsystem engineering team to perform the above tasks. Academic part-time students work out well in this role.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SUBSYSTEM DESIGN DATE 10/30/86

AUTHOR G. K. MAN

PROBLEM SPACECRAFT DESIGN TEAM REPRESENTATIVE

We did not have proper representation at the spacecraft design team all the time. If that position is staffed with someone with a specialized background then only that area will be adequately represented.

LESSON LEARNED

The spacecraft design team should be supported by a person who has a very good overview of the system. This person should have a good technical background, be able to react in real time and be a good liaison.

RECOMMENDATION

Put the subsystem engineer or his delegate in this spot. This person has to be a good communicator but he/she does not have to know everything. Remember, this person has all the cognizant engineers backing him. At times he should bring cognizant engineers with him to the design team to address special issues.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SUBSYSTEM DESIGN DATE 10/30/86

AUTHOR G. K. MAN

PROBLEM INFORMATION FLOW FROM THE SPACECRAFT DESIGN TEAM

Information from the spacecraft design team meetings was not filtered down to the troops all the time. It is very bad for the team morale if they do not know what is going on.

LESSON LEARNED

A lot of times, designers were not aware of issues and constraints discussed in higher level design team meetings. As a result, the designs had to be changed after they were done. Furthermore, productivity can be dampened if the engineers do not feel that they are part of the team. We cannot only rely on documents (FRs, project policies, flight rules and constraints) for communication. They usually come rather late and they are not widely circulated. Verbal communication followed by a memo is the most effective way.

RECOMMENDATION

The design team rep should take notes at the design team meeting. Relevant information should always be passed on to the AACS component cognizant engineer via the AACS S/S engineer.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SUBSYSTEM DESIGN DATE 11/02/86

AUTHOR G. M. BURDICK

PROBLEM SUBSYSTEM DESIGN TEAM MEETINGS

Subsystem design coordination was hampered by the lack of subsystem design team meetings throughout the design activity.

LESSON LEARNED

A subsystem forum for design coordination and change control is essential throughout the design activity. This is especially true while the spacecraft design team is still working.

RECOMMENDATION

The importance of subsystem design team meetings should be reinforced by mandatory attendance by all subsystem cognizant engineers. These meetings should be continued on a frequency and schedule consistent with both spacecraft and internal subsystem needs. The meeting should have an agenda and be held within tight time constraints; normally the meeting should last no longer than an hour. Design issues should be worked in detail outside the meeting with those involved. Even after the spacecraft design team is disbanded, the subsystem design team meeting should continue as a forum for change control (ECR's, SCR's, etc.) on an as needed basis.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SUBSYSTEM DESIGN DATE 10/30/86

AUTHOR G. K. MAN

PROBLEM MEETINGS

Some of the subsystem meetings were too long and tied up too many people.

LESSON LEARNED

In general, real time design should not be done in meetings. Detailed design issues should be worked off-line, especially issues concerning only part of the group. For example, the fault protection meetings were impossible. They were too drawn out so people stopped attending.

RECOMMENDATION

Plan and control all meetings -- have an agenda, publish it and follow it.

Address "special" items off-line.

Avoid doing real time design in meetings. Meetings are great for reviewing a design, not doing one.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SUBSYSTEM DESIGN DATE 10/31/86

AUTHOR E. C. LITTY

PROBLEM REVIEW OF ECR's

Engineering Change Requests did not receive adequate review at subsystem or system level.

At the subsystem level, on several occasions ECR's were originated by subsystem cognizant engineers (not the subsystem design group), signed off by supervisors and the division representative and forwarded to the system office without coordination by the subsystem design group. This resulted in inquiries by systems about ECR's which the subsystem design group had never seen, and could have resulted in unforeseen impacts on other subsystem groups.

At the system level, ECR's were originated by systems on other subsystems which could potentially impact AACS, but were not forwarded to AACS for review or were reviewed by a system engineer with a subsystem cognizant engineer not in the AACS subsystem design group.

LESSON LEARNED

There exist gaps in the ECR system which occasionally allow ECR's to be issued without adequate review.

RECOMMENDATION

At the subsystem level, the Technical Manager should require the subsystem design cognizant engineer's signature on the ECR impact form prior to approval.

At the system level there should be a procedure to ensure that all ECR review is handled through the AACS subsystem design group.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SUBSYSTEM DESIGN DATE 10/31/86

AUTHOR E. C. LITTY

PROBLEM INTERNAL SUBSYSTEM HARDWARE INTERFACES

There was inadequate definition of internal subsystem hardware interfaces. In addition, contractors were often unaware of system/subsystem requirements on issues such as connector assignments or interface nomenclature.

LESSON LEARNED

Internal AACS hardware interfaces were not adequately defined and/or controlled. All pertinent information (such as shielding, grounding and fusing requirements) was not passed on to the contractor by the H/W Cog E.

RECOMMENDATION

Circuit Data Sheets should be used to define all hardware interfaces within the AACS subsystem (rather than only those interfaces external to the subsystem or passing through system cabling).

Information on the Circuit Data Sheets should include signal definition and timing information to a level of detail sufficient to affect control. See IOM 343-86-1594, "GLL What Did You Learn Re: Interfaces," A.L. Pitts, 10/86.

The subsystem design group should be adequately staffed to create the Circuit Data Sheets and explain them to other subsystem personnel who may not be familiar with this format. This will avoid creating some other (easier to understand) document, which would then have to be separately controlled.

The H/W Cog E's should work with the subsystem engineers to define/document the interfaces via Circuit Data Sheets, then they should pass this information on to the contractors.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SUBSYSTEM ENGINEERING DATE 10/30/86

AUTHOR W. O. KEKSZ

PROBLEM TOP DOWN DESIGN

GLL, as is the case with most systems, was actually specified with a bottom-up approach: they started with component specification (e.g., processor selection), probably emphasizing "off-the-shelf" H/W, and then went on to define H/W interfaces. Similarly, RAM was sized before any honest attempt was made to design the FSW. Then came fault protection design, system S/W and last of all, mission planning.

LESSON LEARNED

The traditional approach doesn't work. Early mission planning (encounter scenario walkthrus, etc.) would have identified many of the system and S/S FSW issues before they became problems, and possibly H/W issues (e.g., radiation environment) as well. Doing FMEA before FP design would have simplified the latter task, and may have produced a better design.

RECOMMENDATION

Define the mission requirements at as detailed a level as possible early in the project life and update them periodically. Use these as top level requirements, making sure they get exposure at all levels of the project. Besides identifying issues before they become problems, this would probably also instill more of a team spirit, allowing team members to see how their piece works with the entire project.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SUBSYSTEM DESIGN DATE 11/02/86

AUTHOR G. M. BURDICK

PROBLEM EVOLUTION OF CONFIGURATION CONTROL

Configuration control on Galileo was in some cases exercised at too high a level at a point much too early in the design development.

LESSON LEARNED

Some degree of configuration control is important at all phases of design development. Galileo experience has shown that the degree of configuration control should tighten as the development progresses.

RECOMMENDATION

Future projects should explicitly plan how configuration control will be applied. The plan should allow for graduated changes in rigidity starting with subsystem Cog E autonomy to project-level control. Changes should be keyed to measurable events such as internal and external deliveries.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SUBSYSTEM ENGINEERING DATE 10/30/86

AUTHOR W. O. KEKSZ

PROBLEM DESIGN WORKBOOK

There must be a central depository of documentation for the subsystem design operation. While the official project documents are necessary, they do not describe the rationale for the designs that must be rethought when something goes wrong, or when a sequence comes up that had not been considered before. The resulting file needs maintenance and publicity to work.

LESSON LEARNED

Memos, presentations, personal notes, etc., that are not included in the project documentation need to be maintained in a central file.

RECOMMENDATION

The Galileo Design Workbook, as implemented, is a good starting point. It includes the documents that affect AACS design and operation, along with a data base to find the documents. Items are filed by date only, and key words are assigned (by the S/S Cog E). The data base looks for items by authors or key words, the latter of which are searched for in both the title and added keywords. A problem, however, is that the workbook has never been adequately supported. More emphasis needs to be placed on including the S/S Cog E on all distributions. In addition, subsystem documents such as the Control Analysis Book should be developed to serve as references.

GALILEO AACS LESSONS LEARNED

SECTION C - HARDWARE DESIGN & FABRICATION

TABLE OF CONTENTS

NUMBER	TITLE
C-001	OFF-THE-SHELF HARDWARE
C-002	INHERITED HARDWARE MODIFICATION
C-003	IMPORTANCE OF BREADBOARD/ENGINEERING MODELS
C-004	IMPLEMENTATION OF CHANGES FROM PROTO TO FLIGHT HARDWARE
C-005	HARDWARE PERFORMANCE DATA
C-006	USE OF CONTRACTOR CONTROLS
C-007	PWB DESIGN
C-008	INTERWIRING DESIGN
C-009	USE OF FAB DOCUMENTATION
C-010	USE OF JPL PACKAGING DESIGN
C-011	POLYMERIC COMPATIBILITY
C-012	REWORKABILITY
C-013	IN-HOUSE REWORK CAPABILITY
C-014	BOARD LEVEL, MODULE LEVEL, AND SUBASSEMBLY LEVEL TESTING
C-015	VIBRATION TEST
C-016	VIBRATION TEST SCHEDULING
C-017	LIFE TEST REQUIREMENTS
C-018	ADEQUATE HARDWARE QUANTITIES
C-019	STRUCTURED HARDWARE DEVELOPMENT PROCESS
C-020	SUCCESSFUL STRUCTURED H/W BUILD - SPIN DETECTOR
C-021	CONFIGURATION CONTROL OF PWB'S
C-022	LATE INITIATION OF ENVIRONMENTAL TEST TASKS
C-023	CO-LOCATION OF HARDWARE TEAM MEMBERS

GALILEO AACS LESSONS LEARNED

DISCIPLINE HARDWARE DESIGN & FABRICATION DATE 10/14/86

AUTHOR D. G. CARPENTER

PROBLEM OFF-THE-SHELF HARDWARE

During the initial planning of Galileo, assumptions were made as to the availability of "off-the-shelf" hardware. Implementation plans and cost estimates were developed around these assumptions which later proved to be completely wrong. Examples were the star scanner, SBA (a Hughes BAPTA was assumed) and the ATAC processor.

LESSON LEARNED

THERE IS NO SUCH THING AS "OFF THE SHELF" HARDWARE!
Once the lid is opened to look at the design, the JPL support areas such as QA&R, packaging design, and environmental requirements, as well as the cognizant engineer, will determine that changes are needed to meet JPL requirements. This typically leads to significant redesign which increases cost, schedule and risk.

RECOMMENDATION

Investigate more completely the hardware that is to be proposed. Assume the need for redesign (particularly upgraded parts) in the implementation plan. Do not allow the Project to push the subsystem into assumptions that are known to be totally unrealistic.

GALILEO AACS LESSONS LEARNED

DISCIPLINE HARDWARE DESIGN & FABRICATION DATE 10/14/86

AUTHOR R. O. ALLEN

PROBLEM INHERITED HARDWARE MODIFICATION

Modifying old inherited hardware was not cost effective. Modifications became excessive.

LESSON LEARNED

Considerable damage occurred to printed circuit boards when they were removed from old chassis. Repairs were time consuming.

Modifications to boards and subassemblies were complicated and costly.

Chassis modifications were required to protect radiation sensitive parts. The use of a radiation hardened part and a circuit board change would have been better.

Radiation analysis became a major issue.

RECOMMENDATION

Review past performance when making a decision on using inherited hardware. Try to understand radiation requirements for the mission before the design is complete.

GALILEO AACS LESSONS LEARNED

DISCIPLINE HARDWARE DESIGN & FABRICATION DATE 10/03/86

AUTHOR R. E. WILLIAMSON

PROBLEM IMPORTANCE OF BREADBOARD/ENGINEERING MODELS

Breadboards and Engineering Models were eliminated for the Galileo Star Scanner to "save costs." The electronics designs were inherited with supposedly "minor" changes. These changes in fact were significant, namely:

(1) numerous parts substitutions for GLL power and radiation requirements, and

(2) re-layout of the boards to meet the JPL DM 509306.

LESSON LEARNED

Elimination of the Breadboards and Engineering Model resulted in very large cost increases, not savings. Knowledge of the problems was delayed by waiting for the flight builds. The costs and schedule delays were much much larger to do troubleshooting and rework on flight hardware then it would have been with engineering hardware.

RECOMMENDATION

Breadboards should be built for new designs, even if the changes are supposedly "minor," particularly if a new class of parts is being used.

Engineering models should be used for any significant design change or for new designs. The JPL PWB packaging approach in the DM is a significant change for those contractors who have not worked with this before.

GALILEO AACS LESSONS LEARNED

DISCIPLINE HARDWARE DESIGN & FABRICATION DATE 10/17/86

AUTHOR D. E. BELL

PROBLEM

IMPLEMENTATION OF CHANGES FROM PROTO TO FLIGHT HARDWARE

During the planning of the AACS Electronics design, fab and test at Martin there was insufficient time allowed in the schedule for problems, errors, etc. found in the PPM hardware to be factored into the flight design. Consequently the flight hardware has many "on the board" corrections that could have been avoided.

LESSON LEARNED

Allow sufficient time between the Prototype/PPM phase and the flight fab start to incorporate changes as a result of the Prototype/PPM fab and test results.

RECOMMENDATION

During the planning phase at JPL for a contract or an "in-house" build, allow sufficient time in the schedule between the Prototype/PPM phase and the flight phase to incorporate necessary changes to the flight design, fabrication instructions and/or test details.

GALILEO AACS LESSONS LEARNED

DISCIPLINE HARDWARE DESIGN & FABRICATION DATE 10/17/86

AUTHOR R. L. SHRAKE

PROBLEM HARDWARE PERFORMANCE DATA

Hardware performance data was not available for the control analysts until late in the program after design, fab and test of flight hardware was completed.

LESSON LEARNED

Hardware performance data can be obtained by fabrication and test of a breadboard/brassboard version of the flight hardware using non-flight parts and materials. Data from the breadboard/brassboard version can be used for a preliminary model of the hardware and can be available long before the flight hardware testing is complete. Fabrication, assembly and test of the breadboard/brassboard hardware also serves as a valuable learning experience for the fabrication, assembly and test of the flight hardware.

RECOMMENDATION

Implementation planning for design and fabrication of flight hardware mechanisms should include fabrication, assembly and test of prototype/ engineering model hardware to be completed as soon as possible after the program start. Early completion of this hardware is generally more important than similarity to flight. Interaction with the control analyst should take place to ensure that appropriate performance data is collected during testing.

GALILEO AACS LESSONS LEARNED

DISCIPLINE HARDWARE DESIGN & FABRICATION DATE 10/16/86

AUTHOR P. P. PERIUS / R. F. RATHCKE

PROBLEM USE OF CONTRACTOR CONTROLS

JPL imposes hardware design and fabrication controls on contractors without sufficient flexibility to evaluate the contractor's system, methods and procedures. The contractor is forced to use systems he is unfamiliar with which produces less reliable hardware and increases program costs.

LESSON LEARNED

Directing the contractor to use JPL manufacturing documentation and controls without having first evaluated the contractor's in-place system caused increased costs, extended deliveries and produced less reliable hardware.

RECOMMENDATION

Recommend that RFP contain the request for detailed information of the design, procurement and manufacturing systems in-place or planned to be in place for the JPL build and that the project evaluate its sufficiency since they direct the imposition of JPL controls. Then use the contractor's system with JPL exceptions imposed.

GALILEO AACS LESSONS LEARNED

DISCIPLINE HARDWARE DESIGN & FABRICATION DATE 10/16/86

AUTHOR P. P. PERIUS / R. F. RATHCKE

PROBLEM PWB DESIGN

The same printed wiring board (PWB) design weaknesses and hardware failures occur on each successive JPL program. Hardware reliability is always compromised by this failure to spend money and time wisely up front.

LESSON LEARNED

The design and purchase of PWBs without imposition of all possible controls or without conformation testing has produced costly failures and reduced product reliability.

RECOMMENDATION

It is recommended that all possible controls of design and conformation be implemented in the purchase order and that technical interchange be face-to-face on a weekly basis. Some of the controls are:

1. Board design review
2. Bed-of-nails testing
3. Division 35 support
4. Low packaging density
5. Few multi layers
6. All layout requirements in P.O.
7. Lot control coupons with timely review
8. Confirm all circuits are or are not there -- use schematic
9. Supply parts footprint to mfg.
10. Define all technical requirements in P.O. (circuit cross section etc.)
11. Total lot control of the run.

GALILEO AACS LESSONS LEARNED

DISCIPLINE HARDWARE DESIGN & FABRICATION DATE 10/09/86

AUTHOR F. W. OSBORN

PROBLEM INTERWIRING DESIGN

Wire damage and breakage occurred during normal assembly processes.

LESSON LEARNED

Wherever wires can be subjected to flexing during normal assembly, test, and rework processes it is virtually impossible to preclude damage by requirements for "careful handling." GLL SAS and SBA motor leads and encoder electronics interconnect wires were damaged in spite of repeated warnings of probable damage and cautions to avoid flexing in high stress areas.

RECOMMENDATION

Wiring space, wire strain relief, and wire handling provisions must be a critical element in the design and design review process. Wire joint provision should be mandatory where leads exit from subassemblies such as motors. Connections to electronics boards should not be made via direct wires except as a last resort. Final flight wires should be attached as late in the assembly process as possible followed by immediate and positive strain relief for the wire joint.

GALILEO AACS LESSONS LEARNED

DISCIPLINE HARDWARE DESIGN & FABRICATION DATE 10/16/86

AUTHOR P. P. PERIUS / R. F. RATHCKE

PROBLEM USE OF FAB DOCUMENTATION

Division 35 packaging design manuals and supporting process, procedure and material specs are incorporated into most flight type contracts and are used for flight hardware design and fabrication at JPL. These documents contain errors and inconsistencies and require modernization. This action requires Division 35 support which has not occurred. Constant Division 35 support to assist JPL and contractors in using and incorporating these requirements is needed. Only minimal support has occurred.

LESSON LEARNED

The use of inaccurate, outdated JPL controlling documentation to direct contractor manufacturing activities and the limited availability of responsible Division 35 personnel to implement the documents produces poor hardware at increased costs.

RECOMMENDATION

Recommend that the chosen staff from Division 35 be block funded from the project to support all controlling documentation upkeep and coordination for in-house and contractor manufacturing activities.

GALILEO AACS LESSONS LEARNED

DISCIPLINE HARDWARE DESIGN & FABRICATION DATE 10/15/86

AUTHOR D. G. CARPENTER

PROBLEM USE OF JPL PACKAGING DESIGN

JPL's designs, methods, and processes used in electronics packaging design are different than those used by industry. In some cases this is fully justified by the nature of the JPL mission. However, many times significant cost savings could be realized if more use were made of the contractor's designs, methods, and processes. Out-of-date and error prone JPL documents significantly compound the problem, especially when working with contractors.

LESSON LEARNED

The JPL electronic packaging approach is costly and not well understood by outside contractors.
The JPL materials and processes documentation is not current and has numerous errors.

RECOMMENDATION

Division 35 must be funded to revise and upgrade all materials and processes documentation. Consider using contractor process documentation when reasonable to do so. Obtain early review by the JPL specialist of the processes but hold to as little change as possible to take full advantage of the contractor's way of doing things. Be as specific as possible in the contract RFQ and the contract implementation as to the requirements to be levied on the contractor for packaging design.

GALILEO AACS LESSONS LEARNED

DISCIPLINE HARDWARE DESIGN & FABRICATION DATE 10/16/86

AUTHOR R. L. SHRAKE

PROBLEM POLYMERIC COMPATIBILITY

Polymerics materials and processes were specified for use in locations and situations which were inappropriate to the material or process.

Materials and processes were often selected on the basis of traditional use for similar applications with too little regard for the specific situation.

LESSON LEARNED

Polymerics materials and processes selection should consider compatibility of materials and of local conditions where applied.

RECOMMENDATION

Thoroughly evaluate each application of polymerics before selection of material or process.

GALILEO AACS LESSONS LEARNED

DISCIPLINE HARDWARE DESIGN & FABRICATION DATE 10/17/86

AUTHOR D. E. BELL

PROBLEM REWORKABILITY

Martin used procedures and materials in the fabrication of the AACS Electronics that did not allow for ease of rework or retrofit of the hardware. This was in defiance of DM509306. Their response was "we build it right the first time and then it doesn't have to be reworked."

LESSON LEARNED

All hardware being fabricated must be capable of being reworked and/or retrofitted. The Galileo program demonstrates that even if hardware is built correctly the first time, mission and program changes can occur which bring about hardware design changes.

RECOMMENDATION

Insist during the design phase that reworkability of the hardware is a firm requirement. Utilize materials and adhesives that are suitable to this requirement.

Pursue having the DM509306 updated in those areas (if any) where hardware reworkability has not been taken into account.

GALILEO AACS LESSONS LEARNED

DISCIPLINE HARDWARE DESIGN & FABRICATION DATE 10/13/86

AUTHOR R. H. STANTON

PROBLEM IN-HOUSE REWORK CAPABILITY

The contractor was unable to complete GLL Star Scanners within schedule and cost constraints. The change to ceramic PMTs would have significantly delayed critical subsystem testing and greatly stretched-out the contract.

LESSON LEARNED

Existing in-house manufacturing and rework capability provided ability to recover from serious contractor setbacks, incorporate required hardware upgrades after delivery from contractor and make repairs following component failure.

RECOMMENDATION

Maintain in-house ability to design, manufacture and repair flight hardware as an important component of JPL's planetary program.

GALILEO AACS LESSONS LEARNED

DISCIPLINE HARDWARE DESIGN & FABRICATION DATE 10/17/86

AUTHOR D. E. BELL

PROBLEM

BOARD LEVEL, MODULE LEVEL, AND SUBASSEMBLY LEVEL TESTING

During the contract negotiations with Martin for the design, fab and test of the AACS Electronics, in order to save some costs it was decided to eliminate board and module level testing of the assembly. Martin objected strongly and insisted on at least having module level testing of the power supply, MEM/DMA and the PDE. Later during assembly level testing, a lot of time and costs were expended in troubleshooting which could have been avoided if board and module level testing had been performed.

LESSON LEARNED

Conduct board level tests, module level tests and subassembly tests whenever practical before assembly testing.

RECOMMENDATION

Resist all efforts by Project, etc. to eliminate lower tier testing of assemblies in order to cut costs during proposal assessment and contract negotiations.

For in-house build and test operations, insist on lower tier testing as part of the work package agreement with the project.

GALILEO AACS LESSONS LEARNED

DISCIPLINE HARDWARE DESIGN & FABRICATION DATE 10/09/86

AUTHOR F. W. OSBORN

PROBLEM VIBRATION TEST

Sinusoidal vibration testing can subject the test article to excessive and unjustified stress levels.

LESSON LEARNED

The JPL sine vibration test requirements are often the primary factor in determining the design stress levels for mechanical structures. It is difficult, and costly, to attempt to predict the response of a complex non-linear structure as it passes through its many resonant modes. The designer is therefore forced to the use of excess mass and/or the use of expensive special materials in order to provide an adequate margin for stress levels which will never be experienced during actual flight conditions.

RECOMMENDATION

The JPL sinusoidal test requirements should be reviewed and compared with aerospace industry practice to determine if they are really necessary to prove design integrity and fabrication quality and to determine if they represent a realistic simulation of flight conditions.

GALILEO AACS LESSONS LEARNED

DISCIPLINE HARDWARE DESIGN & FABRICATION DATE 10/14/86

AUTHOR R. O. ALLEN

PROBLEM VIBRATION TEST SCHEDULING

Vibration testing should have taken less time. It was nearly impossible to perform more than two axes of vibration in a single day due to the lacksadasical attitudes of the test personnel.

LESSON LEARNED

Allow extra time and money to perform vibration tests.

RECOMMENDATION

Enforce the coffee breaks and lunch times to those specified by the JPL SPI.

GALILEO AACS LESSONS LEARNED

DISCIPLINE HARDWARE DESIGN & FABRICATION DATE 10/14/86

AUTHOR C. A. MARCHETTO

PROBLEM LIFE TEST REQUIREMENTS

What are the project requirements for life testing hardware? When should these tests commence for their results to be meaningful to the project they support?

LESSON LEARNED

In the case of the SBA slip ring testing the following was recognized:

- Tests did not start early enough in the project development cycle
- Original life test facility was inadequate in representing realistic flight environment
- The costs to design, build and maintain the instrumentation over the test period was grossly underestimated.

RECOMMENDATION

Life testing objectives, performance criteria and environment should be defined by subsystem PDR.

Tests should accommodate both engineering (prototype) and flight hardware if required for tiger team efforts.

Detailed schedule and costs are required with provisions for test reviews and revisions.

GALILEO AACS LESSONS LEARNED

DISCIPLINE HARDWARE DESIGN & FABRICATION DATE 10/30/86

AUTHOR D. G. CARPENTER

PROBLEM ADEQUATE HARDWARE QUANTITIES

At the beginning of a project, or whenever the Project is attempting to reduce the cost estimate, the elimination of spare hardware is always proposed and almost always accomplished. In the long run, this does not save money, but in reality costs money because of the inability to provide hardware for parallel activities that invariably occur because of schedule difficulties.

LESSON LEARNED

Spare hardware is an absolute necessity for a major flight project, and should not be deleted as a so-called cost cutting measure because it will actually result in costing money in the long run.

RECOMMENDATION

Document the use of spare hardware in past projects to use as a negotiating tool with project management to save the spare hardware.

GALILEO AACS LESSONS LEARNED

DISCIPLINE HARDWARE DESIGN & FABRICATION DATE 1/19/90

AUTHOR R. E. WILLIAMSON

PROBLEM STRUCTURED HARDWARE DEVELOPMENT PROCESS

None. Unit delivered in less than a year, under cost, and with no P/FR's on hardware.

LESSON LEARNED

A structured approach to developing and verifying the design was used including:

a. Clearly establishing requirements and interfaces.
b. Breadboarding the circuits.
c. Performing reliability and mechanical analyses.
d. Holding peer reviews with other electronic design engineers prior to submitting the design to PWB fab. (These reviews were in addition to the PDR and CDR.)
e. Building an engineering model with non-flight parts.

RECOMMENDATION

Use the above structured process to minimize problems with the flight hardware.

GALILEO AACS LESSONS LEARNED

DISCIPLINE HARDWARE DESIGN & FABRICATION DATE 1/16/90

AUTHOR R. O. ALLEN

PROBLEM SUCCESSFUL STRUCTURED H/W BUILD - SPIN DETECTOR

None. All previously learned successful techniques were applied to the design, management, fabrication, testing and delivery of the Spin Detector (less than one year from start to delivery).

LESSON LEARNED

The following is a list of items considered important in the success of the Spin Detector effort. The items are listed in the order of importance.

1) The key personnel had experience with previous flight projects.
2) An Engineering Model was authorized to verify form, fit and function before the flight boards were assembled.
3) New circuits were breadboarded before designing the printed circuit boards.
4) The printed circuit boards were double-sided.
5) The electronic components for the design were selected from the available parts in the JPL flight stores.

RECOMMENDATION

Follow this example.

GALILEO AACS LESSONS LEARNED

DISCIPLINE HARDWARE DESIGN & FABRICATION DATE 1/16/90

AUTHOR R. SCHLIESMANN

PROBLEM CONFIGURATION CONTROL OF PWB'S

An obsolete sheet of printed wiring board artwork was used to fabricate one of the multilayer flight boards for the PDE ANNEX. This resulted in an anxious diagnostic effort and a high priority repair.

LESSON LEARNED

An active/alert configuration control system is critical to efficient fabrication.

RECOMMENDATION

Place the current revision letter on each sheet of the PWB artwork as part of the process of updating the plot. This should be done automatically, complete with a date-time group placed on the sheet, each time the artwork is plotted.

The courier, preferably the technical support representative (TSR), should be chartered/empowered to see that the change is effected at all levels of manufacture. The list includes vendor contract, configuration control, engineering, fabrication, QA, and also the cognizant JPL QA representative.

GALILEO AACS LESSONS LEARNED

DISCIPLINE HARDWARE DESIGN & FABRICATION DATE 1/16/90

AUTHOR R. SCHLIESMANN

PROBLEM LATE INITIATION OF ENVIRONMENTAL TEST TASKS

Failure to initiate environmental test tasks on day one of the PDE Annex one-year hardware development cycle resulted in late test plans/procedures. Two specific undesirable outcomes were: increased fabrication cost of test cables and marginal (hurried) reviews of test procedures.

LESSON LEARNED

Avoid the human tendency to put off tasks which have no readily visible near-term product.

RECOMMENDATION

Staff early and well, preferably as soon as the task can be defined.

GALILEO AACS LESSONS LEARNED

DISCIPLINE HARDWARE DESIGN & FABRICATION DATE 1/16/90

AUTHOR R. SCHLIESMANN

PROBLEM CO-LOCATION OF HARDWARE TEAM MEMBERS

The detail design and fabrication of some support equipment was not co-located with the PDE Annex engineering team. Communication was less effective than it might otherwise have been. Symptoms were slow progress toward required due dates and some wasteful rework.

LESSON LEARNED

Co-location is particularly advantageous for tightly scheduled tasks. Communication is facilitated; coordination and teaming in general is enhanced.

Corollary Lesson: A temporary disruption resulting from relocation (and sometimes restaffing) can be a good investment.

RECOMMENDATION

Co-locate engineering team members. Caution: co-location is not a substitute for vigilence.

GALILEO AACS LESSONS LEARNED

SECTION D - CONTRACTING

TABLE OF CONTENTS

NUMBER	TITLE
D-001	CONTRACT REQUIREMENTS DEFINITION
D-002	RFP PREPARATION
D-003	CONTRACT CHANGE PROCEDURES
D-004	PERFORMANCE MEASUREMENT SYSTEM (PMS)
D-005	AWARD FEE CRITERIA
D-006	SUBCONTRACTOR REQUIREMENT PASS-DOWNS
D-007	RESIDENT ENGINEERING AND QA REPS
D-008	DIVISION 35 TSR SUPPORT
D-009	GFP PARTS LABELING

GALILEO AACS LESSONS LEARNED

DISCIPLINE CONTRACTING DATE 10/24/86

AUTHOR D. G. CARPENTER

PROBLEM

CONTRACT REQUIREMENTS DEFINITION

Contracts for S/S elements are often initiated before the S/C system and AACS subsystem designs are mature enough to prevent major design perturbation at the contractors. This translates into major cost and schedule impacts.

LESSON LEARNED

The start of most projects at JPL has all areas running in all directions, designing all elements of the system in parallel. We design by iteration instead of top down. This is the most expensive approach possible, and guarantees many changes throughout the development phase of all elements of the subsystem. The major lesson learned here is that it is imperative to develop requirements from the top down, early in the program and minimize change through the course of the project. This will allow you to know what it is you are contracting for. It is often better to delay contract preparation until the requirements are firm.

RECOMMENDATION

Spend the time required to develop sound system and subsystem designs before contracting for subsystem elements.
Develop complete and detailed RFP's for each contracted element defining exactly what is wanted. The more thought and detail that goes into this effort, the fewer changes that will be needed during the course of the contract.

GALILEO AACS LESSONS LEARNED

DISCIPLINE CONTRACTING DATE 10/24/86

AUTHOR D. G. CARPENTER / D. E. BELL

PROBLEM

RFP PREPARATION

The Requests for Proposals (RFPs), as usually issued, do not contain a sufficiently detailed statement of work that defines clearly what work is to be done. This leads to difficult and lengthy proposal evaluations and negotiations.

An incomplete RFP results in an incomplete or flawed proposal which translates into changes during the course of the contract which means cost and schedule growth.

LESSON LEARNED

Develop a complete and detailed RFP. Be specific on exactly what is required, particularly the technical requirements and description of the element to be procured. Determine carefully what Contract Data Requirements Lists (CDRLs) and Data Requirements Descriptions (DRDs) are necessary.

RECOMMENDATION

Use a team approach to developing the RFP. Assign sections of the RFP to different individuals and allow them to work on it full time until completed. Take particular care in developing a very detailed technical statement of work and technical description of the elements including support equipment. Carefully select the applicable documents to be imposed on the contractor. If possible, pull out the applicable sections of the document and write that directly into the SOW, thereby deleting the need to impose the entire document.

Be specific about what requirements and documents are to be imposed on subtier contractors.

Start preparation of the RFP sufficiently early to allow for review and revision and TAKE THE TIME TO DO IT RIGHT!

GALILEO AACS LESSONS LEARNED

DISCIPLINE CONTRACTING DATE 10/27/86

AUTHOR F. W. OSBORN

PROBLEM

CONTRACT CHANGE PROCEDURES

Development contracts often have inadequate or inappropriate change provisions.

LESSON LEARNED

There will be changes that must be made or at least considered during the course of a development contract. Existing change provisions are cumbersome and time consuming, and represent a considerable burden to both the contractor and to JPL. This results in the accumulation of charges for smaller changes and delays in the evaluation of the impact of major changes.

RECOMMENDATION

Major contracts should contain a level-of-effort pool of engineering hours which can be used, with proper authorization, to incorporate small, out-of-scope changes and to conduct studies on the impact of major changes.

GALILEO AACS LESSONS LEARNED

DISCIPLINE CONTRACTING DATE 10/24/86

AUTHOR D. G. CARPENTER

PROBLEM

PERFORMANCE MEASUREMENT SYSTEM (PMS)

Imposing a formal PMS requirement on a hardware development contract that is undergoing numerous changes produces reports of little value.

LESSON LEARNED

PMS doesn't work in a dynamic contractual environment. The PMS uses the baseline cost estimate to calculate earned value and schedule accomplishment. The baseline cost estimate is not revised to incorporate the latest changes often enough to make the baseline valid. Therefore, the PMS works with an invalid baseline, and produces erroneous data. It cannot be used by a contract technical manager to help manage his contract. Updating the baseline for each change would be impractical.

RECOMMENDATION

Do not impose a formal PMS unless the contract is unlikely to undergo significant change. The CTM must use his own accounting system, incorporate changes, comparing that to the contractor's monthly financial reports, evaluating performance against schedules to understand contractor performance. The CTM should prepare a monthly report evaluating contractor performance against adjusted plans. This report should be distributed to Project, Division, Section management and the contract negotiator.

GALILEO AACS LESSONS LEARNED

DISCIPLINE CONTRACTING DATE 10/27/86

AUTHOR F. W. OSBORN

PROBLEM AWARD FEE CRITERIA

Existing award fee criteria don't allow for evaluation of performance against the level of difficulty of the task.

LESSON LEARNED

Evaluators, judging a contractor against the present award fee criteria, will have to award lower fees when the contractor is struggling with difficult design problems. Such award fees can be a strong dis-incentive. When a contractor is dealing with complex, difficult, technical design problems and he realizes that the best efforts of his best people are not going to earn an adequate (in his mind) award fee, he will move those people to tasks with a higher payoff. He will also make facility assignments on a similar basis. The morale of the personnel on the task will suffer. In the end, JPL will suffer with higher costs, delayed deliveries, and poor hardware.

RECOMMENDATION

Award fee criteria should be modified to include a "level of difficulty factor," particularly for evaluation of a contractor design effort.

GALILEO AACS LESSONS LEARNED

DISCIPLINE CONTRACTING DATE 10/16/86

AUTHOR R. F. RATHCKE

PROBLEM

SUBCONTRACTOR REQUIREMENT PASS-DOWNS

When JPL contractors subcontract for complex items, the supporting documentation does not satisfy JPL requirements. This most likely occurs because the JPL contractor does not yet understand the JPL documentation requirements well enough to pass them on to his subcontractor. Significant program and analysis delays result from the problem. The purchase order does not contain the necessary requirements as a pass-down from the JPL contractor.

LESSON LEARNED

When JPL contractors subcontract for complex items, documentation requirements imposed on the subcontractor are inadequate to meet the JPL imposed requirements.

RECOMMENDATION

The major contract should contain a requirement for JPL to review the total subcontract and to have the option to attend the negotiations. This may be unacceptable to the contractor though. Another approach is to negotiate into the main contract the requirement that the contractor pass down to subcontractors any applicable JPL requirements.

GALILEO AACS LESSONS LEARNED

DISCIPLINE CONTRACTING DATE 10/17/86

AUTHOR R. L. SHRAKE

PROBLEM RESIDENT ENGINEERING AND QA REPS

Lack of a readily available communication link between contractor and JPL Technical Manager and JPL discipline specialists caused unnecessary delays in resolving technical problems.

LESSON LEARNED

For major contracts, a full-time resident engineer/technical representative and an experienced resident QA engineer are both necessary for the smooth implementation of development and fabrication contracts.

RECOMMENDATION

Implementation planning for design and fabrication of flight hardware should include resources to support a full time engineer/technical representative in residence at the contractor's facility for the duration of the design, fabrication and test phases of the effort.

GALILEO AACS LESSONS LEARNED

DISCIPLINE CONTRACTING DATE 10/22/86

AUTHOR D. E. BELL

PROBLEM DIVISION 35 TSR SUPPORT

Inadequate support was obtained from Division 35 Technical Support Representative (TSR) on the Martin AACS Electronics contract. Review and approval of the Contractor's material and processes was late and often inadequate.

LESSON LEARNED

Inadequate support from the TSR assigned to the contract results in designs and processes that violate JPL requirements. This condition contributed to the PWB terminal interference problem that occurred at Martin. The reworkability issue would have been lessened by the usage of JPL "approved" adhesives rather than Martin's preferred adhesives.

RECOMMENDATION

Insist that Division 35 provide sufficient and competent contract support from the TSRs. This implies good coordination during the planning stage of the project to assure adequate staffing from Division 35.

GALILEO AACS LESSONS LEARNED

DISCIPLINE CONTRACTING DATE 10/16/86

AUTHOR R. F. RATHCKE

PROBLEM GFP PARTS LABELING

When flight (class S) electronic parts are GFP to contractors by JPL, the part ID info supplied to the contractor prior to delivery of the part is used on all contractor documentation. When the parts are delivered GFP, many are identified differently and the accompanying paper may include different ID's than the part IDs. As a result, all fab activities are stopped for cross reference information to be included on all contractor documentation. It has not been possible to identify the problem before receipt of part at JPL.

LESSON LEARNED

Delivery of (GFP) electronic parts to contractors with IDs different from contractors produces mass confusion and program stoppage.

RECOMMENDATION

Recommend that Section 514 supply a cross-reference list along with the parts when differences in IDs are detected at JPL part receipt.

GALILEO AACS LESSONS LEARNED

SECTION E - ANALYSIS/SOFTWARE

TABLE OF CONTENTS

NUMBER	TITLE
E-001	LIBRARIES FOR PAST PROJECTS
E-002	DESIGN REQUIREMENTS
E-003	REQUIREMENTS DOCUMENTS
E-004	3-170 FUNCTIONAL REQUIREMENTS
E-005	FAULT PROTECTION REQUIREMENTS VS FMEA
E-006	FAULT PROTECTION AFFECTING H/W DESIGN
E-007	LATE ALGORITHM DEVELOPMENT
E-008	END-TO-END COGNIZANCE
E-009	ALGORITHM INTEGRATION
E-010	ADEQUACY OF ALGORITHM DESIGN ENVIRONMENT
E-011	ANALYSIS DOCUMENTATION
E-012	ANALYSIS S/W TOOLS
E-013	COORDINATION OF COMPUTER MODELS
E-014	ARCHIVAL OF COMPUTER MODELS
E-015	INADEQUATE COMPUTING SUPPORT
E-016	FUNSIM STAFFING
E-017	S/W-ANALYSIS INTERFACE
E-018	LATE REQUIREMENTS
E-019	FSW LIFE CYCLE GUIDELINES
E-020	FLIGHT SOFTWARE SRD

GALILEO AACS LESSONS LEARNED

SECTION E - ANALYSIS/SOFTWARE

TABLE OF CONTENTS CONTINUED

NUMBER	TITLE
E-021	SRD USE FOR FLIGHT S/W ACCEPTANCE TESTING
E-022	MEMORY AND CPU TIMING MARGIN MANAGEMENT
E-023	SOFTWARE PROTOTYPING
E-024	LOW LEVEL SOFTWARE TESTING
E-025	FLIGHT S/W ACCEPTANCE TESTING
E-026	EARLY S/S TESTING
E-027	PARTICIPATION IN S/S TESTING
E-028	ITL AUTOMATION
E-029	EARLY OPS PLANNING
E-030	GROUND S/W LIFE CYCLE
E-031	CALIBRATION S/W
E-032	PROBLEM TRACKING
E-033	PFR/FR USAGE AND QA PARTICIPATION
E-034	PROGRESS REPORTS
E-035	SCHEDULE SLIPS

GALILEO AACS LESSONS LEARNED

DISCIPLINE PROJECTS DOCUMENTATION DATE 10/20/86

AUTHOR E. C. WONG

PROBLEM LIBRARIES FOR PAST PROJECTS

There was no centralized library for past projects.

LESSON LEARNED

A library or libraries of significant documents categorized by projects should be maintained. Custodians could be the section's administrative aides or group supervisors. This would greatly help the engineers to capitalize on previous project experience.

RECOMMENDATION

Maintain libraries for past and on-going projects.

GALILEO AACS LESSONS LEARNED

DISCIPLINE ANALYSIS/FSW DEVELOPMENT DATE 10/14/86

AUTHOR R. G. HAGA

PROBLEM

DESIGN REQUIREMENTS

Design requirements were difficult to decipher and understand. This led to confusion about mission objectives and took a lot of time to clarify.

LESSON LEARNED

In the project functional requirements hierarchy, (especially in level 3 documents, i.e. 270, 280, 290), requirements are often too vague or on the other hand, too detailed. Every requirement should be testable.

RECOMMENDATION

It's difficult to separate requirements from implementation. AACS S/S engineering should produce an AACS design requirements and description document (DRDD) derived from FR's. The DRDD would include CMD & TLM dictionaries. The analysis and fsw areas could then design to the DRDD. If possible, design requirements could be highlighted for verification.

GALILEO AACS LESSONS LEARNED

DISCIPLINE REQUIREMENTS DATE 10/09/86

AUTHOR H. K. BOUVIER

PROBLEM REQUIREMENTS DOCUMENTS

Functional Requirements, Software Requirements Documents and Design Documents are not considered to be important project deliverables. Therefore they are poorly organized and written in the first place and are not maintained well.

LESSON LEARNED

From a management standpoint, requirements and design documents are often considered as level-of-effort tasks. Workers often consider them to be red tape. They are not given proper attention.

RECOMMENDATION

Functional Requirements and Design Documents cannot be developed instantaneously and in serial order. Nevertheless, the contents can be defined and scheduled so that everyone knows what to expect and when they will get it. Slips in these schedules should be considered serious. Missing elements should be defined at PDR's and CDR's with the same follow-up as that given to other scheduled items.

GALILEO AACS LESSONS LEARNED

DISCIPLINE ANALYSIS-ERROR ANALYSIS DATE 10/17/86

AUTHOR E. W. KLUMPE

PROBLEM 3-170 FUNCTIONAL REQUIREMENTS

Error sources in 3-170 were not tracked. Representative numbers were inserted in several places but they were not updated when the necessary information became available.

LESSON LEARNED

Representative numbers generated in early analyses are never updated resulting in inaccurate analysis downstream.

RECOMMENDATION

All performance capabilities numbers must be traced to their respective error sources in a data base with all the representative numbers and TBD's flagged.

GALILEO AACS LESSONS LEARNED

DISCIPLINE FSW REQUIREMENTS DATE 10/20/86

AUTHOR S. M. KRASNER

PROBLEM FAULT PROTECTION REQUIREMENTS VS FMEA

System and subsystem level fault protection requirements were never explicitly stated. They were not traceable to the subsystem FMEA.

LESSON LEARNED

Confusion about definitions of failure modes led to lack of priority for coverage, lack of definition of acceptable fault protection responses by mission phase and difficulties in simulating failures and defining acceptable tests. It was also difficult to coordinate responses between AACS and CDS.

RECOMMENDATION

Complete subsystem FMEA early and maintain it. Trace fault protection requirements to it. Assign relative priorities to failure modes.

Define acceptable fault protection responses early, and by mission phase.

Emphasize coordination of fault protection responses between AACS and CDS.

GALILEO AACS LESSONS LEARNED

DISCIPLINE FSW REQUIREMENTS DATE 10/20/86

AUTHOR S. M. KRASNER

PROBLEM FAULT PROTECTION AFFECTING H/W DESIGN

It is difficult to infer sensor or actuator failure indirectly. Coupling device inputs and outputs (e.g. SBA/SAS torques are required to sample encoder data, star scanner outputs are required to sample the high voltage state and also abort star scanner interrupts in progress) makes isolation of the failure even more difficult.

LESSON LEARNED

Do not rely on indirect means to isolate failures. Provide hardware to test directly for failure modes. Do not couple device inputs and outputs, and decouple asynchronous inputs from synchronous ones. Provide actively redundant devices to allow data comparisons.

RECOMMENDATION

Allow fault protection requirements to influence the design of the hardware by doing it early enough to make a difference.

GALILEO AACS LESSONS LEARNED

DISCIPLINE ALGORITHM DESIGN DATE 10/09/86

AUTHOR H. K. BOUVIER

PROBLEM LATE ALGORITHM DEVELOPMENT

Late development and verification of algorithms caused problems for implementation in software.

LESSON LEARNED

On VGR and GLL, there was a serious problem in the timely delivery of algorithms for the AACS and for their verification.

RECOMMENDATION

At the start of a project, the analysis and algorithm designers should develop and publish an Algorithm Design and Verification Plan (ADVP). It could be modelled after the more traditional Software Management and Development Plan. It should at least contain the following:

- Scope
- Approach
- Facilities and tools (hardware and software)
- Products (all products, including the algorithm verification tools, design work-book memos, delivery memos to S/W engineering, test methodology, test procedure, etc.)
- Staffing plans
- Schedules

GALILEO AACS LESSONS LEARNED

DISCIPLINE G&C COGNIZANCE & INTEGRATION DATE 10/20/86

AUTHOR E. C. WONG

PROBLEM END-TO-END COGNIZANCE

There was a lack of long term cognizance and overall technical responsibility in the various AACS analysis areas.

LESSON LEARNED

A cognizant engineer should be assigned to oversee an assigned subject area in AACS areas, e.g., Attitude Determination, Attitude Control, Scan Pointing Control, Mass Properties, Configuration and Dynamics Modeling.

RECOMMENDATION

The cognizant individual should be responsible for the end-to-end integration of the design, and the coding, testing and documentation of all analysts' work in his assigned area of responsibility.

GALILEO AACS LESSONS LEARNED

DISCIPLINE ANALYSIS & SOFTWARE-INTEGRATION DATE 10/15/86

AUTHOR G. K. MAN

PROBLEM ALGORITHM INTEGRATION

The integration of algorithms, at the algorithm design level, was not done well.

LESSON LEARNED

The attitude determination algorithms were not well integrated because they were not developed following a top-down design and they were not tested in an integrated manner. As a result, a lot of integration problems were encountered during software integration. Contrary to attitude determination, scan platform control algorithms were developed in a structured fashion and they were tested in an integrated way at the algorithm level. Because of these extra steps, relatively few problems were encountered in the scan control area.

RECOMMENDATION

Develop a top-down design for the algorithms from the inception of the project.
Maintain this design structure throughout the project.
Have the analysts test the algorithms at the unit level as well as in an integrated manner.

GALILEO AACS LESSONS LEARNED

DISCIPLINE ANALYSIS & SOFTWARE-INTEGRATION DATE 10/20/86

AUTHOR W. G. BRECKENRIDGE

PROBLEM ADEQUACY OF ALGORITHM DESIGN ENVIRONMENT

The environment in which algorithm design was done was artificial and didn't show up the problems that were encountered later in the real environment.

LESSON LEARNED

1. The slew profile generation algorithm was very sensitive to computational error, but this wasn't detected until it was moved from the CDC (64 bit) to IBM (32 bit). Word length can be important.
2. Slew path control was tested on CDC using ideal AD & Control (no error), but when tested with simulated dynamics there were significant discontinuities at the end of a slew. Operating environment is not ideal.
3. It was found that a wobble estimate had to be added to the rotor attitude estimator to reduce the estimate noise and attitude bias. Algorithm design/analysis/simulation didn't consider the total environment.
4. Rotangle was added to correct for the effects of stator slewing on rotor clock angle.
5. The "polar region," inserted to avoid a mathematical singularity, had to be enlarged because spacecraft motion compensation interacted with vehicle dynamics to cause nutation instability. Must consider the whole picture.

RECOMMENDATION

Need more complete evaluation of algorithm designs by analysis and/or simulation, including the effect of the real operating environment (misalignments, S/C motion, numerical accuracy, ...). An algorithm that "performs its function" in an ideal environment may not perform adequately in the real environment.

GALILEO AACS LESSONS LEARNED

DISCIPLINE ANALYSIS DOCUMENTATION DATE 10/15/86

AUTHOR E. C. WONG

PROBLEM ANALYSIS DOCUMENTATION

There was a lack of a coherent plan for documentation.
This resulted in severe waste of time and documentation.

LESSON LEARNED

The control analysis documentation (e.g., a project document entitled CONTROL ANALYSIS BOOK) should be well defined in the early phase of the project, preferably in the first quarter after the project approval date. Guidelines and format specifications should be published early.

RECOMMENDATION

Analysts shall be fully responsible for any future update of their sections in the documentation. Each analyst should also provide a floppy disk to the documentation cognizant engineer where their documents are being stored. This effort will make the job of later updates much easier and more efficient.

GALILEO AACS LESSONS LEARNED

DISCIPLINE ANALYSIS AND SOFTWARE-TOOL DATE 10/15/86

AUTHOR G. K. MAN

PROBLEM ANALYSIS S/W TOOLS

Analysis tools were inadequate during algorithm design and development.

LESSON LEARNED

Analysis tools for digital control design and flexbody simulation were insufficient. Consequently several ad-hoc software tools had to be developed during algorithm development. This was very time consuming and inefficient.

RECOMMENDATION

A plan that addresses short term and long term software tool needs should be developed. Model reduction, flexbody simulations, integrated control system analysis, and structured software development tools are most needed.

GALILEO AACS LESSONS LEARNED

DISCIPLINE ANALYSIS AND S/W TEST-COORDINATION DATE 10/20/86

AUTHOR G. K. MAN

PROBLEM COORDINATION OF COMPUTER MODELS

Simulation model development was not coordinated within the subsystem.

LESSON LEARNED

Spacecraft and hardware models were developed by different people for different test environments, but the models served the same purpose. Time and effort can be minimized by coordinating the test bed development work. One model should be sufficient for one hardware assembly.

RECOMMENDATION

Establish test bed requirements early and coordinate model development work. Do not duplicate effort within the subsystem.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SIMULATION MODELS/SUBROUTINES DATE 10/20/86

AUTHOR E. C. WONG

PROBLEM ARCHIVAL OF COMPUTER MODELS

Simulation models and subroutines developed from past projects were not readily available for use on Galileo.

LESSON LEARNED

The lack of a coherent and organized plan to archive developed simulation programs and subroutines used in past projects meant that Galileo could not benefit from past experiences.

RECOMMENDATION

Provide standard guidelines to enable analysts to deliver their simulation models, programs and subroutines to a general library in a user friendly format. This could be part of the computer aided design tasks. These programs can be very useful, and can minimize the time spent to 'reinvent the wheel.' Examples are Galileo scan platform/notch filter control loop, gyro integrator, star scanner model, sun sensor model, Earth magnetic field model, and Section 347's large antenna model, etc.

GALILEO AACS LESSONS LEARNED

DISCIPLINE ANALYSIS AND SOFTWARE-TOOL DATE 10/15/86

AUTHOR G. K. MAN

PROBLEM INADEQUATE COMPUTING SUPPORT

Inadequate computing capability hampered algorithm design and development.

LESSON LEARNED

Algorithm development was held back because the analysts did not have adequate computing capabilities. They were forced to use the outdated UNIVAC until difficulties were encountered. A lot of time and money was wasted when the switch from the UNIVAC to the CDC was made (justifications, paper work, conversion time, learning curve...).

RECOMMENDATION

Every analyst must have a personal computer. Computer hardware requirements must be established at the start of the project and an adequate computer budget should be provided. The Laboratory should not impose inferior institutional computer facilities on computer users.

GALILEO AACS LESSONS LEARNED

DISCIPLINE ANALYSIS-TESTING DATE 10/15/86

AUTHOR L. NEEDELS

PROBLEM FUNSIM STAFFING

The Functional Simulator program FUNSIM was inadequately staffed.

LESSON LEARNED

Only one person knew enough about FUNSIM to be able to answer questions and assist in debugging. This support was inadequate considering the extensive use of FUNSIM.

RECOMMENDATION

Provide a backup person for FUNSIM.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SOFTWARE & ALGORITHM INTERFACE DATE 10/09/86

AUTHOR H. K. BOUVIER

PROBLEM S/W-ANALYSIS INTERFACE

The interface between algorithm developers and software designers was difficult on GLL and took too much time to resolve.

LESSON LEARNED

Experience in industry and at JPL shows that the handover of algorithms to software engineering contains the seeds of contention. It has been blamed upon personality problems (a shiboleth good for any occasion) and upon the wrong organizational structure. The usual approach to resolution is to change the organization. Nevertheless, the major culprit is the lack of recognition that a real technical job is involved, not just a transfer of paper.

RECOMMENDATION

A position should be created, staffed by a person with working familiarity in both algorithm development and software engineering. The duties for that person should include:

1. Negotiate a standard format for algorithm deliveries.
2. For each function, track the algorithm development and S/W Requirements Definition, resolving any inconsistencies. Bring the workers together as needed.
3. Hold a walk-through for each delivery of an algorithm.
4. Maintain a document with a description of each algorithm, and how they all work together as an AACS.

GALILEO AACS LESSONS LEARNED

DISCIPLINE REQUIREMENTS DATE 10/08/86

AUTHOR H. K. BOUVIER

PROBLEM LATE REQUIREMENTS

Late requirements delivery to S/W engineering impacted timely development of the software.

LESSON LEARNED

On GLL and most other projects, the definition and delivery of requirements to S/W engineering has had a most serious impact on the timely and cost-effective development of the S/W.

RECOMMENDATION

Early-on in a project, functions that must be performed at the system or subsystem level are identified. From those functions, requirements will flow to S/W. These functions, and the requirements definition from them, must be put on project schedules as definite deliverables with delivery dates that are taken as seriously as hardware delivery dates. Late requirements deliveries should be a truly serious concern for managers.

GALILEO AACS LESSONS LEARNED

DISCIPLINE FSW DEVELOPMENT DATE 10/20/86

AUTHOR S. M. KRASNER

PROBLEM

FSW LIFE CYCLE GUIDELINES

The AACS flight software was developed in a very ad hoc fashion, with the guiding life cycle ignored and late development of a guiding architecture. There was no top-down design process.

LESSON LEARNED

Since several phases were often in process in parallel (a little requirements definition, some architecture definition and general design, a lot of coding, some unit testing), it was very difficult to identify the current requirements baseline, do test planning, identify the current capabilities of the software or to determine whether progress was being made.

RECOMMENDATION

Follow software life cycle guidelines. Use a requirements analysis and design methodology. Recognize the fact that the software development process is iterative and allow for changes due to feedback.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SOFTWARE REQUIREMENTS DATE 10/08/86

AUTHOR H. K. BOUVIER

PROBLEM

FLIGHT SOFTWARE SRD

Requirements were poorly designed (specifically for S/W).

LESSON LEARNED

On GLL, AACS Flight S/W Requirements were written at a low-level of detail that represented design. Small design changes caused innumerable changes in the Software Requirements Document (SRD) which was under configuration control - a terrible (and costly) paper mill.

RECOMMENDATION

Develop a definition of "requirements" for each new project or task, because each one is a little different and lessons learned can be incorporated continuously. An old definition can be used if appropriate. Obtain guidance from recent advances in Computer Science. Grapple with the boundaries of the definition, and answer the questions: what is a too-high-level requirement with insufficient detail for design? What is a too-low-level requirement that goes into design detail?

I recommend a rule-of-thumb definition. If you must go to another group to get needed information, that's a requirement. If you can develop needed information yourself, that's a design.

GALILEO AACS LESSONS LEARNED

DISCIPLINE FLIGHT S/W ACCEPTANCE TESTING DATE 10/20/86

AUTHOR E. C. WONG

PROBLEM SRD USE FOR FLIGHT S/W ACCEPTANCE TESTING

The flight S/W SRD is a necessary tool for Flight S/W Acceptance Testing yet it is rarely developed with that in mind.

LESSON LEARNED

The SRD is the basis for all S/W test requirements. It should be stored in D-Base, and its format should be such that it can be readily turned into a test requirement hierarchy. SRD maintenance should be a clearly defined task.

RECOMMENDATION

The smallest block of statements in the SRD should represent the primitive (lowest level) of the entire requirement hierarchy. The size of each black box for testing should be determined at the SRD level. Reference to high level documents should be cited for traceability.

GALILEO AACS LESSONS LEARNED

DISCIPLINE FSW DEVELOPMENT DATE 10/20/86

AUTHOR S. M. KRASNER

PROBLEM MEMORY AND CPU TIMING MARGIN MANAGEMENT

Problems were encountered early with both memory and timing margins. A project plan existed for memory margin but it was not realistic since it did not accurately estimate the initial memory needed and did not take into account the increasing number of system and subsystem requirements that impacted the flight software. There was no timing margin plan and no effort at prototyping to identify what the critical path was and how long it was taking.

LESSON LEARNED

Realistic memory and timing margin plans should have been used to guide scrubbing efforts. Early scrubbing is counter productive. Too much time was spent trying to scrub implementation before the real needs were understood. Also, resources were wasted scrubbing functions that were not major memory users or on the critical timing path.

RECOMMENDATION

Develop realistic memory and timing margin plans based on rapid prototypes where possible. Ensure that a good understanding of the requirements is in hand before proceeding to the design phase.

Define and prioritize requirements. Develop a list of requirements to scrub if memory and timing margin estimates indicate a need to do so. This list should reflect both the importance of the requirement to mission success and the ease in which it can be scrubbed.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SOFTWARE DEVELOPMENT DATE 10/09/86

AUTHOR H. K. BOUVIER

PROBLEM SOFTWARE PROTOTYPING

On GLL, predictions of important technical and management parameters for flight software were egregiously underestimated, even by the best people involved.

LESSON LEARNED

On GLL, it was impossible to maintain a meaningful schedule for S/W management. Estimates for memory usage and timing (throughput) were disastrously low. The overrun in schedule and cost, to recover, was unacceptable in any terms.

RECOMMENDATION

Fast prototyping of flight S/W, based upon the knowledge of the AACS functional design, should be done prior to the analysis and development of software requirements. A far better understanding of the total task will emerge -- the key to success is the speed of the prototyping, not the quality of the code.

GALILEO AACS LESSONS LEARNED

DISCIPLINE ANALYSIS & SOFTWARE-TESTING DATE 10/20/86

AUTHOR G. K. MAN

PROBLEM LOW LEVEL SOFTWARE TESTING

Low level flight software testing was inadequate.

LESSON LEARNED

Module level S/W testing should not be performed in the ITL. S/W testing should be done step-by-step by unit, by function, and by subsystem on a software test bed before it goes into the ITL. Inadequate low level testing makes acceptance testing impossible because it is not clear how deep the software needs to be tested.

RECOMMENDATION

Test flight S/W on a S/W test bed before delivery to the ITL.

GALILEO AACS LESSONS LEARNED

DISCIPLINE FLIGHT S/W ACCEPTANCE TESTING DATE 10/20/86

AUTHOR E. C. WONG

PROBLEM FLIGHT S/W ACCEPTANCE TESTING

There was no coherent or long term plan for Flight S/W Acceptance Testing.

LESSON LEARNED

Products (e.g., SRD, D-charts, Validation Test Matrix), and the corresponding configuration control method, should be defined for each stage of the S/W life cycle.

Design/code walkthroughs and peer reviews proved to be very beneficial and should be part of any Acceptance Test Plan.

RECOMMENDATION

The whole acceptance testing life cycle should be defined at the start of the subsystem design phase. The entire schedule of AT effort, including planning, requirements verification, traceability and S/W validation should be spread throughout the whole S/W life cycle.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SUBSYSTEM TESTING DATE 10/09/86

AUTHOR H. K. BOUVIER

PROBLEM EARLY S/S TESTING

Subsystem testing uncovers problems at a time that is too late for a timely and cost-effective resolution.

LESSON LEARNED

On every project, subsystem testing has been about the most important task performed. There are two major reasons: 1) things always work together differently than anyone can imagine, and 2) everyone who must deliver something has a real moment-of-truth -- they must deliver something and it must work, or the whole world knows about it. These include hardware, software and plans to do the test.

RECOMMENDATION

Each project should start subsystem testing as soon as possible, using breadboards, prototypes, simulators, or anything that can be pressed into service. Ideally, prototype (or simplified) algorithms and prototype software should be used. Testing should start as soon as any two or three items can somehow be integrated. The experience gained should be incorporated into the remaining cycle of subsystem development.

GALILEO AACS LESSONS LEARNED

DISCIPLINE ANALYSIS/FSW DEVELOPMENT DATE 10/14/86

AUTHOR R. G. HAGA

PROBLEM

PARTICIPATION IN S/S TESTING

The FSW and analysis and hardware design efforts were affected by those areas not knowing how the subsystem works as a whole.

LESSON LEARNED

Subsystem testing: it should be planned that analytical and software engineers participate in subsystem testing (and component engineers also). The learning experience is of great value to the various design efforts.

RECOMMENDATION

Analysis engineers have supported S/S testing. FSW and H/W engineers are required to support S/S testing. S/S testing Dog and Pony shows have described lab capabilities. It's important to train and keep an independent test team utilizing other areas as consultants/reviewers. A formal training program for the test team (per Viking) should be established.

GALILEO AACS LESSONS LEARNED

DISCIPLINE ANALYSIS-TESTING DATE 10/20/86

AUTHOR G. A. MACALA

PROBLEM ITL AUTOMATION

Integration lab testing was very tedious and difficult due to lack of automation capabilities and lack of computer power.

LESSON LEARNED

Many tasks, such as data collection, display, etc., were very tedious and could have been automated via PC's or additional computing power (the PDP). Much needless effort was spent on "optimizing" code for speed, especially the dynamics, due to lack of computing power on the PDP.

RECOMMENDATION

Avoid low level languages, e.g., assembly. The ITS assembly code made the ITS digital control loops 10 times more difficult to debug.
Buy 2 or 3 times the computing power you "think" you need.
It can always be used.

GALILEO AACS LESSONS LEARNED

DISCIPLINE ANALYSIS/FSW DEVELOPMENT DATE 10/14/86

AUTHOR R. G. HAGA/S. M. KRASNER

PROBLEM

EARLY OPS PLANNING

Intended use of subsystem in flight was not coordinated with flight S/W requirements. This led to incorrect and/or missing requirements which had to be updated or incorporated later.

LESSON LEARNED

Mission scenario walkthroughs should start early in the project; operations planning should start early. It is an iterative process and needs early preliminary work.

RECOMMENDATION

Phase operations planning with S/C development i.e., start it early. Integrate OPS planning with S/S engineering and document the OPS pseudo requirements in the suggested S/S DRDD. Early work includes CMD & TLM needs for mission scenarios, calibration, GSW interaction and rule definition.

GALILEO AACS LESSONS LEARNED

DISCIPLINE GROUND SOFTWARE DEVELOPMENT DATE 10/16/86

AUTHOR M. H. JAHANSHAHI

PROBLEM GROUND S/W LIFE CYCLE

Coding and testing the software without clear requirements and a design that has been thought through results in a lot of work having to be redone later.

LESSON LEARNED

Starting implementation phase without close scrutiny of the requirements and design should not be done, even at the expense of missing milestones and over-running the budget.

RECOMMENDATION

No matter how reliable the cognizant engineer, do not deviate from the normal proven S/W development methodology. Follow the life cycle; don't enter the next phase before fully validating the completion of the present phase.

GALILEO AACS LESSONS LEARNED

DISCIPLINE ANALYSIS & S/W INTERFACE DATE 10/20/86

AUTHOR W. G. BRECKENRIDGE/M. H. JAHANSHAHI

PROBLEM CALIBRATION S/W

Calibration interface between ground software and flight software was inadequate.

LESSON LEARNED

Calibration interface between ground software and flight software is very important, but since ground software algorithms were developed early and since personnel were physically separated, the interface fell apart.

Simulation programs, flight software error compensation, and ground calibration programs should be based on a single mathematical error model. They should have same/similar names, signs, etc.

RECOMMENDATION

Define the mathematical model for error sources early enough to be the basis for all S/W development.

Make sure everyone knows about it and uses it.

Define and document the ground-flight S/W interface early in the development process to ensure compatibility and consistency.

Hold inter group review of designs to catch inconsistencies.

GALILEO AACS LESSONS LEARNED

DISCIPLINE ANALYSIS AND SOFTWARE DATE 10/20/86

AUTHOR G. K. MAN

PROBLEM PROBLEM TRACKING

There were too many problem lists and they were impossible to track.

LESSON LEARNED

Problem lists were generated from H/W, ITL, S/W and analysis. They were difficult to monitor and control.

RECOMMENDATION

Do not stop individual groups from generating problem lists. As a matter of fact, this should be encouraged. However, there should be a central location where an official subsystem problem list is kept. Use clerical support for this task, don't waste engineers' time. Automation is definitely needed.

GALILEO AACS LESSONS LEARNED

DISCIPLINE FSW DEVELOPMENT DATE 10/20/86

AUTHOR S. M. KRASNER

PROBLEM PFR/FR USAGE AND QA PARTICIPATION

The PFR system is not set up to address flight software problems, especially with phased software deliveries. This system is redundant with the software failure report system.

LESSON LEARNED

Problems with the PFR system were exacerbated by the participation of QA personnel in verifying and validating the PFR closures. There had been no QA participation in earlier phases of the software development process, so QA personnel did not have any background in the software requirements and related design decisions.

RECOMMENDATION

Clearly define how to track and close flight software problems. Perhaps this is a special case of the PFR or FR system. Specify how the problems are to be closed (acceptable analysis, tests). Educate QA personnel along the way.

GALILEO AACS LESSONS LEARNED

DISCIPLINE PROGRAM REPORTING DATE 10/09/86

AUTHOR H. K. BOUVIER

PROBLEM PROGRESS REPORTS

Consistent and timely progress reports are hard to get and typical management efforts to fix that problem cause much busy-work.

LESSON LEARNED

Managers want and need regular progress reports, which is a burden upon the implementers. A typical management response is to impose schedules with shorter and shorter milestones (i.e., higher resolution). Another approach is to have more meetings. There are times in the life of a project when all that is necessary, but usually not all the time. A way is needed to increase the efficiency of the information dissemination process.

RECOMMENDATION

Weekly, each group supervisor (or leader) should write a very short, informal progress report for the section manager and technical manager. The emphasis is on short and informal. It should be handwritten, with simple, declarative sentences.

If there are no concerns and everything is on schedule, the report should say that, with no need for clarification.

Management can follow-up as they choose. At the end of a month, or for whatever period of time, anyone can combine and condense those into the more formal type reports that are usually required by projects.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SCHEDULES DATE 10/09/86

AUTHOR H. K. BOUDIER

PROBLEM SCHEDULE SLIPS

Milestone slips for algorithm development and flight S/W were allowed to happen without proper management concern or response.

LESSON LEARNED

On GLL early in the program, serious schedule slips became evident. The best engineers were making the schedules and doing the work. Being the best available engineers, they were embarrassed by slips and hastened to assure management that the slips could be managed by them. They thought that if they worked a little harder, they could make it up. Project and section management were willing accomplices in that self-deception. Any ideas to add staff were summarily rejected as being more time consuming than the actual required work.

RECOMMENDATION

Management should have a tolerance on schedule slips, so that when it is exceeded, they do something. They must realize that probably the schedule is too optimistic to begin with. Something new must happen, such as adding resources, reducing requirements, etc. Doing nothing is fatal.

GALILEO AACS LESSONS LEARNED

SECTION F - SUBSYSTEM TESTING

TABLE OF CONTENTS

NUMBER	TITLE
F-001	EVOLUTION OF TEST REQUIREMENTS
F-002	TEST REQUIREMENTS VS. SUCCESS CRITERIA
F-003	TEST ENGINEERING
F-004	TEST BED REQUIREMENTS
F-005	COMMON SPACECRAFT MODELS
F-006	MULTIPLE TEST BEDS
F-007	SE FLEXIBILITY
F-008	TEST BED CERTIFICATION
F-009	TEST TEAM STAFFING
F-010	SOFTWARE SUPPORT TO SUBSYSTEM TESTING
F-011	ANALYST SUPPORT OF ITL
F-012	HARDWARE SUPPORT TO S/S INTEGRATION TEST
F-013	SYSTEM AND SUBSYSTEM TEST EXPERIENCE
F-014	TEST TEAM ACTS AS AACS REPRESENTATIVES
F-015	EFFECTIVE TRAINING FOR TEST TEAM
F-016	SUBSYSTEM TEST DOCUMENTATION
F-017	REVIEW OF TEST RESULTS
F-018	TIMELY DOCUMENTATION OF TEST ANALYSIS
F-019	SE SOFTWARE DOCUMENTATION
F-020	OPS STAFFING
F-021	CONSOLIDATION OF OPS SUPPORT IN ONE GROUP
F-022	CROSS TRAINING FOR OPS POSITIONS

GALILEO AACS LESSONS LEARNED

DISCIPLINE SUBSYSTEM TESTING DATE 10/17/86

AUTHOR G. M. BURDICK

PROBLEM EVOLUTION OF TEST REQUIREMENTS

Galileo AACS test requirements and plans were inadequately developed and did not account for the inherent evolution of the process.

LESSON LEARNED

Subsystem testing evolves through phases much the same way as the development of hardware and software. The integration and testing program for a complex subsystem has a life cycle starting with hardware functional integration through hardware/software acceptance testing.

RECOMMENDATION

The plans for subsystem testing should incorporate provisions for all phases of the test program. The phases that should be included are: development, functional, performance, and acceptance which correspond to the level of maturity of the design and requirements.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SUBSYSTEM TESTING DATE 10/13/86

AUTHOR G. K. MAN

PROBLEM TEST REQUIREMENTS VS. SUCCESS CRITERIA

Test requirements were inadequate. Ideally, test results should not be used to set all success criteria. However if time is a problem, running the test to understand the results and then set the test requirements might be more time efficient.

LESSON LEARNED

Test requirement definition was inadequate. Success criteria should be derived from test requirements. Also, different success criteria should be used for different phases of test (i.e., assembly, subsystem integration and functional).

RECOMMENDATION

Derive test requirements from FRs.

Do calibration (use tests results to set success criteria) only if test requirements are not available.

Tailor test requirements to test objectives (functional, acceptance, performance).

GALILEO AACS LESSONS LEARNED

DISCIPLINE SUBSYSTEM TESTING -OPS PLANNING DATE 10/10/86

AUTHOR L. L. ANDERSON

PROBLEM TEST ENGINEERING

Problems found with an engineered test
- Test approach was not feasible in targeted testbed
- Targeted testbed was not optimal choice to meet test objectives
- More direct, simpler approach existed
- Portions of test design were based on inaccurate assumptions
- Test fell short of verifying requirements, meeting original objectives
- Requirements were verified by previous/existing test outline
- Actual test required more time and workforce than anticipated

LESSON LEARNED

Valuable information for test engineering includes
- Input from test operator
- Thorough knowledge of available testbeds
- Knowledge of past testing
- Technical review
- A good understanding of software implementation and/or hardware behavior

RECOMMENDATION

In order to gather all of the necessary knowledge and information that makes up a good test, test engineers should review their test outlines and objectives with their peers.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SUBSYSTEM TESTING DATE 10/10/86

AUTHOR G. K. MAN

PROBLEM TEST BED REQUIREMENTS

Test bed requirements were ill defined and poorly tested.

LESSON LEARNED

Subsystem test planning was not well coordinated. Test bed requirements were not allocated to different test beds. As a result, test beds were developed by different teams in different directions.

RECOMMENDATION

There should be a subsystem test plan which addresses test bed requirements for different test beds. This test plan should be reviewed by all concerned parties before test bed development starts. This is an up front planning task and will require adequate initial funding.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SUBSYSTEM TESTING - FLT OPS PREP DATE 10/15/86

AUTHOR D. M. COHEN

PROBLEM COMMON SPACECRAFT MODELS

The dynamic equations of motion and the models for the different testbeds were designed by several different people during different time periods. This resulted in model differences and idiosyncrasies which had to be remembered during testing.

LESSON LEARNED

Resources were diluted by having different people reinventing models and by confusion about which model was used in which test bed.

RECOMMENDATION

Minimize problems by developing a master set of equations of motion and models. Simplified models or equations may be needed by some testbeds but they should be derived from the master set.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SUBSYSTEM TESTING - FLT OPS PREP DATE 10/15/86

AUTHOR D. M. COHEN

PROBLEM MULTIPLE TEST BEDS

Each of the subsystem test beds, ITL, GRASIM, FUNSIM, had unique capabilities. It was not possible to run particular tests on certain test beds.

LESSON LEARNED

More than one test bed is needed to test all the subsystem requirements. The tests must be properly allocated to the appropriate test bed.

RECOMMENDATION

Recognize that more than one test bed will be needed. Identify what each test bed will be used for and design it to do that job.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SUBSYSTEM TESTING - FLT OPS PREP DATE 10/15/86

AUTHOR D. M. COHEN

PROBLEM SE FLEXIBILITY

The SE software was forced to change rate groups from 44 ms to 66 ms to be compatible with a flight software change and to provide SE software timing margin. This left the SE software incompatible with the ROM rate groups.

LESSON LEARNED

To avoid costly changes and redesigns, have the SE flexible enough to adapt to flight changes. Ideally, the SE software should run 5-10 times faster than the flight software.

RECOMMENDATION

Technology and money permitting, design the SE to be powerful enough to be flexible to flight changes.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SUBSYSTEM TESTING DATE 10/14/86

AUTHOR G. K. MAN

PROBLEM TEST BED CERTIFICATION

Test bed certification was inadequate.

LESSON LEARNED

A "formal" checkout of the test bed was not performed until very late. As a result, a lot of time was wasted trying to determine if a problem was caused by S/W, H/W or test bed (e.g. ITS, star simulator and sun model). The test environment needs to be understood (through calibration) before it can be used for testing. Fixing it along the way is very costly. This will require a lot of up front funding which may be difficult to obtain.

RECOMMENDATION

Establish a test bed checkout plan based on test bed requirements. This plan should address test bed requirements, shoot out tests (with other test beds), and regression tests.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SUBSYSTEM TESTING - FLT OPS PREP DATE 10/09/86

AUTHOR R. G. HAGA

PROBLEM TEST TEAM STAFFING

Flight software support of subsystem testing was inadequate which resulted in lost test time.

In addition, the test area was understaffed as a result of H/W problems and late S/W (esp. fault protection) development.

LESSON LEARNED

Test (and flight ops) areas should be as autonomous as possible to minimize priority conflicts.

The understaffing usually results in excess overtime and/or robbing other areas (which results in their tasks slipping).

RECOMMENDATION

Add a flight software analyst to the test team.

Increase the testing staff to anticipate extra testing (if possible). Utilize APT's for routine tasks. A test staff should include: group leader, FSW analyst, lab lead plus 2 analysts and SAF lead plus 2* analysts.

*Note: Add 2 to SAF team if the MSA must be staffed.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SUBSYSTEM TESTING -OPS PLANNING DATE 10/18/86

AUTHOR L. L. ANDERSON

PROBLEM SOFTWARE SUPPORT TO SUBSYSTEM TESTING

The turnaround time for problem fixes was slow.

Results of software test and software interactions were not understood by test team at times and held up test sometimes.

There was poor communication with the software group during test time.

LESSON LEARNED

Daily communication with software representative to review test status and problems found was very beneficial.

RECOMMENDATION

Attendance by software group representative at daily test reviews should be compulsory. An even better solution would be to have a flight software analyst on the test team.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SUBSYSTEM TESTING DATE 10/13/86

AUTHOR G. K. MAN

PROBLEM ANALYST SUPPORT OF ITL

Analyst support was not used effectively for troubleshooting.
On-call support makes analysis planning work impossible.

LESSON LEARNED

An analyst should not be used to support testing on an on-call basis because it is very disruptive to planned work and it is not as effective as it can be to the lab.

RECOMMENDATION

Roll design engineers off to testing to minimize hiring new personnel.

Augment ITL test team with long term analyst support at a specified level.

Require analyst who supports ITL testing to be trained on the ITL.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SUBSYSTEM TESTING - FLT OPS PREP DATE 10/16/86

AUTHOR D. G. CARPENTER

PROBLEM HARDWARE SUPPORT TO S/S INTEGRATION TEST

Good support to the subsystem integration test activity by the hardware groups is imperative to ensure adequate and timely testing of the AACS. In general this support was provided during Galileo testing. However some areas of support activities could be improved.

LESSON LEARNED

The quality of S/S testing is enhanced by the level of hardware knowledge possessed by the test team. The individual hardware cognizant engineers must be kept aware of test activities, and the test team must be informed of hardware details (changes, hardware test results, idiosyncrasies, etc.) to insure that communication lines between hardware and S/S test are efficient.

RECOMMENDATION

Each hardware cognizant engineer should be required to prepare and present a tutorial for the test team as early in the test team formation as possible. The tutorial should provide a solid general working knowledge of the hardware design, functional performance, operational constraints, and any other information pertinent to test and operation.

Weekly test status meetings should be conducted with mandatory attendance required for all involved hardware cognizant engineers.

During hardware integration, the hardware cognizant engineers should provide full time support to the integration lab and should attend the daily status meetings.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SUBSYSTEM TESTING - FLT OPS PREP DATE 10/15/86

AUTHOR D. M. COHEN

PROBLEM SYSTEM AND SUBSYSTEM TEST EXPERIENCE

All test personnel were not exposed to both system and subsystem test experience.

LESSON LEARNED

Subsystem test gives personnel in-depth training of the subsystem capabilities. System test gives the personnel information on how the subsystem will be used in flight and an opportunity to learn how it interacts with the other subsystems.

RECOMMENDATION

Cross-train test personnel between system and subsystem level testing.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SUBSYSTEM TESTING - FLT OPS PREP DATE 10/09/86

AUTHOR R. G. HAGA

PROBLEM TEST TEAM ACTS AS AACS REPRESENTATIVES

The AACS test team fully represents the AACS during system (SAF/KSC) testing. All AACS contact with system testing is via the AACS test team.

LESSON LEARNED

There are numerous AACS internal and external organizations that can cause problems if tasks are not coordinated. This includes all H/W and S/W deliveries/removals, troubleshooting and miscellaneous tasks such as baffle installations or actuator motion tests.

RECOMMENDATION

Have management emphasize the coordinating role of the AACS test team during system testing to both AACS and SAF personnel.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SUBSYSTEM TESTING -OPS PLANNING DATE 10/16/86

AUTHOR L. L. ANDERSON

PROBLEM EFECTIVE TRAINING FOR TEST TEAM

Test team may not understand subsystem well enough to thoroughly test software/hardware, understand results/interactions, be involved in problem resolution, or design lien tests. This results from inadequate training tools and training time.

LESSON LEARNED

Effective training tools and training time are required, such as:

- Software design walkthroughs, tutorials
- Mission operations training videos
- Sufficient and up-to-date documentation of all aspects of subsystem
- Hardware tutorials, walkthroughs
- Test time training (on test bed)
- Testbed hardware, software walkthroughs, tutorials

RECOMMENDATION

Keep it a high priority to keep test team well educated and develop plans to include appropriate tools.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SUBSYSTEM TESTING - FLT OPS PREP DATE 10/09/86

AUTHOR R. G. HAGA

PROBLEM SUBSYSTEM TEST DOCUMENTATION

Subsystem test documentation was good. This included log books, problem lists, procedures and reports. Critiques followed each test phase. Daily test status meetings and weekly test summary meetings were held.

LESSON LEARNED

The documentation and critiques not only helped communication within the subsystem but they also provided division and project personnel with detailed status. This openness allowed freer internal test planning and operation since project personnel knew they would be kept informed.

RECOMMENDATION

Emphasize and improve upon all aspects of test documentation. Continue holding test status/summary meetings and test critiques. Impress upon critique presenters that absence of project personnel reflects trust and not indifference.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SUBSYSTEM TESTING -OPS PLANNING DATE 10/16/86

AUTHOR L. L. ANDERSON

PROBLEM REVIEW OF TEST RESULTS

When retest or additional testing is inconvenient, untimely, or impossible:
- It was discovered that test did not meet objectives
- A problem was uncovered that wasn't caught by test personnel
- Insufficient data was captured to properly evaluate test results
- Test bed capabilities were insufficient to fulfill all of test objectives

Test analysis/problem investigations were not performed promptly.

LESSON LEARNED

Quick review of results by test personnel with test engineer within same day of test exposed blatant problems early. Retest for additional data was much easier to obtain.

Thorough review of test results immediately after test by test engineer exposed some of the more subtle problems early. Again, retest and additional data captures were much easier to obtain.

RECOMMENDATION

Hold a quick review of results by test personnel with test engineer present, immediately after test (within same day).

Plan a thorough review of results immediately after test by test engineer.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SUBSYSTEM TESTING -OPS PLANNING DATE 10/16/86

AUTHOR L. L. ANDERSON

PROBLEM TIMELY DOCUMENTATION OF TEST ANALYSIS

Results of analysis or research performed are not available and:
- Responsible person is gone
- Same tools no longer exist to perform task over
- Existing notes are too sketchy to remember important facts and details

LESSON LEARNED

The sooner the results of the task are documented, the less likely these problems are to occur.

RECOMMENDATION

Document analysis, research, fact finding immediately.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SUBSYSTEM TESTING - FLT OPS PREP DATE 10/15/86

AUTHOR D. M. COHEN

PROBLEM SE SOFTWARE DOCUMENTATION

Test personnel were not well educated on the SE software.
One person was relied on for information.

LESSON LEARNED

An SE software user's guide was written but it was needed earlier in the testing program so it could have been used and modified.

RECOMMENDATION

Emphasize documentation at the same time as SE software development. Have it reviewed by the test personnel. Use the documentation and update it as necessary.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SUBSYSTEM TESTING - FLT OPS PREP DATE 10/17/86

AUTHOR J. L. CHODAS

PROBLEM OPS STAFFING

Too few people were available to support Ops planning during the development phase (1 in '83, 3 in '84, 5 in '85). AACS Ops team was assembled too late (10 people 5 months before launch, 17 people 2 months before launch). Some AACS personnel did not transfer from development to operations because of misconceptions about what Ops was, what types of jobs were available, etc.

LESSON LEARNED

The Ops planning area needs to be staffed earlier and with more people. This would allow spacecraft operations and strategies to be planned and coordinated with development products such as flight and ground software and with development activities such as analysis and test.

RECOMMENDATION

Have 1/4 of Ops team start 2 years before the development phase ends. Have 1/2 of Ops team start 1 year before the development phase ends. Assemble the complete Ops team at least 9 months before launch if people are transferring from development to operations, even earlier if people are not from AACS. At least 1 year before the development phase ends, interview personnel about posssible Ops positions to let them know what's available.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SUBSYSTEM TESTING - FLT OPS PREP DATE 10/17/86

AUTHOR J. L. CHODAS

PROBLEM CONSOLIDATION OF OPS SUPPORT IN ONE GROUP

If the people supporting mission operations are in different groups, it is difficult to coordinate their activities because of competing priorities from the various groups. There is no one group supervisor who can direct and prioritize the work for everyone in Ops.

LESSON LEARNED

Consolidate all the Ops support (except hardware) in one group. This provides closer interfaces between areas that must work together in Ops. Since everyone is in the same group, tasks can be prioritized and worked more easily. Having everyone in the same group gives more of a team spirit to mission operations.

RECOMMENDATION

Consolidate all the Ops support (except hardware) in one group. This includes flight software, ground software, subsystem engineering, integration and test, dynamics and control, and support to the Orbiter Engineering Team and Sequence Team.

GALILEO AACS LESSONS LEARNED

DISCIPLINE SUBSYSTEM TESTING - FLT OPS PREP DATE 10/17/86

AUTHOR J. L. CHODAS

PROBLEM CROSS TRAINING FOR OPS POSITIONS

If each position in Ops is covered by only one person, a number of problems result:

- the person feels locked into the job because no one else knows how to do it
- there is no backup coverage if the person is absent
- the person does not develop an overall view of the subsystem
- after the person learns the job, there is little opportunity or incentive to learn new things

LESSON LEARNED

Maintaining a prime and alternate for each position has the following advantages:

- backup coverage is available
- it provides at least two viewpoints into the area which can lead to new insights and contributions
- the repetitiveness of the job is avoided if a new person is doing it
- it enables the person to be an expert in many areas which gives him a better subsystem viewpoint
- it provides opportunity and incentive to learn new areas
- the trainee can generate useful documentation as he learns the area

RECOMMENDATION

Maintain a prime and an alternate person for each position. Rotate the primes and alternates periodically.