NAVAL RESEARCH LABORATORY

NAVAL CENTER FOR SPACE TECHNOLOGY

Full-Sky Astrometric Mapping Explorer (FAME)
Flight Software Management Plan (SMP)

NCST-SMP-FM001 **9 November 2000**
Version 0.1

Approved By:_____ Date: _____
Dr. Kenneth J. Johnston, Principal Investigator, USNO

Approved By:_____ Date: _____
Mark S. Johnson, Program Manager, NRL Code 8133

Approved By:_____ Date: _____
Matthew Hayden, Lead Flight Software Engineer, NRL

Approved By:_____ Date: _____
Cognizant NASA Manager

4555 Overlook Avenue, S.W.
Washington, D.C. 20375-5000

RECORD OF CHANGES

REVISION LETTER	DATE	TITLE OR BRIEF DESCRIPTION	ENTERED BY
Version 0.1	9 November 00	Initial Draft Release	R. Spencer

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1.SCOPE

1.1 Identification.

This Software Management Plan (SMP) describes the development of the Flight Software (FSW) Computer Software Configuration Item (CSCI) for the Full-sky Astrometric Mapping Explorer (FAME) project. This document applies to the FAME spacecraft flight software being developed by the Naval Research Laboratory (NRL). It does not apply to the ground software (also being developed by NRL) or to the instrument software being developed by Lockheed Martin Missiles and Space (LMMS).

This software management plan describes the major activities, schedules, resources, and milestones for developing the FAME spacecraft flight software. The areas addressed are:

- Software Engineering Management,
- Software Engineering Development,
- Test Engineering,
- Software Quality Assurance (SQA),
- Software Configuration Management (SCM),
- Software Support, and
- Operations Support.

Updates to this document will be made as needed and will be delivered at project reviews.

1.2 System Overview.

FAME will measure the positions, proper motions, parallaxes, and four-color magnitudes of 40 million stars brighter than 15th visual magnitude during the observational program. The positional accuracy will be the finest yet achieved. The positional, parallax, and proper motion accuracies will be better than 50 µas, 50 µas, and 50 µas/year, respectively for brighter stars.

The FAME Flight Software CSCI consists of all the software that executes onboard the FAME Integrated Spacecraft Controller (FSC). The FSC has two redundant processors, each executing the same version of FAME flight software. The use of multiple processors is primarily to support Fault Detection, Isolation, and Recovery (FDIR) for a processor failure.

1.3 Document Overview.

This plan identifies and describes the management, development, and verification for the FAME Flight Software CSCI. It serves as a source document for the quality control, configuration management (CM), and defect tracking for the development of the flight software. This plan documents the results of the planning process performed by the software development manager for development activities for the FAME flight software. The sections of this document and their specific purposes follow.

- Section 1.0, Scope, provides the identification and overview of this document as well as the system and software to be developed.
- Section 2.0, Applicable Documents, lists the documents referenced in this plan or that are applicable to the software development products or processes.
- Section 3.0, Software Engineering Management, describes the overall project management approach including the project organization, resources, work breakdown structure, schedule, and deliverables.
- Section 4.0, Software Engineering Development, describes the software development approach including architecture, life cycle model, detailed activities, and unit level testing.
- Section 5.0, Test Engineering, describes the software testing approach including software system testing, and Independent Validation and Verification (IV&V).
- Section 6.0, Software Quality Assurance, describes the plan for quality assurance of the software and related documents.
- Section 7.0, Software Configuration Management, describes the plan for configuration control of software and related documents.
- Section 8.0, Software Support, describes the software development environment and data-handling plan.
- Section 9.0, Operations Support, describes the plan for providing operations support of the delivered software, including software uploads.

- Section 10, Notes, provides a list of the acronyms and abbreviations used in this document.
- Appendix A provides the basis for the flight software.
- Appendix B provides the flight software standards in checklist format.
- Appendix C provides MIL-STD-498 DIDs as tailored for FAME

1.4 Relationship to Other Plans.

Table 1-1 lists the FAME deliverable documentation as specified in Order S-13610-Y. Document numbers beginning with "NCST" are project documents developed and controlled by NRL. This document is shown in bold in Table 1-1. This Software Management Plan is consistent with the Project Management Plan (NCST-D-FM003) and the Configuration Management Plan (NCST-D-FM008).

Table 1-1. FAME Document Deliverables List

Document Name	Number
Science Requirements Document	NCST-D-FM001
Mission Requirements Document	NCST-D-FM002
Project Management Plan	NCST-D-FM003
Systems Engineering Management Plan (SEMP)	NCST-D-FM004
Product Assurance Plan	NCST-D-FM005
SR&QA Plan	NCST-D-FM006
Contamination Control Plan	NCST-D-FM007
Configuration Management Plan	NCST-D-FM008
Software Management Plan	**NCST-SMP-FM001**
Safety Documents	
Preliminary Safety Assessment	NCST-D-FM009
System Safety Implementation Plan (SSIP)	NCST-D-FM010
Ground Operations Procedures (30 days before PER)	
Safety Data Package	
Launch Site Data Plan	
Space Segment Documents	
Instrument	
Instrument Design Specification	LMMS document number
Instrument to S/C ICD	NCST-ICD-FM001
Instrument subsystem and component specifications	LMMS document numbers
Spacecraft	
S/C Design Specification	NCST-S-FM001
S/C subsystem and component specifications	NCST-S-FM002 through NCST-S-FM00*n*
System Integration and Test Plan	NCST-TP-FM001
Verification Matrix	
Environmental Test Matrix	
Verification Procedures	
Integration and Test Procedures	
Software Documents	
Software Requirements Document	NCST-SRS-FM001
Launch Segment Documents	
S/C to L/V ICD	NCST-ICD-FM002

Document Name	Number
Ground Segment Description Document	NCST-D-FM016
Space to Ground ICD	NCST-ICD-FM003
Supporting Documents	
Failure Mode and Effects Analysis (FMEA)	NCST-D-FM011
Preliminary EEE Parts List	NCST-D-FM012
Preliminary Materials List	NCST-D-FM013
Orbital Debris Report (CDR +60 days)	NCST-D-FM014
Space Segment Reliability Analysis	NCST-D-FM015
MO&DA Documents	
Flight Operations Plan	
Software User Guides	
Final B/C/D Technical Report (Launch +60 days)	
Final Phase E Technical Report (End of Mission +60 days)	

2.APPLICABLE DOCUMENTS

This section lists documents that either are referenced in this Software Management Plan or provide additional information applicable to the understanding of this document.

2.1 Government Documents.

2.1.1 NRL Documents.

The following documents are FAME project specific:

Number	Document Title
NCST-D-FM001	Science Requirements Document for the Full-sky Astrometric Mapping Explorer (FAME)
NCST-D-FM002	Mission Requirements Document for the Full-sky Astrometric Mapping Explorer (FAME)
NCST-D-FM003	Project Management Plan for the for the Full-sky Astrometric Mapping Explorer (FAME)
NCST-D-FM004	Systems Engineering Management Plan (SEMP) for the Full-sky Astrometric Mapping Explorer (FAME)
NCST-D-FM008	Configuration Management Plan (CMP) for the Full-sky Astrometric Mapping Explorer (FAME)
NCST-ICD-FM001	Instrument to S/C ICD for the Full-sky Astrometric Mapping Explorer (FAME)
NCST-ICD-FM003	Space to Ground ICD for the Full-sky Astrometric Mapping Explorer (FAME)

2.1.2 Military Standards.

The FAME Flight Software CSCI uses the following military standards (MIL-STD) and international standards:

Number	Document Title
MIL-STD-498	Software Development and Documentation, 5 December 1994
MIL-STD-1553B Notice II	1553 Data Bus Specification
ISO/ICE 9899	ANSI C Standard Programming Language

2.2 Non-Government Documents.

2.2.1 Commercial Documents.

The following commercial documents provide additional information on commercial off-the-shelf (COTS) software:

Vendor	Document Title
Wind River Systems	VxWorks Programmer's Guide
Wind River Systems	VxWorks Reference Manual
Wind River System	VxWorks User's Manual
Rational	ClearCase/ClearQuest Administrator's Manual
Rational	ClearCase/ClearQuest User's Manual

3.SOFTWARE ENGINEERING MANAGEMENT

This section of the Software Management Plan (SMP) addresses the software engineering management approach used by the Naval Research Laboratory (NRL) to manage the development of the FAME flight software.

3.1 Project Organization.

The NRL is responsible for development of the spacecraft bus hardware, flight software, and ground data systems for the FAME project. Figure 3-1 depicts the NRL organization for the FAME project as it relates to the software development effort.

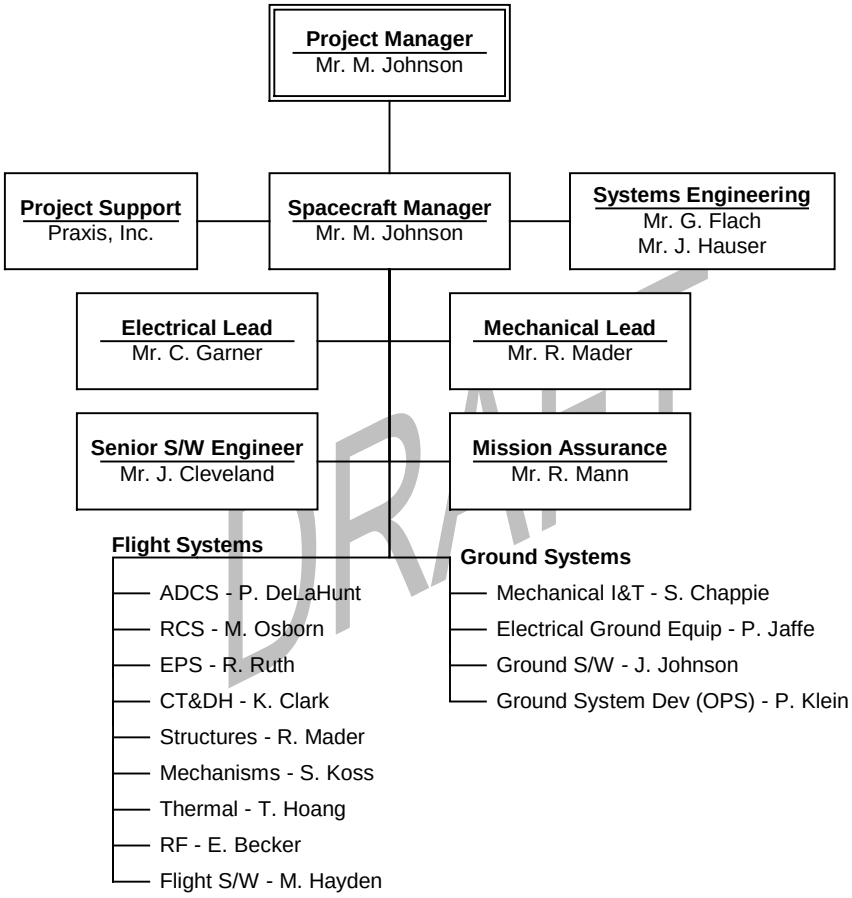

Figure 3-1. FAME Project Organization for Software-related Areas

3.2 Roles and Responsibilities.

The flight software for the FAME project is being developed within the NRL NCST Electrical Systems Segment.

- C. Garner is the Segment Manager responsible for the Electrical Systems Segment.
- Matthew Hayden is the Subsystem Manager responsible for flight software. This encompasses the following software areas:
  - Flight Software (FSW) CSCI
  - Software Only Test Bed (SOTB).

3.2.1 FAME Software Development Organizations.

Figure 3-2 details the structure of the FAME flight software development management team.

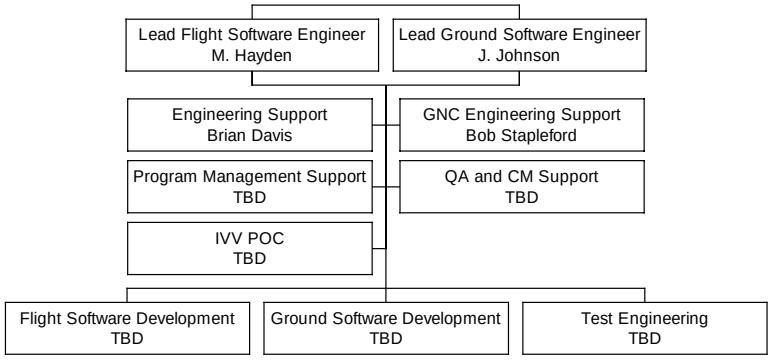

Figure 3-2. Flight Software Organization

In order to gain independence in the software development and testing activities, the FAME software development and testing responsibilities are partitioned into independent groups within the NRL. This provides an independent verification of the FAME flight software. In addition to these separate groups within NRL, a team composed of the National Aeronautics and Space Administration (NASA) Engineering Group (EG) and an independent contractor will perform verification and validation of the flight software. An NRL Engineer will be assigned as the primary Point of Contact to the Independent Validation and Verification (IV&V) Team. Table 3-1 provides a description of the responsibilities of each of these groups.

Table 3-1. Software Groups

Software Group	Responsibility
Flight Software Group	Responsible for the development of the flight software that executes onboard the FAME Integrated Spacecraft Controllers (FSCs).
Ground Software Group	Responsible for the development of the ground software that executes in various test bed configurations as well as the MOC and SOC.
QA and CM Support	This group will also provide Configuration Management (CM) services to the Flight and Ground Software groups. The CM services will perform version and baseline control during development and provide releases for system level testing and flight operations.
Test Group	Responsible for the development of the Software Test Plan (NCST-STP-FM001) and the Software Test Report (NCST-STR-FM001) documents for verification of the FAME flight software. This group will develop the flight software test procedures and descriptions, and perform the flight software system level testing.
IV&V Support	NRL Engineer responsible for providing a single Point of Contact to the IV&V Team and supporting the IV&V of the FAME flight software.
IV&V Team	Responsible for executing the IV&V of the FAME flight software. The IV&V Team is composed of a combination of the EG and an independent subcontractor to NRL. More details of the IV&V are provided in Subsection 5.1.2, Flight Software IV&V Approach.

3.3 Work Breakdown Structure (WBS).

The software functions are partitioned into the various software/systems engineering disciplines as described in Subsection 1.1, Identification. Table 3-2 presents the Work Breakdown Structure (WBS) based on these disciplines, the WBS Products, and the Responsible Individual (RI).

Table 3-2. Work Breakdown Structure

WBS	WBS Name	WBS Products	RI
1.1	Software Management	Software Management Plan (SMP)	M. Hayden
1.2	Systems/Software Engineering	System Requirements Review (SRR) Inputs Flight Software Requirements Specification (SRS) System Preliminary Design Review (PDR) Inputs System Critical Design Review (CDR) Inputs	M. Hayden
1.3	Software Engineering	Flight Software Design Document (SDD) Flight Software Builds Software Version Description Documents (SVDs)	TBD (R.I. - MH)
1.4	Test Engineering	Flight Software Test Plan (STP) Flight Software Test Reports (STRs)	TBD (R.I. - MH)
1.5	Software Support	Configuration Management Software QA Flight Software Releases IV&V Support	TBD (R.I. - MH)

3.4 Master Schedule.

The high-level schedule for the flight software development efforts is maintained at the FAME project level. This schedule is updated at least monthly to reflect in-progress and completion status along with current milestone dates. The high-level schedule tracks timelines for flight software development and flight software documentation. The documentation timelines at this level will only track documents that are called out as deliverables to NASA.

The high-level software development schedule tracks the software development progress using build, system test, and release milestones. Details regarding the most recent project schedule status are available from Mr. Mark Johnson, Project Manager, who may be reached at the NRL FAME Project Office. This detailed schedule will map components and requirements to the build milestones.

Table 3-3 identifies the key functionality to be provided in each of the flight software releases. The ground test software functionality required is driven by the flight software functionality being tested in each of the builds.

Table 3-3. Flight Software Release Descriptions

Release	Description
Build 1.0	TBD (R.I. - MH)
Build 2.0	TBD (R.I. - MH)
Build 3.0	TBD (R.I. - MH)
Build 4.0	TBD (R.I. - MH)
Build X.0	TBD (R.I. - MH)

3.5 Deliverables.

3.5.1 Software.

The Flight Software (FSW) Computer Software Configuration Item (CSCI) is partitioned into a group of software components whose interactions compose the software architecture. The software architecture for the FSW CSCI is formally documented in the FAME Flight Software Design Document, NCST-SDD-FM001.

All FAME Flight Software CSCI source code (this source code set does not include COTS), build procedures and test procedures are delivered. All compiled source code object files and object libraries are delivered. All files needed to load operational code into the FSC are delivered.

3.5.2 Documentation.

The FAME Flight Software CSCI document set addresses the areas of Project Management, System/Software Engineering, Test Engineering, Program Support, and Product Use. Table 3-4 lists the deliverable FAME Flight Software documents. Subsection 3.5.2.1 provides a description of each document.

Table 3-4. FAME Documentation Deliverables

Document Number	Deliverable Product
NCST-SMP-FM001	FAME Software Management Plan
NCST-SRS-FM001	FAME Flight Software Requirements Specification
NCST-ICD-FM008	FAME Flight Software External Interface Control Document
NCST-SDD-FM001	FAME Flight Software Design Document
NCST-STP-FM001	FAME Flight Software Test Plan
NCST-STR-FM001	FAME Flight Software Test Report
NCST-SVD-FM001	FAME Flight Software Version Description Document

3.5.2.1 Deliverable Document Descriptions.

The following describes the deliverable documentation set for the Flight Software CSCI. All documents will remain current as requirements or the design changes. For each version of the CSCI, the documents will be released as a new version. The Software Version Description (SVD) will identify the documents related to the software build.

3.5.2.1.1 FAME Software Management Plan, NCST-SMP-FM001.

This SMP establishes the approach for the development of the FAME Flight Software CSCI. It describes the software organizational framework, software management, deliverables, software development activities, software test and integration activities, flight software IV&V, software quality assurance (SQA), CM, software support, and operations support.

3.5.2.1.2 FAME Flight Software Requirements Specification, NCST-SRS-FM001.

The Flight SRS establishes the functional and performance requirements for the FAME Flight Software CSCI. The Flight SRS will be used as the basis for performing the flight software design, development, and flight software test. It also provides a basis for the IV&V activity.

3.5.2.1.3 FAME Flight Software External Interface Control Document, NCST-ICD-FM008.

The Flight Software External Interface Control Document (ICD) establishes the interfaces between the Flight Software and external systems and devices, including:

- Uplink/Downlink Module (UDM)
- Spacecraft I/O Module (SCIOM)
- Forward Link (FWL) (including command formats)
- Return Link (RTL) (including telemetry formats and contents)
- Solid State Data Recorder (SSDR)
- Fame Instrument
- Attitude/Reaction Controller (ARC)
- Star Camera
- Inertial Measurement Unit (IMU)
- 1553 Bus (including protocols for FSC-Instrument and any other devices on the 1553 bus)
- FSC

The command and data formats and rates for the uplink and downlink are defined and the hardware/software interfaces are described for all FSC interfaces. This document provides the basis for development of the uplink/downlink Space to Ground ICD (NCST-ICD-FM003).

3.5.2.1.4 FAME Flight Software Design Document, NCST-SDD-FM001.

The Flight SDD captures the software component and unit architecture for the Flight Software CSCI. The SDD captures the static view of the architecture as a decomposition of the fight software into components, and the components into a set of logically related software units. The dynamic view of the architecture is captured as data and control flows between the flight software components. The SDD will conform to the FAME tailoring of MIL-STD-498 as identified in DID DI-IPSC-81435 found in appendix C.

3.5.2.1.5 FAME Flight Software Test Plan, NCST-STP-FM001.

The Flight Software Test Plan establishes the plans for software system level Formal Qualification Test (FQT) of the FAME Flight Software. The test plan will identify the software test environment, describe the testing approach, identify the tests to be performed, and provide the schedule for test activities. The test plan will trace the flight software requirements to the test cases. The STP will conform to the FAME tailoring of MIL-STD-498 as identified in DID DI-IPSC-81438 found in appendix C.

3.5.2.1.6 FAME Flight Software Test Report, NCST-STR-FM001.

The Flight STR documents the results of the execution of the Flight Software Test Procedures. The Flight STR documents the actual results vs. the expected results, problems encountered during testing, deviations from test case procedures, and test evaluations/recommendations. The STR will conform to the FAME tailoring of MIL-STD-498 as identified in DID DI-IPSC-81440 found in appendix C.

3.5.2.1.7 FAME Flight Software Version Description Document, NCST-SVD-FM001.

A Flight Software Version Description Document (SVD) will be produced for each release of the FAME Flight Software. The SVD identifies the flight software release for the FAME system. The SVD identifies, at a minimum, the software components being delivered, resolved Flight Software Action Requests (FSWARs), and any known deficiencies. The SVD will also define the configuration of all tools and data sets used to generate the version. The SVD is prepared by CM and released with the deliverable software product. The SVD will conform to the FAME tailoring of MIL-STD-498 as identified in DID DI-IPSC-81442 found in appendix C.

3.5.2.1.8 Flight Software User Manual.

A Flight Software User Manual will not be produced. The information intended for a Flight Software User's Manual will be represented by the Flight Software External ICD (NCST-ICD-FM008). While the entire contents of the FSW External ICD can be used as a reference guide for the Flight Software, section 4 of the ICD will provide user guide summary information.

3.5.2.2 Project Document Formats.

The FAME project documents will be created and managed using Microsoft Office 2000 tools (Word, Excel, PowerPoint, Access, Visio) on Windows 98/Windows NT platforms. Released documents will be available electronically in Portable Document Format (PDF) in Adobe Acrobat.

The FAME software components will be managed using the Rational ClearCase/Multi-site/ClearQuest CM tool set. Source code will be delivered for all NRL-developed software components. All makefiles, build scripts, and utilities will be delivered to support building of the software components for system integration and test, and operations and maintenance.

3.6 Size Estimates.

This subsection describes the software estimation process employed by NRL in developing size, effort, and cost estimates for the FAME software. The basis for the estimates is historically based from previous projects.

3.6.1 Software Size Estimates.

Flight software estimates are based on the software size for the ICM project. Table 3-5 shows the projected flight software size estimates:

Table 3-5. Flight Software Size Estimates

Module	LOC	ICM Reuse
Attitude Determination and Control (ADAC)	7400	4000
ADAC Executive	4000	2700
Delta-V	900	500
Math Library	600	443
Star Tracker Control	3000	1000
Downlink Processing	3000	0
Uplink Processing	3500	1750
Memory Processing	2200	2000
Real Time Command Processing	2500	0
Stored Command Processing	4200	4000
(TM) Limit Processing	700	500
Time Manager	900	500
Diagnostic Manager	1000	0
RM Application Services Library	18900	14700
1553 BC	4700	1500
Device Drivers	4500	0
OS Support	1200	1200
Boot Monitor And Diagnostics	6800	4700
Total	69900	39493

3.7 Project Resources.

3.7.1 Personnel.

Table 3-6 shows the staffing projections by software group over a projected three (3)-year period of performance.

Table 3-6. Staffing Projection by Software Group

WBS/Staff	Flight Software	Test and Release	Total
Software Management	1.0	0.2	**1.2**
Software/System Engineering	1.5	1.0	**2.5**
Software Development Engineering	8.0	0.0	**8.0**
Software Test Engineering	0.0	5.0	**5.0**
Quality Assurance	0.0	1.0	**1.0**
Configuration Management	0.0	1.0	**1.0**
Software Support	0.0	3.5	**3.5**
Totals	10.5	11.7	**22.2**

3.8 Project Budgets.

The project budgets are based on the man-loading, planned travel, and equipment purchases to support software development as well as integration and test. The FAME budgets are maintained separately by the FAME Project Manager and are not included in this SMP.

3.9 Project Management.

3.9.1 Project Planning/Monitoring.

This SMP is a living document that will be maintained throughout the software development phases. As the software requirements are refined and the software architecture designed, the information in this SMP will be updated to reflect the latest software projections and estimates.

3.9.2 Risk Management.

Risks will be managed throughout the FAME software development effort. The Lead FSW Engineer will identify risks. Status reports and Configuration Change Board (CCB) issues will both be used as the means for risk documentation, notification and tracking. This information will be elevated to FAME project management for review. The following information will be used to describe each identified risk:

- Risk Area identifies the area of risk.
- Risk Type identifies the engineering group affected.
- Risk Mitigation Strategy identifies control/reduction strategy for the risk.

3.9.3 Interfacing with IV&V Agents.

3.9.3.1 Interface with Software IV&V Analysis Agents.

TBD (R.I. – BD)

3.9.3.2 Interface Between IV&V Agent and Project.

TBD (R.I. – BD)

3.9.3.3 IV&V Agent Access to Deliverables and Resources.

TBD (R.I. – BD)

3.9.3.4 Participation in Reviews and Tests.

TBD (R.I. – BD)

3.9.4 Reviews and Reports.

3.9.4.1 Management Reviews and Reports.

Internal software reviews will be held on a monthly basis (or as necessary) to review the overall software status. These meetings will be held to review the software schedule, cost, priorities, and issues/concerns.

Status reports will also be submitted monthly and will include an updated software schedule, accomplishments for the previous month, plans for the next month, and any issues/concerns.

3.9.4.2 Technical Reviews and Reports.

3.9.4.2.1 Technical Reviews.

Several technical reviews are planned during the FAME project. The reviews for software are defined below.

In addition to these activities, all deliverable software documents will be made available for review in draft form prior to the first official release of the documents. Once the document is officially released, a Review Item Discrepancy (RID) or Configuration Control Notice (CCN) must be completed to request a change to a software document.

3.9.4.2.1.1 System Requirements Review (SRR).

Review the overall system requirements. The draft FAME Software Management Plan (NCST-SMP-FM001) will be released.

3.9.4.2.1.2 Preliminary Design Review (PDR).

Review the preliminary system design. Final versions of the FAME Flight Software Requirements Specification (NCST-SRS-FM001) and the FAME Software Management Plan (NCST-SMP-FM001) will be released. The initial FAME Flight Software External Interface Control Document (NCST-ICD-FM008), and the FAME Flight Software Test Plan (NCST-STP-FM001) will be released following PDR.

3.9.4.2.1.3 Critical Design Review (CDR).

Review the detailed system design. Review of the top-level software architecture is completed. Internal inspections, along with FAME IV&V inspections, will be performed on a subset of software units, unit tests, components, component tests, and documentation. The initial release of the FAME Flight Software Design Document (NCST-SDD-FM001) follows CDR.

3.9.4.2.1.4 Vehicle Test Readiness Review (VTRR).

Review of FSW readiness in preparation for formal vehicle level testing. The FSW may not reach the Formal Qualification Testing (FQT) milestone prior to FAME vehicle level Test Readiness Review (VTRR). Therefore, an evaluation of the non-FQT flight software must take place prior to the FAME vehicle TRR. The evaluation is preformed to assess the maturity of the FSW in order to protect the hardware, the personnel, and the ability to meet test objectives.

Acceptance criteria:

- The subset of FSW system level testing will be identified by identifying the subset test cases covered in the STP. This subset will cover the functionality needed to support the FAME vehicle testing. Some vehicle level tests may require software that is beyond the scope of the FSW requirements, such as special interface montoring software to be used during EMI/EMC testing. This software will be identified and required to be ready at VTRR.
- Flight SRS (NCST-SRS-FM001) and External ICD (NCST-ICD-FM008) released and maintained in NRL CM. All pending changes addressed to determine impacts.
- The FSW System level testing results covered by the test subset will be available for assessment. The completed FSW system level testing will cover the functionality needed to support FAME vehicle testing.
- The FSW will be under CM and consistent with a draft SVD.
- The Flight Software/Hardware interfaces will be tested.
- The open FSWARs will be addressed to determined impacts to the test.
- The Ground software and attitude simulator will be proven through prior use.
- Autonomous Protective Features will be demonstrated by test.
- Results found satisfactory from a hazard analysis for all software to hardware interfaces.

3.9.4.2.1.5 Test Readiness Review (TRR).

Review of readiness to begin formal FSW system level testing.

NCST-SDD-FM001 will be available in draft form. This version will capture, at a minimum, the high level design with a draft of the detailed design.

Any existing unit design checklists and inspection checklists will be made available for review at NRL. The content and coverage will not be used as part of the success criteria for TRR.

Any existing unit tests, unit test results and unit test inspection checklists will be made available for review at NRL. The content and coverage will not be used as part of the success criteria for TRR.

Component integration build reports will be made available for review at NRL. The content and coverage will not be used as part of the success criteria for TRR.

Acceptance criteria:

- The following documents must be released and under document CM and available for review two weeks prior to TRR:
  - NCST-SRS-FM001,
  - NCST-ICD-FM008,
  - NCST-SDD-FM001,
  - NCST-TP-FM001, and
  - NCST-SVD-FM001.
- The following must be available for review two weeks prior to TRR:
  - FSW source code as built for FQT,
  - Unit design checklists and inspection checklists,
  - Unit tests, unit test results and unit test inspection checklists,
  - Unit path coverage results,
  - Component integration build reports,
  - NCST-STR-FM001 (draft),
  - List of all software deviations and waivers,
  - List of all open work with a closure plan and schedule, and
- The TRR will consist of a review of the above material. Approval to move forward with FSW FQT will be as a result of NRL and NASA approval of the content and status of the above material. The NRL and NASA review of this material will utilize the following criteria to ascertain the readiness of the FSW to proceed into FQT:
  - CSC integration is correct and complete,
  - Formal software test plans, descriptions and procedures are complete and adequate to test requirements,
  - Traceability from all SRS and software ICD requirements to the formal test procedures,
  - FQT test procedures have been exercised through dry-runs and all known problems have been documented,
  - Identification of unused code (excluding COTS),
  - Determination that the software is maintained under proper configuration control and test environment is ready for use in conduct of formal test, and
  - Action plan for TRR issue resolution including assignment of actionee and closure date for each issue.

3.9.4.2.1.6 Flight Software Qualification Test (FQT).

Verify FSW requirement coverage.

Acceptance criteria:

- FAME QA witness of the NRL execution of all FSW system level test procedures.
- NRL and FAME QA Signoff of all FSW compliance verification worksheets.
- Resolution of all open action items.
- Release and CM of NCST-STR-FM001.
- Release and CM of NCST-SVD-FM001.
- Release and CM of all required changes to the products released for TRR.
- Release and CM of the NCST-SDD-FM001.

3.9.4.2.1.7 FAME System Level Acceptance.

Verify applicable Mission Requirements Document (MRD), (NCST-D-FM002), requirements coverage as a result of FSW system level requirement compliance. The FSW Requirements Specification (SRS) contains the FSW requirement to MRD mapping. Each FSW Compliance and Verification worksheet will indicate the FSW requirement mapping back to MRD requirements. The system level requirements (MRD) compliance and verification information is tracked at the system level and the mapping from MRD requirements to FSW requirements is also maintained at the system level.

Acceptance criteria:

- Flight Software Requirements Specification (NCST-SRS-FM001) contains FSW requirements to system level requirement mapping.
- Accepted FSW compliance worksheets.
- FSW requirement compliance applied to the applicable system level requirements.

3.9.4.2.2 Technical Peer Reviews.

In addition to the formal technical reviews, several technical peer reviews are planned. These include:

- Peer reviews of software component design,
- Peer inspection of software unit implementation, and
- Peer review of software unit tests and test results.

These reviews are detailed individually in Subsection 4.3.

3.9.4.2.3 Technical Interchange Memos.

Technical Interchange Memos (TIMs) will be used as the mechanism to address technical issues that are relevant to agencies external to NRL.

3.9.4.2.4 Technical Notes.

TNs will be written as required to document technical approaches and design. These are primarily for internal purposes and provide rationale for a particular software approach or design. Additional notes related to the development of a particular software component are documented in the component SDF.

3.9.4.2.5 Flight Software Action Requests (FSWARs).

FSWARs are the mechanism to be used to request a flight software change and/or report a flight software deficiency. All FSWARs will be tracked via the FAME CM system. A library of documentation, memos, and notes will be provided for linking and tracking technical details concerning FSWARs.

3.9.4.2.6 Configuraiton Change Board.

The Configuration Change Board (CCB) will be used as the forum to present all proposed changes to the FAME Flight Software and changes to the FSW external interfaces (NCST-ICD-FM008). The CCB members will include the NRL FSW Lead, the LMMS Instrument Software Lead, and the NRL GSW Lead. An IV&V representative will attend as a non-voting member but will be responsible for providing feedback with respect to risk, implementation and test assessments. The CCB will review and disposition all changes to the FSW and associated external interfaces. The CCB will be formed when the FSW and interfaces have matured to the extent that the FSW development is complete and the external interfaces are stable. The NRL FSW Lead will determine when the CCB activities are initiated. Proposed FSW changes will be dispositioned by the board members unless technical or cost issues require FAME Project Management (PM) and/or FAME System Engineering (SE) review. CCB issues that require FAME PM/SE review will be presented at the FAME monthly software status meeting.

4.SOFTWARE ENGINEERING DEVELOPMENT

4.1 Software Life Cycle.

The FAME flight software development effort will utilize an incremental software life cycle model. The incremental development model calls for a series of development builds, each of which leads to a product to be used in a test environment until the next incremental build is scheduled. Each incremental build provides more of the functionality required until the full functionality of the product is attained. Each incremental build in this model represents a development cycle including detailed design, unit code and unit test, and integration test. The incremental software builds that have been identified for the FAME development effort are documented in Table 3-3, Flight Software Release Descriptions**.** The FSW versions intended for operational use will undergo system level testing to ensure requirement compliance and operational capabilities.

4.2 Software Engineering Methodology.

4.2.1 Software Development Methods.

The FAME software will be designed using Rapid Object-Oriented Prototyping for Embedded Systems (ROPES). The method employed is a well-defined, step-by-step approach to developing software with rules, tests, guidelines, and examples based on actual experience with the method.

4.2.2 Software Development Terminology.

The terms 'software architecture,' 'software component,' 'software unit,' 'review,' and 'inspection' are used throughout the sections that follow. The following subsections define these terms as used in this document.

4.2.2.1 Software Architecture.

The software architecture is the top-level design of the software system. It involves all external sources of input, software components, and the connection or interface between these components.

4.2.2.2 Software Component.

A software component is defined as the initial partitioning of the software system into software tasks (i.e. VxWorks processes) or shared software libraries.

4.2.2.3 Software Units.

A software unit is a subdivision of the software component that provides a well-defined service of the component. A software unit has a well-defined interface for providing the service. A software unit may be implemented via a combination of both external functions (interface) and internal functions. All software units will be traceable to one (1) software component.

4.2.2.4 Review.

A review is the examination of a work product to determine whether the product has met its requirements and acceptance criteria. The participants and level of detail in the review will vary depending on the work product under review.

4.2.2.5 Inspection.

An inspection is the detailed examination of a software unit. The inspection will determine if the unit satisfies all requirements, and that all design and coding standards have been met. In addition, the unit software source code will be checked for defects in implementation of the component design.

4.2.3 Software Standards.

The software produced by the Software Development Team at NRL will be written to a tailored version of MIL-STD. These software standards are applied to all activities as a driving force during development and as a quality check during the review after each activity. Each standard will be entered as a Configurable Item (CI) and controlled through Configuration Management (CM).

4.2.3.1 Software Unit Design Standards.

Software unit design standards define the methods for designing software units. The design standards checklist, as defined in Appendix B, will be used for verification of the unit design. A completed unit design checklist will be placed in the SDF for a subset of the software units.

4.2.3.2 Software Unit Coding Standards.

Software unit coding standards define the methods for implementation of software units. The unit coding standards checklist, as defined in Appendix B, will be used for verification of the unit implementation.

4.2.3.3 Software Unit Test Standards.

Software unit testing standards specify the testing required at the unit level. The unit test standards checklist, as defined in Appendix B, will be used for verification of the unit test. A completed unit test checklist will be placed in the SDF for a subset of the software units.

4.2.3.4 Software Component Integration Standards.

Software component integration standards specify the testing required at the component interface level. The software component integration test standards checklist, as defined in Appendix B, will be used for verification of the component integration test. A completed component integration checklist will be maintained as input to the Flight Software TRR when necessary.

4.2.4 Software Reuse.

The FAME project will utilize a combination of Commercial Off-The Shelf (COTS) software, as well as software components from previous projects at the Naval Research Laboratory (NRL).

4.2.4.1 Commercial Off-The Shelf (COTS) Software.

Various COTS products will be used to support the FAME project. The following list details some of the significant operational COTS products.

- VxWorks™ Real-Time Operating System will be used on the FAME Integrated Spacecraft Controller (FSC).
- Spacecraft Command Language (SCL) ™ flight system tool kit for providing Flight Command and Telemetry (C&T) capabilities, as well as event-triggered command processing to support Fault Detection, Isolation, and Recovery (FDIR).

4.2.4.2 Non-Developmental Item (NDI).

Software components will be reused as appropriate during both design and coding. The determination of specific component reusability will occur during the top-level design, detailed design, and coding phases of software development.

4.2.4.2.1 Software Reuse Verification and Validation (V&V).

All reuse software components will, at minimum, enter the verification and validation (V&V) phases during component inspection. At this point, all identified critical source code is inspected for defects against the required functionality.

The reuse deficiencies identified during the inspection process are reported in Flight Software Action Requests (FSWARs). Code not acceptable by the FAME requirements will be modified, unit tested, and inspected again for acceptability.

4.3 Software Development Activities.

This section details the software development activities to be performed during FAME flight software development. Each subsection details a phase of the software process to be used by NRL in management and development of the FAME software.

Figure 4-1, Software Development Activities, depicts the incremental software development activities and illustrates the relationships among them. It also shows the System Engineering activities of System Design feeding into the software development process, and the Systems Engineering activities associated with system-level testing of the developed software product(s).

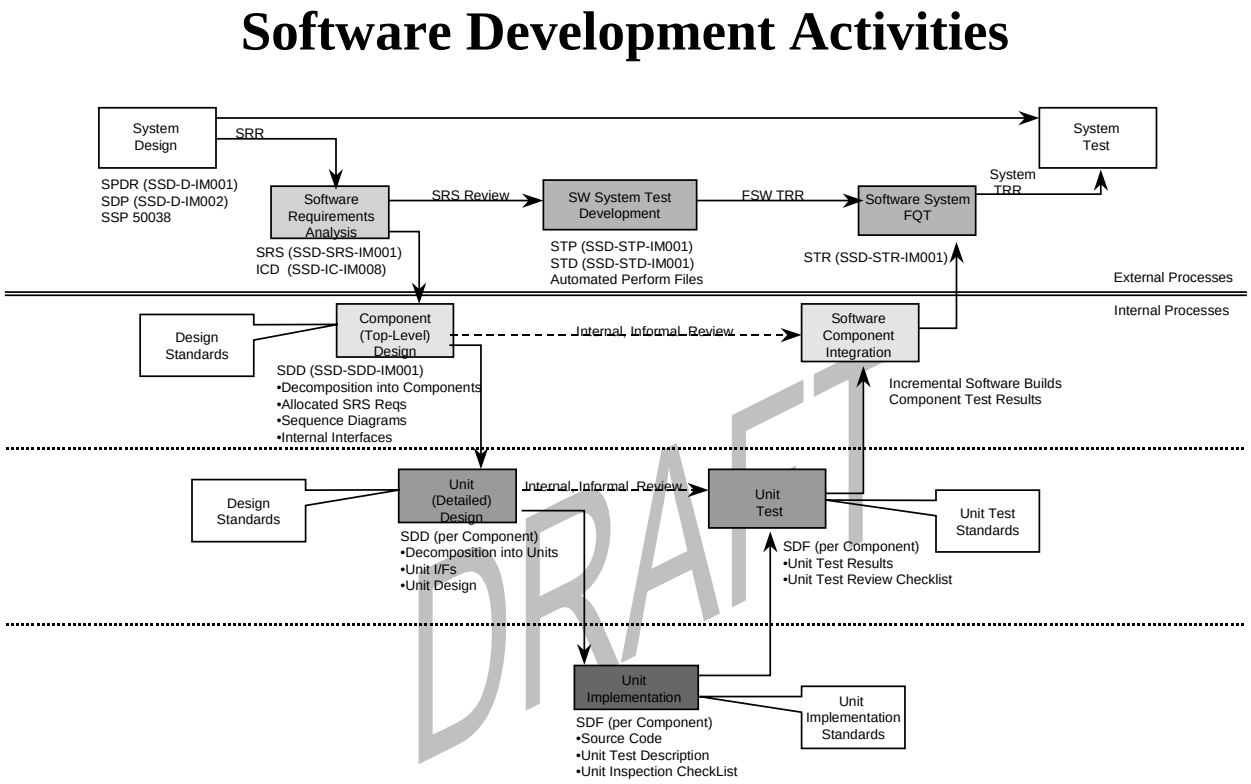

Figure 4-1. Software Development Activities

In this figure System Design, Software Requirements Analysis, and Software Top Level Design are intended to be produced in order and placed under control. All these documents and their dependents will undergo controlled change once placed under CM control.

Software Detailed Design and all of its dependents represent the incremental path for each code unit. Each unit of software will be developed, tested individually, and integrated into its larger component. Each component will be integrated during the component integration activity. Formal Qualification Testing of the Flight Software will performed at the software system FQT.

4.3.1 Software Requirements Analysis.

The Software Requirements Analysis phase identifies specific software requirements and interfaces from the baselined system documents.

4.3.1.1 Resources.

FAME system-level documents as listed below:

- FAME Mission Requirements Document, (NCST-D-FM002)
- FAME Software Management Plan, (NCST-SMP-FM001)
- FAME System Requirements Review (SRR) and Preliminary Design Review (PDR) Presentation Packages.

4.3.1.2 Work Products.

- Flight Software Requirements Specification, (NCST-SRS-FM001),
- Initial Flight Software External Interface Control Document, (NCST-ICD-FM008).
- Instrument to S/C ICD (NCST-ICD-FM001). This document is a LMMS produced and maintained document but requires acceptance (by signature) from both LMMS and NRL representatives.

4.3.1.3 Reviews.

A review of the Flight Software Requirements Document (NCST-SRS-FM001) and the Flight Software External Interface Document (NCST-ICD-FM008) will be performed with NASA.

The development of the Instrument to S/C ICD (NCST-ICD-FM001) is a joint LMMS/NRL effort. LMMS will produce and maintain the document using NRL review inputs and concurrence.

4.3.1.4 Exit Criteria.

All action items identified in the SRS review shall be resolved, and the SRS entered into NRL CM to produce the initial baselined document.

All action items identified in the FSW External ICD review shall be resolved.

All action items identified in the Instrument to S/C ICD review shall be resolved.

4.3.2 Software Component (Top-Level) Design.

During the Software Component Design development phase, the top-level component architecture is developed. The software requirements identified in the SRS are allocated to the identified software components. A Sequence Diagram is generated for each external flight software stimulus identified in the Flight Software External ICD. An SDF will be created for each component.

In conjunction with the development of the FSW external ICD (NCST-ICD-FM008), two deliverable products are generated. The telemetry definition files (spreadsheets) provide the information required to decommutate and interpret the FAME return link data stream. The commanding definition files (spreadsheets) provide the information required to format and transmit the FAME forward link data stream.

4.3.2.1 Resources.

- FAME Mission Requirements Document, (NCST-D-FM002)
- FAME Software Management Plan, (NCST-SMP-FM001)
-
- FAME System Requirements Review (SRR) and Preliminary Design Review (PDR) Presentation Packages
- Flight Software Requirements Specification, (NCST-SRS-FM001)
- Flight Software External Interface Control Document, (NCST-ICD-FM008).
- Instrument to S/C ICD, (NCST-ICD-FM001)

4.3.2.2 Work Products.

- Component Design documented in the Flight Software Design Document, NCST-SDD-FM001, containing at a minimum:
- Identification of all Computer Software Components (CSCs)
- Allocation of SRS Requirements to Components
- Flight Software External and Internal Interfaces (i.e. Static Component to Component Interfaces)
- Sequence Diagram for each External Flight Software Stimulus as identified in the ICD (i.e. Dynamic Component to Component Interfaces)
- Updated Flight Software External ICD, NCST-ICD-FM008.
-
- A draft FSW System level Test Plan (NCST-STP-FM001)
- A CDR package.
- Definition of the Forward Link formats and protocols.
- Definition of the Return Link formats and protocols.

4.3.2.3 Reviews.

A review of the appropriate sections of the SDD will be held to review the software component architecture, allocated requirements, and sequence diagrams for the flight software.

A critical design review (CDR) will be held with NASA to review the Flight Software Top-level Design.

Top-level design information will be presented during the FSW development cycle using various forums. These forums include detailed design review (DDR), technical interchange meetings (TIM), and internal reviews.

The top-level design will be reviewed prior TRR using a distribution of a draft SDD representing the FSW as built.

4.3.2.4 Exit Criteria.

The initial draft version of the Flight Software SDD shall be created that identifies each software component, the allocated SRS requirements, and the sequence diagrams for each external flight software stimulus, and the derived internal interfaces. (A final SDD will be released post FQT in preparation for FSW buyoff.) The Flight Software External ICD shall be updated. Review items relating to the FSW top-level design will be collected at CDR and resolved.

4.3.3 Software Unit (Detailed) Design.

The Software Unit Design phase analyzes and designs the internal structure of each software component. Each component is decomposed into one or more software units. The unit design captures both the unit interfaces and the internal design of unit.

The SDD will include unit descriptions but will not include the internal design of the unit.

4.3.3.1 Resources.

- FAME Mission Requirements Document, (NCST-D-FM002)
- FAME Software Management Plan, (NCST-SMP-FM001)
- Critical Design Review (CDR) Presentation Package.
- Flight Software Requirements Specification, (NCST-SRS-FM001)
- Flight Software External Interface Control Document, (NCST-ICD-FM008).
- Flight Software Design Document draft (NCST-SDD-FM001),
- Instrument to S/C ICD (NCST-ICD-FM001)

4.3.3.2 Work Products.

- The SDD is updated for each software component which contains at a minimum:
- Identification of all the Units within the Component
- Unit Interfaces
- Unit Design
- The FSW External ICD (NCST-ICD-FM008) is updated to reflect ongoing interface definitions.

4.3.3.3 Reviews.

An internal FSW team review of the appropriate sections of the SDD will be held to review the software unit design.

4.3.3.4 Exit Criteria.

All action items from the Flight Software Design Document review are resolved; the SDD is baselined and entered into CM. The SDD will include unit identification, external unit interfaces, and unit descriptions but will not include the detailed design of the units.

The Flight Software External ICD shall be updated.

4.3.4 Software Unit Implementation.

The Software Implementation phase implements the Unit Design and documents the Unit Test Descriptions. Both the Unit Source code and Unit Test Descriptions are reviewed at the Unit Inspection and documented in the

Software Development Folder. The Unit Interfaces will be the primary mechanism for driving the unit tests. Unit Interfaces tests will be repeatable. Intrusive unit testing will be performed as required to satisfy any "path coverage" requirements, but is not required to be repeatable.

Unit implementation and unit test descriptions will be reviewed informally. Documentation from the review process will be captured for a subset of the units. The unit test approach will be based on the guidelines (Appendix B) and tailored per unit based on developer's judgement and experience. Unit testing will utilize a combination of manual and automated procedures.

4.3.4.1 Resources.

- FAME Mission Requirements Document, (NCST-D-FM002)
- FAME Software Management Plan, (NCST-SMP-FM001)
- Critical Design Review (CDR) Presentation Package
- Flight Software Requirements Specification, (NCST-SRS-FM001)
- Flight Software External Interface Control Document, (NCST-ICD-FM008)
- Instrument to S/C ICD, (NCST-ICD-FM001)
- Flight Software Design Document, (NCST-SDD-FM001).

4.3.4.2 Work Products.

A Software Development Folder (SDF) for each component containing at a minimum:

- Source code for each unit identified as part of the component
- Unit-level test descriptions for a subset of units
- Unit Inspection Checklist for a subset of units

4.3.4.3 Reviews.

A peer review will be held to review the source code and unit-level test description.

4.3.4.4 Exit Criteria.

All action items identified in the review shall be addressed. The SDFs shall be updated to contain the unit code, unit test description (if available), and unit inspection checklist (if available). The unit test descriptions and unit inspection checklists will be available in the SDFs for only a subset of units. The Flight Software External ICD shall be updated. The Detailed Design shall be updated in the Flight Software Design Document (SDD). The internal design of units are not captured in the SDD.

4.3.5 Software Unit Test.

The Software Unit Test verifies the implementation of the software unit against the unit design and test standards (Appendix B). Unit Tests are primarily driven via the Unit Interfaces and will be repeatable. Additional, intrusive unit testing may be performed to satisfy path coverage, but are not required to be repeatable as part of the Unit Test. The Unit is a Building Block approach such that previous tested units will be used in testing higher-level units until all the units are integrated into the component. The top-level unit test verifies the fully integrated units into a software component.

The unit test approach will be based on the test guidelines (Appendix B) and tailored per unit based on developer's judgment and experience. Unit testing will utilize a combination of manual and automated procedures.

4.3.5.1 Resources.

- FAME Mission Requirements Document, (NCST-D-FM002)
- FAME Software Management Plan, (NCST-SMP-FM001)
- Critical Design Review (CDR) Presentation Package.
- Flight Software Requirements Specification, (NCST-SRS-FM001)
- Flight Software External Interface Control Document, (NCST-ICD-FM008).
- Instrument to S/C ICD, (NCST-ICD-FM001)
- Flight Software Design Document, (NCST-SDD-FM001),

4.3.5.2 Work Products.

The SDF is updated with:

- Unit test description and test results for a subset of units
- Unit Test Review Checklist for a subset of units

4.3.5.3 Reviews.

A peer review will be held to review the unit test results for a subset of units.

4.3.5.4 Exit Criteria.

All action items identified in the unit test review shall be resolved, and an updated SDF, containing the tested software unit, test results, and unit test review checklist if these have been produced.

4.3.6 Software Component Integration.

The Software Component Integration phase integrates the successfully tested software components into a software system build on the target hardware. The software component integration verifies the software build against the sequence diagrams identified in the SDD. Software component integration will consist of informal sessions using SW external interfaces to validate component integration.

4.3.6.1 Resources.

- FAME Mission Requirements Document, (NCST-D-FM002)
- FAME Software Management Plan, (NCST-SMP-FM001)
- Critical Design Review (CDR) Presentation Package
- Flight Software Requirements Specification, (NCST-SRS-FM001)
- Flight Software External Interface Control Document, (NCST-ICD-FM008).
- Instrument to S/C ICD, (NCST-ICD-FM001)
- Flight Software Design Document, (NCST-SDD-FM001)

4.3.6.2 Work Products.

- A flight software system build tested against the sequence diagrams identified in the SDD.
- A software system build report captured in the SDFs.

4.3.6.3 Review.

An internal review will be held to informally review the component integration activities.

4.3.6.4 Exit Criteria.

The software system build successfully executes the sequence diagrams identified in the SDD. All action items identified in the component integration review shall be resolved. Updated SDFs, containing the software system build report.

4.3.7 Software System Test.

The Software System Test verifies that the integrated software components satisfy the requirements as documented in the SRS. This test verifies at the Computer Software Configuration Item (CSCI) requirement level. The FSW requirement compliance activities will include the test, inspection, analysis and demonstration actions as identified in the SRS. The results of these activities will be captured (or pointed to) in the Software Test Report.

4.3.7.1 Resources.

- FAME Mission Requirements Document, (NCST-D-FM002)
- FAME Software Management Plan, (NCST-SMP-FM001)
- Critical Design Review (CDR) Presentation Package.
- Flight Software Requirements Specification, (NCST-SRS-FM001)
- Flight Software External Interface Control Document, (NCST-ICD-FM008).
- FAME/ISS Software Interface Control Document, (SSP-50278)

- Flight Software Design Document, (NCST-SDD-FM001)
- A draft FSW System level Test Plan (NCST-STP-FM001)

4.3.7.2 Work Products.

A final FSW System level Test Plan (NCST-STP-FM001)

Software Test Reports (NCST-STR-FM001) to document the actual test, analysis, inspection and demonstration activities. This documentation will provide the means to assess the results of these activities.

A FSW Version Description Document (NCST-SVD-FM001)

4.3.7.3 Reviews.

A Flight Software Test Readiness Review (TRR) will be held to assess the readiness to begin the Formal Qualification Test (FQT).

An FQT Review will be held to review the software system test results.

4.3.7.4 Exit Criteria.

All action items identified in the FQT review shall be resolved. Sign-off for the FQT will be performed by NASA and NRL.

5.TEST ENGINEERING

Five (5) levels of software testing have been identified for the FAME project:

- Unit-Level Testing;
- Component Testing;
- Integration Testing;
- System Testing; and
- Formal Qualification Testing.

The individual software developers will perform Unit-Level Testing, which is "white box" testing of a component implementation. Paths in the code that are difficult to exercise at Component Test, Integration Testing or System Test should be exercised by Unit-Level Testing.

The software developer will perform Component Testing, with an informal sell-off to the Software Component Integration and Test Lead Engineer. Component Testing is "black box" testing of the component verifying the component interface and functional and functional requirements. Component Testing will verify an individual component prior to integrating it with other components.

The Software Component Integration and Test Lead Engineer will perform Integration Testing. This testing will test the interaction of tested software Components. Derived system requirements will be verified at this stage of testing. Integration Testing will be performed on the target system.

Software System Testing will be performed to verify that the integrated components of the software system satisfy the Software Requirements Specification (SRS) and Interface Control Document (ICD) for the specified software build. The Test Engineering Group will execute the Software/System Test Plan/Procedures on the target system.

Formal Qualification Testing will be performed to verify that the FAME software satisfies all the requirements of the SRS and ICD. The Test Engineering Group will perform this testing that is witnessed/validated by the customer and Independent Verification and Validation (IV&V) team.

5.1 Software Test Development.

5.1.1 Software/System Test Plan/Procedure Development.

The Software Test Plan/Procedure Development phase partitions assigned requirements into specific test plans and procedures. Test cases are developed, and requirements and interfaces are mapped to the test cases. Expected results are documented in the Test Procedures.

5.1.1.1 Resources.

- The Flight Software Requirements Specification, NCST-SRS-FM001, which identifies the flight software requirements of the FAME system.
- FAME External Interface Control Document, NCST-ICD-FM008.
- A baselined FAME Test Plan, NCST-STP-FM001

5.1.1.2 Work Products.

- A test objective for each planned test case.
- Acceptance criteria for each planned test case.
- A mapping of requirements and interfaces to the test cases.
- OS/COMET™ Perform Files for automated testing of the software.
- OS/COMET™ Displays for visual observation of test execution.

5.1.1.3 Reviews.

A formal review will be held to review the test plan/procedures and to verify that all requirements and interfaces are tested. Execution of the tests will be witnessed in the Formal Qualifications Test (FQT).

5.1.1.4 Exit Criteria.

The exit criteria for sign-off are the successful execution of all test cases, and the resolution of all action items associated with the Test Procedures and Descriptions document and Test Execution**.** Sign-off of the test plan and procedures shall be performed in order to baseline the document.

5.1.2 Flight Software Independent Verification and Validation (IV&V) Approach.

TBD (R.I. – BD)

5.1.2.1 Software System Test IV&V.

TBD (R.I. – BD)

5.1.2.2 Independent System Validation.

TBD (R.I. – BD)

6.SOFTWARE QUALITY ASSURANCE

Software Quality Assurance (SQA) will be an on-going assessment of the software development products and processes during all phases of the software development life cycle. SQA activities will be inherent in the FAME software development activities via reviews and inspections. The SQA process will enable all software developers and users to aid in the achievement of the highest quality software.

6.1 Resources.

6.1.1 Personnel.

The Software QA responsibilities are distributed between the LSWE, FSW developer leads and the Test and Release Engineering Lead. The standards are documented in appendix B as checklists. The checklists will be used as guidelines for the QA process and not as QA requirements

6.1.2 Software Tools.

The Rational ClearQuest software will be used to track all work done during the development process. This will offer the following functionality:

- Tool support for reporting, searching and query of defect status
- Distributed submission and status reporting of defects through a website
- The website, accessible at NRL, will define the procedures for defects reporting in an on-line help file and tutorial.

6.2 Activities.

6.2.1 Software Product Evaluation.

6.2.1.1 Software Quality Reviews.

The Software QA responsibilities are distributed between the LSWE, FSW developer leads and the Test and Release Engineering Lead. One or more persons with assigned QA responsibilities will be present at every review and inspection specified within section 4.4, Software Development Activities. SQA personnel will ensure that:

- Specified checklists and standards have been used as appropriate.
- All specified exit criteria for the development phase have been met.
- A transition may be made to the next development phase.

6.2.1.2 Software Quality Tracking.

Each initial software requirement, derived requirement, requirement change, and action item will be entered into the defect tracking system as a Flight Software Action Request (FSWAR). This will enable tracking of requirements, changes, and action items throughout the development process.

The FSW requirements are published in the Flight Software Requirement Specification (NCST-SRS-FM001) and associated Configuration Change Notices. The requirements are captured in the FAME SW Design Data Base.

6.2.2 Software Process Evaluation.

The LSWE is responsible to assess the quality and effectiveness of all activities with the Software Management Plan (SMP). The SMP may undergo changes and will be baselined in Configuration Management (CM). The most recent baselined version of the SMP is the measure used for assessing the SMP processes and procedures.

6.2.3 Corrective Action.

6.2.3.1 Flight Software Action Request (FSWAR).

All development work will be tracked through the defect tracking system via the use of FSWARs. This will provide a record of required work and accomplished work. Any software user who discovers a defect in the system must submit a FSWAR in order for corrective action to be taken. The FSWAR process will be addressed in detail in the Software Configuration Management User's Guide.

6.2.3.1.1 FSWAR Corrective Action Tracking.

FSWARs are tracked as they move through a life cycle that begins with the submission of a FSWAR to the FSWAR Review Group, and typically ends with the resolution of the issue or problem. The use of a defect tracking system to monitor FSWAR state transitions enables the appropriate personnel to manage defects and follow FSWARs through their life cycle. All transitions of each FSWAR will be recorded. Allowed transitions will be controlled and managed by the following personnel in the following ways:

- FSWAR Review will control the initial entry point and acceptance of a FSWAR.
- The Change Control Board (CCB) will control the postponement of work, the identification of duplicates, and the assignment of the responsible engineer. The LSWE will maintain responsibility for the necessary work to accomplish each build.
- The software engineer will be allowed to open, design, develop, and rework software based on the assigned FSWAR.
- Functional testing and test result reviews will control the transition of any FSWAR into the verified or rework states.
- The CM Manager will close and record the opened FSWAR as verified and complete. Further action will require a new FSWAR to be submitted.

The following subsections detail the life cycle of a FSWAR.

6.2.3.1.1.1 Submitted.

Submission is the initial state of any FSWAR report. The user identifies a problem and submits it to the FSWAR Review, which determines the responsible development group and forwards the defect report. A copy is kept on the submitting machine or file server, and is continually updated as the defective software moves through various states. The person who submitted the FSWAR will receive up-to-date information on the defect status.

6.2.3.1.1.2 New.

The FSWAR enters the ‘New’ state when it arrives on the home system, joins that project’s collection of defects, and is assigned to a development group. At this point, no engineer has been assigned to fix the problem. An acknowledgment is returned to the person who submitted the FSWAR.

6.2.3.1.1.3 Assigned.

The development group manager can take one of three actions regarding the FSWAR: assign the problem to the responsible engineer, identify the problem as a duplicate, or transition the work to be postponed.

6.2.3.1.1.4 Open.

If a FSWAR is assigned to an engineer for resolution, the assigned engineer acknowledges receipt of the problem and begins working on it. Information is gathered about the problem and documentation is attached to the FSWAR. The development group manager and the person who submitted the FSWAR are notified that the FSWAR has been opened.

6.2.3.1.1.5 Duplicate.

If the problem is determined to be a duplicate of an existing problem, or related closely to a previous FSWAR, it is identified as such in the tracking system and the engineer and the person who submitted the FSWAR are notified of this state transition.

6.2.3.1.1.6 Postponed.

If the development group manager determines that the problem will not be corrected immediately, but will be postponed for some later release, it is identified as ‘Postponed’ in the tracking system. The engineer and the person who submitted the FSWAR are notified of this state transition.

6.2.3.1.1.7 Inspect.

After a FSWAR has been opened, and the responsible engineer implemented the solution, the FSWAR enters the "Inspect" state and is ready for peer review. On satisfactory completion of the peer review, the SAW transitions to the "Software Test" state.

6.2.3.1.1.8 Software Test.

After the FSWAR has passed the "Inspection" state, the FSWAR enters the 'Software Test' state. This state indicates that the problem solution is ready or waiting for software testing. Testing is performed on the development system. The engineer and the person who submitted the FSWAR are notified of this state transition.

6.2.3.1.1.9 System Test.

If the solution presented by the responsible engineer has passed through the Software Test state, it enters the 'System Test' state. This state indicates that the problem solution is ready or waiting for system testing. System testing is performed on the target hardware system. The engineer and the person who submitted the FSWAR are notified of this state transition.

6.2.3.1.1.10 Rework.

If the solution presented by the responsible engineer has failed the Software Test state, it enters the 'Rework' state. This state indicates that the problem failed to pass on any of the test beds and rework of the solution is required. The engineer and the person who submitted the FSWAR are notified of this state transition.

6.2.3.1.1.11 Closed.

If the solution presented by the responsible engineer has passed through the System Test state, it is identified in the tracking system as 'Closed'. The closed state allows for CM to attach any documentation or reports required for closure. The engineer and the person who submitted the FSWAR are notified of this state transition.

6.2.3.1.2 Defect Severity.

FSWARs will have a severity level assigned when the defect is in the 'New' state. These severity levels are defined according to Appendix C of MIL-STD-498 and are listed in Table 6-1. The assignment of severity levels will be made at FSWAR Review.

Table 6-1. Default Severity Definitions

Severity Level	Applies if a problem could:
1	Prevent the accomplishment of an operational or mission-essential capability problem. Jeopardize safety, security, or any other requirement designated "critical."
2	Adversely affect the accomplishment of an operational or mission-essential capability and no work-around solution is known. Adversely affect technical, cost, or schedule risks to the project or to life-cycle support of the system, and no work-around solution is known.
3	Adversely affect the accomplishment of an operational or mission-essential capability but a work-around solution is known. Adversely affect technical, cost, or schedule risks to the project or to life-cycle support of the system, but a work-around solution is known.
4	Result in user/operator inconvenience or annoyance but would not affect a required operational or mission-essential capability. Result in inconvenience or annoyance for development or support personnel, but will not prevent the accomplishment of those responsibilities.
5	Produce any other effect.

6.2.3.2 Software Process Compliance.

Software process compliance is the responsibility of the LSWE and the Subsystem Leads. All members of the team perform process evaluations on a continuous basis. Process guidelines are determined by the LSWE.

6.2.4 Software Quality Assurance Records.

- ClearQuest is used to create reports of FSWAR activity and metrics.
- The defect tracking system automatically associates defects with their respective software products.
- All defect tracking system reports and a history of all defect-tracking activities are available to all software personnel (including IV&V) at any time.
- All available results of inspections and reviews are stored under CM in the SDFs. Inspections and reviews at the unit level are performed on a subset of units. Integrated inspections and reviews are performed informally and not documented.

7.SOFTWARE CONFIGURATION MANAGEMENT (SCM)

7.1 SCM Organization.

The Lead Software Engineer (LSWE) will appoint a Software Configuration Manager (SCMGR), who will have the following responsibilities:

- Administering the ClearCase and defect tracking system products
- Performing and delivering product builds
- Providing a Version Description Document (VDD) with product builds
- Providing reports to software management

7.2 SCM Resources.

Rational's ClearCase software product will be used for Configuration Management (CM) activities. ClearCase provides the means for item version control, item change control, project baselining, and the collection of configuration audit data.

ClearCase uses the "sandbox" scheme for development. The baseline versions of CM items are contained in a global database. Individual developers check files out of this database to edit, build, and test in their own area (sandbox) separate from other developers. This allows developers to change software independently from other developers.

ClearCase will allow CM to be done on all types of files as well as directory contents. Source code, ASCII data files, binary data files, and documents can be controlled by the CM system. Build products (object code, executables, etc.) will not be tracked under CM, but since the source code and procedures (makefiles, build scripts) are controlled by CM, these build products are reproducible items by version. The current baseline version, and any previous version can be reproduced.

Third-party products that come in source code form can be tracked in vendor-release branches. This way, any local modification required of these products can be tracked and not lost with new releases.

7.3 SCM Activities.

A FAME Configuration Management User's Guide will be written and available to address SCM issues including interfacing the multi-site communications between the Naval Research Laboratory (NRL) development sites and remote sites, managing baselines, tracking system dependencies, and tracking system defects.

The FAME Flight Software Action Request (FSWAR) establishes a document trail to track software problems, enhancements, and change requests for the FAME software components. In addition, the FSWAR provides a means to measure software quality (metrics), and show when a software component is qualified to become part of a deliverable baseline.

7.3.1 Configuration Identification.

The Configurable Item (CI) is any item of development that may undergo change during the lifetime of development. The following items are identified for configuration:

- All deliverable documents
- The Software Baseline. This most-recent version exists only on the main branch of the development tree, as described in the SCM Plan.

These CIs are maintained as one current version, and all changes performed on the item. In this manner, the initial version, all intermediate versions, and the current version of an item can be accessed.

Deliverable items are a subset of all the items maintained and represent the final version for each delivery.

Figure 7-1 describes the general tree structure for the FAME Flight Software development.

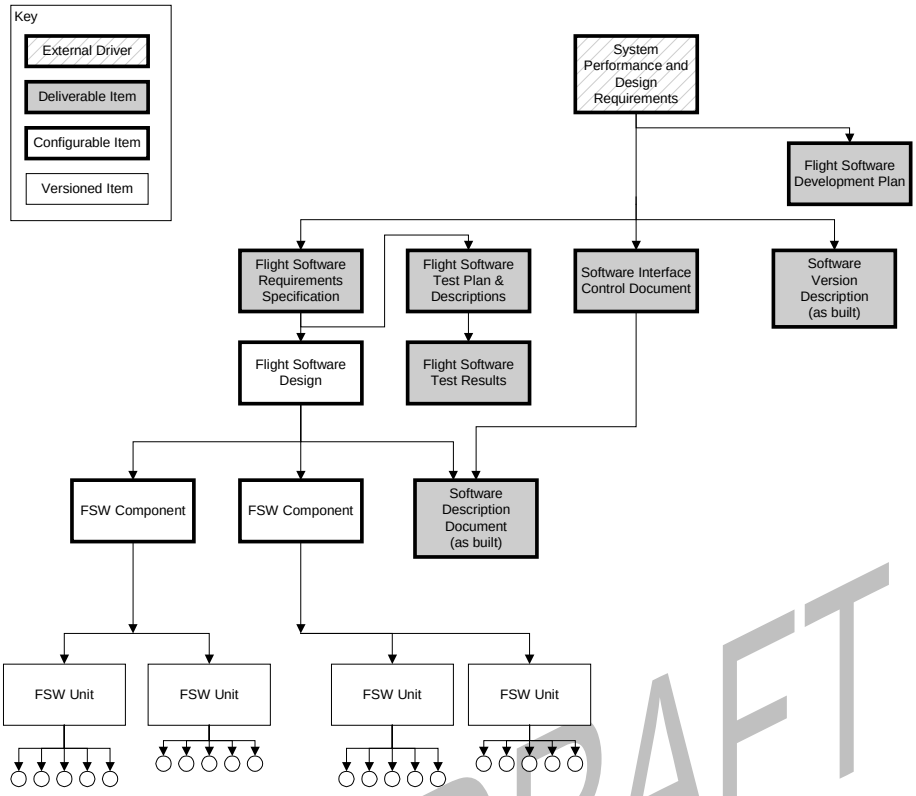

Figure 7-1. Software Configuration Management Tree

7.3.2 Change Control.

A Change Control Board (CCB) will be appointed by the LSWE to decide which change will be made to CIs. Items under Change Control, CIs, are shown in Figure 3-1, Software Configuration Management Tree.

The process to change a CI starts with a FSWAR submitted against the CI. The CCB will review this FSWAR, and either open the change request or reject it. Once the FSWAR is opened, the LSWE can assign the change to a Responsible Individual (RI) to make the change. Changes to the Software Baseline for a CI are made only after the FSWAR for the item has reached the 'Closed' state.

7.3.3 Version Control.

The following are maintained under version control, using ClearCase:

- All items under Change Control
- All files that make up a component
- All software for operations and testing
- All data (tables, constants, etc.) required by component
- All software created and used in Quality Assurance activities
- Directory structure

7.3.4 Configuration Status Accounting.

The history of all CM activity performed by ClearCase is automatically maintained by ClearCase and is available for any user to view at any time. The ClearCase software can be used to provide reports desired by the SCMGR or the LSWE.

7.3.5 Configuration Audits.

The LSWE and SCMGR will determine the versions of all components that will comprise a deliverable build of the software. This will generally include all items in the Software Baseline with any necessary adjustments.

7.3.6 Milestones.

SCM milestones correspond to the software releases described in Subsection 3.4.

7.3.7 Packaging, Storage, Handling, and Delivery.

The SCMGR will be responsible for packaging a software build and delivering it from the development system to the test beds and other systems. A VDD will be delivered with each baseline release of the software.

8.SOFTWARE SUPPORT

8.1 Software Development Environment.

The Software Development Environment will include multiple Sun workstations, IBM compatible personal computers (PCs), an application/Configuration Management (CM) server, commercial μP boards, and printers networked together on 100mBit local area network (LAN). This development network will be connected to the four (4) test bed LANs and the Naval Research Laboratory's (NRL's) NRLNet via the FAME backbone network.

Figure 8-1 illustrates the planned software development network. During the development phase software simulators and commercial μPs running on the engineering workstations will be used to unit test software that will be eventually be run on long lead time hardware.

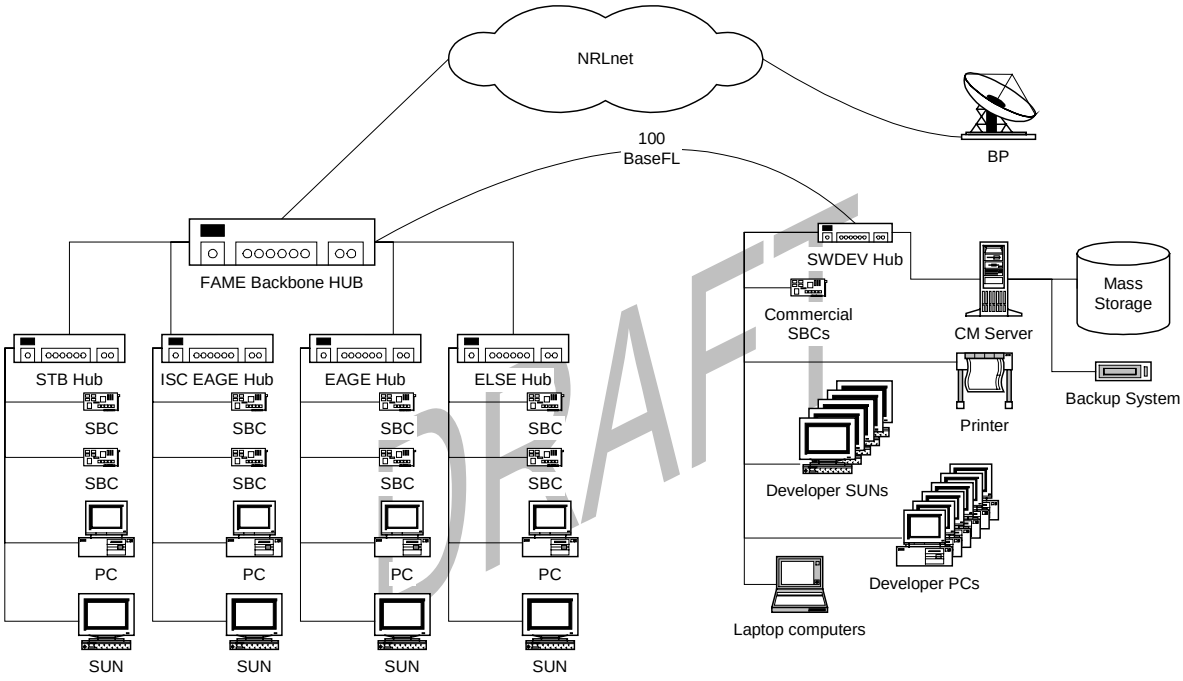

Figure 8-1. Software Development Environment

Table 8-1 details the resources needed for the various phases of development. Software builds will be delivered to the test bed fixtures via the CM tools. This allows changes to be made to software on the test bed that are still under CM. In addition, Flight Software Action Requests (FSWARs) can be submitted directly from the test bed systems. These CM concepts will be described in the Software Configuration Management (SCM) Plan.

Table 8-1. Software Development Resources

Project Phase	Product	Qty	Need Date	Description
Development Hardware	Sun Ultra 5	8	2/1/01 (5 currently available)	Sun Workstations running Solaris 7 operating system, used for the design development of flight, ground, and test bed code
	Sun Ultra 60	1	12/1/00	Sun Server used to host CM system, mass storage and system backup hardware
	Pentium PCs	15	2/1/01	PCs running Windows NT used for documentation and as X Terminals to provide additional access to Sun workstations
	MIPS Single Board Computers	2	Available	SBC used to test flight or test bed code
	SPARCstorage Array Model TBD (R.I. RCC)	1	2/1/01	Mass storage for CM system and software. This model supports and will be operated at RAID level TBD (R.I. – RCC)
	Printer	2	2/1/01	Provide printing from the development system.
Requirements Analysis	TBD (R.I. – MH)	2	2/1/01	Software requirement analysis and modeling tool.
Ground System Software	OS/COMET™	6	2/1/01	**O**pen **S**ystems **/ COM**mon **E**nvironment for **T**est. Ground system software for command and telemetry tests
	SCL™	2	5/1/01	**S**pacecraft **C**ommand **L**anguage. Used to maintain and compile flight software scripts.
Development	Solaris 7 or greater	9	2/1/01	Open Systems Operating System being used as the primary operating system for FAME development.
	VxWorks cross development environment	8 user	2/1/01	C Development environment for flight code and test bed embedded systems.
	VxWorks source code	TBD – R.I. RCC	2/1/01	Is this needed ? We're not supposed to use ICM.
	RHC-3001 Compiler	TBD – R.I. RCC	2/1/01	Only needed if RHC-3001 is chosen.
	SunSoft Visual Workshop for C	8 user	2/1/01	C & C++ Development Environment for the Sun Workstations.
	SunSoft FORTRAN Workshop Compiler	1 user	2/1/01	FORTRAN development for building Attitude Determination and Control (ADAC) models
Networking	100Base-TX Ethernet Hub	6	2/1/01	Provide fast Ethernet for development system
Test / Debugging	WindView and Stethoscope	3	2/1/01	Real-time software characterization and analysis. Will identify and analyze system resource usage, dead-locks, etc.
	VxSim	9	2/1/01	For simulation of embedded systems on the Sun workstations
	Rational Purify/Purelink	4	2/1/01	Run-time code checking and error detection
	Parasoft Insure++ (or equiv)	5 user	2/1/01	Static code checking and error detection
	VxWorks Code Test	2	10/1/01	VxWorks Path coverage testing
Backup	Sun Solstice	1	2/1/01	Backup software to manage automated backup and tape jukebox
	SPARCstorage 140GB 8mm tape library	1	2/1/01	Tape jukebox for automated system backup
Configuration Management	Rational ClearCase	20	2/1/01 (TBD available R.I. RCC)	CM System for the organization of both delivered and developed software
Defect Tracking	Rational ClearQuest	15	2/1/01	(Defect Data Tracking System) Software that will aid in requirements tracking by tracking software requirements as defects and relating these requirements to the CM system
Office Software	MS Office	1 for each PC	2/1/01	Used for requirements and design database as well as all documentation. Include for each user Word, Excel, Powerpoint, Access, Visio and Acrobat)

9.OPERATIONS SUPPORT

The flight software development systems (Software Only Test Bed [SOTB] & Software Test Bed [STB]) and associated ground support software will be maintained under Configuration Management (CM) and remain operational for the operational lifetime of the FAME. These systems will be used to continue development of planned flight software releases after the initial operational release.

These systems will also be available to support flight software anomaly resolution, flight software patch development and software patch qualification testing. All software patches and releases will be delivered as qualified and ready for flight operations.

NRL will maintain a staff of flight software developers and support staff as needed to support FAME operations for the duration of the mission. This support will be limited to short intervals of support at BP for critical operations; completion and delivery of flight software versions required after initial operational capability; and anomaly resolution.

10.NOTES

10.1 Acronyms.

Acronym	Definition
ADAC	Attitude Determination and Control
API	Application Program Interface
C&T, CT	Command and Telemetry
C^3	Ground Communications, Command, and Control
CCB	Change Control Board
CCN	Configuration Change Notice
CDR	Critical Design Review
CI	Configurable Item
CM	Configuration Management
CMP	Configuration Management Plan
CoFR	Certificate of Flight Readiness
COTS	Commercial Off-The-Shelf
CSC	Computer Software Component
CSCI	Computer Software Configuration Item
DDTS	Defect Data Tracking System
DID	Data Item Description
EAGE	Electrical Aerospace Ground Equipment
ELSE	Electrical Launch Support Equipment
FAME	Full-sky Astrometric Mapping Explorer
FDIR	Fault Detection, Isolation, and Recovery
FQT	Formal Qualification Test
FSW	Flight Software
FWL	Forward Link
GNC	Guidance, Navigation, and Control
GS	Ground Software
I/F	Interface
I/O	Input/Output
ICD	Interface Control Document
FAME	Full-sky Astrometric Explorer
IMU	Inertial Measurement Unit
FSC	FAME Integrated Spacecraft Controller
IV&V	Independent Verification and Validation
LMMS	Lockheed Martin Missiles and Space
LSWE	Lead Software Engineer
MO&DA	Mission Operations and Data Analysis
MRD	Mission Requirements Document
NASA	National Aeronautics and Space Administration
NCST	Naval Center for Space Technology
NDI	Non-Developmental Item
NRL	Naval Research Laboratory
OS/COMET™	Open Systems/Common Environment Test (A proprietary computer language)
PDF	Portable Document Format
PDR	Preliminary Design Review
PROM	Programmable Read Only Memory

Acronym	Definition
RCE	Reaction Control Electronics
RI	Responsible Individual
RID	Review Item Discrepancy
RTL	Return Link
RTSTD	Real-Time Structured Analysis and Design
S/W, SW	Software
SCL™	Spacecraft Command Language (A proprietary computer language)
SCM	Software Configuration Management
SCMGR	Software Configuration Manager
SDD	Software Design Document
SEMP	Systems Engineering Management Plan
SMP	Software Management Plan
SOTB	Software Only Test Bed
SQA	Software Quality Assurance
SRR	System Requirements Review
SRS	Software Requirements Specification
STB	Software Test Bed
STP	Software Test Plan
STR	Software Test Report
SUM	Software User's Manual
SW	Software
FSWAR	Flight Software Action Request
TIM	Technical Interchange Memo
TN	Technical Note
TRR	Test Readiness Review
V&V	Verification and Validation
VDD	Version Description Document
WBS	Work Breakdown Structure
XPN	Transponder

A. BASIS FOR FLIGHT SOFTWARE

A.1 NRL Specifications.

Document	Title	Basis for Flight Software				
		Req	Design	Code	Inspection	Test
NCST-D-FM001	FAME System Performance and Design (SPDR) Specification	X				
NCST-D-FM002	FAME Software Management Plan		X	X	X	X
NCST-D-FM004	FAME System Safety Program Plan	X	X	X	X	X
NCST-D-FM090	FAME Flight Software User's Manual				X	X
NCST-D-FM091	FAME Ground Software User's Manual				X	X
NCST-D-FM092	FAME Mathematical Specification for Ground & Onboard Ephemeris SW		X	X	X	X
NCST-D-FM096	FAME Telemetry and Command Report		X	X	X	X
NCST-D-FM000	FAME Attitude Control System Algorithm Description Document		X	X	X	X
NCST-ICD-FM001	Instrument to S/C ICD	X	X		X	X
NCST-ICD-FM002	MIL-STD-1553B Data Bus ICD			X	X	X
NCST-ICD-FM006	Radio Frequency ICD between FAME and the S-Band Transponder		X		X	X
NCST-ICD-FM008	FAME Flight Software External Interface Control Document (ICD)			X	X	X
NCST-RC-5001	Reaction Control Electronics (RCE) CIPS		X		X	X
NCST-SDD-FM001	FAME Flight Software Design Document			X	X	X
NCST-S-FM003	FAME Attitude Control Subsystem Prime Item Development Specification		X		X	X
NCST-S-FM004	FAME Command, Telemetry & Data Handling Subsystem Prime Item Development Specification		X		X	X
NCST-S-FM023	FAME Inertial Measurement Unit Critical Item Product Specification		X		X	X
NCST-S-FM024	FAME Star Tracker Camera Critical Item Product Specification		X		X	X
NCST-S-FM025	FAME S-Band Transponder Critical Item Product Specification		X		X	X
NCST-S-FM028	FAME Attitude Control Electronics Critical Item Product Specification		X	X	X	X
NCST-S-FM030	FAME System Controller Critical Item Specification		X	X	X	X
NCST-S-FM033	Digital Sun Sensor CIPS		X		X	X
NCST-S-FM038	FAME Remote Interface Unit Critical Item Product Specification		X	X	X	X
NCST-S-FM055	Command Decoder Unit Critical Item Product Specification		X	X	X	X
NCST-SRS-FM001	FAME Flight Software Requirements Specification		X		X	X
NCST-S-SD034A	Three Axis Magnetometer CIPS		X		X	X
NCST-STP-FM001	FAME Flight Software Test Plan (STP)				X	X
NCST-STP-FM002	FAME Ephemeris Propagator & Ground Navigation Software Prototypes Test Plan				X	X
NCST-STD-FM001	FAME Flight Software Test Procedures and Description (STD)				X	X
NCST-SVD-FM001	FAME Flight Software Version Description Document (VDD)				X	X
NCST-TP-FM041	FAME Attitude Control Subsystem Verification Plan				X	X

A.2 Military Specifications.

Document	Title	Basis for Flight Software				
		Req	Design	Code	Inspection	Test
MIL-STD-498	Military Standard Software Development and Documentation	X	X	X	X	X
MIL-STD-1553B, Notice 2	Aircraft Internal Time Division Command/Response Multiplex Data Bus	X	X		X	

A.3 NASA Specifications.

Document	Title	Basis for Flight Software				
		Req	Design	Code	Inspection	Test
531-TR-001	User Spacecraft Clock Calibration System (USCCS)		X		X	X
SSP-30309	Safety Analysis and Risk Assessment Requirements Document	X	X	X	X	X

A.4 Vendor/Commercial Specifications.

Document	Title	Basis for Flight Software				
		Req	Design	Code	Inspection	Test
Interface and Control Systems, Inc.	SCL Document Set, Volume 1		X		X	X
Interface and Control Systems, Inc.	SCL Document Set, Volume 2		X		X	X
ISO/ICE 9899	ANSI C Standard Programming Language			X	X	
Litton Memo #2.2/176	LN-200 Core IMU (LN-210) Failure Words		X		X	X
Litton-311875	LN-200 Inertial Measurement Unit, Litton Guidance and Control Systems		X		X	X
OCA-24050	Electrical Interface Control Document – Star Tracker Camera OCA AS		X		X	X
Wind River System	VxWorks User's Manual		X	X	X	X
Wind River Systems	VxWorks Programmer's Guide		X	X	X	X
Wind River Systems	VxWorks Reference Manual		X	X	X	X

A.5 Flight Software Internal Documents.

The following documents are used as guidelines and not applied as requirements to the SW development process.

Document	Title	Basis for Flight Software				
		Req	Design	Code	Inspection	Test
FAME-FSW-0001	FAME Software Development Coding Conventions			X	X	
FAME-FSW-0002	FAME Unit Testing Guidelines			X	X	X
FAME-FSW-0003	FAME Inspection Process Guidelines				X	
FAME-FSW-0004	FAME Inspection Review Packet				X	
FAME-FSW-0005	FAME Design Guidelines		X	X	X	
FAME-FSW-0006	FAME Safety Guidelines	X	X	X	X	X
FAME-FSW-0007	FAME Integration Guidelines			X	X	X

B. FLIGHT SOFTWARE STANDARDS CHECKLISTS

These checklists are used as guidelines and not applied as requirements to the SW development process.

Unit Design Inspection Checklist:
Does the design documentation clearly state the hazardous or safe nature of data modifications?
Does the design support controls for the identified hazardous data modifications?
Does the design documentation clearly state the hazardous or safe nature of the autonomous functions?
Does the design support controls for identified hazardous functions?
Does the design documentation clear state the hazardous or safe nature of the command?
Does the design support controls for identified hazardous commands?
Does the design documentation clear state the nature of any side effects?
Does the design define a safe initial state?
Does the design define a safe state for reset?
Does the design specify the error function for reporting error condition?
Does the design specify error conditions reported?
Does the design support a method to override input validity checking?

Unit Implementation Inspection Checklist:
The code is presented clearly
Header and Function comments are complete
Naming conventions have been followed
All design functionality is implemented
No extra side effects are introduced
No additional functionality is introduced
All TBD sections of the code are marked with #warning
Function call format checked for parameters
Proper access for global shared data checked
Proper mutex usage checked
Minimal critical section checked
Code has been checked for correct message and event selection as per the design
Code has been checked for correct message and event format as per the design
Code has been checked for correct message and event destination as per the design
Case statement checked for proper break statement usage
Case statement checked for complete definition of all normal cases
Case statement checked for complete definition of all error cases
Error handling checked
All errors are handled as per the design
Pointer creation/deletion order and paring checked
==, ||, && logic with =, |, & functions checked
() grouping for logic checked
Code has been checked that type casting is defined, not implied, in source

Unit Test Inspection Checklist:
UNIT_TEST is used correctly
All tests are within the UNIT_TEST block
Variable and parameter ranges checked
An acceptable range has been tested
The border conditions have been tested
All possible received messages are tested
All possible event responses are tested
Unexpected message reporting has been tested for at least one message.
Successful processing paths checked
Erroneous processing paths checked
All return points exercised
Resource cleanup path tested
Verify that the unit test checks design functionality that will be deeply embedded during further testing.
Verify that the unit test provides complete path coverage of the software (i.e. 100%) or notes any exception where the path is not tested.

Component Integration Test Inspection Checklist:
Force errors during command processing and commanded state transitions.
Generate spurious interrupts and verify that they do not affect software behavior
Exercise all operational commands
Force maximum rate of event occurrence
Generate commands at maximum rate from all sources
Generate all detected faults one at a time and concurrent multiple faults and verify the software behavior
Exercise sample functional tests on all applications
Test functions that can be invoked a second time before the previous invocation has been completed
Identify all instances in which a functional application has portions of its function in multiple tasks, assure correct execution sequence occurs in all nominal and off-nominal scenarios
Exercise operation of all hazardous commands
Force instantaneous and steady time update and verify software behavior

C. MIL-STD-498 DIDS AS TAILORED FOR FAME