Navigation and Mission Design Branch
(NASA GSFC, Code 595)

General Mission Analysis Tool (GMAT)

Product Plan (PP)

OCP Version: *Draft.*

OCP Date: *Oct. 2011*

_____ _____

Steven P. Hughes / GMAT Product Lead — Date
Navigation and Mission Design Branch

Approved by:

_____ _____

Mark Beckman / 595 Branch Head — Date
Navigation and Mission Design Branch

_____ _____

Sue Hoge / Director — Date
Flight Dynamics Facility

_____ _____

Cheryl Gramling / Chief Navigation Engineer — Date
Navigation and Mission Design Branch

Reviewed by:

_____ _____

Isaac McGinnis / Software Assurance Engineer — Date
Mission Support Division

Insert additional signatures of stakeholders as needed to document their commitment to this plan. **NOTE:** Approval authorities MUST be specified by name and title.

PLAN UPDATE HISTORY

Version	Date	Description	Affected Pages
1.0	TBD	Initial Baseline Version	All

Table of Contents

Introduction

This document is the GMAT Product Plan (PP), referred to simply as the Plan, for the *General Mission Analysis Tool (GMAT)*. This plan is tailored for operational certification of GMAT.

The major goals for this document are:

1. To describe what products will be delivered
2. To define **who** is responsible for producing the products
3. To describe the baseline **schedule** for completing the effort
4. To specify the estimated resources needed (**cost** and/or effort)
5. To specify the items needed from the customer
6. To describe **how** and **where** the work will be carried out
7. To reach a mutual understanding and agreement with the customer and other stakeholders on the items listed above.

This Plan represents the formal agreement between the Product Development Team (PDT), branch management, and customer management on the scope of the development effort, its cost and schedule, and the manner in which the effort will be managed.

Objectives of This Effort

To demonstrate that:

- GMAT is ready for use by NMDB as a primary mission design tool for support from concept through handoff to FDF or MOC.
- GMAT is ready for use in the FDF by operational missions as a primary mission analysis and maneuver planning tool.

To show this, we will:

- Complete GMAT system and acceptance testing in an analysis environment.
- Perform shadow operations of selected missions in an operational environment.
- Provide all end-user documentation required by GPR 7150.2.
- Pass the Operational Readiness Review (ORR).

GMAT Overview

The General Mission Analysis Tool (GMAT) is a software system designed to model and optimize spacecraft trajectories in flight regimes ranging from low Earth orbit to lunar applications, interplanetary trajectories, and other deep space missions. GMAT has been used to optimize maneuvers for flight projects such as NASA's LCROSS, ARTEMIS, and LRO missions, and for optimization and analysis for the OSIRIS and MMS projects.

The software is licensed using the NASA Open Source Agreement. GMAT is written to run on Windows, Linux and Macintosh platforms, using the wxWidgets cross platform UI Framework, and can be built using either commercial development tools or the GNU Compiler Collection (GCC). The system is implemented in ANSI standard C++ (approximately 380,000 non-comment source lines of code).

GMAT has several user interfaces including an interactive GUI and a textual script interface. The MATLAB interface is a secondary textual interface for running the system via calls from GMAT to MATLAB and allows GMAT to call MATLAB functions from within the GMAT command sequence. An interface currently under development is a C language API for low level calls into the GMAT engine.

The Navigation and Mission Design Branch at NASA's Goddard Space Flight Center performs project management activities for GMAT and is involved in most phases of the development process. The Ground Software Systems Branch and Flight Software Branch perform design, implementation, and integration testing. Current and past contributors include:

- Thinking Systems, Inc. (system architecture and all aspects of development);

- Air Force Research Lab (all aspects of development);
- a.i. solutions (testing);
- The Schafer Corporation (all aspects of development);
- Boeing (algorithms and testing);
- Honeywell Technology Solutions (testing);
- Computer Sciences Corporation (requirements);
- NASA Jet Propulsion Lab (funded SPICE integration);
- European Space Agency (ESA) (Optimizer Plugins)

Document Overview

Section 1, Introduction, presents introductory material, an overview of the software system to be developed, and a list of references used in developing this Plan.

Section 2, Customer Agreement, summarizes the customer requirements, deliverables, and other mutually agreed upon aspects of the relationship between the customer and the software project.

Section 3, Management Approach, describes how the process will be managed, including the work to be performed, planned resources and schedule, initial risks, and project measures.

Section 4, Technical Approach, describes the technical approach to developing, delivering, and maintaining the software products.

Section 5, Product Control, describes how configuration management and data management will be performed and how the quality of the software system will be assured.

Acronyms and Abbreviations

CMMI Policy Compliance Information, describes additional information needed for CMMI

compliance.

NASA Policy Compliance Matrix, documents the compliance of this document to GPR 7150.2 requirements for In-House Software Development projects.

Scope of this Effort

What this effort will deliver

The primary deliverables for this effort are:

- Operationally certified Windows application
- Requirements specification
- Test plan
- Test procedures
- Test results
- Bidirectional requirements to test matrix
- User's guide (with Version Description Document)
- NTR forms
- Delivery Letter

Other deliverables and work products created by this project, such as station reports, risk lists, and similar items are discussed in the Project Monitoring and Control section.

What this effort will not deliver

This project will NOT deliver final versions of the following items. However, draft documents will be available.

- Mathematical specification
- Design specifications

Document Review, Approval, and Update

The plan described in this document was developed by the Product Development Lead (PDL) and reviewed and approved by signatories listed on the document's signature page. This document presents a snapshot of planning information that is current at the time of signature. The PDL will maintain detailed cost, schedule and other planning information throughout certification processes. The planning information in this document will be updated when all signatories agree there is a change in cost, schedule or scope sufficient to merit a re-plan of the effort.

Upon approval, this document will be placed under configuration control. Approved changes will be listed in the document's Plan Update History located immediately after the signature page.

Acceptance Criteria

Customer acceptance of the system will be based on the system meeting the following acceptance criteria:

- All acceptance tests have been formally executed and witnessed by customer
- All acceptance test results have been peer reviewed and passed
- All critical or urgent problem reports are closed or have customer-accepted work-arounds in place

Management Approach

This chapter describes the management processes used by the GMAT project. The sections in this chapter are organized roughly by CMMI process areas with additional sections such as Communications and Safety and Security that are not covered by CMMI.

Objectives and Priorities

Name	Deliverables/Task Areas

Production Release	● GMAT Application
	  ○ Libraries
	  ○ Support Files
	● Requirements
	● Interface Spec
	● Script Tests
	● GUI Tests
	● Critical new features
	● Bug Fixes
	● Test Plans
	● Test Procedures
	● Script requirements to test matrix
	● GUI test matrix
	● User's Guide (with Version Description Document)
	● Delivery Letter
	● NTR Forms

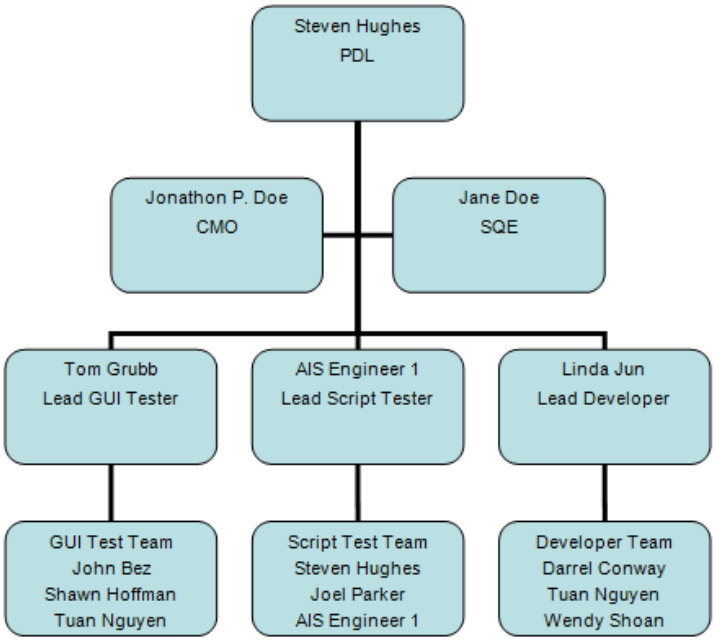
	• Product Plan
Missions	• MMS • OSIRIS • MAVEN • CHOPPER
IRAD	• DE Files • IAU Coord. Sys. • API • Mars-GRAM • Relativistic Corrections • Barycentric Propagation
Collaboration	• KARI • ESA • NRC Presentation • ST Presentation • ICATT Talks/Demos • Public Process Docs
595 Tools	• NITS • OODT • ODTBX

Project Planning

Product Development Team

The GMAT product development team is an integrated team of contractors and civil servants.

The team is composed of area leads, engineers, developers, testers, and support personnel, who together are referred to as the PDT. As members of the integrated PDT, project personnel use the approaches and processes described in this plan.



Roles and Responsibilities

Roles and responsibilities for GMAT project members are defined below and broken down by The hierarchical structure of the organization chart shown in the previous section illustrates the levels of accountability within the PDT, with increasing responsibility and authority assigned from the lowest level to the highest level of the project. The chart also shows assignment of a role for each member of the PDT. The following table describes each PDT role:

Managerial Roles

Role	Acronym	Role Description
Product Development Lead (S)	**PDL**	● Vision ● Prioritization ● Funding ● Task management
Project Manager	**PM**	● Process definition (NPR, CMMI) ● Planning Scheduling ● Project Monitoring & Control of overall project ● Measurement & Analysis of the overall project ● Acquisition Management
External Partnerships Manager (J,S)	**EPM**	● Write agreements ● Work with HQ ● Community development ○ Wiki ○ Forum ○ Mailing lists ○ Bug system
Development Team Lead	**DTL**	● All developers are senior and this role has not been an issue.
GUI Test Lead	**GTL**	● GUI Validation Testing
Script Test Lead (S)	**STL**	● Product Monitoring & Control of Script Verification Test ● Script Validation Testing
User Documentation Lead (T)	**UDL**	●
Configuration	**CMO**	● Configuration Management

Management Officer		
Software Assurance Engineer	**SAE**	● Process & Product Quality Assurance ● Process Audits ● Product Audits
Feature Lead* (J,S,Lots)	**FL**	**Activities** ● Manage feature finalization ○ Maintain work items estimates ○ Work with PM to schedule ○ Feature spec or user's guide input ● Track new and existing bugs ● Report on feature status ● Present feature at ORR **Deliverables** ● Requirements ● Work items ● Bug action recommendations ● Estimates and basis of estimate

Technical Roles

Role	Acronym	Responsibility
Systems Engineer* (S)	**SE**	**Activities** ● Finalize requirements ● Finalize test summaries for unique feature characteristics ● Review specs and/or user's guide **Deliverables** ● Requirements inspection report

		● Feature spec inspection report ● Test summary inspection report
Flight Dynamics Engineer*	**FDE**	**Activities** ● Finalize requirements ● Write math/algorithm spec ● Finalize test summaries for unique feature characteristics ● Write interface description in feature spec **Deliverables** ● x`
Software Engineer	**SwE**	● Finalize requirements ● Finalize test summaries for unique feature characteristics ● Write interface description in feature spec ● Implement new or missing features ● Remove deferred features ● Fix bugs
Test Engineer* (J,S)	**TE**	● Develop test summaries for unique feature characteristics ● Diagnose user issues ● Write tests ● Submit defect reports ● Verify bug fixes
Technical Writer	**TW**	● Write reference material ● Write Tutorials ● Write How To docs
Technical Support*	**TS**	● Answering forum questions ● Helping branch members use GMAT

Work Breakdown Structure

The GMAT Work Breakdown Structure (WBS) is deliverables-based and is combined with the Data Management List. The top two levels of the WBS at the time of writing are shown below. The WBS is used to estimate and track project cost.

1	**Project Management**
1.1	Project Planning
1.2	Project Monitoring and Control
1.3	Configuration Management
1.4	Software Assurance
1.5	Stakeholder Coordination
1.6	Acquisition Management
2	**Engineering and Development**
2.1	System Definition and Design
2.2	Requirements
2.3	Design
2.4	Implementation
2.5	Test/Debug
2.6	Release

3	**Infrastructure**
3.1	People
3.2	Hardware
3.3	Software

Estimated Resource Requirements

The required resources documented in this section have been negotiated with the customer based on initial planning of manpower needs and other costs. On an ongoing basis, the PDL will negotiate customer-funded resource needs and costs and resolve budget issues for this effort with the customer. The cost estimates provided in this Plan was based on using the current software, developed over many years, and the WBS to certify the software for use by the FDF. The details of this cost estimate are documented in the Basis of Estimate (BOE).

Estimated Staffing Requirements

PDT staffing requirements were derived using the WBS described in the previous section. An integrated team of both civil servant and contractor personnel are planned to comprise the PDT.

Staffing requirements contain proprietary and other Sensitive but Unclassified information is maintained in separate document.

Other Resource Requirements

The other required resources for this effort are shown in the table below.

Item	**Cost**
STK	??
MATLAB	??

Test Complete	$1300/yr
Promo material (cups, shirts, etc.)	$2500/yr
Snagit	$150
UML Editor	5x$100
Windows Test Machines	2x$2000
Build Machine	$2000
Camtasia	$320
Hosting Space	$600/yr
Icons	$500/yr
JIRA	$0.0
Forum	$0.0
Wiki	$0.0
Compilers (VS/GCC)	$0.0
Mac Test Machine	$0.0

*Items highlighted in Green have been purchased.

Basis of Estimates

Estimates for GMAT work are based on the WBS and Wideband Delphi estimation process. The Wideband Delphi estimation template is a google document linked from the Project Dashboard. The process description can be found at in the GSFC PAL at

http://software.gsfc.nasa.gov/AssetsApproved/PA1.2.1.2.doc. For more information on the Wideband Delphi process see the treatment by Stellman and Greene.

High Level Schedule

We will perform QA for GMAT 2013a (Production release) using 6 intermediate milestones by grouping features by similarity and degree of interaction. The areas are:

- FG1: Basic Dynamics and Modelling
- FG2: Powered Flight
- FG3: Solver Infrastructure
- FG4: Control Flow/Math/User Variables
- FG5: Output
- FG6: At Risk

The breakdown by feature group is shown below with owners for features for FG1. As QA proceeds, owners will be assigned to features for FG2 and so on.

Feature Group 1 Subteam and Features

Lead: SPH
SDE: WCS
STE: SJH
Exceptions: DJC is SDE for FRR-10,FRR-14, FRR-13

Req. Id	Milestone
FRR-1 Spacecraft Orbit State	M1
FRR-2 Spacecraft orbit epoch	M1

FRR-13 Dynamics models	M1
FRR-14 Numerical integrator	M1
FRC-10 Propagate command	M2
FRC-4 Spacecraft mass properties	M2
FRC-7 Spacecraft attitude	M3
FRR-34 Coordinate system	M3
FRR-16 Solar system	M4
FRR-17 Libration point	M4
FRR-18 Barycenter	M4

Feature Group 2: Powered Flight

Lead: DSC
SDE: DJC
STE: SJH

Req. Id	**Milestone**
FRR-8 Tank model	M1
FRR-9 Thruster model	M1
FRR-11 Impulsive burn	M2
FRR-12 Finite burn	M2

FRC-9 Maneuver command	M2
FRC-8 Finite burn commands	M2
FRR-?? Spaceraft Hardware	M2

Feature Group 2: Solver Infrastructure

Lead: SPH
SDE: DJC
STE: TGG
Exceptions: DSQ is lead for

Req. Id	**Milestone**
FRR-19 Differential Corrector	M3
FRR-3 Target command	M3
FRR-5 Achieve Command	M3
FRR-2 Vary Command	M3/M4
FRR-21 VF13ad Optimizer	M4
FRR-20 fmincon Optimizer	M4
FRR-1 Optimize Command	M4
FRR-6 Minimize Command	M4
FRR-7 Nonlinear Constraint Command	M4

FG3: Solver infrastructure

- Differential corrector
- Fmincon optimizer
- Vf13ad optimizer
- Target command
- Vary command
- Achieve command
- Optimize command
- Nonlinear constraint command
- Minimize command

FG4: Control Flow/Math/User Variables

- Variable
- Array
- String
- MATLAB function
- If command
- Else command
- While command
- If-Else command
- Assignment command
- MATLAB function call
- Script Event

FG5: Output

- ReportFile
- OrbitView

- GroundTrackPlot
- EphemerisFile
- XYPlot
- Report Command
- Toggle
- PenUp/PenDown
- MarkPoint
- GroundTrackPlot
- EphemerisFile
- Spacecraft visualization properties

FG6: At Risk/Unplanned

- Attitude state
- Visualization properties
- Formation
- Stop
- Eclipse Locator
- Global
- Ground station
- GMAT Function
- Call GMAT Function
- Orbit Designer
- SPK Propagator
- Save
- Socket interface
- Matlab interface
- Matlab server interface
- C-API

2012a Beta

- Version 1.0 Analysis Candidate (spring12 Time box)
    - Test Plan review Dec-2011
    - Test Procedure review Dec-2011
    - Training manual peer review Nov-2011
    - Shadow ops requirements review Jan 2012
    - SRS peer review (ongoing)
    - GUI test peer review
        - Resource Tree (Feb. 2012)
        - Mission Tree (Apr. 2012)
        - App control/misc tested (Mar. 2012)
    - Script test peer review
        - Resources
        - Script
    - User's Guide peer review (May 2012)
- Version 2.0 Operational Candidate(Dec. 2012)
- Operational Readiness Review(spring 13)
- Operational Certification(spring 13)

Work Order Authorization

The WOA equivalent document for this effort is provided by this Plan. The Signature Page documents appropriate approvals.

Project Monitoring and Control

Detailed Schedule and Estimates

Detailed estimates are generated using the Wideband Delphi process performed at the beginning of each new milestone.

1. **Define Goals/Deliverables:** Define the scope of the work and what specifically will be delivered. Goals are for the work area and should not be tailored to simply to what you think can fit in the next milestone. Work areas, by definition, are defined to be a 1-3 milestones in scope. Exit Criteria: All participants agree on the goals and deliverables.
2. **Develop Work Items**: Area lead generates initial work items list based on near term goals. Work with estimators to clearly define each work items so that all estimators understand the scope of each item and that all required work items are included in the estimate. The product of this phase is a list of work items with detailed descriptions for each item. Before proceeding to step 2, ensure that all involved agree the work list is complete, and that everyone understands the scope of each work item. Exit Criteria: All participants agree that 100% of the work has been identified.
3. **Create Estimates**: Individuals create their estimates for individual work items. The estimation team works with the lead to rectify estimates and agrees upon an actual estimate. Exit Criteria: You have a agreed-upon estimate with buy-in from the person responsible for the work for each work item.
4. **Examine Work Scope/Prioritize**: Compare the total estimate with the desired next milestone/due date and other work loads on the participants. Work with area lead and/or PDL to determine how to schedule milestone/due date. When will the work be complete? Should we descope if necessary? Should we extend the schedule?
5. **Schedule Items in JIRA**. Add the work items to JIRA mark against the milestones/due dates agreed upon in step 3.

The detailed schedule is generated and tracked using JIRA burn down charts by milestone and next release. The following are JIRA best practices:

- If you are working on something being charged to GMAT, it should have a work item in JIRA marked against a specific milestone.
- If team members will be out of the office, a JIRA task is created to track that time.

- Estimate about 20% of each individuals time in any given milestone is spent on overhead: Goddard training, reporting, time cards, meetings, task switching etc.,
- At the end of the work day, update your progress on the work items in JIRA.

Bug Triage

Bugs are addressed in order of criticality where:

- Blocks mission customers or team processes such as testing.
- Critical
    - Crashes
    - Numeric errors
- Affects all or many users.
- Affects significant number of users and there is no workaround

Issue Triage

There is always work that needs to be done that doesn't map directly to goals of a current milestone. Here is how we handle triaging those issues.

- Team members triage issues they have submitted at the beginning of each milestone.
- From [Berkun, 2005]: "Anything that does not affect the rate of completion of P1 work items should never take priority over the things that do." We'll make an exception: if you can convince a majority of the team that something is important that is not on the schedule, we'll put it on the schedule.
- Present items you would like to see added to the schedule at the first team meeting in a new milestone.

Issues Tracking

Issues for the GMAT project are defined as items that require CCB or team evaluation or approval, or require elevation to management or customers. Issues are logged and maintained in the Issues Tracking Tool. Issues include but are not limited to process problems, classic mistakes, and tool or hardware problems that cause systemic problems. Note that software defects are NOT considered issues and are tracked as described in the Configuration Management section of this document.

Action Items

Action items for the GMAT project are defined as short (<=2 hours of work), miscellaneous work items that do not fall under other work categories such as design, implementation, risk management etc. Action items are recorded in the Action Items Tool and tracked at weekly meetings.

Meeting Minutes

Meeting minutes for weekly, CCB, and branch status meetings are maintained in Google docs and contain required and present attendees, agenda, discussion items and actions. All meeting minutes are written by starting from the Meeting Minutes Template.

Status Reporting

The following will be reported at monthly Branch Status Review meetings and also provided via email. A template will be developed, modeled after the Branch Status Review template in the GSFC PAL, but with changes to reflect the unique character of the GMAT project. BSR will be documented via meeting minutes and stored as indicated in the WBS/DML.

- Management Review
    - Activities
    - Status
    - Risks
    - Issues
- Technical Review
    - Development
    - Testing
    - Documentation
- Metrics

Lessons Learned

GMAT project lessons learned are recorded here. Any team member can submit a lesson learned at any point in the project and regarding any aspect of the project from process to technical issues.

Project Communications

Project Dashboard

Forums

User Help
    Novice User Help
    Flight Regime/Mission Physics Questions
    Modelling/Analysis
    Resources/Commands
    External Interfaces/Plugins
Development

Compiling GMAT
Design
Implementing New Functionality
Testing

Wiki

Mailing Lists

Stakeholder Involvement

Stakeholder identification

The following individuals and roles are GMAT stakeholders:

- FDF Director and Associate
- 590 Managers
- 595 Branch Head and Associates
- 585 Branch Head and Associates
- FDF Analysts
- 595 Branch Members
- SSMO Management
- GMAT Team

Stakeholder Notifications and Reporting

This section describes the planned participation of customers and other key stakeholders in supporting this effort. Signatures on this Plan constitute commitment by the customer, other key stakeholders, and the PDL to participate in activities as described in the section.

The PDL plans what the customer and other key stakeholders will be involved in this effort as described in WBS/DML/Stakeholder Involvement Plan. The PDL will use this table to facilitate stakeholder involvement. For example, the PDL will use this table when creating or revising distribution lists to ensure that PDT-generated materials are made available to appropriate stakeholders. The PDL will monitor stakeholder involvement by methods such as recording attendance of expected attendees in meetings minutes and by noting participation of stakeholders in other activities listed in the table. The PDL will work with the customer to take corrective action if there will be an impact to the effort because the involvement of stakeholders is not meeting the plan. The current version of the WBS/DML is provided here.

Risk Management

The PDL is responsible for risk identification and monitoring and will be supported in this activity by other members of the PDT. Throughout the effort, potential risks including cost, schedule, technical, and process risks will be identified. All identified risks to the effort will be tracked and watched for impact or for change in status until retired. As needed, mitigation plans will be developed and implemented. The PDL and other members of the PDT will use the Risk Management Tool to document and, on a regular basis, monitor risks and associated mitigation plans. At a minimum, the status of all identified, open risks will be presented at both management and technical reviews. See the index page of the Risk Management Tool for complete information on the use of the tool, probability, impact, identification, classification, mitigation, state, status and reporting details.

Risk status shall be reported at monthly Branch Status Reviews and major milestone reviews.

Initial Risks

The following is an initial set of Risks that were identified for the project. The latest detailed information on these Risks and new Risks can be found in the Risk Management Tool located here.

Risk ID	Initial Risk Description
2	New development with significant re-factorization of the system will occur during certification process. (IRAD and key new functionality)
?	Difficult to estimate time to fix bugs because testing is not complete so we don't know all issues.
5	Requirements have not been rigorously reviewed by FDF and Branch and so customers don't know precisely what will be delivered.
?	Unclear how needed but unsupported functionality will be developed after certification. We need a plan in place that identifies what functionality is not supported that is required by missions.
?	No test plan will find all issues and users will find things our approach doesn't find. We need beta testers for the GUI.
?	Testing on Windows 7, Vista, and XP (64 and 32 bit) is challenging. Need to determine if all of these must be supported.
3	GMAT project doesn't have requirements for successful shadow operations for specific missions.

Bulleted items below should be removed after comments are resolved.

- Unclear how needed but unsupported functionality will be developed after certification. We need a plan in place that identifies what functionality is not supported that is required by missions.
- No test plan will find all issues and users will find things our approach doesn't find. We need beta testers for the GUI.

Identification, Management, Reporting

Measurement and Analysis

Project Metrics

The following metrics will be used to track progress for Production Release:

- P1 Bugs Created vs. Resolved
- P2 Bugs Created vs. Resolved
- Percent tests passing
- Specs complete
- User docs complete
- Test complete

Maintenance and reporting for Bug metrics is the responsibility of the Software Development Engineers. Maintenance of Percent test passing is the responsibility of the test team. Maintenance and reporting of the Spec metrics is the responsibility of the Feature Group leads.

Analysis and Reporting

Safety and Security

Occupational Safety

[Describe how the PDL will comply with GPR1700.1]

Software and Safety Classification

Each GMAT software subsystem or component has been classified into one of eight classifications depending on how critical that subsystem or component is to the safety and operation. The software classifications, as specified in NPR 7150.2A (NASA Software Engineering Requirements) are defined below.

Description	Contractor	Software Classification	Safety Critical
Customer Interface Layer	a.i.solutions	Class B	No
Front End Communications Processor	a.i.solutions	Class B	No
Modular Data Processors	a.i.solutions	Class B	No
Data Packager/Viewer	a.i.solutions	Class B	No
System Monitor	a.i.solutions	Class B	No
Control and Automation	a.i.solutions	Class B	No
Data Product Generation	a.i.solutions	Class B	No
Logging	a.i.solutions	Class B	No
Process / Application Recovery	a.i.solutions	Class B	No
Product Center Re-engineering	a.i.solutions	Class B	No
Oracle Database Upgrade	a.i.-solutions	Class B	No

Each of the eleven GMAT software components has a Software Classification of ‘Class B’.

Each component has been examined for Safety Criticality and found not to be safety critical.

Security and Privacy

[Describe how the PDL will comply with of NPR 2810.1C, NASA Information Security Policy; NPR 2810.1A, NASA Security of Information Technology; and GPR 2810.1, GSFC Security of Information Technology.
The software project manager is responsible for requirements associated with the Security of people and NASA assets, Security of information technology, Proper export of controlled hardware, technology, and data (including software), and Involvement of partners, contractors, and citizens of foreign countries.
If the project has a separate Security Plan, reference it here. Otherwise, describe the project's plans for addressing security and privacy considerations, both physically for the facilities involved and electronically for any computer systems being used either for development and testing or as a part of the final product. Address each of the items in the bulleted list above. Also provide input to the Mission Project Security Plan, as needed to comply with NPR 2810.1, Security of Information Technology, located at http://nodis.hq.nasa.gov/.]

Technical Approach

This chapter describes the technical processes used to develop GMAT and is tailored for operational certification of the system. The Development Strategy section contains high level discussions of the project life cycle including major development milestones and artifacts and associated life-cycle reviews. The Verification and Validation section describes the test team organization and roles and responsibilities; the processes used to verify requirements for version 1.0; the process used to finalize software user interfaces; testing methodologies; and processes for defect resolution. The Analysis Environment Testing section describes how GMAT will be tested by mission analysts in the Navigation and Mission Design Branch. The Operational Environment Testing section describes how GMAT will be tested in a shadow operations setting in the Flight Dynamics Facility (FDF). The contents of the Operational Readiness Review and the plan for transitioning to operational use are discussed in the last two

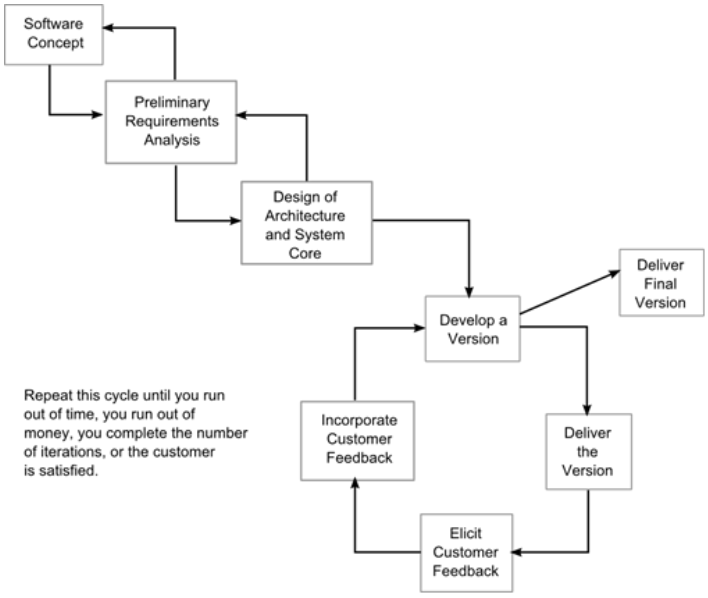
sections of this chapter.

Development Strategy

Project Life-Cycle

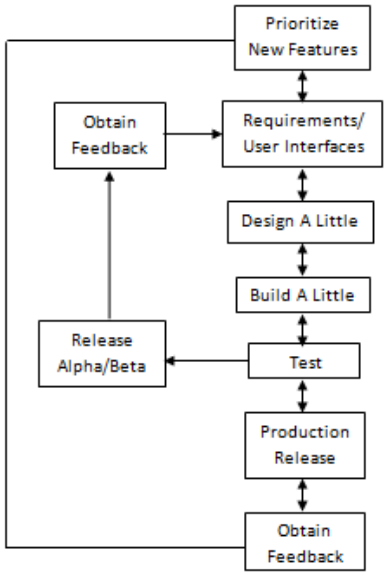
The GMAT project uses a modified Evolutionary Delivery process as its life cycle model (McConnell, 1996). Evolutionary Delivery (EDM) is an incremental development model well suited to situations where it is not possible to know all system requirements up front. Success of EDM depends upon identifying primary functionality early and ensuring that the design of the system's core architecture anticipates as many requirements as possible.

A diagram of McConnell's Evolutionary Delivery model is shown in the Figure below (GMAT uses a tailored build cycle, discussed in the next section, which differs significantly from the original model). There are two primary stages in Evolutionary Delivery. In the first stage, consisting of Software Concept, Preliminary Requirements Analysis, Design of Architecture and System Core, the driving requirements are identified and the architecture is designed. This phase of the GMAT project was completed in 2003. The second life-cycle stage is a tailored iterative build/release cycle that produces incremental builds. This build/release cycle is discussed in detail in the section below.



Build/Release Cycle

The GMAT build/release cycle consists of a series of Major and Minor iterations that use a “build a little, test a little” philosophy. The major iteration cycle is approximately six months long and consists of 2-4 minor iterations of the process as illustrated in below. Minor iterations produce beta software with incremental feature enhancements and the beta versions are provided to users in NMDB for early evaluation and feedback. The major iteration produces an operationally certified build by using the same V&V processes described later in this document and in the GMAT Test Plans and GMAT Test Procedures documents (i.e. an ORR is not required at the end of each major iteration) . While the goal is for a major iteration to take about 6 months and minor iterations to take about about 4-6 weeks., the actual duration of iterations varies depending upon the complexity of the features being developed.



A major iteration begins by reevaluating feature priorities based on user feedback, bug fixes, and new and enhanced requirements from the previous cycle. Next, detailed requirements and user interface design prototypes are developed for high priority features. At this point, a minor iteration begins and design is performed to implement a subset of the highest priority features. At any given time, new features may be at different states of the development cycle. However, at the completion of a major cycle, all features that are either incomplete or of insufficient quality are removed and are completed in the next major cycle.

Note that though Test is listed as a separate action, regression tests are performed nightly (or at least twice weekly) to verify changes do not break other areas of the product or that breakages are tracked in the bug database. The Test cycle refers to testing the changes as well as performing broader end-to-end testing.

Workflows for Routine Development Activities

Most new functionality added to GMAT falls into one of two basic categories: Minor modifications to an existing Resource or Command; or adding a new GMAT Resource or Command object. Less frequently, modifications may be made to GMAT's core infrastructure to support new capabilities. Sections below describe the work flow for common development activities.

Modifying an existing Resource or Command

Owner	Activity
Feature Lead	Draft, review, and finalize the Minor Enhancement template. This document describes requirements, use cases, user-interface modifications and other issues associated with modifying an existing GMAT resource or command and provides the software developer with what they need to implement the feature enhancement.
Software Engineer	Design and implement, and perform developer testing, for the new feature in a local code repository. If needed iterate on the feature by revisiting the Minor Enhancement template.
Software Engineer	When development of the new feature is complete, notify feature lead and tester that new code will soon be committed. (See the PBS for list of feature leads and testers by feature area).
CCB	Update the GMAT Software Requirement Specification with the finalized requirements for the implemented enhancement. *This is the most important step because the SRS is used in the automated regression test system for requirements to test traceability and QA*.
CCB	Update the GMAT User Interface Specification with new interface

	changes. *This is the second most important step, because the interface specification is used in the automated regression test system for interface element to test traceability and QA*.
Software Engineer	Commit the code.
Test Engineer	Write and execute functional and system tests.
Software Engineer	Fix defects identified as must fix.
Test Engineer	Verify tests in the automated test matrices.

Adding a new Resource or Command

To be written.

Major Activities and Deliverables

The major work products, including documentation, data, and code, as well as stakeholder involvement for each phase are documented in the project WBS. The table below briefly summarizes the major activities and deliverables for each phase in a Major iteration. **See the project WBS for a complete, up-to-date list of deliverables including stakeholder involvement and configuration level for each item.**

Phase	Primary Deliverables
Prioritize new Features	● Prioritized feature list ● Votes for new feature suggestions
Requirements/	● Objectives/Requirements/Scope section in the feature spec.

User Interfaces	● User Interface Spec section in the feature spec. ● Mathematical/algorithm specs ● Review/Inspect results
Design	● Updates to design specification ● Basic test scripts/procedures in feature spec ● Review/Inspection docs
Implementation	● Source code ● Data files ● Unit tests ● Integration tests ● Test results ● Review/Inspection docs
Test	● Test plan updates ● Test procedure updates ● Test reports ● Traceability matrix updates ● Bug reports
Release	● Application and installer ● Version description ● User's guide ● NTR
Feedback	● User feedback emails ● Feedback and evaluation forms

Life-Cycle Reviews

The GMAT project has two types of Reviews/Inspections. The first type are major reviews such as the Operational Readiness Review (ORR). The contents of major reviews follow the guidelines from the Software Engineering Laboratory's *Recommended Approach to Software Development.* The review panels are composed of external subject matter experts and major stakeholders. RFAs generated from reviews are tracked to closure as described in the Issue Tracking section of this plan.

The second type of review/inspection is less formal and takes place between steps in the minor iteration cycle. These reviews ensure work products generated in one phase are sufficiently complete to begin work in the next phase. Issues found during these reviews are tracked as either work items (the projects to-do list) or as described in the Action Items section of this plan.

Make/Buy Approach

Early make/buy studies and decisions were made during the concept definition and requirements development phases before the writing of this plan and are covered in other GMAT documentation.

Integration of Customer-Supplied and Acquired Products

Third party components can be integrated into GMAT using the custom plug-in interface described in the GMAT Architectural Design Specification. The GMAT Technical Lead and PDT are available to help users integrate custom plug-in components into the user's mission environment.

Rights and Approvals

GMAT has been approved for open source release by the GSFC Software Release Authority in 2007 and 2011. The GMAT PDL ensures that appropriate approvals are obtained for any new technology or software that is incorporated into GMAT or that the technology is licensed using agreements compatible with the NASA Open Source Agreement (NOSA). Currently, the known compatible licenses are the Academic Free License and the BSD license.

Technology and Commercialization Plan

GMAT is released annually through the New Technology Report system by submitting a NASA Form (NF) 1679, Disclosure of Invention and New Technology.

Development and Test Environment

This section describes the facilities, equipment, libraries, and tools the PDT will use to develop and test the system.

Development and Test Facilities and Equipment

GMAT is desktop software and no specialized development or test facilities are required. For civil servant team members, development and testing is performed on team member's individual computers provided by GSFC infrastructure. For contractor team members, development and test activities are usually performed on machines provided to those members under terms of the FDSS contract. In some instances, additional work stations are provided when activities such as building and regression testing require devoted machines. Operational certification will be performed in the FDF on similar workstations that are connected to the FDFNet.

Development and Test Tools

GMAT is written in ANSI C++ and is compiled using the Gnu Compiler Collection (GCC) on Windows, Mac, and Linux, and using Microsoft Visual C++ 2010 on Windows. Nightly builds and test versions and Release versions are built using Microsoft Visual C++ Express 2010 because performance comparisons have shown that builds from that compiler are 30% faster than builds from GCC.

The following external libraries are required to build the core GMAT application:

- GMAT source code
    - SourceForge repository
    - JAZZ repository

- JPL SPICE libraries
- WxWidgets GUI libraries
- f2c

The following libraries are required for commonly used, but optional, plug-ins

- MATLAB
- MATLAB optimization toolbox
- Harwell Subroutine Library VF13ad optimizer

The following tools are required for GMAT testing:

- TestComplete (by AutomatedQA)
- MATLAB
- Excel

Verification and Validation

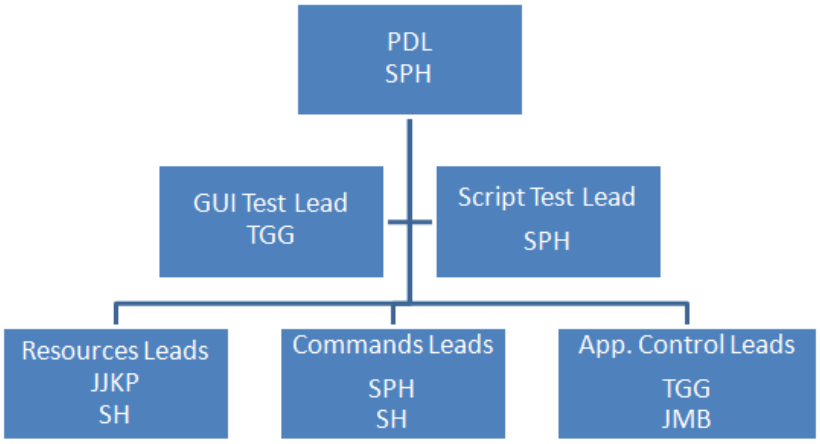
Overview

This section describes how the GMAT test team will perform verification and validation. The purpose of the verification and validation effort is to prepare GMAT for testing in analysis and operational environments so that few, if any, unknown issues are found during FDF shadow operations. For V&V purposes, GMAT is broken down by requirements groups into approximately 115 feature areas: at the time of this writing, 64 Resources features, 29 command features, and 23 application control features.

Verification and validation will occur on a feature-by-feature approach where the steps in the V&V process are Requirements Verification, User Interface Specification, Testing, Defect Resolution, and User Documentation. The PDL, or his designee, shall evaluate and record the results of all verification and validation activities and ensure that corrective actions are tracked to closure. Each of these steps is described in detail in the sections below.

Most GMAT features have two primary interfaces; a script interface and a GUI interface. The organizational structure of the test team is broken down by feature area and interface type as

shown below. Owners of individual feature areas are not shown in the diagram and can be found in the Product Breakdown Structure. The roles for each test team member are described below the organizational chart.



Test Team Roles and Responsibilities

Role	Responsibilities
PDL	• Provide high level feature prioritization to ensure critical features are tested early so non-critical features can be removed from the system if needed to remain on schedule. • Verify requirements

GUI Test Lead	• Manage the GUI test effort • Write test plans • Write test procedures • Work with PDT to prioritize GUI bugs. • Manage GUI regression test process • Present test strategy for GUI at ORR
Script Test Lead	• Manage the script test effort • Present test strategy for script at ORR • Work with PDT to prioritize script bugs. • Verify requirements
Resources Leads	• Manage and coordinate remaining implementation and test efforts for resource components. • Verify requirements • Track new and existing bugs to closure
Commands Lead	• Manage and coordinate remaining implementation and test efforts for resource components. • Verify requirements • Track new and existing bugs to closure
App. Control Lead	• Manage and coordinate remaining implementation and test efforts for resource components. • Verify requirements • Track new and existing bugs to closure
Feature Lead	• Coordinate and support the developer, tester, engineer, and tech writer to see feature to completion. • Check in bugs for existing regression tests that fail • Maintain work items estimates for feature area • Work with PM to schedule • Write or obtain input for user interface spec

	• Validate requirements for feature • Track new and existing bugs to closure • Report on feature status • Present feature at ORR
Developer	• Fix defects • Implement new functionality • Remove untested features • Remove deferred or incomplete features
Tester	• Develop test summaries for unique feature characteristics • Diagnose user issues • Write tests • Submit defect reports • Verify bug fixes
Engineer	•
Technical Writer	• Write reference material • Write tutorials • Write How To docs • Write outreach/marketing material

Role	**Responsibilities/Deliverables**

Feature Area Lead	● Responsibilities ○ On time delivery of feature area ○ Quality of features and docs ○ Review of other area member work products ○ Improve error messages ○ Provide engineering analysis when need to diagnose bugs ○ Verify bug fixes ● Deliverables ○ Requirements ○ Field Spec ○ Functional Spec ○ Script test procedures ○ Detailed bug reports
SDE	● Responsibilities ○ On time fixes of defects ○ Quality of code and application ○ Remove silent or obsolete fields ○ Read parameter Get/Set methods for consistent handling of data types and validation of user inputs. ● Deliverables ○ P1 Bug fixes ○ High Value P2 Bug fixes ○ Improved error messages ○ Spec review comments/suggeestions ○ Spec writing when code reading is required (Field spec for example)
STE	● Responsibilities ○ Quality of GUI interface ○ Run regression tests ○ Verify bug fixes ● Deliverables

	○ GUI Test Procedures ○ Automated GUI tests ○ Detailed bug reports
TW	● Write reference material ● Review tutorials

Requirements Verification

The goals of requirements verification phase are to ensure that requirements are clear, testable, and complete, and to define which requirements will be deferred until after operational certification. Validated requirements will be will be frozen by feature group until after certification is complete and the first operational version of GMAT is delivered. The final deliverable of the requirements verification process is a verified Software Requirements Specification that describes all functionality to be delivered in GMAT v1.0. (The Requirements to Test Matrix and other QA tools are discussed in the Testing section.)

Requirements verification is performed on a feature-by-feature level and is managed by the feature lead. The verification process starts by comparing existing requirements for the feature found in the GMAT SRS with existing functionality in GMAT. The verification process is then performed using the following steps:

1. Identify required functionality that has not been implemented in the system.
2. Identify functionality that has been implemented but is not in the requirements.
3. Develop recommendation for final requirements that include how to handle:

    a. New functionality that should be implemented (only critical new functionality will be considered)
    b. Un-required functionality that has been implemented in the system.
4. Peer review the recommended final requirements obtaining inputs from the PDL, Test Area Lead, Feature Lead, Tester, Engineer, and Developer. (This does not have to occur in an in-person meeting, and can be performed in google docs or using track-changes in a word doc.)
5. Update and finalize the requirements based on peer-review.
6. Provide final requirements to PDL for update in the SRS.

Note to requirements reviewers: the GMAT script-based regression test systems makes extensive use of requirements IDs that are defined in the SRS. Changes to requirements IDs can be made, but they must be made in both the SRS and test case (TC) files. When another solution is possible, do not change requirements IDs. In the case that changing an ID is the best solution, do NOT change requirements IDs without letting the feature lead and feature tester know of the changes before they are actually made in the SRS.

Interface Specification

The goals of the interface specification phase are to ensure that user and external interfaces meet requirements for v1.0 and to provide engineering specifications of those interfaces so that testers and end-user documenters have the necessary information to test and document the system. The final deliverable of the interface specification process is a set of documents that completely describes the behavior of the GMAT script, GUI, and external user interfaces. User interface specifications are written on a feature-by-feature level and are managed by the feature lead using the following steps:

1. Identify all script and GUI fields and interface elements that are implemented for the feature.
2. Determine if any required interface elements are missing and check in bugs so that they will be added during defect resolution.

3. Determine if any non-required interface elements exist and check in bugs so that they will be removed during defect resolution.
4. Document each required interface component (text box, combo box, field name etc.) in the User Interface Specification.
5. Peer review the interface specification with input from PDL, Test Area Lead, Feature Lead, Tester, Engineer, and Developer.
6. Finalize the interface specification based on peer-review input.

Testing

GMAT testing is a complex activity that is described in detail in the GMAT Test Plan and GMAT Test Procedures documents. Due to the complexity of the testing activity only a high level overview of the testing process is presented here. Inputs to the testing process are the Requirements Specification, Interface Specification. The deliverables of testing activities are:

- Test Summaries
- Test cases
- Defect reports
- Requirements to Test Matrix
- Interface to Test Matrix

The Requirements to Test Matrix (RTM) and the Interface to Test Matrix (ITM) accomplish different quality assurance purposes and are automated outputs of the GMAT test process. The RTM ensures that GMAT meets requirements (Acceptance Tests). The ITM maps functional and validation tests to specific user interface elements to ensure all elements perform as designed. For example, GMAT often requires several fields or “Features” to meet a requirement. One GMAT requirement states that “The system shall allow the user to define the spacecraft state using Classical Keplerian Elements”. The design allows the user to specify the Cartesian state as individual values rather than setting the entire state as a six vector at one time. The result is that there are six fields (interfaces) that map to a single requirement. Each of these fields requires for example Input Validation tests and Functional tests. The ITM ensures that each required test type is performed for each interface element.

Defect Resolution

The goal of the defect resolution effort is to resolve software defects, and to document known critical issues that are present in the release version. Defects are addressed in priority (P) order where P1 means must fix, P2 means hope to fix, and P3 and below means the issue will not be fixed. When a feature is fully tested and all issues planned for correction have been addressed, code related specifically to that feature is locked in the version control system. Defect resolution is performed using the following steps.

1. Fix interface bugs first. Interfaces must be complete so that testers and documenters can perform their tasks.
2. Fix defects marked as priority P1 in the bug tracking database.
3. When all P1 bugs are fixed. Address P2 Bugs.

User Documentation

GMAT user documentation includes required User's Guide content and selected portions of the what is commonly located in the Version Description Document. All content required in the Version Description Document that targets GMAT end-users is located in the GMAT User's Guide. The GMAT User's Guide is broken down into the follow main sections:

- Introduction
    - GMAT Overview
    - Getting Started
- Creating Your First Mission
- How To
- Tutorials
- Reference Material

Analysis Environment Testing

To be written.

Operational Environment Testing

To be written.

Operational Readiness Review

To be written

Product Delivery

GMAT will be delivered as a standard desktop application including the following:

- Automated installer
- User's Guide (Manual)
- Version Description Document

Product Control

This section describes the approaches the Product Development Team (PDT) will use to manage and control the project's data and products.

The PDT has tailored the standard configuration management processes in the Organizational Support / Configuration Management area of the GSFC PAL to fit the unique nature of this web-based collaborative project. The guidance on data management, configuration management, and the checklists for configuration audits was consulted to insure completeness of the processes used but was modified to fit the web-based collaborative needs of the project.

Data Management

The location of all GMAT project data including process documentation, technical documentation, software source code, and data files is described in the project's Data Management List (DML), The DML has been combined with a product-based Work Breakdown Structure (WBS) and the Stakeholder Involvement matrix. The WBS/DML is organized by process area for management activities and by engineering areas for technical activities. The WBS/DML will be monitored for completeness on at least a quarterly basis, with each item verified to ensure it is being stored as planned at least once a year. The DML contains the complete list of all GMAT Configuration Items (CI). For each CI, there is a brief description, along with the level of control for each item, and those with change authority. The current version of the WBS/DML is provided here.

Configuration Management

This section describes how the GMAT Product Development Team performs Configuration Management (CM). The CM process is performed under the direction of the Product Development Lead (PDL) and the Configuration Management Officer (CMO). The primary CM activities for the GMAT project are described in the table below and later sections describe how each activity is performed. Day-to-day CM activities are guided by the Data Management List and the CM process described in this Chapter.

Area	Activities
Configuration Identification	• Define levels of control • Roles with change authority • Establish baselines • Baseline creation and storage

Configuration Control	● Establish data storage locations and tools ● Establish CCB ● Define CCB responsibilities, authority, and processes ● Evaluate change requests and approve or disapprove
Configuration Status Accounting	● Establishment process to manage change of controlled items ● Record and monitor changes to evolving CIs
Configuration Audits	● Establishment of validation activities that ensure that the software conforms to released documentation and approved changes

Configuration Identification

This section describes the levels of control and change authority for GMAT configuration items, and describes how baselines for configuration items are created and stored.

Levels of Control

The GMAT Data Management List contains a complete list of all project configuration items along with the level of control and the role with change authority for each item. GMAT configuration items are all assigned one of the following levels of control:

Level of Control Acronym	Policy and Requirements
CM	● A configuration item which requires version control ● Changes must be approved by the CCB

VERA	• A configuration item which requires version control. • Changes can only be made by individuals acting in a role with change authority.
VER	• A configuration item which requires version control. • Changes can be made by any team member. • For CI items defined as VER, the DML contains a (N/A) in the change authority column.
STORED	• An item that is not version controlled, such as email, notes, meeting minutes that are simply stored in a documented location.

Change Authority

GMAT configuration items marked as CM or VERA are all assigned an owner by role and only individuals acting in the assigned roles are permitted to make changes to a configuration item. This process ensures that changes to configuration items are made only by those with the required expertise and if necessary, CCB approval is obtained. For example, only software developers can modify source code and make files, only flight dynamics engineers can modify mathematical and algorithmic specifications. Roles for the GMAT project are defined in the Roles and Responsibilities sections. The DML contains a complete list of configuration items and the role with change authority for each item. The next section describes which roles have change authority for major project configuration items.

Planned Baselines

The table below contains a summary of baselined configuration items along with their level of control, change authority role, and project phase when baselined. As new configuration items are identified, they are added to the DML along with the required level of control and change authority. Only configuration items shown in the table below are baselined. Baselines are

created and maintained through the process described in the next section.

Configuration Item	Level of Control	Change Authority	Phase when Baselined
Feature Prioritization	CM	CCB	Cycle Start
Product Plan	CM	CCB	Cycle Start
Requirements Specification	CM	CCB	Release
User Interface Specification	CM	CCB	Release
Feature Specification	VERA	SE/DE	Release
Mathematical Specification	VERA	FDE	Release
Design Specification	VERA	SE	Release
Source Code	VERA	SE	Release
Build/Make Files	VERA	SE	Release
Doxygen Output	VER	N/A	Release
User's Guide	VERA	TW	Release
Version Description Doc.	VER	N/A	Release
Test Plan	VERA	TE	Release
Requirements to Test Matrix	VER	N/A	Release

Creation and Storage of Baselines

The GMAT team creates and stores baselines for all configuration items stored on SourceForge and Jazz by creating Tags in the revision control software and by creating repository branches for each major release. After code freeze for a release, a tag is created in the revision control system using the release ID (2012a for example). Next, a branch is created in the repository structure with a duplicate of all versioned items at the time of the release. The Tag ensures that all configuration items in the revision control system can be retrieved by release ID. The branch ensures that the GMAT team can address critical issues in a specific release and provide an incremental build for users years after the initial release date. This is important when switching to a new version that is unacceptably risky to a customer and the best solution is to fix a bug in the version approved for use by the customers organization.

Configuration Control

The section below describes policies and procedures for making changes to GMAT configuration items, the primary storage location for those items, and the organization and function of the CCB.

Data Storage Locations

The GMAT team uses the primary data storage locations shown in the table below. Similar types of data are stored in different locations so that code and documents approved for public release are publicly available while code/docs not yet approved for release -- or that are proprietary or potentially ITAR -- are firewalled. Configuration items approved for public release under the NASA Open Source Agreement (NOSA) are located on the project's SourceForge page. All other configuration items that require version control are stored on the JAZZ server maintained by Code 580. Backups of the SourceForge repository are performed monthly by the GMAT team. Backups of the JAZZ repository are performed (Daily ??) by code 580.

The GMAT team uses Google docs for selected project documentation to facilitate collaboration and co-authoring and reviewing of documents. The publicly available google docs service is used for docs related to publicly approved material. The NASA google docs service is used for material not approved for public release and these documents are not available to the public

until approved for release.

Name	What is Stored	Type of Control	Location
Source Forge	Files approved for public release: ● Code ● Data ● Build files ● Prototypes ● Documents	Manual Version Control	https://gmat.svn.sourceforge.net/svnroot/gmat
Jazz	Files not approved for release yet, items that are proprietary, or ITAR: ● Code ● Libraries ● Documents ● Test system	Manual Version Control	https://gs580s-jazz.ndc.nasa.gov/svn/GMAT
Google Docs	Initial file generated for use on the GMAT project.	Automatic Version Control	https://docs.google.com
Personal Hard Disk	● Individual data ● SBU data	Stored Manually	Team member's personal hard disk

Configuration Control Board

The GMAT Configuration Control Board (CCB) is composed of selected members (by role) of the product development team and provides documented authorization for all changes to configuration items defined as CM-level control in the Data Management List. CCB members are selected to ensure broad interdisciplinary representation, and additional subject matter experts are selected as needed for the material under review. The Acting CCB (ACCB) is formed when the PDL or designee selects team members and subject matter experts for the purpose of analyzing or approving a requested change or issue for a CM-level configuration item. The Acting CCB for one type of request may be different than the Acting CCB for another request. For example, approval of changes to software requirements requires may require a different set of individuals than changes to the product plan or test plan.

Most CCB decisions are made during weekly GMAT meetings and do not require a meeting for the sole purpose of performing CCB activities. These decisions are marked as CCB approved in the weekly meeting minutes and the initials of the Acting CCB members are added to the meeting minutes. In the rare event that a meeting is held just for the sole purpose of CCB activites, the minutes are documented in the same manner as the weekly meetings. Additionally, to ensure efficiency of GMAT development process, ACCB concurrence can be documented electronically without an in-person meeting by obtaining electronic signatures in the issues tool discussed Issues Tracking section of the Management Approach chapter.

Defect Reporting

Software defect reports are managed in the project's Bugzilla database (see DML for URL). Processes for reporting and resolving defects are discussed in detail in the GMAT Test Plan and only a high level overview is included here. Defects related to the GMAT application are classified by feature area. There are special categories for documentation, test system, and other types of defects. All defects follow these basic steps:

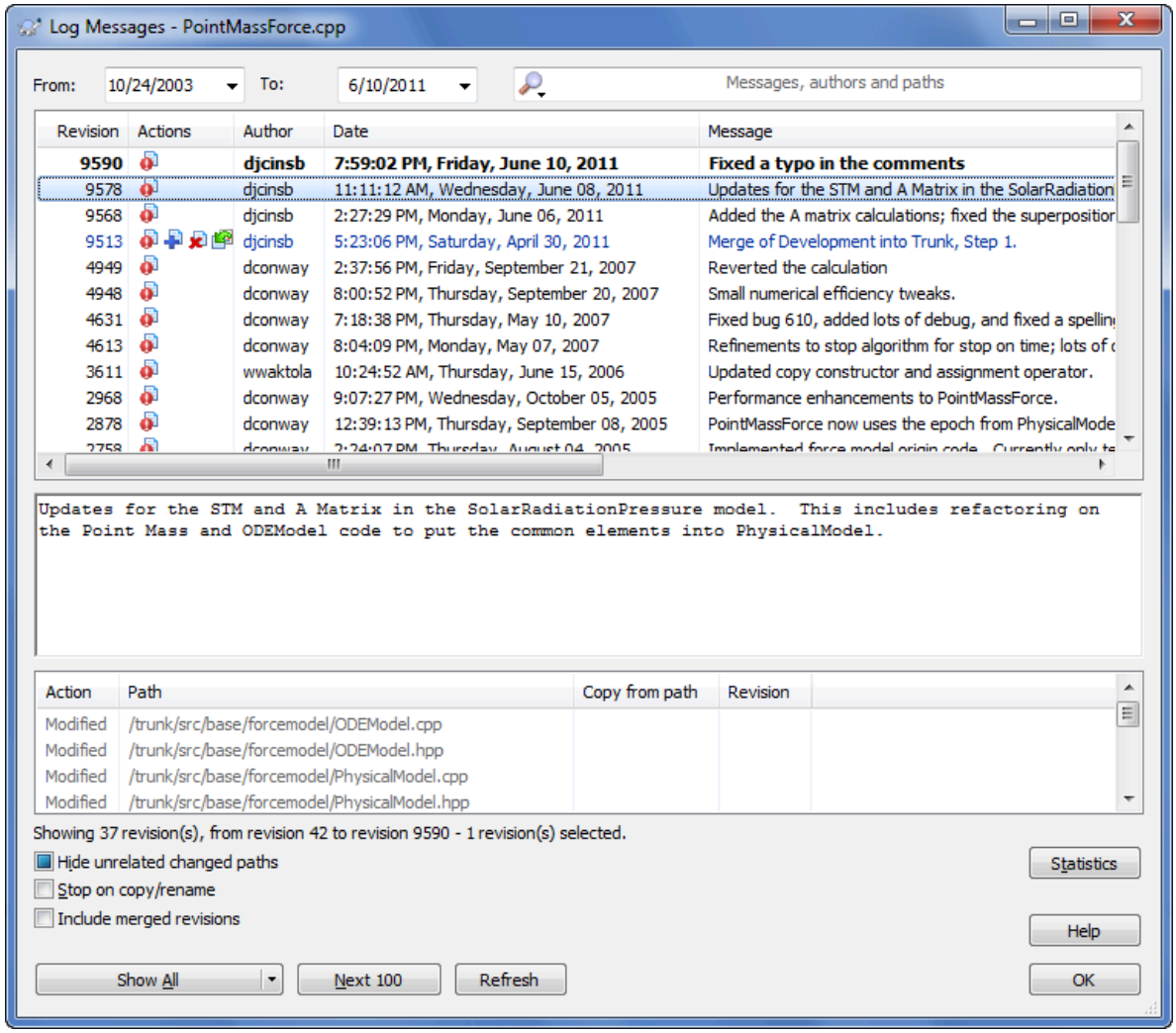
- New -- The initial state of a defect report.
- Assigned -- Assigned to an individual for analysis and tracking. If the bug is marked as P1 (Priority 1), then the assignee is the individual responsible for addressing the defect.

- Needs Test -- The assignee has resolved the defect, unit tested, and passed the issue to tester so if appropriate, a test case is added to the regression system.
- Needs Documentation - The defect has been resolved; if necessary, explain the issue in documentation.
- Resolved -- There is no remaining work; the defect is closed.

The prioritization of a bug is used to indicate that it has been approved for resolution. Defects marked as P1 must be addressed by the next major release. Defects marked as P2 are bugs that should be fixed if the schedule allows. Defects prioritized as P3 or below will not be addressed in the next release.

Configuration Status Accounting

The purpose of Configuration Status Accounting is to record and monitor changes to evolving CIs throughout the project life cycle. The change history for configuration items maintained in the project's SourceForge or Jazz repositories is automatically documented for all of these items by Subversion (SVN). SVN client software such as TortoiseSVN or SmartSVN provide tools to view the change logs of all version controlled files. Below is the change log for a randomly selected source code file. All GMAT configuration items that require configuration status accounting are maintained in a SVN repository.



Additional configuration status accounting items are CCB minutes, discussed in the Meeting Minutes section, Configuration Audit reports, discussed in the configuration audits section.

Configuration Audits

Configuration audits are performed to maintain the integrity of the configuration baselines. Specifically, audits ensure that developed and installed products meet their technical requirements, that the products are accurately described in documentation, and that the products do not include any unauthorized changes. The CMO will conduct configuration audits, document and retain the audit results as indicated in the DML, and forward the audit results to the PDL and/or relevant stakeholders. The PDL and members of the PDT will work closely with the CMO to resolve any problems identified in the audit results.

The CMO will conduct the following configuration audits:

- **Baseline audit** will be conducted for major releases. A Baseline audit verifies that the content and versions of the CIs are consistent with the baseline documentation that defines them (i.e., the Baselines Table). See Planned Baselines for content description.

- **Physical Configuration audit** (PCA) will be conducted for major releases. The PCA verifies that a CI, as built, conforms to the technical documentation that defines it (i.e., it verifies that all baseline elements of the delivery match the documentation). During this audit, the CMO will analyze the following to assure that the documentation is accurate:
    - Baselines Table
    - DML
    - Version Description Document (VDD)
    - Physical contents of the project's CM Libraries (where the CIs reside)

- **Functional Configuration audit** (FCA) will be conducted for major releases. The FCA verifies that the CI has been completed satisfactorily, that the item has achieved the performance and functional characteristics, and that its operational and support documents are complete and satisfactory. During this audit, the CMO will analyze the:
    - Requirements Traceability Matrix

- Requirements documentation
- Test reports
- Change authorizations

Control of Non-conforming Products

Issues related to non-conforming products are managed using the Issues Tool described in the Issues Tracking section. Any member of the PDT may fill out a problem report (i.e., non-conformance) against any system product or tool at any time. Issues are reviewed monthly or at the request of any team member or stakeholder. All non-conformance issues that adversely impacts the cost, schedule, or scope of the effort are reported to stakeholders and customers for approval of the proposed solution.

Control of Customer-Supplied Products

There are currently no customer supplied products.

Quality Assurance

The GMAT Software Assurance Plan (SAP) provides the framework necessary to ensure a consistent approach to software quality assurance throughout the project life cycle. It defines the approach that will be used by the Chief Safety and Mission Assurance Officer (CSO) and Software Assurance Engineers (SAEs) to monitor and assess software development processes and products to provide objective insight into the maturity and quality of the software. The Software Assurance (SA) support will be provided by Code 320 and paid for by the GMAT Project. The PDT will work closely with the Code 320 SAE during SA activities.

Software Assurance Process

The SAE will comply with the Software Assessment Process and use the software quality checklists and forms available on the GSFC Software Assurance web site to conduct product and process assessments.

The SAE will identify and document non-compliances, observations, and risks from all assessments and report these results to the PDL. The PDL will review assessment results, retain assessment results as indicated in the WBS/DML, and work with the SAE and members of the PDT to develop a corrective action plan to resolve identified problems. The PDL will document the audit, its findings, and, as appropriate, the corrective action plan in the Audit Findings and Corrective Action Tool. The PDL will use this tool to track all corrective actions to closure. The SAE will monitor the status of non-compliances to ensure they are resolved and will escalate any non-compliances that cannot be resolved within the PDT to ?????.

Peer Reviews / Inspections

The peer review / inspections process enables rigorous and formal review of work products to certify quality and conformance to standards. The PDT will use the checklists provided in the Product Development area of the GSFC PAL for the various types of peer reviews / inspections.

The PDT will use the peer review / inspections process to prepare for and conduct peer reviews / inspections, to document results, and to certify completion for the following work products:

- Requirements
- Implementation (i.e., design and code work products that ?????)
- Test Plan
- Product Plan
-

The PDT will use the Inspection Moderator Tool to support this process. The PDT will retain the review and certification records and measures produced by this tool as described in the DML.

Code 320 QA Support

This section summarizes the tasks to be performed during the new software development, operations support activities, and sustaining engineering of the software commensurate with the software classification. These tasks are selected based on the project schedule, Software Management Plan (SMP), planned deliverables, contractual deliverables, and identified reviews.

NOTE: Given the maturity of the GMAT product, the level of software assurance has been reduced. Process assessments are conducted annually, while sample products are randomly evaluated.

Product Assessments

Products planned for assessment by the SAE are:

- Product Plan (Includes the Configuration, Requirements, and Risk Management Plans)
- GMAT User's guide (w/ Version Description Documents)
- Risk Management Tool
- Software Metrics
- Branch Status Review Package
- Project Schedule
- Requirements Traceability Matrix (bi-directional between requirements and tests)

Process Assessments

Processes planned for assessment by the SAE are:

- Project Planning
- Project Monitoring and Control
- Measurement and Analysis
- Risk Management
- Configuration Management
- Requirements Management

Additional Activities

- Attends Branch Status Reviews (BSRs).
- *Attends Major Mission Milestone Reviews*
- *Attends Developer Status Meetings, as time permits.*
- *Attends Configuration Control Board (CCB) meetings and reviews Problem Reports*
- Attends PDL Tag-ups for Project Status.
- Maintains GMAT SA Action Item Log.

Acronyms and Abbreviations

CMMI Policy Compliance Information

References

This document was developed using explicit versions of NPRs, GPRs, and organizational process and standards documents. The table below contains a list of these documents. The policy with respect to how changes in these documents affect this Plan is as follows:

The versions referenced are current at the time of writing. If one of the standards documents changes at a point in the development life-cycle after which the referenced process is no longer being used, the reference in this Plan will not be updated.

If there is a change in a standards document relating to a process that has not yet been completed, the change will be evaluated for impact on the content of this Plan and the related development process. In this case, the reference may be updated to reflect the new version, along with required changes (if any) to this Plan. It will not be necessary to update the signature page in this case.

Table: Documents Used in Developing This Plan

Document Name	Document Number	Date
NASA Export Control Program	NPR 2190.1	04/10/03
NASA External Release of NASA Software	NPR 2210.1	01/04/02
NASA Security of Information Technology	NPR 2810.1A	04/07/04
NASA Information Security Policy	NPR 2810.1C	04/07/04

NASA Software Policy	NPR 2820.1C	08/30/05
NASA Software Engineering Requirements	NPR 7150.2A	11/19/09
NASA Risk Management	NPR 8000.4	04/25/02
NASA Software Assurance Standard	NASA-STD-8739.8	07/28/04
NASA Software Safety Standard	NASA-STD-8719.13B	07/08/04
Configuration Management	GPR 1410.2C	03/01/05
Occupational Safety Program at GSFC	GPR 1700.1-	02/21/06
Corrective and Preventive Action	GPR 1710.1H	12/29/04
GSFC Section 508 Web Policy	GPR 2800.1-	10/06/05
~~Security of Information Technology~~	~~GPR 2810.1-~~	~~04/06/03~~
~~Wireless Networks and Access Points~~	~~GPR 2810.2A~~	~~01/19/05~~
Receiving Inspection and Test	GPR 4520.2E	08/08/06
Procurement	GPR 5100.1F	03/14/07
Supplier Performance Evaluations	GPR 5100.2B	11/18/04
Quality Assurance Letter of Delegation	GPR 5100.3F	05/07/07
Supplier Quality Audits	GPR 5100.4D	12/29/04
Government Cost Estimates for Acquisitions Exceeding Micro-Purchase Threshold (Projects	GPR 5100.5A	04/28/05

over $1 Million)		
Identification and Traceability of Products	GPR 5310.4D	12/29/04
Product Processing, Inspection, and Test	GPR 5330.1E	02/15/05
Control of Nonconformances	GPR 5340.2I	01/10/05
Program and Project Management	GPR 7120.1C	01/05/05
Risk Management	GPR 7120.4A	06/14/05
Software Project Process Initiation	GPR 7150.1	
In-House Software Development	GPR 7150.2	
Process Control	GPR 8072.1D	02/22/05
Design Planning and Interface Management	GPR 8700.1D	02/15/05
Design Development	GPR 8700.2E	07/29/05
Design Validation	GPR 8700.3B	02/15/05
Integrated Independent Reviews	GPR 8700.4F	08/02/05
~~In-House Development and Maintenance of Software Products~~	~~GPR 8700.5A~~	~~01/05/05~~
Engineering Peer Reviews	GPR 8700.6	01/26/05
Calibration and Metrology	GPR 8730.1J	02/23/07
Electrostatic Discharge Control	GPR 8730.6	01/05/06

ISD Software Management Plan / SMP/PP (SMP/PP) for Class B & C Software Template	580-TM-033-01	08/25/05
ISD Software Policies	580-PL-002-02	04/20/05
Project Planning Process	580-PC-004-03	04/02/07
ISD Project Monitoring and Control	580-PC-012-01	02/01/05
ISD Software Configuration Management	580-PC-019-01	02/01/05
ISD Requirements Development Process	580-PC-020-01	06/20/05
ISD Requirements Management Process	580-PC-024-01	06/01/05
Measurement and Analysis for ISD Projects	580-PC-048-01	12/05/05
Acquisition Preparation Process for ISD Projects	580-PC-059-01	04/24/06
ISD Lessons Learned Process	580-PC-060-01	05/25/06
Easy Acquisition Process for ISD Projects	580-PC-061-01	04/24/06
Coding and Integration Process	580-PC-066-01	04/01/07
ISD Configuration Control Board (CCB) Process	580-SP-001-01	03/01/04
ISD Software Risk Identification	580-SP-013-01	09/01/04
ISD Software Risk Monitoring and Control	580-SP-014-01	09/01/04
ISD Software Project Estimation	580-SP-026-01	06/20/05
ISD Preliminary Design	580-SP-041-01	02/18/06

ISD Detailed Design	580-SP-042-01	02/18/06
ISD Inspections, Peer Reviews, and Walkthroughs	580-SP-055-01	04/24/06
ISD Sub-processes, Templates, Guidelines, Checklists and other materials in the ISD Process Asset Library (PAL) http://software.gsfc.nasa.gov/	Various	Various dates

Planning Documents References

The table below lists the planning documents referenced in this Plan. During development of these documents, they will be reviewed by the PDL and/or the PDT for consistency among them. When completed, the latest version of these documents may be found at the listed location.

Table: Planning Documents References

Planning Documents	URL
SMP/PP	PDL Desktop (Master Copy containing cost-sensitive data) See WBS/DML (Distribution Copy)
Test Plan	See WBS/DML
Software Assurance Plan (SAP)	See WBS/DML

Websites and Systems References

The table below lists the websites and systems along with their associated web addresses or

locations that are referenced in this document.

Table: Websites and Systems References

Website / Support System	URL
NASA Procedural Requirements (NPRs)	http://nodis.hq.nasa.gov/
Goddard Procedural Requirements (GPRs)	http://gdms.gsfc.nasa.gov/gdmsnew/home.jsp
NASA Lessons Learned Information System (LLIS)	http://llis.nasa.gov
Goddard Space Flight Center (GSFC) Receiving Inspection and Test System (RITS)	http://rits.gsfc.nasa.gov/
~~GSFC Problem Reporting System (GPRS)~~	~~http://gprs.gsfc.nas~~
GSFC Software Assurance website	http://sw-assurance.gsfc.nasa.gov
GSFC Software Process Improvement (SPI) Process Asset Library (PAL) (policies, training materials, guidelines, etc.)	http://software.gsfc.nasa.gov
~~ISD Library of Approved Team Processes (Section 2: Control of Documents and Data & Quality Records)~~	~~http://ISD.gsfc.nasa.gov/iso9k/is09001.htm~~
~~ISD Library of Approved Team Processes (Section 3: Control of Customer Supplied Elements)~~	~~http://ISD.gsfc.nasa.gov/iso9k/iso9001.htm~~

~~Code 400 Project Configuration Management (CM) System~~	~~https://~~
Document Repository	See WBS/DML
Problem/Change Reporting System	See WBS/DML
~~Lab Server~~	~~Server located in the development lab~~

Roles, Responsibilities, Authority & Accountability

The table below defines the primary process responsibilities assigned to project roles. A “P” indicates primary responsibility for the process area. Any listed role may also contribute to any of the listed process areas. The process areas in the table include:

- Project Planning (PP)
- Project Monitoring and Control (PMC)
- Measurement and Analysis (MA)
- Risk Management (RSKM)
- Configuration Management (CM)
- Process and Product Quality Assurance (PPQA)
- Requirements Engineering (REQ), which includes Requirements Development and Requirements Management
- Design and Implementation (D&I), which includes Technical Solution, Product Integration, and Decision Analysis Resolution
- Verification and Validation (V&V)
- Acquisition (ACQ), also known as Supplier Agreement Management

Role	PP	PMC	MA	RSKM	CM	PPQA	REQ	D&I	V&V	ACQ
PDL	P	P	P	P						P
DTL							P	P		
GTL (GUI)									P	
STL (Scripts)									P	
SAE						P				
CMO					P					

I believe the information in this section is covered in other sections and the table below is either obsolete, or needs to be moved to an appendix.

Management Processes and Tools

Unless otherwise specifically stated in this Plan, the PDT will comply with the standard management processes in the Project Management area of the GSFC PAL. This area includes processes that will be used for Project Planning and Project Monitoring and Control. The PDT will use these tools listed to manage this effort:

Planned Use	Tool	Notes
Project Planning, Monitoring, and Control	Evaluating Options	a
Work Identification	WBS/DML Spreadsheet	a

Cost Estimation	Wideband Delpi Form	a b
Schedule Planning and Monitoring	Evaluating Options	c
Staff Planning and Monitoring	Evaluating Options	a
Data Management Planning and Monitoring	WBS/DML Spreadsheet	a
Stakeholder Planning and Monitoring	Stakeholder Involvement Tool	a
Detailed Schedule Activities Planning and Monitoring	Evaluating Options	a
Training Planning and Monitoring	Training Tool	a
Risk Management	Risk Management Tool	a
Action Item Tracking	Action Items Tracking Tool	a
Issues Tracking	Issues Tracking Tool	a
Peer Reviews / Inspections	Inspection Moderator Tool	a
Audit Findings and Corrective Actions Tracking	Audit Findings and Corrective Actions Tool	a
Measurement and Analysis	MS Office-based Measurement and Analysis Tools	a
Requirements Measures	Requirements Metrics	a

	Tool	
Inspection Measures	Inspection Metrics Tool	a

Notes:

a. No purchase needed – tool uses MS Office programs; MS Office is provided with the standard GSFC desktop configuration; tool is free and available from the TOOLS area of the GSFC PAL

b. No purchase needed – tool is available using a GSFC site license

c. No purchase needed – tool is provided by the Branch

NASA Policy Compliance Matrix

This matrix was taken from GPR 7150.2, dated Jan 10, 2011, for Class B software. This matrix is to be used by the PDT to determine the minimum requirements for developing a class B software system.

Entries under the heading “Class B Software” are interpreted as follows:

- **“X”** indicates the requirement must be implemented as stated
- **“SO”** (safety only) indicates the requirement applies to the extent necessary to satisfy safety critical aspects of the software. GMAT is rated as non-safety critical so these requirements are considered to be non-applicable.
- **“ “** blank boxes indicate the requirement is non-applicable per GPR 7150.2

The following table presents the mapping between GPR 7150.2 and the GMAT SMP/PP.

Req. #	Requirements	Class B Software	SMP/PP Section or Other Location
G2SW-001	Comply with requirements by Software Classification and Safety Criticality	X	Software and Safety Classification
G2SW-002	Use Software Classification and Safety Criticality to Determine Requirements for Planning, Managing, and Implementing the in-house development/maintenance of software	X	Software and Safety Classification
G2SW-003	Use Center-defined *“Requirements for the Minimum Content of Software Documents”* on GSFC/EPG website to Determine	X	This table

	Documentation Requirements **(**Asset tailors document contents for P(C) documents)		
G2SW-004	Software Development or Software Management Plan /Product Plan (SMP/PP)	X	This document
G2SW-005	Software Configuration Management Plan	X (SO)	Configuration Management
G2SW-006	Software Test Plan	X (SO)	Jazz\trunk\docs\TestPlan
G2SW-007	Software Maintenance Plan	X (SO)	Not Applicable
G2SW-008	Software Assurance Plan	X	Quality Assurance
G2SW-009	Software Safety Plan	X	Software and Safety Classification
G2SW-010	Software Requirements Specification	X	Jazz\trunk\docs\SystemDocs\ Requirements
G2SW-011	Software Dictionary (as applicable)	X (SO)	Not Applicable
G2SW-012	Software Design Description	X	Feature Specs on Project Wiki
G2SW-013	Interface Design Description	X	Feature Specs on Project Wiki
G2SW-014	Software Change Request/Problem Report	X	Defect Resolution

G2SW-015	Software Test Procedures	X	Jazz\trunk\docs\Test
G2SW-016	Software User Manual	X	SourceForge\trunk\doc\help
G2SW-017	Software Version Description	X	Contained in user's manual
G2SW-018	Software Metrics Report	X	Jazz\trunk\doc\ProcessDocs\ Monitoring and Control
G2SW-019	Software Test Report	X	emails
G2SW-020	Software Peer Review/Inspection Report	X	Peer Reviews / Inspections
G2SW-021	Plan the Software Development/Maintenance Project	X	This document
G2SW-022	Document and maintain the Software Management Plan/Maintenance Plan	X	This document
G2SW-023	Plan for operations, maintenance and retirement	X	?
G2SW-024	Re-plan When Relevant Stakeholders Agree and When Software or Safety Classifications Evolve to a Higher-Level	X	?
G2SW-025	Review SMP/PP at least at major reviews and replan as needed	X	?
G2SW-026	Review of plans for consistency among them	X	?

G2SW-027	Review of plan changes by stakeholders	X	?
G2SW-028	Approval of SMP/PP and associated plans	X	See signature page of this and associated documents
G2SW-029	Manage according to SMP/PP and/or maintenance plan and associated plans	X	?
G2SW-030	Plan, document and implement software assurance according to NASA-STD-8739.8	X	?
G2SW-031	SQE plans and performs process and product audits	X	Quality Assurance
G2SW-032	Resolve audit findings, track to closure	X	Software Assurance Process
G2SW-033	Document and Maintain the Software Safety Plan	X	Not applicable See Software and Safety Classification
G2SW-034	Determine which tasks, activities, processes are required for the project	X	This document
G2SW-035	Document tasks, activities, processes required for the project in a WBS	X	See WBS/DML
G2SW-036	Create and maintain the project schedule	X	Schedule and Milestones
G2SW-037	Ensure schedule coordinates with overall project schedule	X	Not applicable, there is no “Mission Project”
G2SW-038	Schedule documents interactions of milestones and deliverables	X	Not applicable, there is no “Mission Project”

G2SW-039	Schedule reflects critical path for software development activities	X	Not applicable, there is no “Mission Project”
G2SW-040	Schedule documents life cycle activities	X	Schedule and Milestones
G2SW-041	Schedule identifies dependencies between/among life cycle activities	X	Schedule and Milestones
G2SW-042	Define and document cost estimate process, including BOE, tools,	X	Resources Required
G2SW-043	Develop cost estimate that covers entire life cycle	X	Resources Required
G2SW-044	Develop cost estimate that is based on project attributes	X	Resources Required
G2SW-045	Develop cost estimate that considers implications of technology	X	Resources Required
G2SW-046	Reconcile cost estimates and project’s available budget	X	Resources Required
G2SW-047	Review cost estimate yearly and at milestones; update as needed	X	Resources Required
G2SW-048	Identify and document all roles and responsibilities to perform management, technical and process activities	X	Roles and Responsibilities
G2SW-049	Assign primary responsibility for performance of each CMMI process area to only one person	X	Process Roles
G2SW-	Develop and maintain a project organization	X	PDT Organization

050	chart		
G2SW-051	Develop and maintain a staffing plan showing planned versus actual staffing	X	Staffing Plan
G2SW-052	Plan and document project-specific training	X	Training Plan
G2SW-053	Ensure personnel have taken project-specific training	X	Training Plan
G2SW-054	Determine processes required for the project	X	This Document
G2SW-055	Document project-specific descriptions for managing requirements and requirements changes	X	Verification and Validation
G2SW-056	Document project-specific descriptions for storing, handling, delivering, and releasing products	X	Build/Release Cycle
G2SW-057	Achieve concurrence on deliverables, receivables, commitments	X	Acceptance Criteria
G2SW-058	Obtain concurrence on estimated budget and resources	X	Resources Required
G2SW-059	Agree on documented acceptance criteria	X	Acceptance Criteria
G2SW-060	Ensure changes to commitments are agreed upon	X	Customer Agreement Stakeholder Involvement
G2SW-061	Define, document process for monitoring and control of project	X	Project Monitoring and Control

G2SW-062	Monitor project commitments	X	Project Monitoring and Control
G2SW-063	Solicit project progress, status, track against plan	X	Project Monitoring and Control
G2SW-064	Analysis progress and status to identify issues	X	Issue Tracking
G2SW-065	Establish corrective actions, track to closure	X	
G2SW-066	Conduct status reviews	X	Status Reporting and Communication
G2SW-067	Use of Branch Status Review template	X	Status Reporting and Communication
G2SW-068	Track issues identified during status reviews to closure	X	Status Reporting and Communication
G2SW-069	Record minutes of status reviews, stakeholder meetings, CCB meetings	X	Status Reporting and Communication
G2SW-070	Define and document the project's measurement objectives	X	Project Metrics
G2SW-071	Choose measures in each of the 5 areas: software progress tracking, software functionality, software quality, software requirements volatility, and software characteristics.	X	
G2SW-072	Document a measurement procedure for collecting, storing, analyzing and reporting measures	X	

G2SW-073	Use *Measurement and Storage Template* or equivalent	X	
G2SW-074	Collect and store project measures	X	Project Metrics
G2SW-075	Analyze measures according to analysis and reporting procedures	X	Project Metrics
G2SW-076	Report measurement analysis results	X	Status Reporting and Communication
G2SW-077	Use *Measurement Summary Tool* to submit measures to organization	X	
G2SW-078	Provide measures for OCE inventory	X	
G2SW-079	Define and document approach for identifying, classifying, analyzing, tracking, mitigating and documenting risks.	X	Risk Monitoring and Control
G2SW-080	Define risk sources and categories	X	Risk Monitoring and Control
G2SW-081	Identify and document project risks	X	Initial Risks Risk Monitoring and Control
G2SW-082	Manage risks	X	Risk Monitoring and Control
G2SW-083	Escalate risks; report status of risks	X	Risk Monitoring and Control
G2SW-084	Plan and document stakeholder involvement	X	Stakeholder Involvement

G2SW-085	Monitor stakeholder involvement	X	Stakeholder Involvement
G2SW-086	Identify and document lifecycle model	X	Project Lifecycle Build / Release Cycle
G2SW-087	Ensure that life cycle includes exit criteria and phase transition reviews	X	Project Lifecycle Build / Release Cycle
G2SW-088	Ensure that phase transition reviews are completed	X	Project Lifecycle Build / Release Cycle
GS2SW-089	Prepare and present SRR, PDR, CDR, ATRR using PAL checklists	X	Not Applicable see Scope
G2SW-090	Plan for engineering environments	X	Development and Test Environment
G2SW-091	Use of data management tool or equivalent to plan project data management	X	Data Management
G2SW-092	Monitor data management	X	Data Management
G2SW-093	Plan and document CM activities, responsibilities and authority	X	Configuration Management
G2SW-094	Describe configuration items, configuration control, configuration status, audits	X	Configuration Management
G2SW-095	Review and update planning parameters to reflect impact of approved changes	X	Configuration Management
G2SW-096	Establish and implement procedures designating levels of control, approval authority, process for authorizing and tracking	X	Configuration Management

	changes		
G2SW-097	Identify configuration items and configured baselines	X	Configuration Management
G2SW-098	Track and analyze changes to software products	X	Configuration Management
G2SW-099	Authorize changes to software products in accordance with the CM plan	X	Change Authority
G2SW-100	Generate and maintain CM records and reports	X	Configuration Status Accounting
G2SW-101	Conduct configuration audits	X	Configuration Audits
G2SW-102	Plan and document verification and validation approach	X	Verification and Validation
G2SW-103	Evaluate and record the results of all verification and validation; Track corrective actions to closure	X	Test Team Roles and Responsibilities
G2SW-104	Document levels of testing to be performed in compliance with Table 2	X	See G2SW-105 through G2SW-109
G2SW-105	Perform unit testing	X	
G2SW-106	Perform testing as planned in test plan	X	Jazz\trunk\docs\TestPlan
G2SW-107	Implementation Verified to Requirements	X	Requirements Verification

G2SW-108	Verification, Validation, and Accreditation of Software Models, Simulations, and Analysis Tools	X	Verification and Validation
G2SW-109	Software Validation on a Target Platform or in a High-Fidelity Simulation	X	Operational Environment Testing
G2SW-110	Use a nonconformance tracking system to track to closure all defects identified during verification and validation	X	Control of Non-conforming Products
G2SW-111	Document the plan for peer review, inspections, walkthroughs		Peer Reviews / Inspections
G2SW-112	Inspections/peer reviews shall have a) identified participants b) use checklists c) use readiness and completion criteria d) document results e) track defects to closure	X	Peer Reviews / Inspections
G2SW-113	Collect measures in Peer Review/Inspection Report	X	Peer Reviews / Inspections
G2SW-114	Inspections/peer reviews of software requirements	X	Requirements Verification
G2SW-115	Inspections/peer reviews of project-selected design work products	X	
G2SW-116	Inspections/peer reviews of project-selected software code	X	
G2SW-117	Inspections/peer reviews of software test plans	X	
G2SW-118	Inspections/peer reviews of software management plans/software maintenance	X	

	plans		
G2SW-119	Inspections/peer reviews of safety plan	X	
G2SW-120	Inspections/peer reviews of CM plan, assurance plans	X	
G2SW-121	Evaluate the options for acquisition versus development	X	Make / Buy Approach
G2SW-122	If acquiring COTs, use COTS acquisition process	X	Not Applicable
G2SW-123	If acquiring custom software, use GPR 7150.3	X	Not Applicable
G2SW-124	Accept the custom built software in accordance with the acceptance plan	X	Integration of Customer-Supplied and Acquired Products
G2SW-125	Transition the acquired software into use	X	Integration of Customer-Supplied and Acquired Products
G2SW-126	Develop, Document and maintain software requirements and specifications	X	Requirements Verification
G2SW-127	Work with the customer to understand high level requirements and operations concepts	X	Not Applicable - has been done in previous work
G2SW-128	Validate the requirements as intended in the customer environment	X	Operational Environment Testing
G2SW-129	Perform requirements analysis based on flowed down requirements and derived	X	Not Applicable -Has been done in previous

	requirements and the design and specification of the hardware.		work -Hardware not part of GMAT
G2SW-130	Work with stakeholders to ensure that interfaces are documented	X	User Interface Specification
G2SW-131	Obtain approval of the requirements	X	Requirements Verification
G2SW-132	Identify inconsistencies among requirements, project plans, and software products, initiate corrective actions and track them to closure	X	Requirements Verification
G2SW-133	COTs, GOTS, MOTS, requirements	X	Not Applicable
G2SW-134	Manage requirements changes	X	Requirements Verification
G2SW-135	Analyze requirements changes to determine impact	X	Requirements Verification
G2SW-136	Establish, document and maintain bi-directional traceability throughout the life cycle in accordance with the minimum requirements shown in Table 4.	X	See G2SW-136 through G2SW-144 for details of compliance
G2SW-137, GSW-138	Bidirectional traceability between System (or Agency-level) requirements and software requirements	X	Not Applicable, the two levels indicated here do not exist
G2SW-139, G2SW-	Bidirectional Traceability between software requirements and software design	X	**We will request a waiver. This is a waste of time.**

140			
G2SW-141, G2SW-142	Bidirectional Traceability between software design and code	X	**We will request a waiver. This is a waste of time.**
G2SW-143, G2SW-144	Bidirectional Traceability between test procedures and software requirements	X	Testing
G2SW-145	Use bidirectional traceability to verify that all source requirements have been addressed and that all lower level requirements trace to a source requirement	X	
G2SW-146	Use bidirectional traceability to verify that all requirements trace to a test and that all test trace to a source requirement	X	Testing
G2SW-147	Document and maintain the project's software design	X	Feature Specs on Project Wiki
G2SW-148	Document the software architectural design based on the allocated and derived requirement	X	Jazz\trunk\doc\SystemDocs\ArchitecturalSpecification
G2SW-149	Document and maintain a detailed software design that describes the lower level units so they can be coded, compiled and tested	X	GMAT uses iterative process and detailed design is determined during implementation and inspection. Detailed Design is then documented by running DoxyGen and the detailed

			design spec is here: http://gmat.sourceforge.net/docs/nightly/base/index.html and here http://gmat.sourceforge.net/docs/nightly/gui/index.html
G2SW-150	Safety-critical software is initialized, at first start and at restarts, to a known safe state		Not Applicable
G2SW-151	Safety-critical software safely transitions between all allowed predefined known states		Not Applicable
G2SW-152	Termination performed by software of safety critical functions is performed to a known safe state		Not Applicable
G2SW-153	Operator overrides of safety-critical software functions require at least two independent actions by an operator		Not Applicable
G2SW-154	Safety-critical software rejects commands received out of sequence, when execution of those commands out of sequence can cause a hazard		Not Applicable
G2SW-155	Safety-critical software detects inadvertent memory modification and recovers to a known safe state		Not Applicable
G2SW-156	Safety-critical software performs integrity checks on inputs to the software system.		Not Applicable
G2SW-157	Safety-critical software performs integrity checks on outputs from the software system		Not Applicable

G2SW-158	Safety-critical software performs prerequisite checks prior to the execution of safety-critical software commands		Not Applicable
G2SW-159	No single software event or action is allowed to initiate an identified hazard		Not Applicable
G2SW-160	Safety-critical software responds to an off nominal condition within the time needed to prevent a hazardous event		Not Applicable
G2SW-161	Software provides error handling of safety-critical functions		Not Applicable
G2SW-162	Safety-critical software has the capability to place the system into a safe state		Not Applicable
G2SW-163	Safety-critical elements (requirements, design elements, code components, and interfaces) are uniquely identified as safety-critical	X	Not Applicable
G2SW-164	Incorporate requirements in the coding methods, standards, and/or criteria to clearly identify safety-critical code and data within source code comments		Not Applicable
G2SW-165	Implement the software design into software code	X	
G2SW-166	Develop and maintain appropriate system and user documentation	X	User Documentation
G2SW-167	Verify that software units adhere to software coding methods, standards, and/or criteria	X	
G2SW-	Use results from static analysis tool(s) to	X	

168	verifying and validating software code		
G2SW-169	Validate software development/maintenance tool(s)	X	
G2SW-170	Document and maintain software test plan(s) and associated software test procedures and test report	X	Jazz\trunk\docs\TestPlan AND Jazz\trunk\docs\Test
G2SW-171	Update test plan(s) and test procedures to be consistent with software requirement changes	X	Requirements Verification
G2SW-172	Verify and validate software models, simulations, analysis tools used to qualify flight software or equipment	X	Operational Environment Testing
G2SW-173	Conduct software tests as documented in the SMP/PP or the stand-alone Test Plan(s) and Test Procedure	X	Testing
G2SW-174	Evaluate and record results of all software verification testing and system validation testing performed	X	Peer Reviews / Inspections
G2SW-175	Identify, record, and analyze defects and track them to closure	X	Defect Resolution
G2SW-176	Prepare and deliver software products as in project's process(es) for storing, handling, delivering and releasing software products	X	Product Delivery
G2SW-177	Prepare and deliver a Product Release Letter and Version Description Document	X	Product Delivery
G2SW-178	Prepare and deliver system/software documentation that will be used to operate or	X	User Documentation

	maintain the software		
G2SW-179	Prepare and deliver as-built system/software documentation that will be used to maintain the software	X	
G2SW-180	Perform software operations, maintenance and retirement activities	X	

Training Plan

The PDL planned for training the PDT and documented the training needs in a Training Plan that is maintained and stored as indicated in the DML. The PDL will also use the Training Plan to monitor PDT training. Both mandatory and optional training are identified based on role assignments. For training monitoring purposes, training for individual PDT members will be identified based on all of their assigned roles. The PDL, or designee, will determine whether any individual PDT member will be exempted from mandatory training based on the individual's proficiency in the skills required for the role. For example, PDT members may be exempted from training that covers basic skills (i.e., skills required as a condition of employment) or training that covers skills in which the individual has extensive on-the-job experience and demonstrable proficiency.

For this effort, the PDL will have the option to exempt PDT members with at least 3 years of experience using the general skills required for the role they are assigned. The required

experience may be less for certain training (e.g., use of a specific tool for several months may constitute sufficient experience for proficiency. Specific explanation for exempting personnel from a training requirement based on the PDT member's proficiency will be noted in the Training Plan.

The PDL, or designee, will be responsible for monitoring and tracking completion of training for each member the PDT based on the role each performs.

Customer Agreement

The customer's signature on the signature page indicates agreement with this section. The primary customers, referred to as simply the customer in this Plan, are the Navigation and Mission Design Branch (595) and the Flight Dynamics Facility (FDF).

The customer's primary goals and objectives as they relate to this effort are that the system be designed, built, tested, and delivered within the allotted time and within the allotted budget.

The primary customer point of contact for this effort is Randall Beckman, branch head of 595, and Sue Hoge, FDF Director. The PDL, Steve Hughes, will be the primary point of contact for all management- and budget-related interactions with the customer for this effort.

Customer-Supplied Elements

The following sections describe the agreements reached between the PDT and the customer on items that will be provided by the customer to the PDT.

What FDF Will Provide

The FDF shall provide the following items to the GMAT project:

- A list of missions for shadow operations that cover desired flight regimes.
- A list of required products and criteria for successful shadow operations for each mission.
- Review of GMAT Requirements Specification with feedback on missing but needed functionality.
- Delivery requirements negotiated with other in-house-built FDF systems such as GMSEC

What 595 Will Provide

Code 595 will provide the following items to the GMAT project:

- List of possible missions potentially to be the first to use GMAT from analysis through operations.
- Review of GMAT requirements specification with feedback on missing but needed functionality.

Required Resources

(* means already paid for)

- No unique facilities are required. GMAT is desktop software.
- Required Software
    - MATLAB
    - STK
    - FreeFlyer

    - Test Complete
    - Clarizen
- Required Hardware
    - Windows 7 Test Machine*
    - Windows XP Test Machine*
    - FDF machines for operational certification.