Updates per MSFC Software Audit findings dated 6/11/01 by George Mitchell

Stanford University

Relativity Mission

Software Quality Assurance Plan

P0630 Rev -

November 1, 2002

Contract No. NAS8-39225

Kelly Burlingham
SU Software Quality Engineer

Dorrene M. Ross
System Effectiveness Manager

Ron Sharbaugh
MOC Software Manager

Bill Bencze
Payload Electronics Manager

Mac Keiser
Chief Scientist

Gaylord Green
Program Manager

Document Revision Record

Document Title: Stanford University, Gravity Probe B Relativity Mission, Software Quality Assurance Plan

Document Number: **P 0630, Rev -**

Document Approved By NASA/MSFC:

Release Date:

REV	Date	Authorization for Change - ECO #	Section	Change Description
-	07/16/01			Initial release upon approval by MSFC.
	9/12/02		2.1	Software Documentation
	9/12/02		3	Added Reference Documents
	9/12/02		5.3	Augmented Documentation
	9/12/02		6.2	Changed Audits to meet the Direction from Marshal as well as the applicable Appendix
	9/12/02		6.6	Changed Requirements Traceability to meet the Direction from Marshal
	9/12/02		6.3	Added Data Archival
	9/12/02		6.12	Changed Problem Reporting and applicable Appendices to meet the Direction from Marshal.
	10/25/02		6.7	Added Minor Releases to the Release Process.

1 INTRODUCTION

This Software Quality Assurance Plan describes the organization and procedures used at Stanford University for the Gravity Probe B project to assure that the software developed and/or procured for Stanford University complies with all project technical and standards requirements. When properly implemented, the techniques outlined within this document will provide assurance that high quality software is delivered for ground and flight operations in accordance with project requirements.

This procedure is the planning and control document for Software Quality Assurance (SQA) activities performed on Stanford University software. It describes the responsibilities, specific procedures and techniques used for defining, formulating, implementing, and managing quality assurance activities throughout the Stanford University software developmental lifecycle.

2 APPLICABILITY

The SQA requirements, as specified in this plan, are applicable to Gravity Probe B project software within the following software categories: Stanford University developed deliverable software, sub-contractor developed software which interfaces with SU software, Operating System software, Science Data Analysis software, and other off the shelf software.

2.1 Software Developed by Stanford University

The SQA requirements imposed by this plan shall apply to the SU software developer, subprograms, and QA organization.

The teams described in paragraph 5.3 are responsible their applicable software development, release and testing. Each team is responsible for the overall work schedule milestones, the activity involved in carrying out the schedule and any risk areas. Members of the teams are responsible for the following as they apply:

- Ensure traceability of design and test cases to requirements
- Keep all databases up-to-date as work is completed
- Requirements
- Build list status
- Change Request status
- Action Item status
- Test cases
- Software Design Documentation as the software is developed or changed
- Database Design Documentation as the databases are developed or changed
- Released and controlled documents are the latest revision found on the GP-B server
- Testing
- Test plan
- Test procedures
- Test results are archived
- TAR/DR written for discrepancies
- Closing action items
- Complete design and test traceability to requirements

3 REFERENCE DOCUMENTS

Document Name	Document Number
T002 12 Science Requirements	
T003 System Design Performance Requirements	
SCSE 08 Volume I Software Management Plan	LMMS/F440425
SCSE 08 Volume II Software Development Plan	LMMS/F440426
SCSE 08 Volume III Software Test and Description	LMMS/F440427
SCSE 08 Volume IV Software Verification and Validation Test Reports and Procedures	LMMS/P086724
SU Science Mission Configuration Management Plan	P0098
SU Mission Operations Plan	MO-01
SU Mission Operations Specifications	S0457/MO-02
SU GP-B Data Management Plan	S0331
SU Post Processing Operations for Science Mission Data	S0401
SU MOC Configuration Management Plan	S0667
SU POD H/W and S/W Configuration	S0475
SU MOC Configuration Control, IONET LAN	S0476
SU MOC Configuration Control, Science LAN	S0477
SU GP-B Mission Operational Network Security Plan	LM P480289
SU MOC Software Development Plan	S0501
SU MOC Configuration Control Plan, Firewall to IONET	S0502
SU GPB Version Description Document Guide	P0937

4 Definitions

Acceptance Testing - Testing conducted to determine whether or not a system satisfies its requirements and to determine whether or not to accept the system.
Architectural Design - The process of defining the set of software components and their interfaces as the framework for a software system.
Audit - an independent review that addresses compliance with software development standards, procedures, and contractual agreements.
Baseline - A product that has been formally reviewed and approved that can be changed only through formal change control procedures.
Change Control - The process by which a change to a baseline is proposed, evaluated, approved or rejected, scheduled, and tracked.
Component - A distinct part or element of a computer software configuration item or software product.
Configuration - A set of software, documentation, and data elements that meets a set of requirements or contractual obligations. The totality of items making up a baseline.
Configuration Control - Configuration control is a process to provide the administrative mechanism for precipitating, preparing, evaluation, and approving or disapproving all change proposals throughout the system life cycle. That is, software configuration control is change proposal processing.
Configuration Identification - Configuration identification includes the specifications and their associated diagrams, flow charts, drawings, parts lists, etc., that are used to describe the functional and physical characteristics of an CSCI.
Computer Software Configuration Item (CSCI) – A collection of software elements treated as a unit for the purpose of configuration management.
Debugging - The process of locating, analyzing, and correcting suspected faults in software.
Defect – Any occurrence in a software product that is determined to be incomplete or incorrect relative to the software requirements and/or program requirements.
Defect Classification – The process where all defects identified during an inspection or tests are classified by severity and type.

Deliverable Software – The code and corresponding documentation that is turned over to the customer at specific points throughout the life of the contract.
Discrepancy – A formally documented deviation of an actual result from its expected result.
Discrepancy Report (DR) – An instrument used to record, research, and track resolution of a defect found in a baseline.
Design - The process of defining the software architecture, components, modules, interfaces, test approach, and data for a software system.
Error – A discrepancy between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition.
Failure – The behavior of the software or system component when a fault is encountered, producing an incorrect or undesired effect of a specified severity.
Fault - An accidental condition that causes a functional part of a software system to fail to perform a required function or to perform unwanted functions.
Firmware – The programmed instructions and/or computer data that reside in some form of storage element and are required for proper operation of a hardware unit. There are two common types: (1) firmware that requires an integral understanding of the hardware design and its operation, and/or is design implementation-dependent (e.g., machine instructions, control logic, etc.); and (2) firmware that implements system applications and/or support functions that do not fall within the limitations in (1) (e.g., database services, task scheduling, etc.), but is packaged in a form of Read Only Memory (ROM) for reasons such as performance, capacity, etc.
Functional Configuration Audit (FCA) - The FCA is a means of validating that development of a configuration item has been completed satisfactorily. FCAs shall be conducted on configuration items to assure that: a) Test/analysis data for configuration item verify that the configuration item has achieved the performance specified in its functional or allocated configuration identification; and b) The contractor maintains internal technical documentation that describes the physical configuration of each unit of the configuration item for which test/analysis data are verified.
Inspection - A formal technique in which s/w requirements, design, or code are reviewed in detail by a person or group other than the author to detect faults, violations of standards, and other problems.
Interface – A shared boundary across which information is passed; may be a hardware component, a portion of storage, or registers accessed by two or more computer programs.
Module - A part of a computer program that is separable and identifiable with respect to compiling, combining with other parts, and loading. A subroutine is an example of a module.
Multiple Operation Procedure – A procedure that would be considered a combination of 2 or more procedures or contains modules (sections) that perform multiple separate but related tests.
Nonconformance - Any deviation of hardware, software, or documentation from its functional, performance, or interface requirements or from the standards to which it is to be developed.
Performance – A measure of the ability of a computer system or subsystem to exercise its functions; for example response time, throughput, number of transactions, etc.
Physical Configuration Audit - The PCA is a means of establishing the product configuration identification used initially for the production and acceptance of configuration items. The PCA is used to establish an "As Built" configuration of the CSCI against the technical documentation that establishes a product physical configuration baseline. The PCA will also assure that the acceptance testing requirements prescribed by the documentation are adequate for acceptance of production units of a configuration item by the quality assurance activities.
Procedure Document (PDOC) – A document outlining work instructions, test operates, functional operations, etc.
Regression Testing - Re-testing done after program modification to verify that the modifications have not introduced faults or unintended adverse side effects.
Security - The protection of computer hardware and software from accidental and deliberate unauthorized access, use, modification, destruction, or disclosure.
Single Operation Procedure – Runs 1 test that involves 1 system only or 1 set of closely related functions.
Software - Computer programs, procedures, associated documentation and data, including firmware.
Software Change Request (SCR) - A form used to document errors, inconsistencies, or omissions discovered in a document, drawing, or procedure being evaluated for acceptability during a formal project review.
The Software Configuration Item - A defined software product that satisfies an end use function and is designated for configuration management.
Software Configuration Management (SCM) is a discipline applying technical and administrative direction and surveillance to (1) identify and document the functional and physical characteristics of software configuration items and baseline, (2) control changes to those characteristics, and (3) record and report change processing and

implementation status of software configuration items and baseline.
Software Development Folder - A formalized set of records documenting the development of a software unit. It includes schedules, reviews, approvals, and supporting documentation.
Software Life Cycle – The period of time that starts when a software product is conceived and ends when the product is no longer available for use. The software life cycle typically traditionally includes the following eight phases:

- Concept and Initiation Phase
- Requirements Phase
- Architectural Design Phase
- Detailed Design Phase
- Implementation Phase
- Integration and Test Phase
- Acceptance and Delivery Phase
- Sustaining Engineering and Operations Phase

Software Quality Assurance (SQA) is the planned, systematic process that ensures that desired procedures, standards, requirements, and quality attributes are:

- Established prior to software acquisition/development
- Followed during each phase of acquisition/development

Ultimately, the basic Software Quality Assurance function is to ensure that both software products and acquisition process comply with established standards, practices, and procedures.
Specification Document (SDOC) – A document giving detailed specifications, analysis, data summary, etc.
System - A collection of software programs organized to accomplish a specific set of functions or to meet a set of requirements.
Test Anomaly Report (TAR) – An instrument used for the documenting, dispositioning and reworking of a simple defect back to specification.
Test Documentation – The documentation describing the plans for, or results of, the testing of a system or component. Types include test incident report, test log, test plan, test procedure, and test report.
Test Plan - A document describing the approach to be taken for intended testing activities. The plan typically identifies the items to be tested, the testing to be performed, test schedules, personnel requirements, reporting requirements, evaluation criteria, the level of acceptable risk, and any risk requiring contingency planning.
Test Readiness Review (TRR) – A formal evaluation of the state of readiness of test hardware/software, support equipment, test facility, test requirements specification and procedures, and test operations.
Unit - Separately named and accessible elements of software, which perform specific functions. Also called subroutines, functions, procedures, or modules.
Validation - The process of evaluation of software to assure that it meets its requirements. It is normally done by reviews and testing.
Verification - The process of evaluating a system or component to determine whether the product of a given life cycle phase satisfies the conditions imposed at the start of that phase.

5 Organizations and Resources

5.1 Stanford University Facilities

This plan was developed in accordance with Stanford University, and will assure the establishment, implementation, and maintenance of a suitable software quality assurance program for the Stanford University Gravity Probe B project.

Software will be developed at Stanford University or at sub-contractors facilities. This software will be tested in the MOC area located on the second floor of the Stanford campus GP-B building. In addition to the Stanford MOC area there will be testing performed at the Lockheed Integrated Test Facility (ITF) and GSFC ground stations.

Communications links with the ITF and space vehicle during system test permit Stanford and GSFC to be active participants in software and space vehicle integration, test, data analysis, mission operation procedure checkouts, and launch rehearsals.

5.2 Organizational Structure

The Stanford University shall develop an organizational structure, which shall assure effective management and implementation of the SQA program. The SQA organization shall be defined such that it has the resources, responsibilities, authority and organizational freedom to perform objective evaluations of project activities, to identify problems and to recommend and/or implement appropriate action. The Stanford University shall designate one individual who shall have the responsibility and authority for directing and managing the SQA program. That individual shall have direct, unimpeded access to the development effort.

The software is divided into three systemic groups of development/testing and is as follows:

a) Ground Station. Ron Sharbaugh is the Team Leader.
b) GPS (Global Positioning System). Bill Bencze is the Team Leader.
c) Science Data. Mac Keiser is the Team Leader.

These teams may include contractors as deemed necessary by the team leader and the Program Manager. The team leaders are completely responsible for the software development and test activity.

5.3 Training

SQA personnel are trained to ensure project processes and products meet contractual and Stanford University software quality assurance requirements. Training needs are periodically assessed to determine requirements for additional training. This includes all training required for operators assigned to the Mission Operations Center.

Personnel developing and implementing the software quality assurance process shall be trained and/or experienced in software quality assurance. Software quality assurance training shall be obtained and/or originated and maintained as necessary for management, engineering, and assurance personnel. Records shall be maintained and readily available for review of the training, testing, and certification/re-certification status of personnel.

5.3.1 Certification

Certified personnel will perform all MOC operations. It is the responsibility of the MOC Manager to ensure operators are qualified to operate the computers in the MOC area. Training will consist of procedures, training videos and comprehensive testing.

5.3.2 Re-Certification

Re-certification will be performed as deemed necessary by the MOC Manager based on workforce turn over, software revisions, etc.

5.3.3 Training Records

Records of course completion and test results will be maintained by the MOC Manager.

6 PROGRAM REQUIREMENTS

6.1 Program Resource Allocation Monitoring

Stanford University Team Leaders shall monitor computer resource (memory and timing) allocation and utilization to assure that required spares or budgets for these resources are not exceeded. They shall notify Stanford University management when required budgets may be compromised.

6.2 SQA Program Audits

Stanford University QA shall perform semi annual and unannounced audits in order to effectively assess actual operations and conditions. As SU QA is a member of all software development teams, these audits will be restricted to documentation audits only. Stanford University QA will perform reviews of the peer reviews but will not perform line by line software code audits. Code audits will be conducted in the Peer Reviews phase of the Software Development Life Cycle. See Appendix F for the audit schedule.

Stanford University QA will perform audits to subcontractor's documentation only such as described below for SU developed software. SU reserves the right to request assistance for subcontractor audits from outside sources should the need arise.

Audits will include as a minimum:

a) Review all operations and associated procedures run during verification.
b) Review of test reports for operations run.
c) Review of anomaly and software discrepancy reports.
d) Review of TRR packages and test requirements.
e) Review of supporting development documentation, peer reviews, etc.
f) Review of as run verses as planned testing, such as during MOC SIMS.
g) Review completed test procedures for closure.
h) Verify software versions run and release status.

6.2.1 External Audits

Stanford University will be subject to periodic audits to assure compliance with approved SQA standards and procedures. These audits shall be conducted and chaired by personnel from organizations (MSFC, DCMC, etc.) external to Stanford University QA. Internal audits by the customer or the customer's designee.

6.2.2 Audit Reports

The assigned audit chairman will prepare an audit report, which documents the results of each audit. The report shall be forwarded to Stanford University management for review with recommendations for preventive and corrective action.

Audits performed under the requirements of the Stanford University Quality Assurance Plan will be documented and reported in accordance with these requirements. SQA shall prepare records, which contain the descriptions and results of all SQA audits, activities and evaluations required by this plan, including recommended preventive measures and corrective actions. These records shall be maintained for the life of the program.

6.3 Retrieval of Records

The System Effectiveness Manager shall ensure that software procurement, processing, inspection, and test records are related to the articles. It shall be organized so that these records and the related articles and materials may be located and retrieved in the event verification of, or removal of articles or materials become necessary. Hard copies of all documentation (e.g. Specifications, Manuals, etc.) and testing data of Flight-related Software will be stored in limited access area for five years after launch.

6.3.1 Documents

SU Document Control will archive all original documents in a location separate of the Document Control Center. This area will have limited access. The Document Control Center will retain copies and a list of all archived original documents.

Documents retained by the Document Control include, but are not limited to the following:

a) Released Procedures (PDOCS)
b) Released Science Documents (SDOCS)
c) Completed "As-Built" Procedures
d) Acceptance Data Packages

Deliverable Documents from Lockheed, such as Documentation Requirement SCSE 08, Software Development Plan (all Volumes) are to be forwarded via official submittal from Stanford University to MSFC.

6.3.2 Test Data

Test Data shall be delivered for archival in a location separate of the Document Control Center to a secure server location or stored on Tapes. It shall be archived along with all pertinent test information, including but not limited to:

a) VDD
b) Test Cases
c) TRR
d) Summary reports

6.4 SQA Status Reports

SQA shall provide the status of the SQA Program to the Stanford University on a periodic basis.
The status shall include information, such as:

a) Organization and key personnel.
b) SQA costs, recommendations and lessons learned.
c) Significant problems, their solutions and remedial and corrective actions.
d) SQA accomplishments, such as audits completed, participation in software life cycle reviews, etc.
e) Supplier SQA accomplishments, if applicable, plus acceptance and rejection rates of any supplier produced software products.

6.5 Software Documentation

SQA shall review all baseline software documentation prior to test readiness reviews and formal implementation. The Software Team Leader is responsible for the objective criteria and guidelines for documentation reviews. SQA shall review subsequent changes to baselined documentation.

6.6 Requirements Traceability

SQA shall assure that traceability exists between the Contract-End-Items (CEI) Specification and/or appropriate database, Software Requirements Specification, Software Design Specification, and Software Test Procedures. The FoxPro Database on the Hardware drive will house all requirements and specifications as well as the Test Cases. Test Cases will also be documented in either the test case Pdocs or in the SU Version Description Guide Sdoc.

Requirements satisfied by test cases will be additionally called out in TRR packages to include the requirement and the test case that maps to it.

6.7 Software Development Process

The Software Development Process shall follow a modified Waterfall Model. Phase one shall consist of a Design Review with Design Documents complete with code called out to be designed or modified, requirements satisfied, proposed test cases, as well as any Action Items, SCRs or MCRs to be closed with the new code. Phase Two is Code and Code review by Peers. Phase Three is to Unit Test, System Integration Test, and Regression Test as applicable. Phase Four is to release the software complete with a Version Description Document. Small releases with the sole purpose of minor bug fixes may be released via and abridge Waterfall method finalized with a Release Memo instead of a Version Description Document.

SQA shall perform audits throughout the software development process to assure compliance with the Project Plan, standards and released procedures. The Software Team Leaders are responsible for developing methods of software back-ups to be performed on a periodic basis.

6.8 Development Tools and Techniques

Software Team Leaders are responsible for defining and documenting all tools and techniques to be used in the development and testing processes for their respective software. SQA shall verify all development and test tools are documented, tested/evaluated, and approved prior to implementation and use.

Note: This applies only to software developed at Stanford University; it does not apply to COTS software.

6.9 Software Configuration Management

Software Configuration Management plans, standards, procedures, and activities shall be reviewed and monitored by SQA to assure that the system is in compliance with appropriate Stanford University program requirements. SQA shall be responsible for evaluating and approving proposed changes to baselined software documentation.

There will be primarily two different types of released documents associated with the development, testing and analysis of Stanford University generated software.

6.9.1 Procedural Document (PDOC)

SU's Procedures (PDOC) describe activities such as: development, integration, testing and so on and are referenced in travel sheets, test plans and test readiness reviews. The list of the Released Procedures will establish the Baseline for Procedures. This list of Procedures is stored in the GP-B Configuration & Test Database shown under "Documents" and then subcategory "Procedures", along with their electronic copies (when available). All Procedures are numbered in sequential order and are provided by and the signed off hard copy will maintained by the Document Control Center.

The release process is as follows:

a) Operation and Integration type Procedures shall have a cover sheet signed and dated by the minimum of the persons listed below:
   - The author of the procedure, the Responsible Team Leader, the Software Quality Engineer, and when required by the Safety Engineer.

b) Test Procedures shall have a cover sheet signed and dated by the minimum of the persons listed below:

- The author of the procedure, the Responsible Team Leader, the Chief System Engineer, the Software Quality Engineer, and when required the Safety Engineer.

c) All Procedures that are contract required must be approved by MSFC, i.e. the Software Quality Plan, etc.
- SU shall submit the procedure to MSFC for approval and a copy of the submittal letter will be attached to the file copy of said procedure.

d) All revision changes to released PDOCS will be done using an Engineering Change Order (ECO) as defined in the SU Configuration Management Plan, P0098.

e) PDOCS that are As Builts, As Runs, Regression Tests, or QA approved configuration deviations for Mini Sims will be kept on file in the MOC Library.

6.9.2 Science Document (SDOC)

"Science Documents" describe software content, configuration control, methodology, test readiness, analysis, requirements specifications, etc. There are two types of SDOCs; those that define specification requirements and those that do not. The initial release process is the same for both types of SDOCs. The document shall have a cover sheet (similar to a Operations and Integration Procedure**,** signed and dated by the minimum of the persons listed below, the author of the document, the Responsible Team Leader, System Effectiveness Manager or designee, and others as requested by the Team Leader.
The definition of types, maintenance, control and revision of these documents are as follows:

a) All SDOCs that define "specifications" (for example S0457/MO-02, Mission Operations Specifications) will be maintained in the GP-B Configuration & Test Database located on the SU GP-B server under "Documents" and then subcategory "Specifications". All specification type SDOCs are numbered in sequential order (such as MO-01, MO-02, etc.) when stored in this database. A link to the SDOC number will be shown in database for each specification type document. The signed off hard copy will maintained by the Document Control Center. The electronic copy will be stored in the GP-B Configuration & Test Database located on the SU GP-B server under "Documents" and then subcategory "Science Documents". A PCB will be use when making revision changes to all SDOCs that define specifications.

NOTE: This requirement becomes effective once this procedure is approved. All changes to the existing database documents will reflect an SDOC number in addition to the original specification number.

b) All other SDOCs will be reviewed and approved by the individuals listed on the cover page. The signed off hard copy will be maintained by the Document Control Center. The electronic copy will be stored in the GP-B Configuration & Test Database located on the SU GP-B server under "Documents" and then subcategory "Science Documents". The original approvers, their designees, or their replacements will approve revisions of all these SDOCs. The revision block will reflect all changes made to the original document.

6.10 Test Readiness Reviews

Prior to a system level verification, the team will generate a Test Readiness Review package for approval by the review committee. The committee will consist of System Engineering (the chairman), the Team Leader (co-chairman), Quality, the customer, and any other parties deemed necessary by the chairman.

Specific subjects and tasks covered by the TRR are as follows:

a) Test Requirements and Specifications: Identify and review the current revision of the baseline test requirements and specifications. Assess the proposed test program and methods for compliance with objectives, intent, and philosophy. Assure that all prerequisite test and pretest analysis has been successfully completed or acceptable waivers/deviations exist.

b) Test Documentation: Determine the status of the test documentation and test related preparation activities such as facility activation and test and checkout procedures. Assess the procedures against the approved test requirements. Assure that test documentation has been reviewed and approved.
c) Facility: The requirements, which the test facility must meet, will be reviewed and it must be shown that the facility has been configured and certified to meet these requirements.
d) Test Article Configuration: The as built versus the design configuration of the test article will be reviewed by assessing all deviations and waivers.
e) Test Equipment: Assure that all equipment utilized in performing the test meets requirements and is ready to support the test.
f) Test Team Certification: Personnel training and certification shall be reviewed. Adequate staffing levels shall be assured.
g) Hazard Analysis: The approved hazard analysis and resulting open item shall be summarized.
h) Open Work: Open work items shall be reviewed to assure satisfactory completion prior to test.

The committee will:

a) Evaluate the state of readiness of test hardware/software, support equipment, test facility, test requirements specification and procedures, and test operations.
b) Provide authority to commence test and identify specific constraints to start tests as appropriate.
c) Assign action items to appropriate organizations. Designate as “constraint to test” or “no constraint to test”.
d) Document results of the TRR including authorization to proceed and approval by the chairman and co-chairman.

SQA shall participate in formal reviews to verify that the documentation to be presented has been reviewed per the requirements of the TRR.

SQA shall resolve any discrepancies or action items related to the SQA program or implementing plans, standards, or procedures and update the SQAP as necessary.

6.11 Software Testing

SQA shall witness the software testing to verify compliance with approved plans and procedures. Specifically, SQA shall:

a) Assure the software to be tested is the correct version.
b) Witness tests to assure that testing is performed in accordance with Stanford University approved test plans and procedures.
c) Review all test results to certify actual results meet established acceptance criteria.
d) Assure all nonconformities are documented and tracked to resolution.
e) Assure that test reports are generated at the conclusion of software testing, if required by Stanford University project requirements.
f) Assure all changes to the software are made in accordance with approved Stanford University software configuration management procedures.

6.11.1 Software Upgrade Process

The policy for retest and penalty testing after software changes have been made to both Flight software and GSE software are as follows:

Flight Software: Any procedure that is started with one set of flight software must be completely re-run if the flight software is changed before the completion of the procedure. Exceptions will be dependent upon testing performed as part of closure to a SCR. Verification of the acceptance of the module change re-test is grounds to consider waiving a complete re-test.

Ground Support Equipment (GSE) Software or Test Set Software:

a) Single operation procedure: Complete re-run of the procedure if the test set changes after procedure has been started.
b) Multiple operations procedure: A committee consisting of the SEM, Payload Electronics Manager, Payload Program Manager, System Engineer and Program Manager (as requested) must define applicability of test set software to each segment of the procedure. The re-test criteria will consist of the minimum data as follows:
    1. List Pdoc and all segments/modules.
    2. For each module complete the status.
    3. For each module show the detailed relationship to the test set being changed.
    4. For each module state whether re-test is recommended - yes or no.
    5. For each module state whether penalty testing is recommended - yes or no.
    6. For each module requiring penalty testing, define the penalty test.

6.12 Problem Reporting and Corrective Action

SQA shall assure that a closed loop system for reporting, analyzing, tracking and resolving problems and deficiencies are established.

6.12.1 Discrepancy Log

The first level of problem reporting will be the Discrepancy Log form (D-Log). See Appendix A. A D-Log will be generated for each test anomaly found during any formal testing phase. The Team Leader and System Effectiveness Manager will review all test anomalies reported on this form. The form will be sequentially number and recorded in the Discrepancy logbook. See Appendix B. The Team Leader and the System Effectiveness Manager will be responsible for determining the severity of the anomaly and its resolution.

6.12.2 Software Change Request (SCR)

If it is determined that the software must be changed, a Software Change Request (SCR) form (Reference Appendix D) will be completed and submitted to the appropriate developer. Completion of the change will prove evidence that the discrepancy has been reworked and the D-Log can then be closed.

SQA shall assure provisions exist for:
a) Reporting problems and deficiencies to proper technical and management levels.
b) Analysis of problem reports to identify the scope, causes, and impacts of software problems and deficiencies.
c) Identification of corrective action and preventive measures along with an implementation method and schedule.
d) Tracking of identified problems and deficiencies to assure disposition and closure.

The SCR shall be tracked in the FoxPro Database. The information tracked will include but not be limited to:
a) Descriptions of the discrepancy
b) Subsystem
c) Change Type
d) Severity
e) Corrective Action
f) Test cases run for validation of the Corrective Action

6.13 Trend Analysis

SQA shall identify and conduct trend analysis, where feasible, on flight and ground software whose anomalous performance could have a detrimental effect on safety or mission success. The FMEA may be used for selection of candidate items for trend analysis.

6.14 Supplier Controls

The Stanford University GP-B Program is responsible for the adequacy and quality of all software, associated documentation and services procured from suppliers for the project. This includes non-deliverable software that is used to accept deliverable hardware and/or software. SQA shall assure acceptable supplier performance by:

a) Supporting development of software quality requirements for the contract.
b) Participating in the selection of procurement sources.
c) Performing/participating in periodic surveys of each supplier's facility and software quality system to determine their capability of satisfying software quality requirements. A pre-award survey is not required if the supplier has a previous and continuous record of supplying quality software, documentation and services of the type being procured.
d) Assuring the existence or development of a complete set of software requirements for supplier developed software.
e) Examining supplier software development and quality assurance procedures and methodologies to determine compliance with contract requirements and approved standards and procedures.
f) When possible, auditing of supplier software development and quality assurance activities to assure compliance with approved procedures and methodologies.

6.15 Commercially Available, Reusable, and Government Furnished Software (GFS)

If these types of software are used as part of a deliverable product, SQA shall, prior to implementation and use:

a) Review accompanying user documentation to assure it is complete, accurate, and sufficient to meet documentation requirements for deliverable software.
b) Ensure the software has been tested/evaluated to verify its characteristics are consistent with established software requirements.

Commercial off-the-shelf (COTS) products will be procured and integrated into the software development and test environment. These products form the basis for Stanford University development and are upgraded and extended for software development and testing.

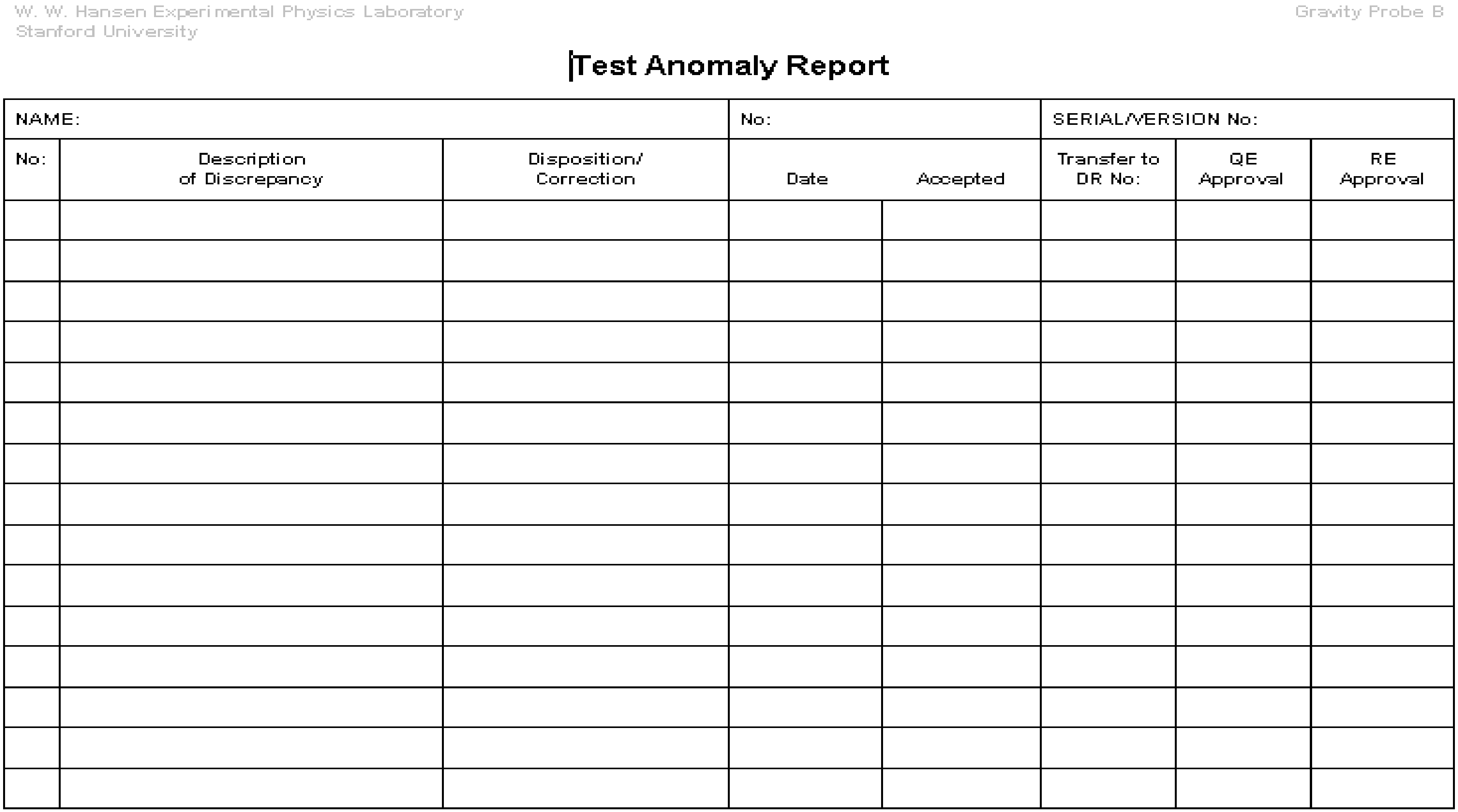
6.16 Acceptance and Delivery of Software to the Customer

SQA shall be responsible for certifying, prior to delivery:

a) Delivered software is the same configuration as that tested and accepted.
b) No open problems or that all open problems have been clearly identified.
c) All test results for the delivered software meet established acceptance criteria.
d) All requirements have been verified by approved methods.
e) Acceptance data package, if applicable, contains all required documentation.

Appendix A

Example of a Discrepancy Log (D-Log)



W. W. Hansen Experimental Physics Laboratory
Stanford University

Gravity Probe B

Test Anomaly Report

NAME:			No:		SERIAL/VERSION No:		
No:	Description of Discrepancy	Disposition/ Correction	Date	Accepted	Transfer to DR No:	QE Approval	RE Approval

Appendix B

Example of Software Discrepancy Report (SDR)

See GP-B database to Software Discrepancy Folder and Forms.

Appendix C:
Yearly Audit Schedule

#	Work Area	RE	Next Audit
1	GPS	B. Bencze	First Quarter 2003
2	Ground Station Operations	R. Sharbaugh v	First Quarter 2003
3	Science Data Analysis	M. Keiser	First Quarter 2003
1	GPS	B. Bencze	Third Quarter 2003
2	Ground Station Operations	R. Sharbaugh	Third Quarter 2003
3	Science Data Analysis	M. Keiser	Third Quarter 2003
1	GPS	B. Bencze	First Quarter 2004
2	Ground Station Operations	R. Sharbaugh	First Quarter2004
3	Science Data Analysis	M. Keiser	First Quarter 2004

Subcontractor audits

#	Subcontractor	Date
1	LMMS	First Quarter 2003

External audits

#	Audit type	Date
1	MSFC annual Audit	N/A
2	ONR annual audit	N/A