NASA Contractor Report 182045

SOFTWARE QUALITY ASSURANCE PLAN FOR GCS

Stephen E. Duncan
Elizabeth K. Bailey

RESEARCH TRIANGLE INSTITUTE
Research Triangle Park, North Carolina

Contract NAS1-17964
May 1990

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665-5225

SOFTWARE QUALITY ASSURANCE PLAN FOR GCS

Guidance and Control Software

RTCA DO-178A Document Number 5B

Release number: 1.0

Prepared for:

NASA-Langley Research Center under contract
NAS1-17964; Task Assignment No. 8.

Prepared by:

Author(s): RTI – Stephen E. Duncan
Consultant – Elizabeth K. Bailey
Reviewer(s): RTI – Anita M. Shagnea
Janet R. Dunham

NASA – George B. Finelli

Software R & D Department
Center for Digital Systems Research
Research Triangle Institute
Research Triangle Park, North Carolina 27709

ACKNOWLEDGEMENT

We are grateful to Elizabeth Bailey, who wrote the initial drafts of this document, and to Leslie Dent, whose reviews and comments throughout the writing of this document largely determined its final organization and contents.

Preface

The Software Quality Assurance Plan for GCS is document # 5B in a series of fifteen documents which fulfill the Radio Technical Commission for Aeronautics RTCA/DO-178A guidelines, "Software Considerations in Airborne Systems and Equipment Certification [1]." The documents are numbered as specified in the DO-178A guidelines. The documents in the series are used to demonstrate compliance with the DO-178A guidelines by describing the application of the procedures and techniques used during the development of flight software. These documents were prepared under contract with NASA-Langley Research Center as a part of their long term research program addressing the fundamentals of the software failure process.

This project consists of two complementary goals: first, to develop software for use by the Research Triangle Institute (RTI) in the software error studies research program sponsored by NASA-Langley Research Center [3]; second, to use and assess the RTCA/DO-178A guidelines for the Federal Aviation Administration (FAA). The two goals are complementary in that the use of the structured DO-178A guidelines in the development of the software will ensure that the test specimens of software have been developed according to the industry standards for flight critical software. The error studies research analyses will then be conducted using high quality software specimens.

The implementations will be subjected to two different software testing environments: verification of each implementation according to the RTCA/DO-178A guidelines and replicated random testing in a configuration which runs more than one test specimen at a time. The term *implementations* refers to bodies of code written by different programmers, while a *version* is a piece of code at a particular state (i.e., version 2.0 is the result of code review). This research effort involves the gathering of product and process data from every phase of software development for later analysis. More information on the goals of the Guidance and Control Software (GCS) project are available in the *GCS Plan for Software Aspects of Certification.*

The series consists of the following documents:

- *GCS Configuration Index* Document no. 1
- *GCS Development Specification* Document no. 2

- *GCS Design Descriptions* One for each software implementation. Document no. 3

- *GCS Programmer's Manual* Document no. 4, includes Software Design Standards, document no. 12.

- *GCS Configuration Management Plan* Document no. 5A

- *Software Quality Assurance Plan for GCS* Document no. 5B

- *GCS Source Listing* One for each software implementation. Document no. 6

- *GCS Source Code* One for each software implementation. Document no. 7

- *GCS Executable Object Code* One for each software implementation. Not available on hardcopy. Document no. 8

- *GCS Support/Development System Configuration Description* Document no. 9

- *GCS Accomplishment Summary* Document no. 10

- *Software Verification Plan for GCS* Document no. 11

- *GCS Development Specification Review Description* Document no. 11A

- *GCS Simulator (GCS_SIM) System Description* Document no. 13

- *GCS Simulator (GCS_SIM) Certification Plan* Document no. 13A

- *GCS Plan for Software Aspects of Certification* Document no. 14

Contents

List of Figures

1 Project Background

Development of the Guidance and Control Software (GCS) has been undertaken as part of a series of studies to characterize the software failure process and provide data on which to base the development of methods for assessing software reliability. For the current study, three implementations of GCS will be designed, coded, and tested. The three implementations are based on a common software requirements specification. One programmer and one tester are assigned as a team to develop each implementation independently of the other two teams. All three implementations will undergo the design, code, and test phases described in this plan. Additional details about the rationale underlying the study can be found in the *GCS Plan for Software Aspects of Certification[7]* and the *Software Verification Plan for GCS*[2].

2 SQA Function

2.1 Purpose

The purpose of the Software Quality Assurance (*SQA*) function is to promote product quality by ensuring that all development, verification, and configuration management activities and products adhere to published policies, procedures, and standards. It is not the responsibility of the SQA function to generate these policies, procedures, and standards but rather to ensure that they are enforceable and that they are followed.[1]

2.2 SQA Organization

The *Software Quality Assurance Plan for GCS* (henceforth referred to as *the Plan*) was written by an outside consultant and an on-site representative. The original organization had the consultant serve as the SQA Director and report to the NASA Contract Monitor, and had the on-site representative, who actually conducts the audits and other SQA activities, report to the SQA Director. Due to changes in staffing considerations, the on-site representative now reports directly to the Contract Monitor, and has subsumed

[1]The policies, procedures, and standards are presented in the RTCA/DO-178A documents for GCS. For a list of these documents, see the Preface.

the responsibilities of the SQA Director. Reporting relationships are shown in Figure 1.

2.3 Scope and Organization of the *Software Quality Assurance Plan for GCS*

The *Software Quality Assurance Plan for GCS* outlines all procedures, controls and audits to be carried out by the SQA organization to ensure adherence to documented procedures and standards. The Plan was written according to the guidelines contained in RTCA/DO-178A [1]. It is assumed that all GCS functions are classified in the DO-178A *critical* category and, hence, GCS represents *Level 1* software. All quality assurance activities and reports described in this plan are intended to support this level of software certification.

The Plan is organized by lifecycle phase. For the GCS project, there are three development phases and four test phases. The development phases include Software Requirements, Design, and Code. For each of these phases, the Plan lists the document that is produced, the associated verification activities, all applicable and documented standards and procedures (including procedures governing conduct of verification activities), and SQA's role in ensuring adherence to those standards and procedures.

The Test Phases include Unit Test, Sub-Frame Test, Frame Test and System Test (definitions of these are found under the appropriate headings). Conduct of these tests is governed by the *Software Verification Plan for GCS*[2]. For each test phase, the Plan gives a brief description of the testing to be conducted and of the applicable policies contained in the *Software Verification Plan for GCS*. There is also a description of the Test Readiness and Test Completion Reviews to be conducted by the SQA representative prior to and following each test phase.

The SQA representative is responsible for ensuring that all problems identified during the various verification activities are documented and corrected and that all change control procedures are followed. The Plan contains a section on problem reporting and correction as well as a section on software configuration management (SCM).

Finally, the SQA representative is responsible for reviewing all deliverable documentation for adherence to project standards and to the DO-178A

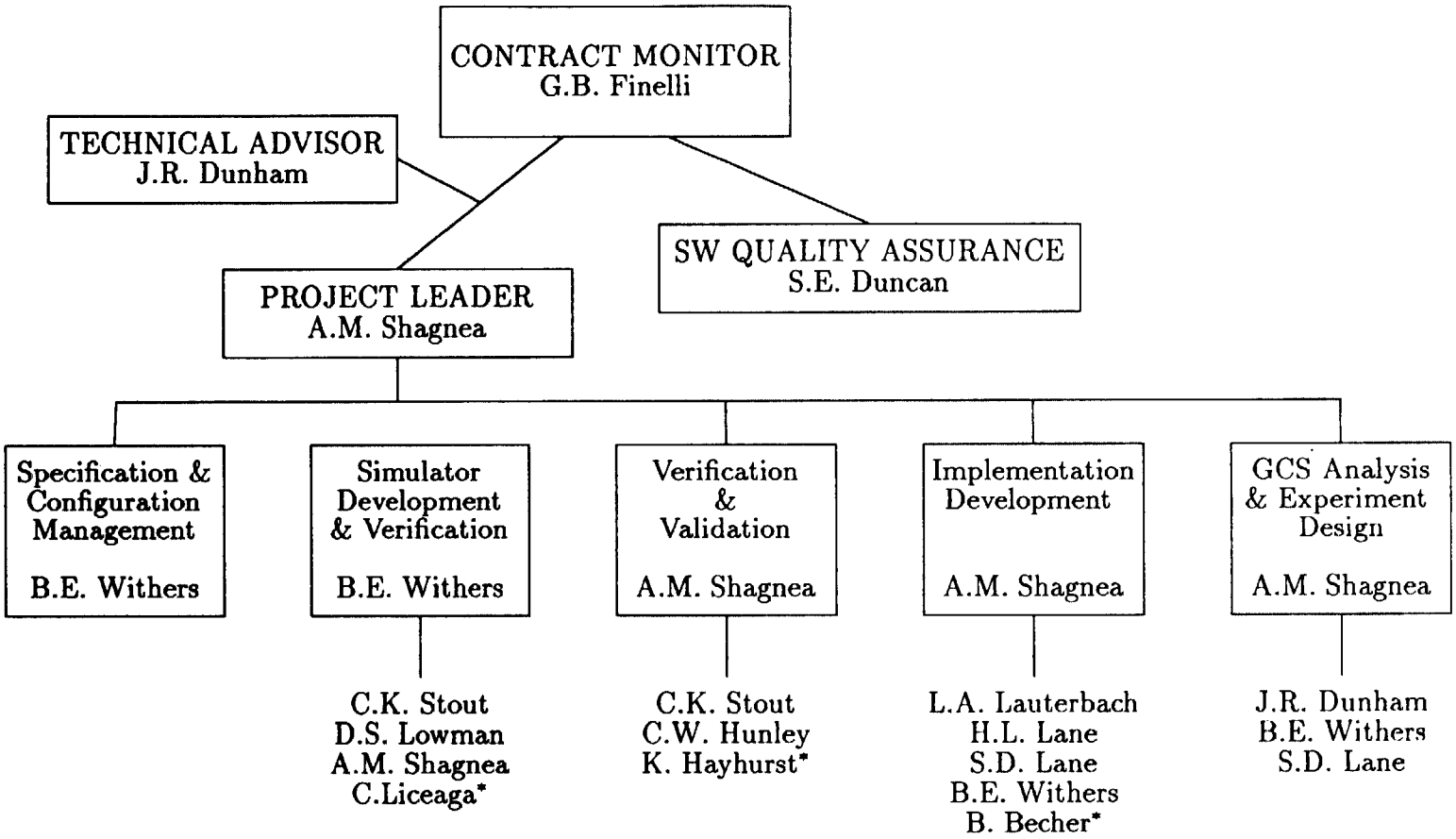

Figure 1: GCS Organizational Chart

*NASA LaRC

guidelines. The final section of the Plan summarizes the set of reports and approvals to be provided by the SQA representative.

3 Development Phases

3.1 Software Requirements

3.1.1 Document Produced

Software requirements are contained in a document entitled *Guidance and Control Software Development Specification* (RTCA/DO-178A Document no. 2)[8].

3.1.2 Applicable Standards

The software requirements were generated using a method which entails the use of data flow diagrams along with control flow information. This method is described in *Strategies for Real-Time System Specification.*[4]

3.1.3 Validation

The software requirements were subjected to a verification process which was outside of the formal verification procedures carried out by the Test organization. The correctness and completeness of the requirements were verified three ways: conducting walkthroughs and peer reviews, coding two prototype programs from the requirements, and applying a CASE tool to the requirements. The results of the requirements review are summarized in the *GCS Development Specification Review Description* (RTCA DO-187A Document no. 11A).

3.1.4 Quality Assurance

The *GCS Development Specification* was written and validated independently of the Software Quality Assurance group. Only modifications to the specification, driven by problem reports, are subject to review by the SQA representative.

3.2 Design

3.2.1 Document Produced

Each of the three independently-produced GCS designs will be contained in documents entitled *GCS Design Description Document.*

3.2.2 Applicable Standards

Design standards and guidelines are contained in the following project documents, found in the *GCS Programmer's Manual* [6]:

Programmer Instruction no. 6: Design Documentation Outline
: This instruction contains a suggested outline for the design document. While strict adherence to the outline is not required, all items listed in the outline must be addressed in the design document.

Programmer Instruction no. 7: Design Standards
: This instruction outlines requirements concerning the design method and a specific tool to be used. Specifically, structured design methods as described by Hatley and Pirbhai [4] are required along with the use of the CASE tool team*work*[2].

Programmer Instruction no. 5: The Use of Error Handlers
: This document contains guidelines concerning the use of error handlers.

3.2.3 Verification

For the GCS, the Preliminary Design Review (PDR) and Critical Design Review (CDR) will be replaced by a single design review. This is feasible because the total number of executable lines of code is only about 2,000, enabling architectural and detailed design to be carried out for each version by individual programmers.

The purpose of the design review is to verify that the requirements have been correctly translated into the design and that design standards have been followed. The procedures to be followed during the review are outlined in the *Software Verification Plan for GCS.* That document also contains

[2]Team*work* is a registered trademark of Cadre Technologies, Inc.

the GCS Requirements Traceability Matrix, which is used to verify that all requirements are addressed by the design, and the GCS Design Review Checklist, which is used to verify that the design adheres to all applicable standards.

3.2.4 Quality Assurance

The SQA representative will attend the Design Review and will be responsible for filling out the GCS Requirements Traceability Matrix and a GCS Design Review Checklist for each review session. The SQA representative will also make sure that a problem report is filled out for all cases of missing or excess functionality or for non-conformance to standards. The SQA representative may grant a waiver for minor cases of non-conformance by initialing the appropriate items on the GCS Design Review Checklist. The SQA representative will keep minutes of the review which will list the generated problem reports, comments, and any action items that do not merit a problem report.

Approval from the SQA representative is required before a programmer is permitted to transition from the Design Phase to the Coding Phase. Before this approval will be granted, all problems identified during the Design Review must be corrected and all action items noted in the minutes must be acted upon. The design changes (or specification changes) made in the course of these corrections must be documented on the problem report and approved by the SQA representative.

The SQA representative generates a Design Review Report, which contains a summary of SQA activity, the minutes of the design reviews, and the GCS Design Review Checklist from each session.

3.3 Code

3.3.1 Document Produced

The GCS FORTRAN code listing will become RTCA/DO-178A Document no. 7, *GCS Source Listing*, and will be under the control of Digital Equipment Corporation's Code Management System (CMS) as described in Section 7.0 *(Software Configuration Management)* of this document.

3.3.2 Applicable Standards

Coding standards and guidelines are contained in the following documents:

VMS FORTRAN Code Generation Guidelines
This document contains coding standards designed to improve code readability.

Programmer Instruction no. 3: Coding Standards for GCS Applications
This document contains modifications to the VMS FORTRAN Code Generation Guidelines, including requirements about the notation of on-line software problem reports and code changes and about the documentation of embedded error handling.

3.3.3 Verification

The purpose of code reviews is to verify that the code has properly implemented the requirements and design as specified and that it meets coding standards. Code reviews are scheduled after all units (a unit consists of a single function or subroutine) have been written and compiled without errors (but not executed).

The procedures to be followed during the review are outlined in the *Software Verification Plan for GCS*. That document also contains the GCS Requirements Traceability Matrix, which is used to verify that all requirements are addressed by the code, and the GCS Code Review Checklist, which is used to verify that the code adheres to all applicable standards. A problem report is filled out for all cases of missing or excess functionality or for non-conformance to standards. The SQA representative may grant a waiver for minor cases of non-conformance by initialing the appropriate items on the GCS Code Review Checklist. If it is determined that a problem originates in the design, the programmer is responsible for filling out a problem report prior to making the design correction and generating a second problem report for the code. All completed problem reports must be approved by the SQA representative.

3.3.4 Quality Assurance

The SQA representative will attend all code reviews and will be responsible for filling out the GCS Requirements Traceability Matrix and the GCS Code Review Checklist. The SQA representative will keep minutes of the review, listing the problem reports generated, any comments, and any action items generated during the review.

Approval from the SQA representative is required before a programmer is permitted to procede to the Unit Test Phase. Before this approval will be granted, all problems identified during the Code Review must be corrected, and all action items settled.

The SQA representative generates a Code Review Report, which contains a summary of SQA activity, the minutes of the code reviews, and the GCS Code Review Checklist from each session.

4 Test Phases

For all test phases, each tester will be required to execute the appropriate test cases within DTM.[3] DTM will act as a test log. A problem report must be filled out whenever the expected result does not match the actual results. The GCS Requirements Traceability Matrix will be used to cross reference test cases and problem reports to the requirements.

The programmer is responsible for making code changes to correct all problems. If it is determined that a problem originates in the design, the programmer must fill out a problem report prior to making the design correction and generate a second problem report for the corresponding code changes. This second report must be completed by the programmer after the code corrections have been made. All completed problem reports must be approved by the SQA representative.

4.1 Unit Test

The programmer is free to use any testing method on the code provided a minimum of 20 test cases are executed per sub-frame, including three for

[3]The use of the DEC/Test Manager (DTM) is described in the *Software Verification Plan for GCS.*

each unit.

During the Unit Test, the programmer will have his own CMS library as described in the *GCS Configuration Management Plan* [9]. The programmer will be required to keep a GCS Unit Test Log which will list each test case, its expected result, and its actual result. (A copy of the GCS Unit Test Log is contained in the *Software Verification Plan for GCS.*)

During the Unit Test, no approval will be required to make any necessary changes to the code but the reason for the change must be documented in a comment for the CMS library according to the procedures described in *Programmer Instruction #4: Change Management.* If the programmer makes any changes to the test cases (including the addition of new cases), he must document the reasons for these changes in the test log.

4.1.1 Quality Assurance for Unit Test

Prior to the start of Unit Testing, the SQA representative will verify that all problems identified during the Code Review have been corrected (i.e., all problem reports have been completed and approved by the SQA representative). The SQA representative will also verify that the programmer has a GCS Unit Test Log which documents each test case along with its expected result and that the minimum number of test cases required by the *Software Verification Plan for GCS*, have been specified.

At the conclusion of Unit Test, the SQA representative is responsible for holding a Test Completion Review. The procedures for this review are as follows:

1. The SQA representative checks the GCS Unit Test Log to ensure that

   (a) the expected results are recorded for each test case (with any appropriate calculations).

   (b) the actual test results are recorded for each test case.

   (c) the reasons for any changes to the test cases (including the addition of new cases) are documented in the test log.

   (d) all discrepancies between actual and expected test results have been documented in a problem report and that the problem report number is contained in the test log.

(e) all problems are corrected (i.e., on the final test run, all test cases were associated with the expected result).

(f) a minimum of three test cases per unit for a total of twenty per sub-frame were executed.

2. The SQA representative verifies that all problem reports have been completed and approved.

3. The SQA representative also checks the programmer's CMS library to ensure that comments are included with all code changes describing the reason for the change along with the associated problem report identification number.

4. The SQA representative produces the Test Completion Review Report for Units, signifying approval that the Unit Test Phase is complete. Once this approval has been obtained, the source code will enter the general CMS library. After that point, a programmer will need SQA approval to make any code changes.

4.2 Sub-Frame Test

For this project, Sub-Frame Testing is roughly equivalent to Module Testing as described in RTCA/DO-178A [1]. Both black-box (requirements-based) and white-box (structure-based) testing will be performed on the sub-frame level. White-box testing will be performed first followed by black-box testing. The *Software Verification Plan for GCS* contains the detailed set of procedures to be followed for these tests.

4.2.1 White-Box Testing

The white-box test cases for each of the three GCS implementations will be designed by the tester assigned to that implementation. A well-known path analysis technique will be used to identify linearly independent paths and to generate test cases to achieve 100% multiple-condition coverage (see the *Software Verification Plan for GCS* for details).

4.2.2 Black-Box Testing

The three testers will design the black-box test cases together. The actual testing of each implementation will be carried out independently by the tester assigned to that implementation. Every requirement will be covered by at least one test case.

4.2.3 Quality Assurance for Sub-Frame Testing

Prior to the start of Sub-Frame Testing, the SQA representative is responsible for conducting a Test Readiness Review, where the SQA representative will verify that:

1. separate versions have been and will be maintained for the black box version, white box version, and baseline version (used in regression analysis).

2. both the white-box and black-box test cases are documented, including all inputs and expected results, and entered in the DTM.

3. the set of white-box test cases meet the coverage criteria outlined in the *Software Verification Plan for GCS.*

4. there is at least one black-box test case for each requirement. The SQA representative will fill in the GCS Requirements Traceability Matrix with the identification number of the test case associated with each requirement.

At the conclusion of Sub-Frame Testing, the SQA representative is responsible for conducting a Test Completion Review according to the following procedures:

1. The SQA representative will check DTM for white-box testing and for black-box testing to ensure that the actual test results are recorded for each test case.

2. The SQA representative will verify that all changes to the test cases (including the addition of new cases) are documented in a problem report.

3. The SQA representative will verify that all discrepancies between actual and expected test results have been documented in a problem report and that the problem report number is contained in the GCS Requirements Traceability Matrix (for black-box testing).

4. The SQA representative will verify that all problems are corrected (i.e., on the final white-box test run and on the final black-box test run, all test cases produce the expected result).

5. The SQA representative will verify that all problem reports have been completed and approved.

6. The version of code which is passed on to the Frame Test phase will incorporate corrections from both the white-box and the black-box tests, *via* the following procedures.

   (a) The version of code with no corrections will serve as the baseline version.

   (b) For each problem report form on the white-box testing, search for a corresponding problem report from the black-box testing.

   (c) All fixes described for that particular problem will be combined and put into the code.

   (d) Repeat the procedure for all white-box problem reports.

   (e) If any black-box problem reports have not been used, apply their fixes to the code.

7. The SQA representative will verify that all corrections have been properly incorporated by witnessing execution of all white-box and black-box tests on the final version of code as describe in the *Software Verification Plan for GCS.*

8. The SQA representative will produce the Test Completion Review Report for Sub-Frames, signifying approval that the Sub-Frame Test Phase is complete.

4.3 Frame Test

For this project, Frame Testing is roughly equivalent to Integration Testing as described in RTCA/DO-178A [1]. The three testers will design the frame tests together. Each requirement will be covered by at least one test case. The tests will be executed on each implementation independently by the appropriate tester.

4.3.1 Quality Assurance for Frame Testing

Prior to the start of Frame Testing, the SQA representative is responsible for conducting a Test Readiness Review according to the following procedures.

1. The SQA representative will verify that the test cases are documented, including all inputs and expected results, and entered in DTM.

2. The SQA representative will verify that there is at least one test case for each requirement. The SQA representative will fill in the GCS Requirements Traceability Matrix with the identification number of the test case associated with each requirement.

At the conclusion of Frame Testing, the SQA representative is responsible for conducting a Test Completion Review according to the following procedures:

1. The SQA representative will check DTM to ensure that the actual test results are recorded for each test case.

2. The SQA representative will verify that all changes to the test cases (including the addition of new cases) are documented in a problem report.

3. The SQA representative will verify that all discrepancies between actual and expected test results have been documented in a problem report and that the problem report number is contained in the GCS Requirements Traceability Matrix.

4. The SQA representative will verify that all problems are corrected (i.e., on the final test run, all test cases produce the expected result).

5. The SQA representative will verify that all problem reports have been completed and approved.

6. The SQA representative will produce the Test Completion Review Report for Frames, signifying approval that the Frame Test Phase is complete.

4.4 System Test

System testing will consist of executing an entire trajectory in a simulator. This phase of testing is roughly equivalent to hardware/software integration testing as described in RTCA/DO-178A [1]. The test cases will be designed by the three testers together. The tests will be executed on each implementation independently by the appropriate tester. The test cases will be divided equally between stress test cases and random test cases. Additional details about the System Test are contained in the *Software Verification Plan for GCS*.

4.4.1 Quality Assurance for System Test

Prior to the start of the System Test, the SQA representative is responsible for conducting a Test Readiness Review according to the following procedures.

1. The SQA representative will verify that the test cases are documented, including all inputs and expected results, and entered in DTM.

2. The SQA representative will verify that there are fifty stress test cases and at least fifty random test cases.

At the conclusion of System Testing, the SQA representative is responsible for conducting a Test Completion Review according to the following procedures:

1. The SQA representative will check DTM to ensure that the actual test results are recorded for each test case.

2. The SQA representative will verify that all changes to the test cases (including the addition of new cases) are documented in a problem report.

3. The SQA representative will verify that all discrepancies between actual and expected test results have been documented in a problem report and that the problem report number is contained in the test log.

4. The SQA representative will verify that all problems are corrected (i.e., on the final test run, all test cases produce the expected result).

5. The SQA representative will verify that all problem reports have been completed and approved.

6. The SQA representative will produce the System Test Completion Review Report, signifying approval that the System Test Phase is complete.

5 Problem Reporting and Correction

One of the cornerstones of an effective software quality program is a systematic, disciplined set of procedures for problem reporting and correction. These procedures ensure that all problems are documented, that problem status at any given time can be readily determined, and that all changes to documentation and code resulting from problem correction follow established configuration control procedures. The problem reporting and correction procedures to be used on the GCS project are outlined in this section.

5.1 Procedures

5.1.1 Specification Problems

The procedures to be followed for reporting specification problems are outlined in Programmer Instruction #2. Suspected problems are reported online via VAXNOTES[4] to the Specification and Configuration Control Manager who makes any needed modifications. In addition, specification problems discovered after the design review will be recorded on a problem report form. These modifications reside in a special CMS library which can be accessed by all programmers as well as all members of the management team (Contract Monitor, Project Leader, SQA Organization, Test Organization.)

[4]Descriptions of VAXNOTES and CMS can be found in [5].

5.1.2 Code Problems Identified during Unit Testing

When problems are discovered during the programmer's Unit Test that impact only code that is not yet entered under formal configuration control, a problem report must be filled out but SQA approval is not needed to make changes (as described in Programmer Instruction #4). Appendix B.2 of the *Software Verification Plan for GCS* contains a Problem Report Form.

5.1.3 All Other Problems

Most problem reports will be generated by a member of the test team as a result of formal verification activities, i.e., during a design or code review or during Sub-Frame, Frame, or System Test. Problem reports will also be generated by programmers when a problem is discovered in design or code that has entered formal configuration control. The following discussion applies to problems which require changes to design or code that is under formal configuration control.

The problem report is assigned a number by the test-team member. The programmer sends a request to the tester for access to specific design or code document(s). The tester is authorized to send files containing the programmer's design or code to the programmer's directory but no changes can be made to the library copy. Only the SQA representative and the configuration manager can release changeable files. Thus, if changes are anticipated, the tester must send a request to SQA or to the configuration manager who can then send a changeable copy to the programmer's directory. (No changes can be made to any document that is under configuration control without an accompanying problem report.) When this request is approved, the document will be sent to the programmer's directory. When the resulting changes have been approved by the SQA representative, this new version will enter the library. All problem reports are turned over to SQA for approval and storage.

Code changes of more than twenty lines require a formal code review which will be conducted by the same procedure as the original code reviews. See the *Software Verification Plan for GCS* for additional information.

An additional source of problem reports occurs when changes are made to test cases. The problem report must document the reasons for the change and must be approved by the SQA representative.

6 Software Configuration Management

The *GCS Configuration Management Plan* [9] outlines the procedures to be followed to control access and changes to documents. The configuration management procedures are supported by Digital Equipment Corporation's Code Management System (CMS). CMS allows one to define various libraries, each of which contains all versions of the documents within that library. Specific users can be authorized to access but not change the documents within a library while other users can be authorized to make changes as well. The design and code documents produced by each of the three programmers will reside in three separate libraries so that the programmer-tester teams can only access their own documents. Testers will be authorized to access the design and code documents of the programmer assigned to them through the *fetch* command. The *fetch* command will allow the testers to place a document in their own or in a programmer's directory but no changes to the document can be entered into the library. Change control will be achieved by authorizing only the SQA representative and the configuration manager to *reserve* documents. This command allows them to place a document in a directory and to mark the copy within the library so that no changes can be entered. The *replace* command is then used to replace the marked document with the changed file.

A total of seven different libraries have been created for the GCS project. The *GCS Configuration Management Plan* outlines the access and change authorizations for the documents within each library and contains a list of the documents to be placed within the various libraries.

7 SQA Reports and Approvals

7.1 Reports

There will be an SQA report after each major review. The signed report is sent to the Project Leader, who may disburse copies as appropriate. Some reports may also be included as part of the documentation, as indicated below. Appendix A contains copies of the review forms for those activities that do not have a full report.

The basic form of all the reports is an introduction followed by the minutes

of the review sessions and any checklists and traceability forms that are appropriate. Below is a synopsis of the context for each report and an outline of its contents. Each report documents the SQA approval for a particular stage of the implementation's development, and contains an acceptance statement signed by the SQA representative as part of the introductory comments.

The SQA documents generated for each implementation are:

Design Review Report

The Design Review Report is the formal acceptance of the design, signifying that the design stage has ended, and the coding stage begun. It is generated when all problem reports and action items generated during the design reviews and subsequent investigations have been closed. A completed copy of the report for each implementation is included in Document no. 3.

- introductory and acceptance remarks
- minutes of review sessions
- GCS Design Review Checklist for each session
- GCS Requirements Traceability Matrix

Code Review Report

This is issued when all problem reports and action items generated during code reviews and subsequent investigations have been closed, including any problem reports and action items for the design. It is the formal acceptance of the code and indicated the inception of the Unit Test stage. It has the same format as the Design Review Report, but can contain minutes from both code reviews and additional design review sessions, if so required. A completed copy of the report for each implementation is included in Document no. 6.

- introductory and acceptance remarks
- minutes of review sessions
- GCS Code Review Checklist for each session
- GCS Requirements Traceability Matrix

Test Completion Review Report for Units

This report indicates that Unit Testing has been completed. There

is no checklist for Unit Testing and no entry in the GCS Requirements Traceability Matrix. While completed GCS Unit Test Logs are archived, their size precludes attaching them to the report.

- introductory and acceptance remarks
- minutes of review sessions

Test Readiness Review Report for Sub-Frames
This report records that all test cases necessary for sub-frame testing, both White Box and Black Box,[5] have been developed. White box testing is not recorded in the GCS Requirements Traceability Matrix.

- introductory and acceptance remarks
- minutes of review sessions
- GCS Requirements Traceability Matrix

Test Completion Review Report for Sub-Frames
This report signifies that all tests have been satisfied for Sub-Frame testing, both White Box and Black Box, and have been documented. White box testing is not recorded in the GCS Requirements Traceability Matrix.

- introductory and acceptance remarks
- minutes of review sessions
- GCS Requirements Traceability Matrix

Test Readiness Review Report for Frames
This is issued when all test cases for Frame testing have been developed.

- introductory and acceptance remarks
- minutes of review sessions
- GCS Requirements Traceability Matrix

[5]See *Software Verification Plan for GCS* for descriptions of White Box and Black Box testing

Test Completion Review Report for Frames

This report is issued when SQA has determined that all procedures have been adhered to, all problem reports and action items found up through Frame testing have been closed, and all tests are satisfied.

- introductory and acceptance remarks
- minutes of review sessions
- GCS Requirements Traceability Matrix

System Test Readiness Review Report

This report is issued when the required number and type of test cases have been developed.

- introductory and acceptance remarks
- minutes of review sessions

System Test Completion Review Report

This is the final SQA report issued for an implementation and is issued when SQA has determined that all procedures have been adhered to, all problem reports and action items logged during System testing have been closed, and all test cases are satisfied.

- introductory and acceptance remarks
- minutes of review sessions

7.2 Approvals

The SQA representative will approve all project documents prior to delivery to the Contract Monitor. A list of project documents which, together, fulfill the RTCA/DO-178A guidelines is contained in the Preface to this and all documents. The signature page of each document contains a line for the SQA representative's signature.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-182045		

4. Title and Subtitle	5. Report Date
Software Quality Assurance Plan for GCS	May 1990
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Stephen E. Duncan Elizabeth K. Bailey	10. Work Unit No. 505-66-21-01

9. Performing Organization Name and Address	11. Contract or Grant No.
Research Triangle Institute P.O. Box 12194 Research Triangle Park, NC 27709-2194	NAS1-17964
	13. Type of Report and Period Covered Contractor Report
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Langley Research Center Hampton, VA 23665-5225	14. Sponsoring Agency Code

15. Supplementary Notes

Technical Monitor: George B. Finelli, Langley Research Center
Task 8 Report

16. Abstract

This document describes the Software Quality Assurance (SQA) function for the Guidance and Control Software (GCS) project which is part of a software error studies research program conducted by the Research Triangle Institute (RTI) and the NASA Langley Research Center. The SQA plan outlines all of the procedures, controls, and audits to be carried out by the SQA organization to ensure adherence to the policies, procedures, and standards for the GCS project. In addition, the SQA representative is responsible for ensuring that all problems identified throughout the development and verification processes are documented and corrected and is also responsible for reviewing all project documentation. This plan fulfills the Radio Technical Commission for Aeronautics RTCA/DO-178A guidelines requirements for document #5B.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Software Quality Assurance (SQA) Guidance and Control Software (GCS) Verification	Unclassified-Unlimited Subject Category 61

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	27	A03