National Aeronautics and
Space Administration

GRC-CONN-PLAN-0024
EFFECTIVE DATE: 02/25/2010

**Communications, Navigation, and Networking reConfigurable Testbed (CoNNeCT) Project**

National Aeronautics and Space Administration
John H. Glenn Research Center at Lewis Field, OH 44135

# CoNNeCT PROJECT

# SOFTWARE MANAGEMENT AND DEVELOPMENT PLAN

| AUTHORIZED by CM when under FORMAL Configuration Control | |
|---|---|
| **Date** | **Signature** |
| *03/16/2010* | */s/ James A. Drury* |

***Distribution:***

[ ] NASA (U.S. Gov. Only) [ ] Project Only [ ] Government and Contractors

***Availability:***

[ ] Public (No Restriction) [ ] Export Controlled [ ] Confidential/ Commercial [ ] Internal Use Only

# PREFACE

National Aeronautics and Space Administration (NASA) is developing an on-orbit, adaptable, Software Defined Radios (SDR)/Space Telecommunications Radio System (STRS)-based testbed facility to conduct a suite of experiments to advance technologies, reduce risk, and enable future mission capabilities on the International Space Station (ISS). The Communications, Navigation, and Networking reConfigurable Testbed (CoNNeCT) Project will provide NASA, industry, other Government agencies, and academic partners the opportunity to develop and field communications, navigation, and networking technologies in the laboratory and space environment based on reconfigurable, software defined radio platforms and the STRS Architecture. The CoNNeCT Payload Operations Nomenclature is "SCAN Testbed" and this nomenclature will be used in all ISS integration, safety, verification, and operations documentation. Also included are the required support efforts for Mission Integration and Operations, consisting of a ground system and the Glenn Telescience Support Center (GRC TSC). This document has been prepared in accordance with NASA Glenn's Configuration Management Procedural Requirements GLPR 8040.1 and applies to the CoNNeCT configuration management activities performed at NASA's Glenn Research Center (GRC). This document is consistent with the requirements of SSP 41170, Configuration Management Requirements, International Space Station, and Space Assurance and Requirements Guideline (SARG).

This document specifies the Software System requirements for the entire project and is the highest-level software requirements document for the project.

This document was prepared as a Software Management and Development Plan per requirement SWE-102 of NPR 7150.2, "NASA Software Engineering Requirements". It defines the CoNNeCT Project software development organization and responsibilities, the software components and their classification, WBS structure, management approach and planes, schedule and budget allocations, verification and validation approach, and acquisition strategy. The document was prepared using a combination of the Software Management Plan Template GRC-SW-TPLT-SMP, found at Software@Glenn (http://software.grc.nasa.gov), and the CoNNeCT standard document template, CoNNeCT_General Plan template (Rev3).dot found in eRoom.

# DOCUMENT HISTORY LOG

| Status (Preliminary/ Baseline/ Revision/ Canceled) | Document Revision | Effective Date | Description |
|---|---|---|---|
| Baseline | – | 02/25/2010 | Baseline prior to System CDR. |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

# SIGNATURE PAGE

***Prepared By:***

*/s/ Denise M. Varga* — *03/15/2010*

**Denise Varga** — Date
CoNNeCT Software Engineering Lead
NASA Glenn Research Center

***Concurred By:***

*/s/ Mark Luboski* — *03/15/2010*

**Mark Luboski** — Date
CoNNeCT Software Product Assurance
NASA Glenn Research Center

*/s/ Todd Tofil* — *03/15/2010*

**Todd Tofil** — Date
CoNNeCT Lead Systems Engineer
NASA Glenn Research Center

*/s/ Michael J. Barrett* — *03/16/2010*

**Michael J. Barrett** — Date
CoNNeCT Chief Engineer
NASA Glenn Research Center

***Approved By:***

*/s/ Ann P. Over* — *03/15/2010*

**Ann P. Over** — Date
CoNNeCT Project Manager
NASA Glenn Research Center

# TABLE OF CONTENTS

1.0 INTRODUCTION ..... 1
1.1 Document Overview ..... 1
1.2 Software Project Summary ..... 2
1.2.1 System Identification ..... 2
1.2.2 Purpose, Scope, and Objectives ..... 3
1.2.3 Project Deliverables ..... 3
1.2.3.1 Payload Avionics Software (PAS) ..... 3
1.2.3.2 SDR Software ..... 4
1.2.3.2.1 GD SDR Operating Environment (GD OE) Software ..... 4
1.2.3.2.2 JPL SDR Operating Environment (JPL OE) Software ..... 5
1.2.3.2.3 Harris SDR Operating Environment (HAR OE) Software ..... 5
1.2.3.3 Waveform Software ..... 5
1.2.3.3.1 TDRSS Ka-Band Waveform (HAR WF Software) ..... 6
1.2.3.3.2 TDRS S-Band Waveform (GD WF, GGT WF) Software ..... 6
1.2.3.3.3 GPS L-Band Waveform (JPL WF) Software ..... 6
1.2.3.4 Ground Software ..... 6
1.2.3.4.1 Ground Software (GSW) ..... 6
1.2.3.4.2 Test Software (TST SW) ..... 7
1.2.3.5 Follow On Experiment Software ..... 7
1.2.3.5.1 Waveform Software ..... 7
1.2.3.5.2 Experiment Software ..... 7
1.3 Classification of Systems and Subsystems ..... 8
1.3.1 Software Defined Radios Software ..... 8
1.3.2 Payload Avionics Flight Software ..... 8
1.3.3 Ground Software ..... 9
1.4 Project Compliance to NPR 7150.2 ..... 9
2.0 REFERENCE DOCUMENTS ..... 10
2.1 Reference Documents ..... 10
2.2 Order of Precedence for Documents ..... 11
3.0 PROJECT ORGANIZATION ..... 12
3.1 Organizational Structure ..... 12
3.2 Roles and Responsibilities ..... 13
3.2.1 CoNNeCT Software Engineering Lead ..... 13
3.2.2 CoNNeCT Payload Avionics Lead Software Engineer ..... 13
3.2.3 CoNNeCT Payload Avionics Software Engineers ..... 14
3.2.4 CoNNeCT Lead Ground Software Engineer ..... 14
3.2.5 Software Defined Radios Operating Environments and Bootcode ..... 15
3.2.5.1 Harris SDR Software Developers ..... 15
3.2.5.2 JPL SDR Software Developers ..... 15
3.2.5.3 GD SDR Software Developers ..... 15
3.2.6 CoNNeCT SDR Waveforms ..... 15
3.2.6.1 GRC/GSFC Waveform Developers ..... 15
3.2.6.2 JPL Waveform Developers ..... 16
3.2.6.3 Harris Waveform Developers ..... 16
3.2.6.4 GD Waveform Developers ..... 16
4.0 MANAGERIAL PROCESS PLANS ..... 17

4.1 Project Start-Up ..... 17
4.1.1 Estimation ..... 17
4.1.1.1 Payload Avionics Flight Software Estimation ..... 17
4.1.1.2 Ground Software Estimation ..... 18
4.1.1.3 Test Software Estimation ..... 18
4.1.1.4 SDR and Waveform Software Estimation ..... 18
4.1.1.5 Software Estimation Summary ..... 18
4.1.2 Staffing ..... 19
4.1.3 Resource Acquisition ..... 19
4.1.3.1 Development Systems and Workstations ..... 19
4.1.3.1.1 Desktop Computers ..... 19
4.1.3.1.2 Development Systems ..... 20
4.1.3.2 Software Development Tools ..... 20
4.1.3.2.1 VxWorks and Associated Tools ..... 20
4.1.3.2.2 Enterprise Architect ..... 20
4.1.3.2.3 Visual Studio .NET ..... 20
4.1.3.2.4 Subversion ..... 20
4.1.3.2.5 JIRA ..... 21
4.1.3.2.6 Fisheye, Crucible ..... 21
4.1.3.3 Facilities ..... 21
4.1.4 Personnel Training ..... 21
4.2 Project Planning ..... 22
4.2.1 Work Breakdown Structure (WBS) ..... 22
4.2.2 Schedule Allocation ..... 22
4.2.3 Resource Allocation ..... 23
4.2.4 Budget Allocation ..... 23
4.3 Management Controls ..... 24
4.3.1 Requirements Control ..... 24
4.3.2 Schedule Control ..... 25
4.3.2.1 GRC Flight Software Schedule Control ..... 25
4.3.3 Budget Control ..... 25
4.3.4 Reporting ..... 25
4.3.5 Metrics ..... 26
4.4 Risk Management ..... 26
5.0 TECHNICAL PROCESS PLANS ..... 27
5.1 Life Cycle Model ..... 27
5.2 Methods, Tools, and Techniques ..... 28
5.2.1 Object Oriented Analysis and Design ..... 28
5.2.2 Coding Standards ..... 28
5.2.3 Waveform FPGA Development Tools ..... 29
5.2.3.1 Model Prototyping ..... 29
5.2.3.2 HDL Synthesis ..... 29
5.2.3.3 HDL Simulation ..... 29
5.2.3.4 Place and Route ..... 29
5.2.3.5 Debugging ..... 29
5.2.4 Simulation Tools ..... 29
5.2.4.1 Matlab/Simulink with Real-Time Workshop Embedded Coder ..... 29
5.2.5 Software Development Facilities ..... 30

5.2.6 Local Area Networks ..... 30
5.2.7 Office Space ..... 30
5.2.7.1 Support Service Contractor Developers ..... 30
5.3 Software Acceptance ..... 30
5.4 Software Maintenance Plan ..... 30
6.0 SUPPORTING PROCESS PLANS ..... 31
6.1 Configuration Management Plan ..... 31
6.2 Verification and Validation ..... 31
6.2.1 Hardware Interface and Subsystem Testing ..... 31
6.2.2 Final Software Verification and Validation ..... 33
6.2.3 Software Development Systems and Ground Integration Unit (GUI) Verification & Validation (V&V) ..... 33
6.2.4 Responsibility for Approval of Software Verification & Validation ..... 34
6.3 Documentation ..... 35
6.3.1 Documentation Tree ..... 35
6.3.2 List of Documentation ..... 37
6.4 Quality Assurance Plan ..... 39
6.5 Reviews and Audits ..... 39
6.5.1 Management Progress Reviews ..... 39
6.5.2 Developer Peer Reviews and Inspections ..... 39
6.5.3 Quality Assurance Audits ..... 39
6.6 Problem Resolution ..... 40
6.7 Subcontractor Management Plans ..... 40
APPENDIX A ACRONYMS AND ABBREVIATIONS ..... 41
A.1 Scope ..... 41
A.2 List of Acronyms and Abbreviations ..... 41
APPENDIX B DEFINITIONS ..... 43
B.1 Scope ..... 43
B.2 List of Definitions ..... 43
APPENDIX C NPR 7150.2 COMPLIANCE MATRIX FOR SWR-102 ..... 44
C.1 Master Monthly Metrics Spreadsheet – Example ..... 46

# TABLE OF FIGURES

Figure 3.1-1—CoNNeCT Project Organizational Structure ..... 12
Figure 4.2.2-1—Schedule Allocation for GRC CoNNeCT Software Development as of November 2009 ..... 22
Figure 5.1-1—Incremental Waterfall Life Cycle for CoNNeCT flight software development ... 28
Figure 6.2.1-1—Use of Software Development Systems for Flight Software Development as of November 2009 ..... 32
Figure 6.3.1-1—CoNNeCT Software Documentation Tree, Management Branch ..... 35
Figure 6.3.1-2—CoNNeCT Software Documentation Tree, Technical Branch ..... 36

# TABLE OF TABLES

Table 1.2.1-1—CoNNeCT CSCI List ..... 2
Table 3.1-1—CoNNeCT Organizations Developing Software ..... 13
Table 4.1.1.1-1—CoNNeCT Comparison to Fluid Combustion Facility ..... 17
Table 4.1.1.5-1—Summary of CoNNeCT Software Estimation ..... 18
Table 4.1.2-1—Staffing for GRC CoNNeCT Software Development, December 2009 ..... 19
Table 4.2.3-1—Resource Allocation for GRC CoNNeCT Software Development ..... 23
Table 4.2.3-1—Budget Allocation for GRC CoNNeCT Software Development as of November 2009 ..... 23
Table A-1—Acronyms ..... 41
Table B-1—Definitions ..... 43
Table C-1—Traceability from the SMP Template to SWE-102 of NPR 7150.2 ..... 44

# 1.0 INTRODUCTION

## 1.1 Document Overview

This document is the Software Management and Development Plan (SMDP) for management and development of the payload avionics and ground software for the Communications, Navigation, and Networking re-Configurable Testbed (CoNNeCT) project. The development of software for the Software Defined Radios (SDRs) and their Waveforms are covered under other plans and are managed by partner organizations. The S-band Waveform for the JPL radio is being developed at GRC under a different work breakdown structure (WBS) and has its own development plan. For the Software Defined Radios and their Waveforms, this also includes firmware. Firmware is defined as *"The combination of a hardware device and computer instructions and data that reside as read-only software on that device."[IEEE]* There is no firmware being developed for payload avionics or ground software, so this plan does not address firmware development. This plan does cover integration and testing with SDR software and firmware to achieve the complete CoNNeCT software system.

The major goals for this document are:

1. To describe **what** products will be delivered as the CoNNeCT project's software system.
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) to produce the software, as a function of time.
5. To describe **how** and **where** the work will be carried out.
6. To reach a mutual understanding and agreement with our customer and other stakeholders on items (1) through (5).

## 1.2 Software Project Summary

### 1.2.1 System Identification

CoNNeCT will provide an on-orbit Software Defined Radio (SDR) and Space Telecommunications Radio System (STRS)-based testbed facility on the International Space Station (ISS). It will be used to conduct a suite of SDR experiments in the environments of space to reduce risk and enable future missions.

The CoNNeCT software articles will be comprised of:

1. The CoNNeCT Payload Avionics flight software
2. The GD Radio Operating Environment, TDRSS-compliant/STRS-compliant Waveform, and ROM codes
3. The Harris Radio Operating Environment, TDRSS-compliant/STRS-compliant Waveform, and ROM codes
4. The JPL Radio Operating Environment, L-band Waveform, and ROM codes
5. The GRC/GFSC TDRSS-compliant/STRS-compliant Waveform for the JPL SDR
6. The CoNNeCT Control Center ground software for mission operations, the PI Center operations, and White Sands (if necessary).
7. Miscellaneous test and support software

The following table lists the CoNNeCT Computer Software Configuration Items (CSCIs). The shaded rows represent CSCIs that are not covered by this plan.

**Table 1.2.1-1—CoNNeCT CSCI List**

| ID | CSCI Name | Developer | WBS |
|---|---|---|---|
| PAS | Payload Avionics Software | GRC | 6.2 |
| GD OE | GD Operating Environment | GD | 5.4 |
| GD WF | GD Waveform | GD | 5.4 |
| JPL OE | JPL Operating Environment | JPL | 5.4 |
| GGT WF | GRC GSFC TDRSS Waveform | GRC, GFSC | 4.3 |
| GGT OE/WF | GGT Waveform compiled with the JPL Operating Environment | GRC | 4.3 |
| JPL WF | JPL Waveform | JPL | PHASE II |
| HAR OE | HAR Operating Environment | Harris | 5.4 |
| HAR WF | HAR Waveform | Harris | 5.4 |
| GSW | CoNNeCT Ground Software | GRC | 6.3 |
| TST SW | Test Software | GRC | 6.2 |
| PIC Ground SW/White Sands SW | CoNNeCT Ground Software for PIC/White Sands | GRC | 4.0 |

### 1.2.2 Purpose, Scope, and Objectives

This Software Management and Development Plan (SMDP) applies to CoNNeCT payload avionics software, ground software, and the developing organizations at GRC during all phases of the software project life cycle during Phase I of the project. Phase I activities include developing, integrating, testing, and delivering the system up to launch, including on-orbit checkout and mission operations planning. Phase II activities include experiment development and mission operations. Phase II activities will be described in a separate plan. For completeness, the following sections describe ALL software and firmware deliverables for the project, even those being developed by partner organizations and thus not covered by this plan.

### 1.2.3 Project Deliverables

#### 1.2.3.1 Payload Avionics Software (PAS)

The PAS is a flight software Computer Software Configuration Item (CSCI) that will reside and operate in the Payload Avionics of the CoNNeCT payload. The primary functions of the PAS are:

- Receives and parses incoming commands. These commands may be from a number of sources including:
    - From the CoNNeCT Control Center (CCC) via ISS and ExPRESS Logistics Carrier (ELC) arriving at the Payload Avionics on the 1553 bus
    - From the CCC via ground stations as part of the Near-Earth Network, CoNNeCT RF subsystem and antenna, and one of the Software Defined Radios (SDR)
    - From the CCC via TDRSS as part of the Space Network, CoNNeCT RF subsystem and antenna, and one of the Software Defined Radios (SDR)
- Collects all CoNNeCT subsystem telemetry and report back to the CCC via the ISS ELC
- Processes and routes experimental data between the SDRs and the ground or ISS
- Monitors the health of the CoNNeCT payload
- Interfaces with the communication busses including: MIL-STD-1553, Ethernet, and SpaceWire
- Performs command and control of the SDR based on commands from the CCC
- Provides storage and uploading of SDR waveforms and data files
- Provides for uploading and replacing any onboard software
- Provides an operational platform for future networking and communications experiments
- Performs time synchronization among all CoNNeCT flight and ground elements
- Perform evaluation of the quality of communications links

#### 1.2.3.2 SDR Software

The SDR platforms on CoNNeCT will each be compliant with the STRS Architecture Standard. The STRS Standard is an open, published architecture framework to abstract application and waveform software from the SDR hardware platform. All CoNNeCT SDRs will be delivered with an STRS Operating Environment (OE) and the accompanying Hardware Abstraction Layer (HAL). This middleware enables waveform software reuse and portability. All software and firmware will be delivered to GRC electronically, either on compact disk or remote transfer from partner site, and stored in the GRC software repository under configuration management. During the integration and testing phase, the partner organizations must provide support to perform any updates to software and firmware required to fix bugs found during testing. Reprogramming firmware in flight is part of the normal operational scenarios and will be verified on the ground before uploading new code to the payload.

The three SDR developers are General Dynamics (GD), Jet Propulsion Laboratory (JPL), and Harris Corporation. The primary functions of the OEs are:

- Communicate with the PAS over MIL-STD-1553 and/or SpaceWire for experiment data, command, and telemetry exchange
- Provide an STRS-compliant application interface for radio operation
- Provide SDR health and status data to the telemetry stream
- Provide a hardware abstraction layer for the STRS interface
- Perform evaluation of the quality of communications links

##### 1.2.3.2.1 GD SDR Operating Environment (GD OE) Software

The SDR will have the STRS operating environment (OE) software installed upon delivery. The OE provides a POSIX layer as required by the STRS architecture. The STRS application programmer interfaces (APIs) will be implemented as specified in the STRS architecture 1.02. The middleware will be developed using native services of the VxWorks real time operating system (RTOS). The hardware abstraction layer (HAL) implementation will provide drivers to support bus communication and interfaces to the complex electronics and other hardware modules. The OE drivers that communicate to the FPGA Framework Source Code are provided, and their APIs are part of the GD Platform STRS OE.

#### 1.2.3.2.2 JPL SDR Operating Environment (JPL OE) Software

The SDR will be delivered from the vendor (Cincinnati Electronics) without software installed. The SDR hardware is shipped to JPL and JPL personnel install/test/update the software prior to delivery to GRC. The STRS operating environment (OE) will need to be recompiled with the GRC GSFC TDRSS (GGT) Waveform before loading. The OE STRS application programmer interfaces (APIs) will be implemented as specified in the STRS architecture 1.02. The hardware abstraction layer (HAL) implementation will provide drivers to support bus communication and interfaces to the complex electronics and other hardware modules. The Baseband Processing Module Boot Code and STRS Operating Environment will have as a minimum the following functionality:

- Resident PROM code boots into OE
- Software and firmware uploads via boot code or OE
- Turns S- Band transponder power amplifier on/off
- Turns L- Band transponder power amplifier on/off

#### 1.2.3.2.3 Harris SDR Operating Environment (HAR OE) Software

The Harris SDR will have the STRS operating environment (OE) software installed upon delivery. The STRS application programmer interfaces (APIs) will be implemented as specified in the STRS architecture 1.02. The hardware abstraction layer (HAL) implementation will provide drivers to support bus communication and interfaces to the complex electronics and other hardware modules.

#### 1.2.3.3 Waveform Software

Waveform software and firmware defines functionality for most of the SDR. This code represents a significant effort in terms of development cost and schedule. STRS defines standards for SDRs in order to maximize reuse and reduce porting effort between various radios. All three CoNNeCT radios will be launched with baseline STRS-compliant waveforms. These waveforms will provide functionality in the three CoNNeCT frequency bands. GD is delivering a single S-band TDRSS Waveform for their SDR at launch. A joint GRC/GSFC team is developing an S-band Waveform for the JPL radio for launch. JPL is producing Global Positioning System (GPS) L-band Waveforms which may be uploaded after launch. Harris Corporation is producing two Waveforms: one S-band Waveform, and one Ka-band Waveform for launch. The primary functions of the Waveforms are:

- Implement radio transmissions and receptions in the designated communication band at the designated data rates in compliance with TDRSS and the STRS standard
- Achieve STRS-compliant application interface for radio operation
- Demonstrate the capabilities of the SDRs
- Demonstrate portability of Waveforms onto different hardware platforms
- Demonstrate software reconfigurability of the SDRs

#### 1.2.3.3.1 TDRSS Ka-Band Waveform (HAR WF Software)

The Harris radio Ka-band waveform will have TDRSS Data Group 2, non-coherent mode capability. A data rate of up to 100 Mbps will be possible. There will be no despreading in the receiver, but the transmitter will have SQPSK capability. Post-launch, other Ka-band waveforms may be developed or ported to the Harris radio.

#### 1.2.3.3.2 TDRS S-Band Waveform (GD WF, GGT WF) Software

The GD and JPL radios will have S-band capability, and will launch with an S-band TDRSS waveform set, transmit and receive functionality. Each partner is bringing a different waveform capability to CoNNeCT, exercising different aspects of the TDRSS. GD will provide the SQPN modulation at a relatively low data rate. A variable data rate BPSK spread waveform capability will be provided by the baseline waveform on the JPL SDR. As part of the Experiment Program, other waveforms with different capabilities or features will be uploaded when available after launch.

#### 1.2.3.3.3 GPS L-Band Waveform (JPL WF) Software

Due to project cost and schedule constraints the L-band waveforms for the JPL radio will not be available until after launch. The L1 GPS waveform software/firmware will be developed and uploaded when available post-launch. The waveform will have standard GPS capabilities, with raw and PVT (position, velocity, time) potential outputs. As part of the Experiment Program, other waveforms utilizing the L2 and L5 GPS services will be developed and uploaded during the on-orbit operations.

### 1.2.3.4 Ground Software

#### 1.2.3.4.1 Ground Software (GSW)

The ground software will be written to support the ground operations, commanding and telemetry monitoring of the CoNNeCT payload. The software will operate at the GRC CoNNeCT Control Center (CCC) on project-provided hardware platforms. The GSW will serve as the primary control and operations software for the CoNNeCT Payload, and will also operate the Ground Interface Unit (GIU), the flight-like ground testbed.

The Ground Software is composed of the following elements:

1. CTADS – CoNNeCT Telemetry Acquisition and Display System
    a. Displays system health and status and telemetry received from the payload
2. TReK – Telescience Resource Kit
    a. Builds command strings from the database and sends to the MSFC POCC
3. Command and Telemetry Database
    a. Contains the information needed to build commands and display telemetry

4. Data Storage and Distribution System

    a. Stores data on a local server and pushes data out to local and remote users (PIs)

5. PI Center specific software and potential White Sands software to be defined.

#### 1.2.3.4.2 Test Software (TST SW)

Integration and test of the payload will occur during the months of January through September 2010. Environmental testing is scheduled during the last four months prior to shipment, nominally October 2010-January 2011. During the hardware integration and later environmental tests, it will be necessary to deliver specialized software to support hardware and software checkout. This software will be developed using flight components and thoroughly tested with breadboard and engineering model hardware prior to use on the flight system. A Test Software Request form will be used to collect requirements and the software builds will be maintained in the software configuration management system so that changes and versions can be tracked.

### 1.2.3.5 Follow On Experiment Software

Note that there is no funding in Phase I of CoNNeCT to develop this software. Funds will need to be allocated in Phase II for this activity, but the capability must be developed in Phase I to support this follow-on work.

#### 1.2.3.5.1 Waveform Software

The reconfigurable capability of the CoNNeCT on-orbit system is based on the reconfigurable SDRs and portion of the avionics made available to experimenters. Principal Investigators (PIs) will develop application or waveform software and firmware to conduct their experiments. The SDR platforms on CoNNeCT will each be compliant with the STRS Standard. The STRS Standard is an open, published architecture framework to abstract application and waveform software from the SDR hardware platform. The STRS Architecture is intended to facilitate the development and porting of firmware and software developed for the SDRs.

CoNNeCT will have breadboards of each SDR and avionics software development system available for use to support experiment software and firmware development. Experimenters will work with developers to port their code to the breadboards to begin verifying functionality and compliance to the STRS Standard.

Once development work on the breadboards is complete and STRS compliance assured, the software and firmware will be moved to the CoNNeCT Ground Integration Unit (GIU) for TDRSS or Near-Earth Network compatibility testing and final verification before uploading the STRS compliant waveforms to the respective SDR(s).

#### 1.2.3.5.2 Experiment Software

The process for developing experimental software for the Payload Avionics is similar to the development of software for the waveforms. The software applications developed on the ground will be developed and tested using the CoNNeCT development system available to the experimenters. Once development work is complete, the software will be moved to the CoNNeCT Ground Integration Unit for compatibility testing and final verification before uploading the application to the payload.

## 1.3 Classification of Systems and Subsystems

### 1.3.1 Software Defined Radios Software

The Software Defined Radios are part of the payload and the item of interest to the CoNNeCT experiments. One of the main objectives of CoNNeCT is to bring the Software Defined Radios to TRL-7. Therefore their software, both the Operating Environment and any Waveforms that run on the Software Defined Radios are classified as **Class C**.

Per GLPR 7150.1, this software is treated as acquired software, since it is being delivered as part of an acquired software defined radio. GLPR 7150.2 states that all organizations developing Class A or B software must meet CMMI CL2 (Capability Maturity Model Integrated, Capability Level 2) or higher. The GD software development organization has achieved a CMMI Level 5 rating. The Harris software development organization had achieved a CMMI Level 3 rating. JPL software development has achieved a CMMI Level 3 rating. The GGT Waveform software being developed at GRC is being developed as Class C software following the GRC processes. Therefore, the SDR software meets the requirements of GLPR 7150.2 Class C software.

### 1.3.2 Payload Avionics Flight Software

The Payload Avionics is part of the payload. It will be responsible for command handling, controlling and monitoring the state of the Software Defined Radios, and controlling the antennas. As such, it is flight software that is necessary for the operation of the experiment. Therefore, the Payload Avionics software is classified as **Class C**.

Some of the software will be experimental in nature and some will provide some basic functionality to the infrastructure. It will perform many of the networking experiments planned for CoNNeCT. CoNNeCT is developed as a platform so that future experimenters can develop new experimental software to be uploaded as new experiments are developed. Since it is experimental in nature but still flight software and part of the payload infrastructure it will be classified as **Class C**. Thus all Payload Avionics software is classified as **Class C.**

In addition to the NPR 7150.2 classification, the Payload Avionics Flight Software has also been classified as Safety Critical per NASA-STD-8719.13, Software Safety Standard. The software "Resides in a safety-critical system (as determined by a hazard analysis) AND provides control or mitigation for hazards." CoNNeCT has a potential hazard in radiating RF (radio frequency) energy at ISS hardware, vehicles, or crew members. Software will control the hardware that allows radiation from happening, and thus will be used as a control to avoid inadvertent activation of RF radiation during extravehicular activity, vehicle docking or undocking operations, or extravehicular robotics. Since this hazard is classified as catastrophic, the system must be two-fault tolerant to the hazard, and thus have three controls. Payload Avionics software will implement two of these controls. This requires Control Path Separation per SSP 50038 Rev. B, Computer-Based Control System Safety Requirements.

### 1.3.3 Ground Software

The ground software will operate from the CoNNeCT Control Center (CCC) and will command, control and display telemetry from the payload and experiment. Since it provides control and monitoring of the payload it is classified as **Class C**.

## 1.4 Project Compliance to NPR 7150.2

Projects must be compliant with the NPR 7150.2 Software Engineering Requirements that are applicable to the project's classification noted in Section 1.3. If the project will not meet any of the requirements, then variant, waiver, or exception to NPR 7150.2 Software Engineering Requirements must be requested from the designated ITA Warrant Authority for Flight Software, which is the Branch Chief of the Flight Software Engineering Branch. If any variants, waivers, or exceptions are requested, then those shall be documented in this section in the form of a compliance matrix.

The CoNNeCT project will maintain a compliance matrix against Class C requirements in NPR 7150.2, per requirement SWE-125. The matrix will show compliance and traceability for a Class C software development project. The compliance matrix is kept in a separate document on eRoom, CoNNeCT NPR 7150.2 Compliance Matrix.xls.

# 2.0 REFERENCE DOCUMENTS

This section lists the NASA/Government and non-NASA/Government specifications, standards, guidelines, handbooks, or other special publications applicable to the application of this document.

## 2.1 Reference Documents

Reference documents are those documents that, though not a part of this document, serve to clarify the intent and contents of this document.

| Document Number | Title |
|---|---|
| 452-NENUG-GN, February 2001 | Near-Earth Network User's Guide |
| GLPR 7150.1 | GRC Software Engineering Requirements |
| GRC-CONN-PLAN-0001 | CoNNeCT Software Configuration Management Plan |
| GRC-CONN-PLAN-0004 | CoNNeCT Project Plan |
| GRC-CONN-PLAN-0006 | CoNNeCT Product Assurance Plan |
| GRC-CONN-PLAN-0007 | CoNNeCT Risk Management Plan |
| GRC-CONN-PLAN-0012 | CoNNeCT Verification and Validation Plan |
| GRC-CONN-PLAN-0076 | GGT Waveform Development Plan |
| GRC-CONN-PLAN-0085 | CoNNeCT Software Product Assurance Plan |
| GRC-CONN-PLAN-0118 | CoNNeCT Software Verification and Validation Plan |
| GRC-CONN-REQ-0019 | CoNNeCT Level 1 Requirements |
| GRC-CONN-REQ-0031 Rev. 2 (5/21/08) | CoNNeCT Science and Technology Requirements Document |
| GRC-CONN-SRD-0013 | CoNNeCT Systems Requirements Document |
| GRC-CONN-SRD-0035 | CONNECT Flight System Requirements Document |
| GRC-CONN-SRD-0036 | CoNNeCT Ground System Requirements Document |
| GRC-SW-TPLT-SMP | Software Management Plan Template |
| MIL-STD-1553B | Department of Defense Interface Standard for Digital Time Division Command/Response Multiplex Data Bus |
| NASA-STD-8719.13 | Software Safety Standard |
| NPR 7150.2 | NASA Software Engineering Requirements |
| SNUG-450, Rev 9 | Space Network User's Guide |
| SSP 41175-02 Rev H | Software Interface Control Document, Station Management and Control to International Space Station, General Software Interface Requirements |
| SSP 50038 Rev. B | Computer-Based Control System Safety Requirements |
| SSP 53112 (draft) | CoNNeCT Payload Interface Control Document (ICD) |
| SSP 57002 Rev D | Payload Software Interface Control Document |
| SSP 57003-ELC | Attached Payload Interface Requirements Document - EXpedite the PRocessing of Experiments to Space Station (EXPRESS) Logistics Carrier (ELC) Cargo Interface Requirements |
| STRS-AR-0002 Revision 1.02 | Space Telecommunications Radio Architecture Standard, September 2008 |

## 2.2 Order of Precedence for Documents

In the event of a conflict between this document and other documents specified herein, the requirements of this document shall apply. In the event of a conflict between this document and higher level documents, the higher level documents shall take precedence over this document.

All documents used, applicable or reference, are to be the approved versions released as of the contract start date. All document changes issued after contract baseline establishment shall be reviewed for impact on scope. Nothing in this document supersedes applicable laws and regulations unless a specific exemption has been obtained.

# 3.0 PROJECT ORGANIZATION

## 3.1 Organizational Structure

Figure 3.1-1 shows the CoNNeCT Project organizational structure. The Software Engineering Lead (SWEL) manages WBS 6.0 is responsible for development, integration, testing, verification and validation, and delivery of the payload avionics software and the ground software to the flight hardware. The Experiment Communication System organization (WBS 4.0) is responsible for developing the S-band waveform for the JPL SDR (GGT WF) and experiments for CoNNeCT. The radio software, except the waveform for the JPL radio, will be developed under Flight Systems (WBS 5.0) at the partner sites. The partner organizations are General Dynamics (Phoenix, Arizona), Jet Propulsion Laboratory (Pasadena, California), Harris Corporation (Melbourne, Florida), and Goddard Space Flight Center (Greenbelt, Maryland). Safety and Mission Assurance (WBS 3.0) will provide the software assurance role. Organizations that are developing software are summarized in Table 3.1-1.

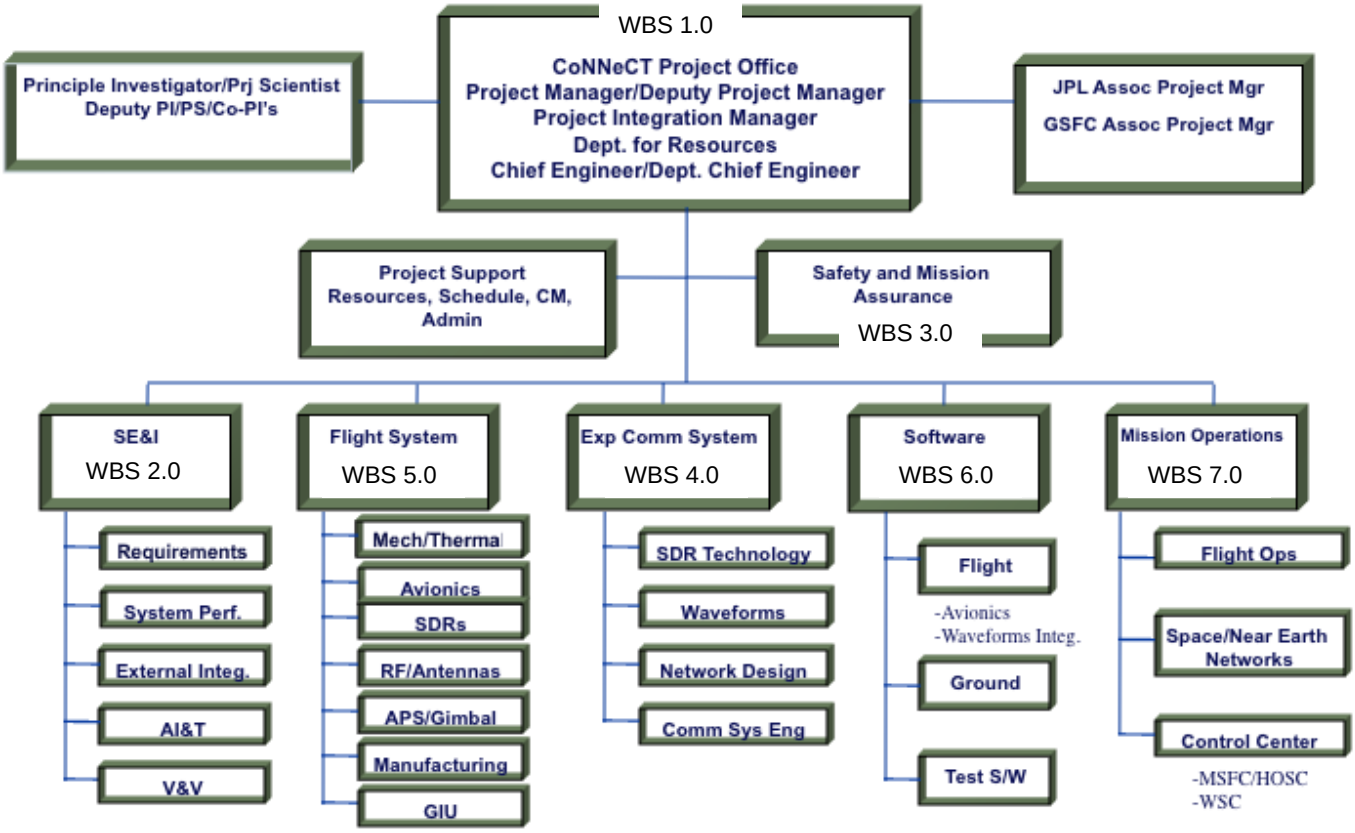



**Figure 3.1-1—CoNNeCT Project Organizational Structure**

**Table 3.1-1—CoNNeCT Organizations Developing Software**

| WBS Element | Software Deliverables | Development Location |
|---|---|---|
| WBS 4.3 Waveforms | Waveforms | GRC: Cleveland, OH<br>GSFC: Greenbelt, MD |
| WBS 5.4 Software Defined Radios | OEs, Waveforms | GD: Phoenix, AZ<br>JPL: Pasadena, CA<br>Harris Corp: Melbourne, FL |
| WBS 6.0 Software | PAS, GSW, TST SW | GRC: Cleveland, OH |

## 3.2 Roles and Responsibilities

### 3.2.1 CoNNeCT Software Engineering Lead

The CoNNeCT Software Engineering Lead (SWEL) will oversee the development and integration of the software components of the CoNNeCT system and ensure that all software articles developed for CoNNeCT are developed to the correct process requirements and standards. She will also ensure that all interfaces and software integration issues are resolved early in the development process to ensure a smooth and seamless integration. The SWEL manages WBS 6.0 and authors and maintains this CoNNeCT Software Management and Development Plan (SMDP).

### 3.2.2 CoNNeCT Payload Avionics Lead Software Engineer

The CoNNeCT Payload Avionics Lead Software Engineer oversees the development of all the payload avionics flight software developed internally at the Glenn Research Center.

His duties include but are not limited to:

- Estimating, tracking and modifying budget and resources for the flight software development
- Developing, tracking and modifying the flight software development schedule for the project
- Overseeing the flight software requirements development
- Overseeing the flight software design development
- Gathering the flight software metrics for the project and reporting them to the Software Engineering Lead
- Overseeing the development of the software code developed at GRC

### 3.2.3 CoNNeCT Payload Avionics Software Engineers

The CoNNeCT Payload Avionics Software Engineers will be responsible, under the guidance of the CoNNeCT Payload Avionics Lead Software Engineer, for the development of all the payload avionics flight software.

The development of flight software includes:

- Gathering and documenting the requirements for the flight software systems
- Designing the software architecture for the flight software systems using the UML and the Unified Process
- Coding of the flight software systems
- Developing test plans and unit testing of the flight software systems
- Assisting with the verification and validation of the flight software systems
- Maintenance both before and after launch of the flight software systems

### 3.2.4 CoNNeCT Lead Ground Software Engineer

The CoNNeCT Lead Ground Software Engineer will support a variety of tasks related to the ground support equipment, hardware testing, and software testing.

This person will be responsible for:

- Purchasing the equipment for the CCC, including computers, data storage, supplies, software, and development tools
- Interfacing with personnel from the Enhanced Huntsville Operation Support Center (EHOSC), Near-Earth Network, and Space Network to establish necessary services
- Managing the development of displays for data and telemetry
- Managing the development of the software to generate and send commands to the payload
- Managing the development of the data management services, including collection, storage, and distribution of project data for use in Phase II
- Managing the development of the interface with the Ground Integration Unit
- Supporting software verification and validation
- Support project reviews and software reviews

Some of the skills that might be necessary for this developer are:

- Software scripting languages
- Software GUI development languages for Windows (such as .NET) or for Linux
- Data management and database programming
- Some understanding of embedded system development

### 3.2.5 Software Defined Radios Operating Environments and Bootcode

#### 3.2.5.1 Harris SDR Software Developers

The Harris Corporation is developing one of the Software Defined Radios. There will be software development groups at Harris who are responsible for developing all the low level bootcode and FPGA code for the Harris radio as well as a STRS compliant Operating Environment for the SDR.

#### 3.2.5.2 JPL SDR Software Developers

The Jet Propulsion Laboratory is developing one of the Software Defined Radios. There will be software development groups at JPL who are responsible for developing all the low level bootcode and firmware for the JPL radio as well as a STRS compliant Operating Environment for the SDR.

#### 3.2.5.3 GD SDR Software Developers

The General Dynamics Corporation is developing one of the Software Defined Radios. There will be software development groups at GD who are responsible for developing all the low level bootcode and firmware for the GD radio as well as a STRS compliant Operating Environment for the SDR.

### 3.2.6 CoNNeCT SDR Waveforms

#### 3.2.6.1 GRC/GSFC Waveform Developers

Collaboration between developers at GSFC and GRC will produce a STRS-compliant S-band TDRSS waveform, whose first target will be the JPL SDR. GSFC already has TDRSS firmware that will be reused as much as possible to create a portable STRS-compliant waveform for CoNNeCT. Initially, commercial SDR develop platforms available at both GSFC and GRC will be utilized for development. Then the waveform will be ported to the JPL SDR when it becomes available. The JPL SDR will have a STRS Operating Environment, providing the first test of waveform portability. The waveform development will consider the JPL SDR specifics and interfaces from the start. Porting this same waveform to the Harris and Starlight platforms are planned post-launch experiments.

#### 3.2.6.2 JPL Waveform Developers

JPL will utilize the S-band waveform developed by GSFC and GRC as described above. L-band GPS baseline waveform software/firmware will be developed by JPL for post-launch upload during the experiment period. This activity must be funded in Phase II, since there is no budget in Phase I for this development.

#### 3.2.6.3 Harris Waveform Developers

Harris waveforms will center on several firmware based co-processors resident in the FPGAs of the SDR. There is currently one Ka-band waveform planned for the Harris SDR.

#### 3.2.6.4 GD Waveform Developers

The GD CoNNeCT Starlight will be delivered with a DG1, Mode 1 TDRS compatible waveform with heritage from previous transponders. The waveform software will reside in both the reconfigurable FPGA (firmware) and the Coldfire processor (general purpose processor code). The original portion of the GD TDRSS waveform that was in an ASIC for previous TDRSS transponders was developed using an FPGA. These FPGA signal processing functions (e.g., Carrier Acquisition and Tracking, Transmitter Carrier Tracking and Modulation, and Command Demodulation) will be located in the Xilinx FPGA in the digital signal processing module (SPM). The Coldfire will host the logic for software configuration and control for reconfigurable elements (e.g., inputs to tracking loops, bandwidth selection) that was originally in a processor.

# 4.0 MANAGERIAL PROCESS PLANS

## 4.1 Project Start-Up

### 4.1.1 Estimation

Below is a summary of the software cost estimation activities for CoNNeCT. For more detail, see the CoNNeCT Software Estimation report in eRoom in the following folder:

https://collaboration.grc.nasa.gov/eRoom/NASAm1f1/CoNNeCT/0_9e1c7.

#### 4.1.1.1 Payload Avionics Flight Software Estimation

The estimation for the software task size / budget for the CoNNeCT payload avionics flight software was performed using the "Ask Pete" software cost estimation tool. "Ask Pete" was developed by NASA to aide in software project planning and estimation based on the COCOMO II Software Cost Estimation Model. The "Ask Pete" estimation for the CoNNeCT Payload Avionics Software was 98.36 person months as calculated in early 2008.

One key input to the "Ask Pete" tool is the Source Lines of Code (SLOC) estimate. The estimate of 30,000 SLOC that was used as an input was developed by comparing this system to known developed systems of comparable functionality. The Fluid and Combustion Facility (FCF) Input/Output Processor (IOP) was chosen as the comparable system. The FCF IOP was 32,000 SLOC. It is expected that CoNNeCT will not have the complexity or number of commands as FCF. A comparison of the functionality of the CoNNeCT Payload Avionics Software versus the FCF IOP is shown in Table 4.1.1.1-1.

**Table 4.1.1.1-1—CoNNeCT Comparison to Fluid Combustion Facility**

| CoNNeCT Payload Avionics Software | FCF IOP |
|---|---|
| 1553 Interface to ELC | 1553 Interface to ISS |
| 1553 Interface to SDR | 1553 Interface to EPCU |
| Ethernet Interface to ELC | Ethernet Interface to all sub-packages (other computers in the rack) |
| Spacewire Interface | |
| Digital I/O (payload control / monitor) | Digital I/O (payload control / monitor) |
| Analog to Digital (monitoring) | Analog to Digital (monitoring) |
| | Digital to Analog (control) |
| Command Parsing from ELC | Command Parsing from ISS |
| Health & Status / Telemetry Monitoring | Health & Status / Telemetry Monitoring |
| Telemetry Reporting | Telemetry Reporting |
| Frame/Deframe CCSDS packets | Frame/Deframe CCSDS packets |
| Frame/Deframe other packet protocols? | |
| Solid state mass storage file system | Hard drive memory storage file system |
| Command Control to SDRs | Command Control to sub-packages (other computers in the rack) |
| Collect status and telemetry from the SDRs and report it back through ELC and ISS to the ground control center | Collects telemetry from the sub-packages |
| Provide platform and API for future experiments to run | Experiment Scripting Language |
| Upload new software updates | Upload new software updates |
| Upload waveforms | |
| Store waveforms and configuration files | Store software and configuration files for other packages |

#### 4.1.1.2 Ground Software Estimation

Ground software was also estimated based on the FCF project due to the similarity in functionality from a ground (commanding and telemetry) perspective. The TReK (Tele-science Resource Kit) at the Tele-science Support Center (TSC) will be used to develop this software.

The estimate for CoNNeCT Ground Software is 50K, with 34K SLOC of code reuse and 16K SLOC of new code. This estimate was done using the SEER-SEM (Software Evaluation and Estimation of Resources - Software Estimating Model) estimation tool.

#### 4.1.1.3 Test Software Estimation

Test software is estimated to be about one-half of the total payload avionics software, or 15K SLOCs. The rationale for this estimate is that test software will be written to test interfaces to subsystem hardware and these interfaces are approximately one-half of the functionality of the payload avionics. Interface software and ground software will be used in the testing, but specialized drivers or loops may be necessary to force complete path coverage during testing. During environmental testing, code may need to be written to stress the performance of the hardware under extreme conditions and to collect data on that performance. Therefore, 15K SLOCs, or approximately one-half the payload avionics software estimate is used for the test software estimate.

#### 4.1.1.4 SDR and Waveform Software Estimation

Detailed estimates for the SDR software and Waveforms being developed by the partners are not provided in this document. These estimates are managed and reported at the partner sites within their own management structure. Progress with these software items is measured by receiving intermediate deliverables as shown in the schedule portion of this document. Another difficulty with estimates for the SDRs and Waveforms is the different methods of counting FPGA functionality. An estimate of 10K SLOCs plus FPGA code has been given for the JPL SDR OE. Assuming the functionality of the different radios will be similar, that number can be used to give an overall project estimate for software size in SLOCs as shown in the next summary section.

#### 4.1.1.5 Software Estimation Summary

The following table shows a summary of the software estimation exercise performed in early 2008, along with characteristics such as number of developers, programming languages, and development methods.

**Table 4.1.1.5-1—Summary of CoNNeCT Software Estimation**

| Location | #of Dev | SLOC (K) | Programming Language(s) | Development Method |
|---|---|---|---|---|
| GRC – Avionics, Ground, Test | 7.5 | 95 | C++ | Incremental |
| GRC/GSFC – Waveform | 4-5 | 10K + FPGA | C, VHDL, XML | Waterfall |
| JPL – Radio w/ test waveforms | 3-4 | ~10K + FPGA | C, C++, Verilog, VHDL | Incremental |
| Harris - Radio & Waveform | 3 | 15K | C/C++ | Incremental |
| General Dynamics – Radio & Waveform | 2 | ~10K + FPGA | C/C++, Verilog, VHDL | Incremental |
| **TOTAL** | **19.5** | **~140K + FPGA** | | |

### 4.1.2 Staffing

The following spreadsheet shows the CoNNeCT staffing plan for software in WBS 6.0. Note that this only represents staffing at GRC, as staffing at other sites is managed by their local management. This staffing is detailed in the CoNNeCT Phasing Plan in eRoom and is managed by the CoNNeCT Software Engineering Lead. The staffing plan was developed based on the software estimate described in Section 4.1.1 and updated based on several CoNNeCT Control Board (CCB) requests for additional resources due to new requirements, revised project milestone dates, delayed hardware deliveries, and requests for additional development resources These CCB presentations are stored in the CoNNeCT eRoom in the CCB folder. FTE (Full-Time Equivalent) is the Civil Service workforce. WYE (Work-Year Equivalent) is the Support Service Contractor workforce. The 23 WYE total for Flight Software is about 276 person-months, or about 2.8 times the Ask Pete estimate, representing this growth.

**Table 4.1.2-1—Staffing for GRC CoNNeCT Software Development, December 2009**

| WBS Level | Personnel | FY 09 | FY10 | FY11 | TOTAL (WYE) | TOTAL (FTE) |
|---|---|---|---|---|---|---|
| 6.1 | Software Management – FTE | 1.0 | 1.9 | 1.5 | **-** | **4.4** |
| 6.2 | Flight Software – WYE | 6.0 | 9.0 | 8.0 | **23.0** | **-** |
| 6.3 | Ground Software - FTE | 0.5 | 0.5 | 0.5 | **-** | **1.5** |
| 6.3 | Ground Software - WYE | 1.0 | 2.1 | 2.1 | **5.2** | **-** |
| 6.4 | Test Software - WYE | 0.5 | 0.5 | 0.5 | **1.5** | **-** |
| **TOTALS** | | **9.0** | **14.00** | **12.60** | **29.7** | **5.9** |

### 4.1.3 Resource Acquisition

This section on resource acquisition applies only to the Payload Avionics and Ground Software being developed at GRC. The GGT Waveform information will be in the Waveform Development Plan.

#### 4.1.3.1 Development Systems and Workstations

##### 4.1.3.1.1 Desktop Computers

Each developer and software engineer on the CoNNeCT software team will have his own desktop PC or Workstation which will be provided through the ODIN contract at GRC. The normal desktop setup will be a laptop computer with a docking station at the developer's desk. This will provide mobility when moving between the office/desk, the lab, meetings, and travel.

#### 4.1.3.1.2 Development Systems

The Avionics group has purchased 3 software developments systems. Two will be commercial grade systems with a complete set of cards similar to the flight configuration. One system will be a qualification-grade system with a complete set of cards. The software team has purchased a fourth system with a subset of commercial grade cards to provide additional development resources, as a result of a CCB decision. These systems will be available for the software development team to develop code on during most of the early phases of the project. When the custom I/O cards for avionics are available in March/April 2010, the qualification-grade system will be converted to the Ground Integration Unit (GIU). One of the commercial grade systems will migrate to the Science and Technology team for Waveform testing and experiment development. The final two systems will remain with the software development team for debugging and testing during pre-launch or on-orbit checkout, or for developing future software upgrades as part of Phase II. These resources have been acquired by the avionics group under the WBS 5.0 budget.

### 4.1.3.2 Software Development Tools

#### 4.1.3.2.1 VxWorks and Associated Tools

VxWorks is the selected operating system for the payload avionics single-board processor, the AiTech S950. The VxWorks version 6.3operating system will be purchased with the Workbench development environment. Each software developer creating software for the payload avionics will require their own license for VxWorks. This will be provided as a dedicated license for the developer. Workbench is available with a site license to all developers at GRC. This software and tool suite will be purchased with WBS 6.0 funding. Yearly maintenance fees also apply.

#### 4.1.3.2.2 Enterprise Architect

Enterprise Architect is used as the Unified Process and UML tool for the all the project's lifecycle phases, requirements through verification and validation. Each software engineer that joins the team at GRC is provided their own licensed copy of Enterprise Architect. The acquisition of these software licenses is performed through the Glenn Engineering Support Services – 2 (GESS-2) contract and the CoNNeCT task order to the GESS-2 contract with WBS 6.0 funding.

#### 4.1.3.2.3 Visual Studio .NET

Some of the software engineers on CoNNeCT need copies and licenses for Visual Studio .NET. The acquisition of these licenses is performed through the CoNNeCT SpaceDoc delivery order contract with WBS 6.0 funding.

#### 4.1.3.2.4 Subversion

Subversion is used for configuration control of the CoNNeCT software. This is an open source version control system that is installed on a server in the CoNNeCT software development laboratory. Developers can access subversion via an internet connection to the server. Another open source tool, TortoiseSVN, can be installed on Windows systems to generate the repository structure in a Windows file structure. Subversion functions such as *update* and *commit* can then be invoked as commands within Windows.

#### 4.1.3.2.5 JIRA

JIRA is used as a bug tracking tool. It was developed by Atlassian (www.atlassian.com) and is integrated with Subversion. Bugs found in the code can be entered into JIRA, assigned to a developer, and used to track the status. Developers can access JIRA over the internet in a Web page and automated emails can be generated to inform developers about the status of bugs.

#### 4.1.3.2.6 Fisheye, Crucible

Licenses for other tools have been purchased. These tools allow in-line code inspections to be performed and provide visibility into the code repository that enhances productivity. These tools are being used on the CoNNeCT Project.

### 4.1.3.3 Facilities

The CoNNeCT software laboratory is located in Building 333, Room 302A at GRC. This room will house three software development systems (SDS) and is the primary location for payload avionics software development. As breadboard SDRs are delivered, they are integrated into a SDS for interface testing. The SDS that will become the GIU will initially be located in the high bay in building 110, and will move to a location in the high bay of building 333 for build-up with the engineering hardware for operational proximity to the CCC. The software team maintains a server for hosting the CoNNeCT software repository, license keys, tools, and other shared resources. Internet connectivity is required to allow remote development on the SDSs, access to the server, and connectivity to GRC.

## 4.1.4 Personnel Training

Because of the challenging technical nature of the embedded flight software development it is essential that the engineering staff is adequately trained.

The major training components include the following:

- An intensive one week training course for both the flight hardware and software development on that hardware for the entire team provided by the hardware vendor
- WindRiver's one week introduction to VxWorks and RTOS development for any embedded developers that are unfamiliar with VxWorks or RTOS
- Object Oriented Analysis and Design and UML training for any developers on the team that are unfamiliar with these techniques
- Some budget has been allocated for miscellaneous training that might become necessary for any members of the software development team

Note that this document does not address training for partner organizations, as that is covered in the agreements between GRC and the partner organization and is covered out of direct funding to that organization.

## 4.2 Project Planning

### 4.2.1 Work Breakdown Structure (WBS)

Software is Work Breakdown Structure 6.0. Section 3.1 describes the project WBS and where software is being developed. The project Master Schedule shows the detailed work being performed under each WBS.

### 4.2.2 Schedule Allocation

Figure 4.2.2-1 below shows how software development is planned during the allocated development schedule. The schedule is driven by delivery of hardware items, such as SDR breadboards and interface simulators, and is bounded by delivery to flight hardware for system verification and validation. Software development effort has been allocated across the schedule to meet the delivery of the hardware and to support testing as soon as hardware is available. Development has been split among the software development systems to allow parallel development and distribution of hardware among the systems in order to isolate interface testing. Note that details of the SDR software development is shown here only as it pertains to deliveries of software to GRC. The latest detailed schedule is kept by the Flight and Ground Software Leads and is rolled up into the Master Schedule. The Master Schedule is available on eRoom in the CoNNeCT Schedules folder.

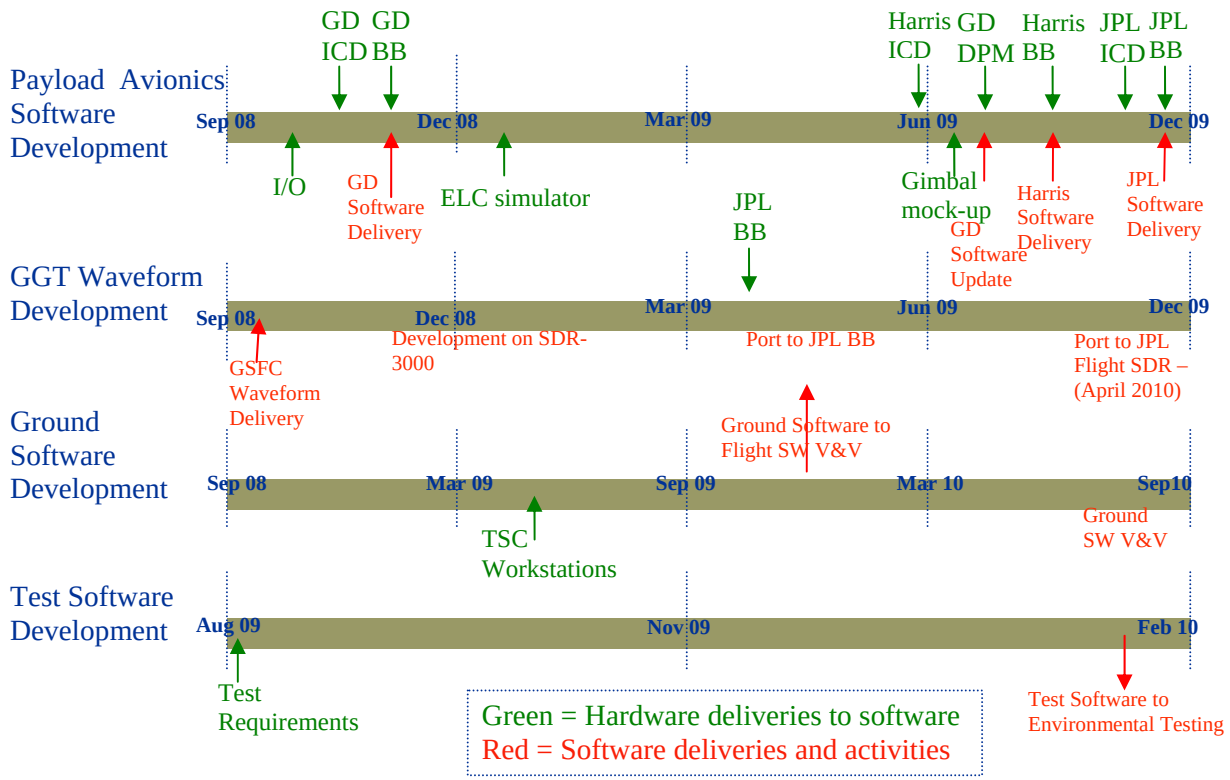



**Figure 4.2.2-1—Schedule Allocation for GRC CoNNeCT Software Development as of November 2009**

### 4.2.3 Resource Allocation

Resources requirements have been described in Section 3.2 Roles and Responsibilities and Section 4.1.2 Staffing. The table below summarizes the skills that are required for each of the resource requirements in each WBS for software.

**Table 4.2.3-1—Resource Allocation for GRC CoNNeCT Software Development**

| WBS Level | Personnel | Skills Required |
|---|---|---|
| 6.2 | Flight Software | Embedded software development; C/C++; real-time programming |
| 6.3 | Ground Software | GUI programming; Web-based development; data management/database |
| 6.4 | Test Software | Embedded software development; C/C++; real-time programming |
| 4.3 | Waveform | Embedded software development; C/C++; FPGA programming |

### 4.2.4 Budget Allocation

WBS 6.0 Software as has its own budget for labor, purchases, and travel. This budget is described in the CoNNeCT Phasing Plan that is in eRoom. The majority of the budget is allocated to labor, as most hardware is purchased either in WBS 5.0 Flight System, or WBS 7.0 Mission Operations. A miscellaneous hardware budget has been established to outfit the Software Development Lab with expected equipment purchases that are needed throughout development and budget is allocated for software licenses and maintenance. A travel budget has also been established based on expected travel.

**Table 4.2.3-1—Budget Allocation for GRC CoNNeCT Software Development as of November 2009**

| WBS Level | Personnel | FY09 | FY10 | FY11 | TOTAL ($K) |
|---|---|---|---|---|---|
| 6.1 | Software Management - Labor | 150K | 285K | 225K | **660K** |
| 6.1 | Software Management – Procurement | 0K | 10K | 0K | **10K** |
| 6.1 | Software Management – Travel | 8K | 13K | 8K | **29K** |
| 6.2 | Flight Software – Procurement | 959K | 1324K | 900K | **3183K** |
| 6.3 | Ground Software - Labor | 75K | 75K | 75K | **225K** |
| 6.3 | Ground Software – Procurement | 275K | 409K | 25K | **709K** |
| 6.3 | Ground Software – Travel | 6K | 3K | 0K | **9K** |
| 6.4 | Test Software - Procurement | 75K | 75K | 35K | **185K** |
| **TOTALS** | | **1548K** | **2194K** | **1268K** | **5010K** |

The detailed phasing plan for CoNNeCT, which contains all budget and resource allocations for the entire project lifecycle is kept in eRoom under the Financial folder and is only accessible to WBS Leads, Resources team, and Project Management.

## 4.3 Management Controls

### 4.3.1 Requirements Control

Software requirements for the CoNNeCT project will be managed per GRC-SW-7150.6 'Requirements Management' process.

The Requirements Management process states that the following procedure or checklist be followed:

- Identify and document new or modified requirements
- Assess impact of changes
- Present proposed changes to the Configuration Control Board
- If the Configuration Control Board approves the changes, then implement changes
- If the Configuration Control Board disapproves the changes, document the rationale and do not implement the change
- Maintain a Requirements Traceability Matrix with which to track requirements throughout the software development life cycle

The CoNNeCT Configuration Management Plan, GRC-CONN-PLAN-0002, describes the CoNNeCT Control Board (CCB) process. The CoNNeCT Software Configuration Management Plan, GRC-CONN-PLAN-0001, contains a Software Change Request Form and a Software Change Impact Assessment Form.

The software requirements are baselined following the Software Design Review. After the baseline has been released, any software changes must follow the above process and be approved by Project Management for budget and schedule impacts. Metrics will be collected on requirements all throughout the project and reported to the Software Discipline Lead Expert on a monthly basis.

### 4.3.2 Schedule Control

#### 4.3.2.1 GRC Flight Software Schedule Control

The CoNNeCT software development schedule will be monitored and controlled in several ways. The Master Schedule will be used to track progress against the plan at the project level. This schedule is reviewed weekly at the WBS Leads Meeting. Software Leads will maintain more detailed software subsystem schedules. A weekly Software Leads Meeting with the Payload Avionics and Ground Software Leads is held and the schedule will be assessed every other week at these meetings. A weekly CoNNeCT Software Working Group telecon is held with all the software developers on the project. This forum will be used to track the progress of software development at the partner sites and the Waveform development. Mitigation strategies have been developed during Risk Management exercises, and any deviation to the schedule of over a month will require invoking mitigation activities. The responsibility of monitoring and invoking mitigation strategies lies with the SWEL.

The final control is a monthly meeting with Software Engineering management at GRC, which includes presentation of monthly metrics showing progress against the plan. The CoNNeCT SWEL will request support from Software Engineering management should the need arise to invoke mitigation strategies.

The SWEL reports software schedule progress in the Monthly Project Report.

### 4.3.3 Budget Control

The Software Engineering Lead controls the WBS 6.0 budget. Twice a year, the Project Resources personnel hold a workshop to revise the phasing plans of each WBS. These workshops are to assess actual costs against planned costs, and to replan for the coming year or remainder of current year. At times, mitigation strategies which cost money, often to save schedule, might be invoked. Any budget liens are prepared and presented along with mitigation strategies.

The WBS 6.0 budget includes labor, procurements, and travel dollars. Support service contractors are accounted for as procurements. Refer to the CoNNeCT Project Plan, GRC-CONN-PLAN-0004 for the budget control processes.

The SWEL reports software budget status in the Monthly Project Report.

### 4.3.4 Reporting

Software progress will be reported on a monthly basis to Software Engineering management. The CoNNeCT Monthly Software Metrics spreadsheet will be used to collect and report data. This spreadsheet will be kept in eRoom and each submittal will be kept as a record.

The Project submits a Monthly Project Report to its Headquarters customer. This report has a fixed format for all WBS Leads to report schedule, budget, risks, and technical status. These reports are stored in eRoom.

Other reporting will include presentations to GRC upper management upon request, presentations as part of major project reviews, and the major software reviews per Section 6.5.1 Management Progress Reviews.

### 4.3.5 Metrics

The metrics for the PAS and Ground Software portions of the project will be tracked and reported to the Chief of the Flight Software Branch, the Chief Engineer and Project Management, and software team members on a monthly basis. The metrics will be gathered and placed in the Master Monthly Metrics Worksheet provided by the GRC Software Engineering Process Group (SEPG). Responsibility for collecting and reporting metrics resides with the Flight Software Lead and the Ground Software Lead. Some of the metrics that will be collected are: Source Lines of Code (SLOC); number of requirements; hours worked on the project; number of CSCIs and CSCs; test procedures; verification items; CPARs. Section 8.2 shows an example of the Master Monthly Metrics Spreadsheet. These spreadsheets will be stored in the Software folder on eRoom and delivered to the Chief of the Flight Software Branch.

## 4.4 Risk Management

The CoNNeCT project will use the Software Risk Evaluation method, from the Software Engineering Institute, to generate an initial set of software risks, with modifications to be consistent with GRC-CONN-PLAN-0007, CoNNeCT Risk Management Plan for the project. Through this process, a set of software risks will be developed, ranked, and mitigation plans defined. From this list, the top 3-5 risks, as ranked by the team, will be forwarded on to the Project Manager for management at the project level.

Risks are stored in RMIT (Risk Management Implementation Tool), the GRC online risk management tool. WBS Leads are responsible for entering and maintaining their risks. Only the Risk Review Board Chair may accept, close, or retire a risk, however.

On a monthly basis, the software team will meet and review current risks and add any new risks. Risks may be retired if they are no longer valid, or closed if they have been mitigated. The project holds weekly Risk Review Boards. New software risks will be presented at this board for acceptance as a project risk. This is also the formal board for closing or retiring a risk. See GRC-CONN-PLAN-0007 for more detail.

# 5.0 TECHNICAL PROCESS PLANS

This section specifies the lifecycle model; the technical methods, tools, and techniques to be used to develop the various software products; plans for establishing and maintaining the project development environment; the software acceptance plan; and the software maintenance plan.

## 5.1 Life Cycle Model

The in house software development at GRC for CoNNeCT will use a modified version of the Waterfall Life Cycle Model, called Incremental, for software development. In a traditional Waterfall Model the development starts with a Requirements Definition phase then moves to a Architecture Design phase, followed by the Implementation or Coding phase and finally to the Software Validation and Verification Phase. In the traditional Waterfall Model each phase is completed before moving on to the next phase. This model is popular because it is felt that time spent early on in software production can lead to greater economy later on in the software lifecycle; that is, it has been shown many times that a bug found in the early stages of the production lifecycle (such as requirements specification or design) is more economical (lower cost in terms of money, effort and time) to fix than the same bug found later on in the process.

One criticism of the traditional Waterfall model is that often some requirements are unknown or difficult to determine at the beginning of the project. The Software Team on CoNNeCT has prioritized the software requirements into three sets: 1) requirements that must be implemented prior to shipping the payload (incompressible test list), 2) requirements that ideally would be implemented and tested by ship date, but could be uploaded after launch, and 3) priorities that are needed for Phase II of the project and could be uploaded after launch with little or no impact. The Incremental process used on CoNNeCT will allow for some feedback to the requirement phase from both the design and implementation phases. This will allow some flexibility and compensate for unknown or ill defined requirements early in the project's lifecycle. The CoNNeCT software development will also take advantage of some limited code prototyping activities during both the requirements and design phases to help better define both requirements and design. The CoNNeCT software requirements have been prioritized to ensure that the most important functionality is implemented prior to shipping the payload to Japan. So while the highest priority requirements will be implemented first, with lower priority requirements being implemented in subsequent increments, the architecture must be developed to support all requirements, including those coming in later increments. The software design documents will describe the architecture.

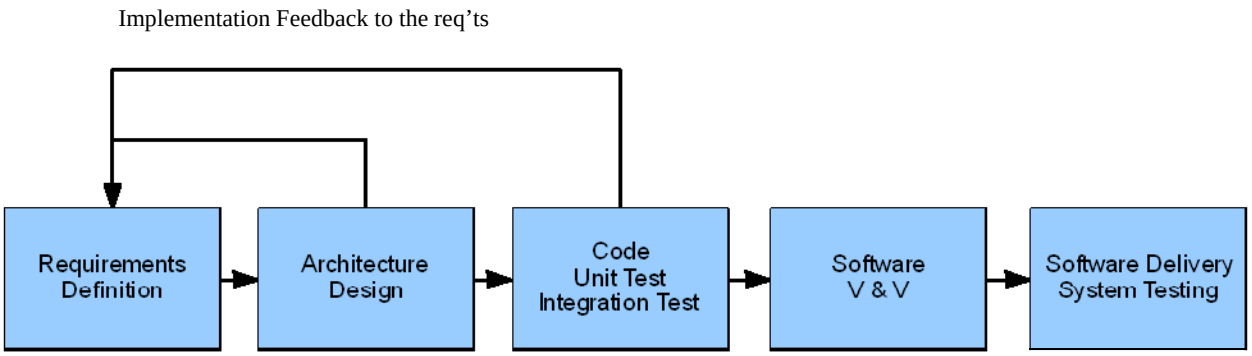



**Figure 5.1-1—Incremental Waterfall Life Cycle for CoNNeCT flight software development**

## 5.2 Methods, Tools, and Techniques

### 5.2.1 Object Oriented Analysis and Design

The software code for the embedded flight systems and ground software will be object oriented and written in C++. An object oriented analysis and design methodology will be used based on the Unified Software Development Process and UML. The Unified Software Development Process is a popular iterative and incremental software development process. It is Use Case driven and Architecture Centric. The UML Computer Aided Software Engineering (CASE) tool, Enterprise Architect, has been chosen to support this development methodology.

### 5.2.2 Coding Standards

All flight software written for the project will be written to conform to the CoNNeCT software flight coding standard which can be found in eRoom. This coding standard was based on the Ellemtel Coding Standard (http://www.doc.ic.ac.uk/lab/cplus/c++.rules), but customized for the project. The ground software will follow the ZIN coding standards that were used for the Fluid/Combustion Facility, as ground software from that project will be reused and modified for CoNNeCT. Each development organization has its own proven standards, and as such, no one standard will be mandated for all software. As part of peer reviews and inspections, adherence to these coding standards will be assessed.

### 5.2.3 Waveform FPGA Development Tools

#### 5.2.3.1 Model Prototyping

Matlab/Simulink tools will be used by GRC and GSFC to help develop and test waveform functions. A model of the waveform will be designed in Matlab Simulink to verify the performance of each of the waveform functions through the various modes. The model will allow testing of some real world link degradations. Similar modeling will be done by the other CoNNeCT partners.

#### 5.2.3.2 HDL Synthesis

Synthesis of the hardware description language (HDL) can be done using the Xilinx tool XST or other third party tools, such as Synplify Pro. Each CoNNeCT partner may use a different tool. GRC and GSFC will be using the Synplify Pro tool.

#### 5.2.3.3 HDL Simulation

The waveform development process GSFC and GRC includes modeling the waveform in a high level fixed-point model and then translating the model into hand coded HDL. GRC will use the ModelSim tool to verify and validate that the HDL code matches the model. Other CoNNeCT partners may use the same or similar tool.1

#### 5.2.3.4 Place and Route

Since all CoNNeCT SDRs have FGPAs manufactured by Xilinx, the ISE Foundation tool will be used for FPGA firmware development. This tool performs the place and route function of the application firmware, and yields the final target-specific configuration file (a.k.a. bitfile). This is usually where the platform specific wrapper and the waveform application are integrated.

#### 5.2.3.5 Debugging

There are many approaches to debugging an FPGA application, and these are closing tied to the SDR's design. Each CoNNeCT partner may use a different approach. GRC will be using the Xilinx Chipscope Pro tool to aide in the debugging process during waveform development. Chipscope Pro creates an internal logic analyzer that is accessed through the JTAG port. This adds the ability to probe lines that are not easily available and without additional test equipment. This also reduces the number of iterative configuration file generations necessary during development because many signals can be selectively analyzed without regenerating the bitfile.

### 5.2.4 Simulation Tools

#### 5.2.4.1 Matlab/Simulink with Real-Time Workshop Embedded Coder

Code for implementing the gimbal control pointing and tracking algorithms to move the antennas to track TDRS will be auto-generated from the Matlab/Simulink model using the Real-Time Workshop Embedded Coder add-on tool. This auto-generated code will be used as the basis for flight code, modified to conform to the CoNNeCT coding standards, and incorporated into flight code. This flight code will be verified like any other flight code in the system. Any changes required to the model will be done in Matlab/Simulink and new code will be generated, modified, tested, and replace existing code. Engineering Environment

### 5.2.5 Software Development Facilities

Section 4.1.3.3 describes the CoNNeCT software development lab in Building 333 that will be used to develop payload avionics flight software.

### 5.2.6 Local Area Networks

All of the CoNNeCT development systems, developer workstations, developer desktops, the CoNNeCT GIU and the CCC must have network interconnectivity. The Project will have to work with ODIN and GRC Network Operations Center (GNOC) to ensure that the network connectivity desired can be met while maintaining the correct level of network security. The Software Development Lab is currently covered by the Code D Security Plan for Labs. A Project IT Security Plan may need to be developed and on file at GRC in order to release flight software.

### 5.2.7 Office Space

#### 5.2.7.1 Support Service Contractor Developers

Office space, desks, chairs, and office supplies for the software engineers on the development team from the support service contractor work force are provided by the GESS-2 contract and the partner companies supporting the GESS-2 contract. Currently the office space for most support service contractors on the GESS-2 contract is in building 500, the Developmental Engineering Building (DEB).

## 5.3 Software Acceptance

A formal software acceptance test shall be performed on the flight system to verify/validate that the software system is ready to support flight operations. The Project will witness a requirements-driven demonstration of the software to show that it meets the baselined requirements. Flight simulations will test the software, using requirements-driven tests based on operational scenarios. At the end of the testing and any retesting required by nonconformance, the Project will conduct an acceptance review. The review will evaluate the test results, resolutions of CPARs (Corrective and Preventive Action Report), and any open issues or concerns from the Project. This process will be done as part of the final system verification and validation as the last step in the verification/validation process for software and will conclude with the FCA/PCA.

## 5.4 Software Maintenance Plan

A Software Maintenance Plan will be developed for final delivery of the software. This maintenance plan will detail the methods of developing new software for uploading to the payload, how that software will be tested before upload, how the Science and Technology team requests software uploads, and how new software development will be funded. This plan will also cover changes to Ground Software over time. At the end of the project life, the software will need to be retired and archived. This will also be covered in this plan, along with the legal rights to the software at that point and delivery of final project data.

# 6.0 SUPPORTING PROCESS PLANS

## 6.1 Configuration Management Plan

The CoNNeCT Software Team has chosen to use Subversion to implement configuration management of source code. All project source code, including that delivered by partners, will be stored and managed using Subversion in the CoNNeCT software repository. All formal builds will be verified against this repository. This repository will maintain all formal configuration releases of software for the life of the project.

For documentation, the Software Team will use the CoNNeCT Project eRoom. Software has its own folder for storing team documents, and formally released documents will be stored in the Configuration Released folder of eRoom. In addition, for each milestone review, there will be a folder in eRoom containing the milestone version of the documents under review.

CoNNeCT software development will follow the CoNNeCT Software Configuration Management Plan, GRC-CONN-PLAN-0001.

## 6.2 Verification and Validation

A Software Verification and Validation Plan (SV&VP) will be written for the CoNNeCT software system, document number GRC-CONN-PLAN-0118. This plan follows the CoNNeCT Systems Verification & Validation Plan, GRC-CONN-PLAN-0012, format, which was used as the template for the software document. This document will serve as the “Test Plan” document for the Payload Avionics Software (PAS). It will describe the planned PAS unit test cases, the PAS integration test cases and the ground to flight formal verifications.

For the formal verification and validation of the software a Software Test Procedure (STPr) will be written. This test procedure will follow the Software Test Procedure (STPr) Template, GRC-SW-TPLT-STPr. A Software Test Report (STR) will be written for the STPr using the Software Test Report (STR) Template, GRC-SW-TPLT-STR. The software test report will consist of the executed software test procedure, data collected during the test, test notes, signatures, and test dates. It will be scanned and assigned a separate CM controlled document number.

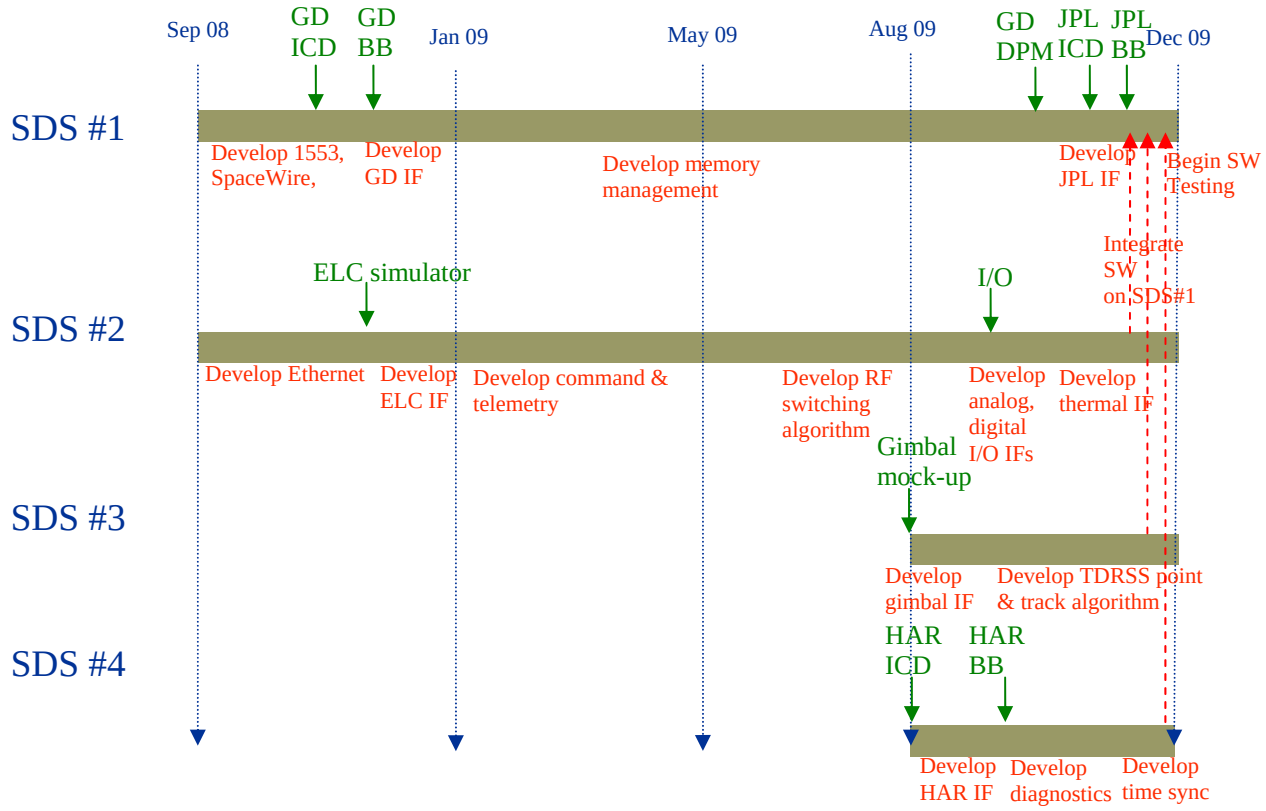
Software development and testing will take place in the CoNNeCT software lab, the CoNNeCT Control Center, and the high bay of building 333 (build-up location for the GIU).

### 6.2.1 Hardware Interface and Subsystem Testing

Software testing and verification/validation is driven and constrained by hardware availability. As hardware becomes available to the GRC software team to work with, it will be integrated into one of the software development systems (SDS) and interface testing will be performed. Once the interface has been tested and shown to work properly, functional tests of the hardware item and any associated software will be done. These are considered “interface” and “subsystem” tests and all test documentation will be kept and configuration controlled.

As interface and subsystem tests are used to prove out the functionality of hardware and software, the software used in those tests will be tagged in the version control system and migrated to the main software development system, which will contain the integrated code from the other systems. Note that not all subsystem hardware will be present on all software development systems, so some simulation, stimulation, or stubs may be required in order to be able to run the full software load on the main SDS.

The figure below depicts the use of the three SDSs for software development and testing. The GRC software team will work closely with the partner teams to define the capability that will be delivered in the SDR breadboards in order to phase the development and testing in the schedule time allotted. In addition, one of the SDSs and the JPL breadboard will be required for Waveform development. The Waveform development schedule is found in the Waveform Development Plan, GRC-CONN-PLAN-0076, at the following eRoom location: https://collaboration.grc.nasa.gov/eRoomReq/Files/NASAm1f1/CoNNeCT/0_8c6c8/GRC-CONN-PLAN-0076_GGT_WF_Dev_Plan-preliminary.doc.




Green = hardware or document deliveries to software; Red = software development activities

**Figure 6.2.1-1—Use of Software Development Systems for Flight Software Development as of November 2009**

### 6.2.2 Final Software Verification and Validation

The hardware available to the software team for testing may not be flight-fidelity. Therefore, as higher fidelity hardware arrives, tests will need to be rerun to verify operation with the flight hardware. In addition, not all flight hardware will be available to the software team for integration with the SDS. The Ground Integration Unit (GIU) will be a high fidelity unit that emulates the flight systems and will serve as a verification platform for the flight software. However, the final verification and validation of the software will not be complete until the software is migrated to the flight hardware and all flight components have been received. Therefore, the schedule time allotted for performing software verification and validation will be used for software testing on the SDS with breadboard radios. A full software test suite will be run during this time to verify performance of the software with the non-flight hardware. When that test has successfully completed, the software will be delivered to the flight hardware and software verification and validation will be performed.

The flight SDRs will arrive at GRC with working bootcode and Operating Environment. The bootcode and OE requirements will be verified and validated to the SDR Prime Item Functional Specification requirements by the supplying organization before delivery to GRC. If there are requirements that are difficult or impossible for the SDR vendors to verify without the complete system, arrangements can be made to verify these requirements at GRC, or, in some cases equipment may be taken to the partner site from GRC to complete testing.

The initial set of deliverable Waveforms will also have their requirements verified to the Waveform Specifications by the providing organizations. The CoNNeCT GIU, which will be used to verify software uploads after launch, can be used to verify and validate SDR Waveforms as necessary. The CoNNeCT GIU itself will need to be verified and validated to ensure it provides a valid platform for software verification.

### 6.2.3 Software Development Systems and Ground Integration Unit (GUI) Verification & Validation (V&V)

After the bulk of the software development is complete and the project moves into V&V, the software development systems will be delivered for other use. One SDS will remain with the software team for recreating any bugs that are found, developing future capabilities under Phase II, and for testing scenarios with the current flight load. Another SDS will be delivered to the Science & Technology Team (WBS 4.0) for developing waveforms and experiments. The final, qualification (non-commercial) unit will be delivered to the Mission Operations Team (WBS 7.0) for development of the Ground Integration Unit that will be used to verify new software builds prior to uploading to the payload.

All of these systems will need a verified and validated set of flight software on them. Once the software is verified on the flight system, that same software load will go through a similar verification on the two SDSs and the GIU. Any problems found will need to be investigated as to their nature. If a software problem is detected, a change may need to be made to the flight load and the software re-verified. If the problem is due to a hardware difference, a software modification (change in timing, simulated hardware response, etc.) may need to be developed to account for this. This software modification will require a new branch of software in the version control system that is associated with the hardware platform (SDS #1, SDS #2, or GIU). All branches of the flight software will need to be maintained throughout the life of the project.

The flight software team will be responsible for maintaining the software configuration on each SDS. Each SDS has a log book associated with it and the software team will keep spreadsheets containing hardware configuration information for each system over time. These spreadsheets will be configuration controlled in eRoom and updated as hardware configuration changes. Once the SDS is delivered to another organization, the hardware configuration management will be transferred to that organization (e.g. Mission Ops, Systems). The payload avionics software team will continue to maintain software configuration on each SDS for the life of the project.

### 6.2.4 Responsibility for Approval of Software Verification & Validation

The CoNNeCT Project Manager is ultimately responsible for signing off on the final software verification and validation. This will occur when the following are complete and available for review:

- Final Software Test Reports
- FCA/PCA Report
- Software Requirements Traceability and Verification Matrix
- Safety and Mission Assurance has signed off on above documents

Additional sign-off authority should be given to the CoNNeCT Software Engineering Lead, and Software Assurance and Software Engineering management.

## 6.3 Documentation

### 6.3.1 Documentation Tree

The two documentation trees below show the management branch of the documentation tree and the technical branch. The management branch shows only the GRC documents, as the management of the SDRs is performed at the partner sites.

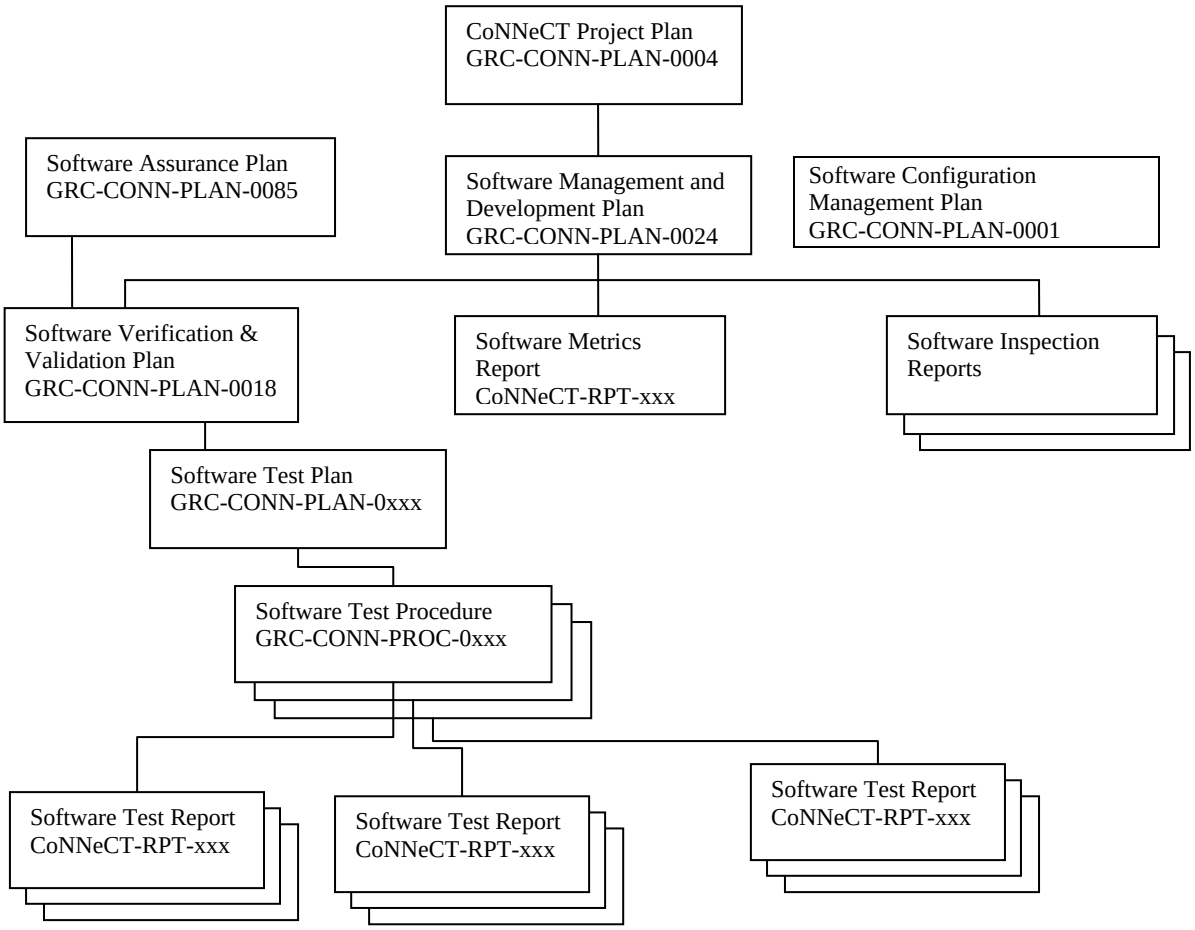



**Figure 6.3.1-1—CoNNeCT Software Documentation Tree, Management Branch**

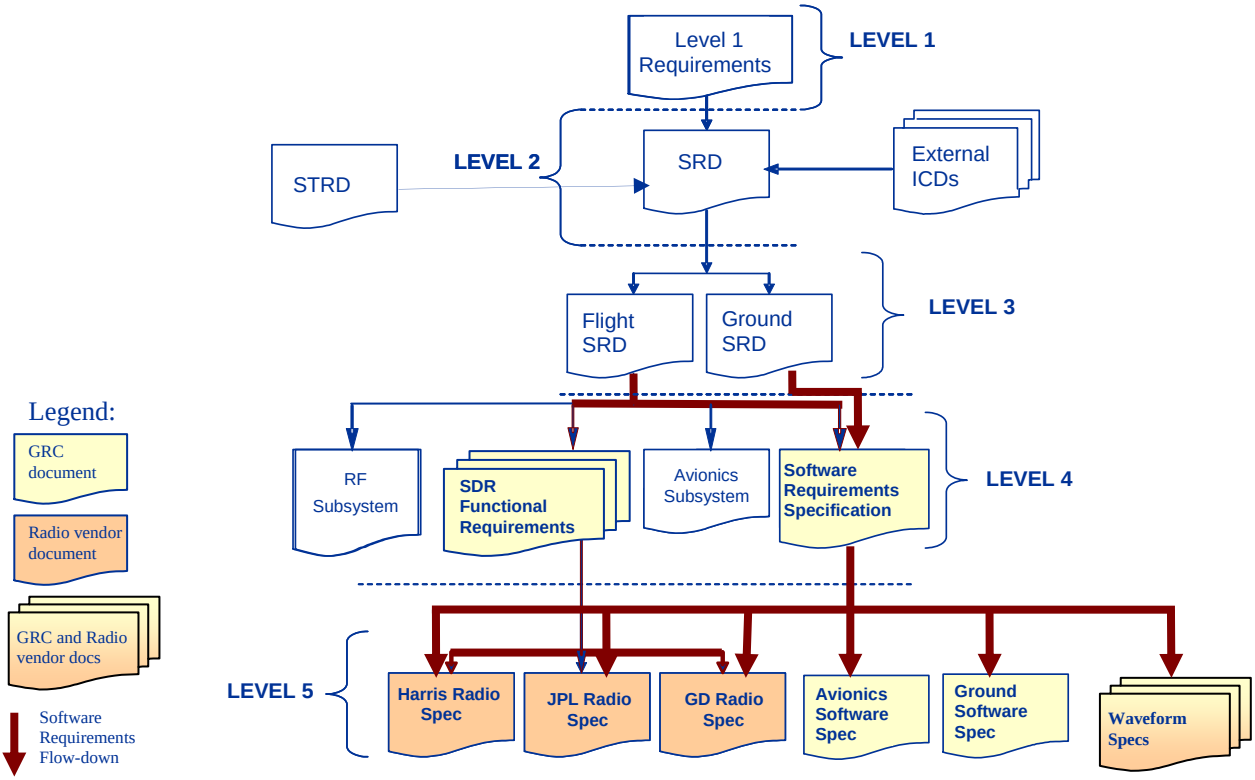



**Figure 6.3.1-2—CoNNeCT Software Documentation Tree, Technical Branch**

### 6.3.2 List of Documentation

The following documents will be developed in support of the GRC CoNNeCT class C software CSCIs.

| Software Documentation | Document Number | Template | Author(s) | Approver(s) | Due |
|---|---|---|---|---|---|
| Software Management and Development Plan | GRC-CONN-PLAN-0024 | GRC-SW-TPLT-SMP | Denise Varga | Project Manager | SRR, review<br>CDR, baseline |
| GRC GSFC TDRSS Waveform Development Plan | GRC-CONN-PLAN-0076 | GRC-SW-TPLT-SMP | Dale Mortensen | PI, SWEL | SRR, review<br>CDR, baseline |
| Software Configuration Management Plan | GRC-CONN-PLAN-0001 | GRC-SW-TPLT-SCMP | Denise Varga | Project Manager | SWRR, review<br>CDR, baseline |
| Software Assurance Plan | GRC-CONN-PLAN-0085 | | Mark Luboski | Project Manager,<br>S&MA Manager | SRR, review<br>CDR, baseline |
| Software Requirements Specification | GRC-CONN-REQ-0084 | GRC-SW-TPLT-SRS | Denise Varga | Project Manager, APMs,<br>Chief Eng. | SWRR, review<br>CDR, baseline |
| Payload Avionics Software Requirements Specification | GRC-CONN-REQ-0113 | GRC-SW-TPLT-SRS | Pronay Paul | Project Manager, SWEL,<br>Chief Eng. | FSWDR, review<br>CDR, baseline |
| Ground Software Requirements | GRC-CONN-SPEC-00129 | GRC-SW-TPLT-SRS | Lindsey Flash | Project Manager, SWEL,<br>Chief Eng. | GSWDR, review<br>CDR, baseline |
| GRCGSFC TDRSS Waveform Requirements | GRC-CONN-REQ-0077 | GRC-SW-TPLT-SRS | Dale Mortensen | PI, SWEL | PDR, review<br>CDR, baseline |
| Payload Avionics Software Design Document | GRC-CONN-DES-0149 | | Dale Walter | Project Manager, SWEL,<br>Chief Eng. | FSWDR, review<br>CDR, baseline |
| Ground Software Design Description | | | Jennifer Keller | Project Manager, SWEL,<br>Chief Eng. | GSWDR, review<br>GCDR, baseline |
| GRCGSFC TDRSS Waveform Design Description | GRC-CONN-BDD-0079 | | Dale Mortensen | PI, SWEL | PDR, review<br>CDR, baseline |
| Software Verification & Validation Plan | GRC-CONN-PLAN-0018 | | Larry Boyer | PM, SWEL, V&V Lead,<br>Sys. Eng. Lead | FSWDR, review<br>CDR, baseline |
| Software Test Plan | GRC-CONN-PLAN-xxx | GRC-SW-TPLT-STP | Software Test Lead | SWEL, Chief Eng. | CDR, review<br>TRR, baseline |
| Software Test Procedures | GRC-CONN-PROC-xxx | GRC-SW-TPLT-STPr | Software Test Lead | SWEL, Chief Eng. | CDR, review<br>TRR, baseline |
| Software Test Reports | GRC-CONN-RPT-xxx | GRC-SW-TPLT-STR | Software Test Lead | SWEL, Chief Eng. | FCA, review |
| Software Inspection/Peer Review Report | GRC-CONN-RPT-xxx | eRoom | Software Test Lead | None | Monthly |
| Software Users Manual | | GRC-SW-TPLT-SUM | Lindsey Flash | SWEL, Chief Eng. | CDR, review<br>FCA/PCA, baseline |
| Software Maintenance Plan | GRC-CONN-PLAN-xxx | GRC-SW-TPLT-SMPL | Dale Walter | Project Manager,<br>SWEL, Chief Eng. | CDR, review<br>FCA/PCA, baseline |

| Software Documentation | Document Number | Template | Author(s) | Approver(s) | Due |
|---|---|---|---|---|---|
| Software Version Description | | GRC-SW-TPLT-SVD | Software Leads | SWEL, Chief Eng. | FCA/PCA, baseline |
| Software Metrics Report | GRC-CONN-RPT-xxx | | Software Leads | None | Monthly |

## 6.4 Quality Assurance Plan

The CoNNeCT Software Quality Assurance Plan is contained in GRC-CONN-PLAN-0085, CoNNeCT Software Product Assurance Plan.

## 6.5 Reviews and Audits

### 6.5.1 Management Progress Reviews

The following software milestone reviews have been identified and scheduled for CoNNeCT:

- Software Requirements Review – November 2008
- Software Architecture Review – June 2009
- Flight Software Design Review – December 2009
- Ground Software Design Review – February 2010
- Functional Configuration Audit/Physical Configuration Audit – August 2010

Software milestone reviews follow the project reviews by approximately 2 months. GRC-CONN-PLAN-0001, CoNNeCT Software Configuration Management Plan details the baseline and associated configuration items that are under review at each of these milestones.

### 6.5.2 Developer Peer Reviews and Inspections

Peer Reviews and Inspections will be performed on software development products. When conducting peer reviews and inspections, GRC-SW-7150.12 'Formal Inspection and Peer Review' process will be followed. Some of the items that will be reviewed either by inspection or peer review are:

- Software requirements documents
- Interface documents
- Design documents
- Software code
- Test plans

Peer reviews and inspections will be identified for each software CSCI and will be part of the detailed subsystem schedule. Metrics on inspections will be collected and stored in the software development folder for the CSCI.

### 6.5.3 Quality Assurance Audits

The software quality assurance audits are addressed in GRC-CONN-PLAN-0085, CoNNeCT Software Product Assurance Plan.

## 6.6 Problem Resolution

Procedures for reporting, analyzing, prioritizing, and processing software problems are contained in GRC-CONN-PLAN-0001, CoNNeCT Software Configuration Management Plan.

## 6.7 Subcontractor Management Plans

The CoNNeCT Project has chosen to subcontract out the ground software development to ZIN Technologies. At one time, GRC considered contracting out part of the payload avionics flight software and ground software to BioServe at the University of Colorado. BioServe offers a Linux solution for command and telemetry on ISS, having about 10 years experience with life science experiments onboard ISS. Porting that solution to VxWorks proved to be expensive and required BioServe to have a software development system for their use, with little to no VxWorks experience on the BioServe team. However, considering the short schedule for CoNNeCT software development, the tight budget of CoNNeCT, and the experience and code base in ISS command and telemetry that already exists at GRC and ZIN Technologies, the decision was made to do the flight development in-house and the ground development at ZIN.

The SDR developers are not subcontractors, but partners with cooperative agreements with either GRC or NASA Headquarters. These agreements do not dictate the management of software development at the partner sites, but contain information on funding, schedule, technical capabilities, and deliverables of the flight SDRs, breadboards, and engineering models. The Software Engineering Lead holds a weekly telecon with software developers from all CoNNeCT development organizations to discuss interfaces, exchange technical information, and coordinate deliveries and technical capabilities. Formal deliveries are not negotiated at this telecon, as this requires the involvement of the Associate Project Manager (APM) at the partner organization. These types of requests must be made to the CoNNeCT Project Manager for negotiation with the APMs.

# APPENDIX A ACRONYMS AND ABBREVIATIONS

## A.1 Scope

This appendix lists the acronyms and abbreviations used in this document.

## A.2 List of Acronyms and Abbreviations

### Table A-1—Acronyms

| | |
|---|---|
| APM | Associate Project Manager |
| CASE | Computer Aided Software Engineering |
| CCC | CoNNeCT Control Center |
| CM | Configuration Management |
| CMMI | Capability Maturity Model Integration |
| CoNNeCT | Communications, Navigation, and Networking re-Configurable Testbed |
| CPAR | Corrective and Preventive Action Report |
| DEB | Developmental Engineering Building |
| ELC | ExPRESS Logistics Carrier |
| ExPA | ExPRESS Payload Adapter |
| ExPRESS | Expedite the PRocessing of Experiments to Space Station |
| FCF | Fluid and Combustion Facility |
| FTE | Full-Time Equivalent |
| GD | General Dynamics |
| GESS | Glenn Engineering Support Services |
| GFSC | Goddard Space Flight Center |
| GGT | GRC GFSC TDRSS (Waveform) |
| GIU | Ground Integration Unit |
| GNOC | GRC Network Operations Center |
| GPP | General Purpose Processor |
| GPS | Global Positioning System |
| GRC | Glenn Research Center |
| GSW | Ground Software |
| GUI | Graphical User Interface |
| IOP | Input/Output Processor |
| ISS | International Space Station |
| JPL | Jet Propulsion Laboratory |
| NEN | Near-Earth Network |
| PAS | Payload Avionics Software |
| PI | Principal Investigator |
| POCC | Payload Operations Control Center |
| PVT | Position, Velocity, Time |
| RMIT | Risk Management Implementation Tool |
| SDR | Software Defined Radio |
| SEER-SEM | Software Evaluation and Estimation of Resources - Software Estimating Model |
| SEPG | Software Engineering Process Group |
| SLOC | Source Lines of Code |

| | |
|---|---|
| SMDP | Software Management and Development Plan |
| STRS | Space Telecommunications Radio System |
| SW | Software |
| SWEL | Software Engineering Lead |
| UML | Universal Modeling Language |
| WBS | Work Breakdown Structure |
| WYE | Work Year Equivalent |

# APPENDIX B DEFINITIONS

## B.1 Scope

This appendix lists the definitions used in this document.

## B.2 List of Definitions

**Table B-1—Definitions**

Firmware *The combination of a hardware device and computer instructions and data that reside as read-only software on that device."[IEEE]*

# APPENDIX C NPR 7150.2 COMPLIANCE MATRIX FOR SWR-102

Requirement SWE-102 of NPR 7150.2 'NASA Software Engineering Requirements' levies a set of requirements that must be met by a project's Software Management Plan. The table below lists the NPR 7150.2 requirements for an SMP and the corresponding section(s) of this CoNNeCT Software Management and Development Plan that address those requirements.

### Table C-1—Traceability from the SMP Template to SWE-102 of NPR 7150.2

| SMP Section(s) | 7150.2 Requirement |
|---|---|
| | The Software Development or Management Plan shall contain: [SWE-102] |
| 3 Project Organization | a. Project organizational structure showing authority and responsibility of each organizational unit, including external organizations (i.e., Safety and Mission Assurance, Independent Verification and Validation (IV&V), Independent Technical Authority (ITA), and NASA Engineering and Safety Center (NESC)). |
| 1.3 Classification of Systems and Subsystems | b. The classification of each of the systems and subsystems containing software as defined in Appendix B. |
| 1.4 Project Compliance to NPR 7150.2 | c. Tailoring compliance matrix for approval by the designated ITA Warrant Authority, if the project has any variants, waivers, or exceptions to this NPR. |
| 5.2 Methods, Tools, and Techniques<br>5.3 Engineering Environment | d. Engineering environment (for development, operation, or maintenance, as applicable), including test environment, library, equipment, facilities, standards, procedures, and tools. |
| 4.2.1 Work Breakdown Structure<br>4.2.2 Schedule Allocation<br>4.2.3 Resource Allocation<br>4.2.4 Budget Allocation | e. Work breakdown structure of the life cycle processes and activities, including the software products, software services, non-deliverable items to be performed, budgets, staffing, physical resources, software size, and schedules associated with the tasks. |
| 4.3.1 Requirements Control | f. Management of the quality characteristics of the software products or services. |
| 4.3.1 Requirements Control | g. Management of safety, security, privacy, and other critical requirements of the software products or services. |
| 6.7 Subcontractor Management Plan | h. Subcontractor management, including subcontractor selection and involvement between the subcontractor and the acquirer, if any. |
| 6.2 Verification and Validation | i. Verification and validation approach. |
| 1.3.3 Project Deliverables<br>3.1 Organizational Structure<br>3.2 Roles and Responsibilities<br>4.4 Risk Management<br>5.4 Software Acceptance<br>6.5 Reviews and Audits | j. Acquirer involvement. |
| 3.1 Organizational Structure<br>3.2 Roles and Responsibilities | k. User involvement. |
| 4.4 Risk Management | l. Risk management. |

| SMP Section(s) | 7150.2 Requirement |
|---|---|
| 5.3 Engineering Environment | m. Security policy. |
| 5.1 Life Cycle Model<br>6.5 Reviews and Audits | n. Approval required by such means as regulations, required certifications, proprietary, usage, ownership, warranty, and licensing rights. |
| 4.3.2 Schedule Control<br>4.3.4 Reporting | o. Process for scheduling, tracking, and reporting. |
| 4.1.4 Personnel Training | p. Training of personnel, including project unique software training needs. |
| 5.1 Life Cycle Model | q. Software life cycle model including description of software integration and hardware/software integration processes, software delivery, and maintenance. |
| 6.1 Configuration Management Plan | r. Configuration management. |
| 6.3 Documentation | s. Software documentation tree. |
| 6.5 Reviews and Audits | t. Peer review/inspection process of software work products. |
| 6.2 Verification and Validation | u. Process for early identification of testing requirements that drive software design decisions, e.g., special system-level timing requirements/checkpoint restart. |
| 4.3.5 Metrics | v. Software metrics. |
| 6.3 Documentation | w. Content of software documentation to be developed on the project. |

## C.1 Master Monthly Metrics Spreadsheet – Example

| | |
|---|---|
| **Report Date:** | |

| | |
|---|---|
| **No. of Hrs Spent on Metrics this Month:** | |

| | |
|---|---|
| **Total CS Hours Used to Date:** | |
| **Total WYE Hours Used to Date:** | |
| **Total Cost to Date:** | |

| **Size** | **Total to Date** |
|---|---|
| Source Lines of Code: | |

| **Requirements Volatility** | **Total** | **Added *** | **Modified *** | **Deleted *** | **Total Changes** |
|---|---|---|---|---|---|
| No. of Requirements: | | | | | 0 |
| No. of TBDs/TBRs: | | | | | 0 |

| | |
|---|---|
| **Total # of Requirements** | |
| Implemented in Design: | |
| Implemented in Code: | |
| Tested: | |

| **Quality** | **Total** | **New *** | **Open** | **Closed** | **Withdrawn** |
|---|---|---|---|---|---|
| Inspection Defects: | | | | | |
| RIDs: | | | | | |
| PRACAs: | | | | | |

| **Risk Management** | **Total** | **New *** | **Mitigated *** | **Accepted *** | **Closed *** | **Total Changes** |
|---|---|---|---|---|---|---|
| No. of Risks: | | | | | | 0 |