National Aeronautics and Space Administration

Independent Verification and Validation (IV&V)

James Webb Space Telescope (JWST) IV&V Technical Scope and Rigor (TS&R) (Mid-Year Update)

IV&V Technical Scope and Rigor

JWST

FY2016 (TSR-94)

Updated Date: 6/6/2016

Status: Final

NASA Independent Verification and Validation Facility

100 University Drive, Fairmont WV 26554

Purpose

To document where IV&V effort will be applied on the JWST IV&V Project, the approaches considered for performing technical tasks that will start in the next planning period (2ND half of FY16), the basis for deciding on an approach or approaches, and detailed information regarding each selected approach.

Assessment of Technical Scope

The JWST Observatory is comprised of a spacecraft platform, 4 science instruments and Ground Segment, which includes optical corrective calculations – called Wavefront Sensing, which are sent to the Observatory.

The development of JWST software follows the Rational Unified Process (RUP) for the development of the Spacecraft, ISIM, MIRI, NIRCam, NIRSpec and FGS subsystems. The process incorporates an iterative build approach where each build includes requirements, design (UML) and source code.

The IV&V team has assessed the JWST mission according to its capabilities in order to focus on those areas deemed to be of highest importance and greatest risk to mission success.

Table 1 below identifies the JWST Mission thread and its associated rationale for being mission critical or not.

Table 1 – JWST Mission Thread/Capabilities

Mission Thread	Description
J1 Launch Ops	As part of launch ops, observatory performs incremental activation and configuration of JWST via on-board sequencer. Observatory provides telemetry via S-band transmitter. Using this capability, ground operators assess the state of health of the observatory. Towards the end of launch phase, observatory confirms the breakwire separation and sends confirmation to the ground. Observatory then fires off thrusters for stabilization
J2 Deployment & TCM	The goal of this capability is to deploy the solar arrays, radiator shield, Sunshields, OTE, HGA and perform trajectory correction to set the Observatory on its journey to L-2 orbit.
J3 Cruise & Commissioning	The goal of this capability is to fully calibrate the JWST instruments (NIRCam, NIRSpec, MIRI, Guider, and NIRISS) and perform mirror alignment (using Wavefront Sensing and Control capability), and place the Observatory into L2 orbit.

Mission Thread	Description
J4 Real-Time Ops	Using this capability, observatory performs real time operations during a planned communication contact between the Ground and the Observatory. After completing the cruise and commissioning phase, the observatory will have 4 hours of realtime contact with the ground in a typical 24 hour period (However observatory is designed to operate without ground contact for up to 10 days or longer if necessary). During this 4 hour window, ground will receive science and engineering data stored on the solid state recorder (SSR), ground will upload or modify observation plan files, perform health checks, wheel momentum updates. Ground performs these actions via realtime commands.
J5 Event Driven Ops	The goal of this capability is to perform autonomous (without Ground contact) Event Driven (ED) ops at L2 orbit. ED ops are the ordered sequence of operation steps. ED ops examples include: Perform Science measurements, Science Instruments (SI) calibration, momentum unloading etc.
J6 Onboard Fault Mgmt	The Observatory's On-board Fault Management capability detects a fault on the Observatory, classifies it depending on severity, and reconfigures the Observatory to minimize the fault's impact and ensure Observatory safety.
J7 Science Operations Planning	The goal of this capability is to prepare science and operations plans for uplink. As part of this capability ground gathers input from astronomers, evaluates and chooses observation programs, schedules the selected observations, generates the Observation Plan with associated schedules & products (which includes selected engineering and calibration operations activities).
J8 Post-Pass Ops	The goal of this capability is for the ground control to perform processing, archiving, and distribution (to the scientists) of JWST data after the end of Real Time contact with the Observatory. Ground controllers also perform engineering data evaluation to support long-term trending, analysis or anomaly investigation, and generation of SI calibration sequences.
J9 Grd Based Fault Mgmt – OBS Faults	This capability helps ground operators, to recover the Observatory from a fault condition. Based on the most recent engineering telemetry, ground operators determine that a Level 1, Level 2 or Level 3 fault has occurred and determine corrective action. Once the fault-specific actions have been tested and validated on the ground, the operators perform these actions during the next available realtime contact.
J10 Grd Based Fault Mgmt – Grd Faults	This capability helps ground operators, to recover from a fault condition with the ground based equipment. Ground operators determine if the fault is with the communication equipment, or the general operation and archiving servers. When the fault-specific actions have been taken, ground operators verify the status of ground based equipment during the next realtime contact.

The mission threads, or capabilities, become the context by which the JWST CSCIs are assessed in the IV&V team's Risk Based Assessment (RBA). The role that each CSCI contributes to each thread is determined and assessed, especially where software is concerned and the extent to which it is involved. Based on this assessment, the criticality of the system and the priority of analysis are determined. Figure 1 shows the results of this analysis. The set of chosen activities is determined in part to the results of this analysis, in conjunction with previous analysis on past JWST flight software builds which have proven to be very successful.

JWST Capabilities / CSCIs	J1: Launch Operations	J2: Deployment & TCM	J3: Cruise & Commission	J4: Real-time Operations	J5: Event Driven Operations	J6: Onboard Fault Mgmt	J7: Science Operations Planning	J8: Post-Pass Operations	J9: Grd Based Fault Mgmt – Obs Faults	J10: Grd Based Fault Mgmt – Grd Faults
Analysis Priority -->	4	2	3	5	6	1	N/A	N/A	7	N/A
SC	P	XX	P	P	XX	P			x	
DEU		P	x			x				
ADU		x	XX		XX	x				
CCE			XX		XX	x				
ISIM			P		P	XX				
NIRCam			XX		XX	x				
NIRSpec MSS			XX		XX	x				
NIRSpec DS			XX		XX	x				
NIRSpec ICE			XX		XX	x				
MIRI Opt System			XX		XX	x				
MIRI Cryocooler App			XX		XX	x				
FGS/TFI			XX		XX	x				
OSS			P	x	P					
FOS	x	XX	x	P	x		x	P	P	P
WEx			XX		x		x			
WAS			XX		x		x			
MCS			XX		x		x			
PPS							P			
DMS								x		
PRDS							x			

LEGEND	
x	FSW involved
XX	CSCI plays a key role in capability
P	Primary CSCI(s) for given capability or aspect of the capability

Figure 1 – JWST Capability to CSCI Mapping and Analysis Priority

The JWST IV&V Project scope has remained largely static for the past five years. There was some re-scoping performed around the time of model-based IV&V introduction, but that was minimal. The PBRA and RBA had been updated in March 2016 to reflect the current state of the development project and to incorporate IV&V understanding of the project.

The table below indicates the JWST analysis activities to be performed in FY16.

Table 2 – JWST Analysis Activities for FY16

Key	
1	Code Quality Only
2	Code Inspection Only
Δ	Change Impact Analysis (CIA) Only
✓	Planned
E	Pending evalua ion
	N/A or performed in another FY

	Requirements	Design	Implementation	Test	Integration	Security	Independent Test
Ground Segment							
FOS – Build 5			✓	✓			
FOS – Build 6	✓	✓				✓	
WSS-WEx – Build 5	Δ		1	✓			
WSS-MMS – Build 5	Δ	Δ	2Δ	Δ			
OSS – Build 4				✓			
OSS – Build 5	✓	✓	✓	✓	✓		
Observatory Segment							
FM-BTA – Build 2.2		✓			✓		
SCSs – B2.2	✓			✓	✓		✓
Spacecraft Element							
EMTB Build 2.2				✓	✓		
SC FSW – Build 2.2					✓		✓
SC FSW – Build 3.0	✓	✓	✓	✓		E	✓
DEU FSW – Build 1.1	✓	✓	✓	✓			
DEU FSW – Build 1.2	✓	✓	✓	✓			
ISIM							
ISIM Risk Reduction Analysis				✓	✓		
ISIM FSW – Build IC14.22	Δ		Δ	Δ	✓		✓
ISIM FSW – Build IC14.3	Δ		2Δ	Δ	✓		E
FGS – Build 5.5	Δ		2	✓			
FGS – Build 6.0	Δ		2Δ	✓			
MIRI CC – Build 1.1				Δ			
ISIM FSW – Build IC14.31	Δ		Δ	Δ	✓		E
NIRCam – Build 4.2.01	Δ			Δ			
NIRSpec.ICE – Build 3.1.12	Δ			Δ			
NIRSpec.MS – Build 6.2	Δ			Δ			
NIRSpec.DS – Build 6.2	Δ			Δ			

ISIM FSW – Build IC14.31+	Δ		Δ	Δ	✓		E

For the sake of determining technical scope, all products delivered for I&T in previous FYs are assumed to be complete. If these products are modified then IV&V will assess the change for software impact, which in turn could result in additional work needing to be performed to ensure the validity of past assurance statements.

Technical Rigor

Given the numerous CSCIs and development organizations, portions of the JWST program can be in various phases in any given FY. Not all TF Goals are targeted in any single FY due to these schedule constraints. The following describes rationale for TF Goals that are not fully covered in FY16.

The 1.x TF Goals (1.1, 1.2, 1.3, 1.4, 1.5, and 1.6) are management related and are therefore not tracked in the TS&R. Additionally, TF Goals, 1.0, 2.0, 3.0, 4.0, 4.1, 5.0, 6.0, and 7.0 are not method related and are not tracked in the TS&R. These items appear blank in the TF Coverage table, but these are not gaps in coverage.

Concept:

The "Concept" oriented TF Goals of 2.1, 2.2, 2.3, 2.4, 2.5 and 2.6 are not covered in FY16. The segments and subsystems of the project have already surpassed the concept phase (coverage of these TF Goals in prior years).

Requirements:

All of the Capabilities and CSCIs within the JWST project receive full TF coverage of all Requirements related goals (3.x). The partial or absent coverage in the table is a side effect from deferring analysis of multiple CSCIs in the FM portion of the analysis. Conversely, the FM coverage gaps are covered by the incremental analysis of individual components that have occurred in prior years or the current FY16.

The one caveat is the high impact SCS entity, which is not derived from parent requirements. In this sense, there is no way to achieve TF Goal 3.2 and it is not targeted.

Test:

All of the capabilities and CSCIs within the JWST project receive full TF coverage of all Test related goals (4.x). Regarding regression testing covered by TF Goal 4.3, IV&V reviews the regression test plans and reviews regression test products; however, IV&V does not decide the tests that will be executed. Therefore, only partial coverage is targeted. Regarding Simulations, IV&V does not validate or verify the simulations directly except under very specific circumstances. This has occurred for the SC simulator and EMTB (SCE) simulators where the nearly all capabilities and entities scored high in Impact and some scored high in likelihood. Additional Rigor has been applied to this area. Similarly, TF Goal 4.8 (validation of the test environment) is not fully covered by IV&V and only partial coverage is targeted. The analysis performed on the test approach and the test results partially cover this goal, but there is no explicit validation of the test environment. The exception is the SC, EMTB (SCE), and SCS test environments where impact scores were exceptionally high. The FM entity coverage of the 4.x TF goals is expressed in the EMTB mapping.

Design:

TF Goal 5.3 and 5.5 are only partially targeted for Yellow entities (FOS, MMS, and DEU). Full coverage of the Design (5.x) TF Goals is targeted for all red entities and for FY16 this includes SC and FM entities. The coverage of TF Goal 5.5 for FM is only partial because the FM system is larger than the sum of its parts and IV&V cannot verify a nearly infinite set of off-nominal scenarios. No FY16 target for WEx, ISIM, NIRSpec, or FGS. These were covered in prior years.

Implementation:
Full coverage of the Implementation TF Goals (6.x) are targeted for all Red (critical) entities and this include SC in FY16. The exception in FY16 is that DEU, a yellow entity is treated like a red entity and receives full coverage of the 6.x TF Goals. Yellow entities (MMS, OSS, ISIM, and FGS) target full coverage of 6.1 and 6.6 with partial coverage of 6.2 using code quality checks. Only Code Quality Checks (TF 6.2) are targeted for FOS and WEx in FY16. No FY16 target for NIRSpec. This item was covered in prior years. SCS is a high impact (Red) entity, but given the data centric manner in which it is designed and implemented, only partial coverage of Implementation TF Goals are targeted.

Operations and Maintenance:
Operations and Maintenance TF Goals are not usually targeted by IV&V analysis, prior to or even after launch. Due to the large number of development organizations and reliance or deferral of responsibility to ground operations rather than FSW automation, the JWST IV&V has scoped partial coverage of TF Goal 7.2 (to ensure portions of the developed software are ready for integration) for the high impact (Red) entities SC and SCS. Full coverage of TF Goal 7.8 (to ensure that documentation is sufficient for operations and maintenance) has been scoped for high impact entities (Red) SC, SCS, and EMTB (SCE). The following TF Goals are not covered in FY16 and never will be targeted: 7.1, 7.3, 7.4, 7.5, 7.6, and 7.7.

The figure below provides the technical framework coverage for each technical task that will start in FY16.

IVV 09-1 Rev O. Full and Partial Coverage for Current Fiscal Year FY2015

Subsystem Mappings	1.0	1.1	1.2	1.3	1.4	1.5	1.6	2.0	2.1	2.2	2.3	2.4	2.5	2.6	3.0	3.1	3.2	3.3	3.4	3.5	4.0	4.1	4.1.	4.1.	4.1.	4.1.	4.1.	4.2	4.3	4.4	4.5	4.6	4.7	4.8	5.0	5.1	5.2	5.3	5.4	5.5	5.6	6.0	6.1	6.2	6.3	6.4	6.5	6.6	7.0	7.1	7.2	7.3	7.4	7.5	7.6	7.7	7.8
GND.FOS																F	F	F	F				F	F	F	F	F	F	P		F	F	F	P		F	F	P	F	P	F			P													
GND.WSS.WEx																F	F	F	F				F	F	F	F	F	F	P		F	F	F	P										P												
GND.WSS.MMS																F	F	F	F				F	F	F	F	F	F	P		F	F	F	P		F	F	P	F	P	F		F	P		F		F									
GND.OSS																F	F	F	F				F	F	F	F	F	F	P		F	F	F	P		P			F				F	P				F									
OBS.SCE.CTP.SC																F	F	F	F	F			F	F	F	F	F	F	P	F	F	F	F	F		F	F	F	F	F	F		F	F	F	F	F	F			P						F
OBS.SCE.DEU																F	F	F	F	P			F	F	F	F	F	F	P		F	F	F	P		F	F	P	F	P	F		F	F	F	F	F	F									
OBS.ISIM.ISIM																F	F	F	F				F	F	F	F	F	F	P		F	F	F	F									F	F		F	F	F									
OBS.ISIM.NIRSpec.ICE																F	F	F	F				F	F	F	F	F	F	P		F	F	F	P																							
OBS.ISIM.NIRSpec.MS																F	F	F	F				F	F	F	F	F	F	P		F	F	F	F																							
OBS.ISIM.NIRSpec.DS																F	F	F	F				F	F	F	F	F	F	P		F	F	F	F																							
OBS.ISIM.FGS																F	F	F	F				F	F	F	F	F	F	P		F	F	F	P									F	F		P	F	F									
OBS.SCE.SCS																F		F	F	P			F	F	F	F	F	F	P		F	F	F	F			P	P	P	P	P		P	P	P	F	P	P			P						
OBS.SCE.FM																P	F		F	F			P	P	P	P	P									F	F	F	F	P	F		P		F		F	P									F
OBS.SCE.EMTB																							F	F	F	F	F	F	P	F	F	F	F	F																							

Figure 3. Technical Framework for Current Fiscal Year

TF Coverage Key:

- = Full coverage of TF element.
- = Partial coverage of TF element.
- = No coverage of TF element.

Figure 2 – JWST Technical Framework Coverage for FY16

Activity 1: Validate Requirements via Bidirectional Tracing

Method:	M-3, Version 1.3 (Current Status: Approved)
Method Title:	Validate Requirements by Inspecting Bidirectional Traces
Method Synopsis	Method for tool-supported manual inspection of a set of requirements to assess and document the degree to which they adequately specify a logical decomposition of the parent requirements, and any functional allocations identified by the developer. This method addresses the integrity of the requirements structure, and identifies faults in correctness, completeness, consistency, and bi-directional tracing of parent to child requirements.
Subsystem/Entity	GND.FOS; GND.WSS.WEx; GND.WSS.MMS; GND.OSS; OBS.SCE.CTP.SC; OBS.SCE.DEU; OBS.ISIM.ISIM; OBS.ISIM.NIRSpec.ICE; OBS.ISIM.NIRSpec.MS; OBS.ISIM.NIRSpec.DS; OBS.ISIM.FGS
Required Method Revisions (if any)	None
Technical Goal:	Assess the quality of the requirements (set) and the degree to which they adequately specify a logical decomposition of the parent requirements 3.1: Ensure that the system requirements are of high quality and are consistent with acquirer needs as they relate to the system's software. (Partial) 3.2: Ensure that all (in-scope) parent requirements are represented in the appropriate child requirements and that the child requirements do not introduce capability that is not required. 3.3: Ensure that the software requirements are of high quality and adequately meet the needs of the system with respect to expectations of its customer and users, operational environment, and both functional and non-functional perspectives. (Partial)
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev 0
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	FOS - Build 6 WSS-WEx - Build 5 (Change Impact Analysis (CIA) only) WSS-MMS - Build 5 (CIA Only)

	OSS - Build 5 OSS - Build 6 SC FSW - Build 3.0 DEU FSW - Build 1.1 ISIM IC14.21+ (CIA Only) ISIM IC15.0 (CIA Only) ISIM IC15.0+ (CIA Only) NIRSpec.ICE - Build 3.1.12 (CIA Only) NIRSpec.MS - Build 6.2 (CIA Only) NIRSpec.DS - Build 6.2 (CIA Only) FGS - Build 4.97 FGS - Build 5.0 FGS - Build 5.0+
Inputs (includes Technical Reference):	1. Requirement traces developed by the Mission Project 2. Additional Reference Artifacts to understand the requirements to be assessed, including IV&V Project Technical Reference 3. Capabilities defined to level of analysis (PBRA, RBA) [scope]
Prerequisites:	Requirements and developer provided traces loaded into traceability tool (spreadsheet / analysis tool)
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution. Success Criteria: • All requirements have been reviewed for software applicability, and any obvious defects • The quality of each software-related performance and functional requirement has been evaluated • For each collection of software-related performance and functional requirements associated to a parent requirement, that collection has been evaluated for completeness, correctness and consistency in the context of the parent requirement • All issues have been synthesized into concerns • All requirement assessments with no issues have been synthesized into confirmations
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	JWST has been under IV&V for many years, so the analysis methods have evolved over time and the rigor

	has had a tendency to grow as capability is added. The following is a breakdown of the analysis methods and assurance levels by phase consistent with the current methods in Compass and consistent with the current and past practices in IV&V analyses. The assurance increases by adding additional layers of assurance, and in some cases, the assurance in one level covers lower levels. Req 1. M-3: Validate Requirements by Inspecting Bidirectional Traces a. Assurance Objective: Requirement set is complete and maps up, down, and laterally consistent with the parents, children, and siblings. b. TF: 3.2 (F), 3.1 (P), 3.3 (P) 2. M-2: Validate Requirements by Inspecting Against Quality Criteria and System/Software Background Artifacts a. Assurance Objective: Each Requirement is of high quality (complete, correct, unambiguous, etc). b. TF: 3.1 (F), 3.3 (F), 3.4 (F) 3. M-1: Validate Interface Requirements by Inspection Against Component Interfaces a. Assurance Objective: Necessary Interface Requirements are specified. b. TF: 3.4 (F) 4. M-103: Verify and Validate Requirement Implementation using Flow Diagrams to Uncover Missing, Conflicting, or Unnecessary Behavior a. Assurance Objective: The Requirement Set (intent) is complete and correct (no conflicts, gaps, or unknowns) b. Note: Method encompasses Assurance Level 2 and 3 mentioned above c. TF: 3.1 (P), 3.3 (F), 3.4 (F), 3.5 (P) 5. M-37: End-to-End Fault Management Verification through Database Development and Analysis a. Assurance Objective: (FM Specific) Validate that each FM related requirement is identified and there are no missing SW behaves to achieve the necessary reliability of the System. b. TF: 3.2 (F), 3.4 (F), 3.5 (F)
Concerns:	None
Method Application Notes: None	

Required Tools:	Engineering worksheets or analysis tool to document results of tracing analysis
Empirical Evidence:	Completeness/correctness/consistency status in engineering worksheets (or analysis tools) for each requirement, list of orphans, list of childless parents
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	None

Activity 2: Validate Requirements via Quality Requirement Analysis (QRA)

Method:	M-2, Version 1.3 (Current Status: Approved)
Method Title:	Validate Requirements by Inspecting Against Quality Criteria and System/Software Background Artifacts
Method Synopsis	Method for tool-supported manual inspection of a set of requirements to assess and document the degree to which they individually and collectively exhibit desired quality attributes (Unambiguous, Verifiable, Consistent, Correct, Complete, Design Independent, Feasible). Use documents that inform the validation target to insure that the requirements are complete and correct.
Subsystem/Entity	GND.FOS; GND.WSS.WEx; GND.WSS.MMS; GND.OSS; OBS.SCE.SCS; OBS.SCE.CTP.SC; OBS.SCE.DEU; OBS.ISIM.ISIM; OBS.ISIM.NIRSpec.ICE; OBS.ISIM.NIRSpec.MS; OBS.ISIM.NIRSpec.DS; OBS.ISIM.FGS
Required Method Revisions (if any)	None
Technical Goal:	3.1 Ensure that the system requirements are of high quality and are consistent with acquirer needs as they relate to the system's software. 3.3 Ensure that the software requirements are of high quality and adequately meet the needs of the system with respect to expectations of its customer and users, operational environment, and both functional and non-functional perspectives. 3.4 Ensure that the requirements for software interfaces with hardware, user, operator, and other systems are adequate to meet the needs of the system with respect to expectations of its customer and users, operational environment, dependability and fault tolerance, and both functional and non-functional perspectives.
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev 0
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	FOS - Build 6 WSS-WEx - Build 5 (Change Impact Analysis (CIA) only) WSS-MMS - Build 5 (CIA Only) OSS - Build 5

	OSS - Build 6 Stored Command Sequences (B2.2) SC FSW - Build 3.0 DEU FSW - Build 1.1 ISIM IC14.21+ (CIA Only) ISIM IC15.0 (CIA Only) ISIM IC15.0+ (CIA Only) NIRSpec.ICE - Build 3.1.12 (CIA Only) NIRSpec.MS - Build 6.2 (CIA Only) NIRSpec.DS - Build 6.2 (CIA Only) FGS - Build 4.97 FGS - Build 5.0 FGS - Build 5.0+
Inputs (includes Technical Reference):	Needs, Goals and Objectives document, opsCon, trades, higher level requirements, and any other additional background materials to understand the requirements to be assessed
Prerequisites:	None
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	JWST has been under IV&V for many years, so the analysis methods have evolved over time and the rigor has had a tendency to grow as capability is added. The following is a breakdown of the analysis methods and assurance levels by phase consistent with the current methods in Compass and consistent with the current and past practices in IV&V analyses. The assurance increases by adding additional layers of assurance, and in some cases, the assurance in one level covers lower levels. Req 1. M-3: Validate Requirements by Inspecting Bidirectional Traces a. Assurance Objective: Requirement set is complete and maps up, down, and laterally consistent with the parents, children, and siblings. b. TF: 3.2 (F), 3.1 (P), 3.3 (P) 2. M-2: Validate Requirements by Inspecting Against Quality Criteria and System/Software Background Artifacts

	a. Assurance Objective: Each Requirement is of high quality (complete, correct, unambiguous, etc). b. TF: 3.1 (F), 3.3 (F), 3.4 (F) 3. M-1: Validate Interface Requirements by Inspection Against Component Interfaces a. Assurance Objective: Necessary Interface Requirements are specified. b. TF: 3.4 (F) 4. M-103: Verify and Validate Requirement Implementation using Flow Diagrams to Uncover Missing, Conflicting, or Unnecessary Behavior a. Assurance Objective: The Requirement Set (intent) is complete and correct (no conflicts, gaps, or unknowns) b. Note: Method encompasses Assurance Level 2 and 3 mentioned above c. TF: 3.1 (P), 3.3 (F), 3.4 (F), 3.5 (P) 5. M-37: End-to-End Fault Management Verification through Database Development and Analysis a. Assurance Objective: (FM Specific) Validate that each FM related requirement is identified and there are no missing SW behaves to achieve the necessary reliability of the System. b. TF: 3.2 (F), 3.4 (F), 3.5 (F)
Concerns:	None
Method Application Notes: None	
Required Tools:	Engineering worksheets (or database) document the assessment of the quality attributes
Empirical Evidence:	Engineering worksheets (or database) documenting the results of the assessment of the quality attributes for each requirement and conclusions about the completeness and correctness of the set(s) of analyzed requirements. Evidence must include an indication that each requirement was examined for every qualitative attribute (i.e. correctness, completeness, etc.) and the version of the requirements that was assessed.
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	While this method's effectiveness is largely a function of the analyst(s) performing it, it can

	nevertheless be applied in a relatively short time period to provide valuable feedback to a mission project Other methods may need to be applied to garner additional rigor and confidence in the correctness, completeness, and overall consistency of the requirements

Activity 3: Validate Requirements via Inspection of Component Interfaces

Method:	M-1, Version 1.0 (Current Status: Approved)
Method Title:	Validate Interface Requirements by Inspection Against Component Interfaces
Method Synopsis	Manual method that focuses attention on evaluating integration requirements against specific interfaces to assess the coverage of those interfaces by the requirements. Faults reported by this method include interface components not specified in requirements, requirements with no implementation in defined interfaces, and integration requirements that fail to exhibit the five quality attributes (correctness, consistency, completeness, accuracy, verifiability) in context of the 3 questions.
Subsystem/Entity	GND.FOS; GND.WSS.WEx; GND.WSS.MMS; GND.OSS; OBS.SCE.SCS; OBS.SCE.CTP.SC; OBS.SCE.DEU; OBS.ISIM.ISIM; OBS.ISIM.NIRSpec.ICE; OBS.ISIM.NIRSpec.MS; OBS.ISIM.NIRSpec.DS; OBS.ISIM.FGS
Required Method Revisions (if any)	None
Technical Goal:	Considering the Three Questions, assess requirements for - Correctness - Completeness - Consistency - Accuracy - Verifiability 3.4 Ensure that the requirements for software interfaces with hardware, user, operator, and other systems are adequate to meet the needs of the system with respect to expectations of its customer and users, operational environment, reliability and fault tolerance, and both functional and non-functional perspectives
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev O
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	FOS - Build 6 WSS-WEx - Build 5 (Change Impact Analysis (CIA) only) WSS-MMS - Build 5 (CIA Only) OSS - Build 5

	OSS - Build 6 Stored Command Sequences (B2.2) SC FSW - Build 3.0 DEU FSW - Build 1.1 ISIM IC14.21+ (CIA Only) ISIM IC15.0 (CIA Only) ISIM IC15.0+ (CIA Only) NIRSpec.ICE - Build 3.1.12 (CIA Only) NIRSpec.MS - Build 6.2 (CIA Only) NIRSpec.DS - Build 6.2 (CIA Only) FGS - Build 4.97 FGS - Build 5.0 FGS - Build 5.0+
Inputs (includes Technical Reference):	None identified
Prerequisites:	None identified
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	JWST has been under IV&V for many years, so the analysis methods have evolved over time and the rigor has had a tendency to grow as capability is added. The following is a breakdown of the analysis methods and assurance levels by phase consistent with the current methods in Compass and consistent with the current and past practices in IV&V analyses. The assurance increases by adding additional layers of assurance, and in some cases, the assurance in one level covers lower levels. Req 1. M-3: Validate Requirements by Inspecting Bidirectional Traces a. Assurance Objective: Requirement set is complete and maps up, down, and laterally consistent with the parents, children, and siblings. b. TF: 3.2 (F), 3.1 (P), 3.3 (P) 2. M-2: Validate Requirements by Inspecting Against Quality Criteria and System/Software Background Artifacts a. Assurance Objective: Each Requirement is of high quality (complete, correct, unambiguous, etc).

	b. TF: 3.1 (F), 3.3 (F), 3.4 (F) 3. M-1: Validate Interface Requirements by Inspection Against Component Interfaces a. Assurance Objective: Necessary Interface Requirements are specified. b. TF: 3.4 (F) 4. M-103: Verify and Validate Requirement Implementation using Flow Diagrams to Uncover Missing, Conflicting, or Unnecessary Behavior a. Assurance Objective: The Requirement Set (intent) is complete and correct (no conflicts, gaps, or unknowns) b. Note: Method encompasses Assurance Level 2 and 3 mentioned above c. TF: 3.1 (P), 3.3 (F), 3.4 (F), 3.5 (P) 5. M-37: End-to-End Fault Management Verification through Database Development and Analysis a. Assurance Objective: (FM Specific) Validate that each FM related requirement is identified and there are no missing SW behaves to achieve the necessary reliability of the System. b. TF: 3.2 (F), 3.4 (F), 3.5 (F)
Concerns:	None
Method Application Notes: None	
Required Tools:	Engineering worksheets (or database) to document the assessment
Empirical Evidence:	Assessment of completeness with respect to Step A (coverage of interfaces) Assessment of completeness with respect to Step B (coverage of 3 questions perspectives) For each requirement, assessment of each quality attribute.
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	Interface - The hardware and supporting software for the physical interface itself (e.g. RS-422, packet definitions) Integration - Everything needed to make two systems work together (e.g. physical interface, data requirements of each system

	Difference - Integration includes interface - Interface defines the protocols so that the systems can talk to each other - Integration adds concept of when, exactly what data is needed For best results, a qualitative assessment (to meet goal 3.3) of the integration requirements should precede this method.

Activity 4: Verify and Validate Requirements via Flow Diagrams

Method:	M-103, Version 1.0 (Current Status: Approved)
Method Title:	Verify and Validate Requirement Implementation using Flow Diagrams to Uncover Missing, Conflicting, or Unnecessary Behavior
Method Synopsis	Method uses Flow Diagrams to analyze software implementation of requirements to ensure the correct and complete implementation of requirements on a system level as well as an atomic level. (Level is dependent upon the abstraction in the modeling chosen by the analyst as well as the available level of artifacts being targeted). Further, the method is applied to the source code that is not specified by requirements or not specified directly.
Subsystem/Entity	OBS.SCE.CTP.SC; OBS.SCE.DEU
Required Method Revisions (if any)	None
Technical Goal:	1. Gain system/software level understanding 2. Uncover ambiguous or missing behaviors 3. Uncover conflicting or undesired behaviors 4. Uncover failure scenarios (identify whether the FSW is protected against off-nominal/adverse conditions/inputs) 3.1 Ensure that the system requirements are of high quality and are consistent with acquirer needs as they relate to the system's software. 3.3 Ensure that the software requirements are of high quality and adequately meet the needs of the system with respect to expectations of its customer and users, operational environment, and both functional and non-functional perspectives. 3.4 Ensure that the requirements for software interfaces with hardware, user, operator, and other systems are adequate to meet the needs of the system with respect to expectations of its customer and users, operational environment, dependability and fault tolerance, and both functional and non-functional perspectives. 3.5 Ensure that software requirements meet the dependability and fault tolerance required by the system and provide the capability of controlling

	identified hazards and do not create hazardous conditions.
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev 0
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	SC FSW - Build 3.0 DEU FSW - Build 1.1
Inputs (includes Technical Reference):	Applicable Technical Reference (i.e. use case scenarios)
Prerequisites:	FSW is not developed using behavior models (uml activity, state, or similar), the models are not of high enough fidelity to analyze code behavior, or the models are not sufficient to provide system level understanding. Note: The process of generating the diagrams can be used to gain system level understanding even when the diagrams duplicate developer products.
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	JWST has been under IV&V for many years, so the analysis methods have evolved over time and the rigor has had a tendency to grow as capability is added. The following is a breakdown of the analysis methods and assurance levels by phase consistent with the current methods in Compass and consistent with the current and past practices in IV&V analyses. The assurance increases by adding additional layers of assurance, and in some cases, the assurance in one level covers lower levels. Req 1. M-3: Validate Requirements by Inspecting Bidirectional Traces a. Assurance Objective: Requirement set is complete and maps up, down, and laterally consistent with the parents, children, and siblings. b. TF: 3.2 (F), 3.1 (P), 3.3 (P) 2. M-2: Validate Requirements by Inspecting Against Quality Criteria and System/Software Background Artifacts a. Assurance Objective: Each Requirement is of high quality (complete, correct, unambiguous, etc). b. TF: 3.1 (F), 3.3 (F), 3.4 (F)

	3. M-1: Validate Interface Requirements by Inspection Against Component Interfaces a. Assurance Objective: Necessary Interface Requirements are specified. b. TF: 3.4 (F) 4. M-103: Verify and Validate Requirement Implementation using Flow Diagrams to Uncover Missing, Conflicting, or Unnecessary Behavior a. Assurance Objective: The Requirement Set (intent) is complete and correct (no conflicts, gaps, or unknowns) b. Note: Method encompasses Assurance Level 2 and 3 mentioned above c. TF: 3.1 (P), 3.3 (F), 3.4 (F), 3.5 (P) 5. M-37: End-to-End Fault Management Verification through Database Development and Analysis a. Assurance Objective: (FM Specific) Validate that each FM related requirement is identified and there are no missing SW behaves to achieve the necessary reliability of the System. b. TF: 3.2 (F), 3.4 (F), 3.5 (F)
Concerns:	None
Method Application Notes: This method applies to multiple phases of development; however, the mapping here is to Requirement phase artifacts thus the Requirement TF Goals are mapped.	
Required Tools:	Drawing Package (i.e. Visio, Together, etc) Code Viewing and analysis tool (i.e. Understand)
Empirical Evidence:	1. Control Flow diagrams and corresponding notes capture IV&V analysis, questions, and concerns. - "Analysis/Findings Report" 2. Diagrams show/trace the behaviors and their source - "Coverage Reports." 2) Test/Analysis Scenarios (nominal and off-nominal) derived from flow paths - "Scenario Report" Example Assurance Claims/Statements: The FSW does not contain any unnecessary functions or "features" The FSW implements all required behaviors with appropriate response to off-nominal conditions The FSW verification suite completely covers the required behaviors as well as the nominal, off-

	nominal, and non-required (i.e. feature) paths in the FSW
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	None

Activity 5: Verify Requirements via Construct of Fault Management Database

Method:	M-37, Version 1.1 (Current Status: Approved)
Method Title:	End-to-End Fault Management Verification through Database Development and Analysis
Method Synopsis	Method applies to the development of a System Fault Management Database that captures relationships and behaviors to aid in the analysis of Fault Management Systems in large distributed systems. A series of incremental databases are built, maintained, and integrated to derive a total system perspective. The database is an extension of the SMART/AWB traceability database where Fault Management Requirements are identiifed along with their drivers (i.e. parent requirements, Failure Mode and Effect Analysis, and Fault Tree Analysis). In addition, the database is further developed to include an "Event Network" that provides a quasi-dynamic (i.e. executable) abstraction of the system. Queries are used to produce scenarios where interactions are more complex, and potential resource conflicts are likely. Manual analysis is then used to determine the validity of such scenarios and uncover defects. For scenarios that are too complex or time-dependent to analyze manually, test procedures are developed for execution by either the developer or IV&V. Primarily, the method is intended to reduce the set of large or infinite scenarios for analysis/test to a reduced set that either has errors identified, or has a higher likelihood of error.
Subsystem/Entity	OBS.SCE.CTP.SC
Required Method Revisions (if any)	None
Technical Goal:	1. Gain system level understanding

	2. Uncover ambiguous or missing behaviors 3. Uncover conflicting or undesired behaviors 4. Uncover failure scenarios Ensure the system is capable of identifying, controlling, preventing, or properly responding to any credible fault scenario. Ensure every fault is properly controlled by a requirement. Uncover credible Failure Scenarios to target analysis and independent tests Maintain dependencies and conflicts between system entities to aid in change impact analysis 3.1 Ensure that the system requirements are of high quality and are consistent with acquirer needs as they relate to the system's software. 3.2 Ensure that all (in-scope) parent requirements are represented in the appropriate child requirements and that the child requirements do not introduce capability that is not required. 3.4 Ensure that the requirements for software interfaces with hardware, user, operator, and other systems are adequate to meet the needs of the system with respect to expectations of its customer and users, operational environment, dependability and fault tolerance, and both functional and non-functional perspectives. 3.5 Ensure that software requirements meet the dependability and fault tolerance required by the system and provide the capability of controlling identified hazards and do not create hazardous conditions.
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev O
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	SC FSW - Build 3.0
Inputs (includes Technical Reference):	Minimum Required: 1. Failure Modes and Effects Analysis (FMEA) 2. Fault Tree Analysis (FTA) 3. FSW Requirements and parents 4. Master Command And Telemetry Database (MCTDB) 5. Stored Command Sequence (SCS) Database (Relative Time or Absolute Time sequence of actions)

	Recommended for completeness: 1. Fault Management Algorithm Document (FMAD) 2. FSW Algorithms Requirement Documents 3. FSW Source Code 4. Fault Management Control Flows (or equivalent system and scenario designs) 5. EQ (Hardware) Specs
Prerequisites:	Database Engineer (Design, Maintain, Administer) Subject Matter Expert (Fault Management, Sub-systems)
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	JWST has been under IV&V for many years, so the analysis methods have evolved over time and the rigor has had a tendency to grow as capability is added. The following is a breakdown of the analysis methods and assurance levels by phase consistent with the current methods in Compass and consistent with the current and past practices in IV&V analyses. The assurance increases by adding additional layers of assurance, and in some cases, the assurance in one level covers lower levels. Req 1. M-3: Validate Requirements by Inspecting Bidirectional Traces a. Assurance Objective: Requirement set is complete and maps up, down, and laterally consistent with the parents, children, and siblings. b. TF: 3.2 (F), 3.1 (P), 3.3 (P) 2. M-2: Validate Requirements by Inspecting Against Quality Criteria and System/Software Background Artifacts a. Assurance Objective: Each Requirement is of high quality (complete, correct, unambiguous, etc). b. TF: 3.1 (F), 3.3 (F), 3.4 (F) 3. M-1: Validate Interface Requirements by Inspection Against Component Interfaces a. Assurance Objective: Necessary Interface Requirements are specified. b. TF: 3.4 (F)

	4. M-103: Verify and Validate Requirement Implementation using Flow Diagrams to Uncover Missing, Conflicting, or Unnecessary Behavior a. Assurance Objective: The Requirement Set (intent) is complete and correct (no conflicts, gaps, or unknowns) b. Note: Method encompasses Assurance Level 2 and 3 mentioned above c. TF: 3.1 (P), 3.3 (F), 3.4 (F), 3.5 (P) 5. M-37: End-to-End Fault Management Verification through Database Development and Analysis a. Assurance Objective: (FM Specific) Validate that each FM related requirement is identified and there are no missing SW behaves to achieve the necessary reliability of the System. b. TF: 3.2 (F), 3.4 (F), 3.5 (F)
Concerns:	None
Method Application Notes: This method applies to multiple phases of development; however, the mapping here is to Requirement phase artifacts thus the Requirement TF Goals are mapped.	
Required Tools:	Database Tool (Access, MySQL, SQL Server, etc) (Database Conversion Tools - as needed to convert project databases to IV&V's DB format) MS Excel (to capture results, ingest inputs, etc)
Empirical Evidence:	1. Requirement/Behavior deficiencies (incomplete, missing, conflicting) uncovered during database development and analysis 2. Database entry deficiencies (incomplete, missing, conflicting) uncovered during database development and analysis 3. Independent Test Scenarios and Results
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	None

Activity 6: Validate Software Design via Algorityhm Qualiative Attribute Assessment

Method:	M-42, Version 1.0 (Current Status: Approved)
Method Title:	Validate algorithm design through algorithm qualitative attribute assessment
Method Synopsis	Manually analyze algorithm qualitative attributes (unambiguous, logically independent, verifiable, consistent, correct, feasible) against algorithm description documentation.
Subsystem/Entity	OBS.SCE.CTP.SC
Required Method Revisions (if any)	None
Technical Goal:	Evaluate the adequacy of the software algorithms (both theory and logical design) for meeting the needs of the system. Software algorithms may include autonomous decision making using rule-based logic, high critical items such as Guidance Navigation and Control along with Fault Protection. 5.5: Partial: Ensure that complex algorithms have been correctly derived, provide the needed behavior under off nominal conditions and assumed conditions, and that the derivation approach is known and understood to support future maintenance.
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev 0
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	SC FSW - Build 3.0
Inputs (includes Technical Reference):	Software Requirements Specification System Requirements Specification Algorithm description (design documentation, algorithm handbooks, etc.) Failure Modes documentation (if available) Digital signal filtering concept documentation Safety Rule documentation (if available)
Prerequisites:	System Requirements Review (SRR) has been completed and the Software Requirements Specification and applicable Algorithm description documentation has matured to a point to conduct analysis.
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.

Rationale for Approach:	JWST has been under IV&V for many years, so the analysis methods have evolved over time and the rigor has had a tendency to grow as capability is added. The following is a breakdown of the analysis methods and assurance levels by phase consistent with the current methods in Compass and consistent with the current and past practices in IV&V analyses. The assurance increases by adding additional layers of assurance, and in some cases, the assurance in one level covers lower levels. 1. M-41: Verify Software Interface Design by Inspection Against Interface Requirements a. Assurance Objective: Necessary Interfaces are designed. b. TF: 5.1 (P), 5.4 (F) 2. (New) M-39: Verify Software Design by Inspecting Traces to Requirements and Software Architecture a. Assurance Objective: Each Requirement is captured in the design and the design is of high quality (complete, correct, unambiguous, etc) b. TF: 5.1 (F), 5.2 (F), 5.3 (P), 5.4 (P), 5.5 (P), 5.6 (F) c. Alternate Method for MBD i. M-8: Verify Autogenerated Software Implementation by Inspecting Traces from Requirements to Rational Rose Design Model ii. TF: 5.1 (F) 3. M-103: Verify and Validate Requirement Implementation using Flow Diagrams to Uncover Missing, Conflicting, or Unnecessary Behavior a. Assurance Objective: Requirement (and Design Element) Set is implemented correctly and completely. b. Note: Method encompasses Assurance Level 2 mentioned above c. TF: 5.1 (F), 5.2 (F), 5.3 (F), 5.4 (F), 5.5 (F), 5.6 (P) 4. M-42: Validate algorithm design through algorithm qualitative attribute assessment a. Assurance Objective: Complex algorithm is complete and correct. b. Note: SME to assess algorithms for correctness and completeness c. TF: 5.5 (P)
Concerns:	None
Method Application Notes: None	

Required Tools:	Engineering worksheets to document the assessment of the quality attributes. ORBIT (as appropriate) to record deficiencies communicated to the project. Risks (as appropriate) to record potential deficiencies. IV&V Lessons Learned database.
Empirical Evidence:	Engineering worksheets documenting the results of the assessment of the quality attributes for each algorithm and conclusions about the quality of the set(s) of analyzed algorithms. Evidence must include an indication that each algorithm was examined for applicable qualitative attribute (i.e. correctness, completeness, etc.). Also, as knowledge is acquired by performing this method, it should be considered as a potential update to the existing Technical Reference.
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	Wikipedia search on "Algorithm" yielded industry focused links to resources used to develop the algorithm characteristics for this analysis. Also found that algorithm analysis is referenced in NPR-7150.2A (SWE-111).

Activity 7: Verify and Validate Software Design via Flow Diagrams

Method:	M-103, Version 1.0 (Current Status: Approved)
Method Title:	Verify and Validate Requirement Implementation using Flow Diagrams to Uncover Missing, Conflicting, or Unnecessary Behavior
Method Synopsis	Method uses Flow Diagrams to analyze software implementation of requirements to ensure the correct and complete implementation of requirements on a system level as well as an atomic level. (Level is dependent upon the abstraction in the modeling chosen by the analyst as well as the available level of artifacts being targeted). Further, the method is applied to the source code that is not specified by requirements or not specified directly.
Subsystem/Entity	OBS.SCE.CTP.SC
Required Method Revisions (if any)	None
Technical Goal:	1. Gain system/software level understanding 2. Uncover ambiguous or missing behaviors 3. Uncover conflicting or undesired behaviors 4. Uncover failure scenarios (identify whether the FSW is protected against off-nominal/adverse conditions/inputs) 5.1 Ensure that all requirements (e.g. SRS and IRS) are represented in the appropriate elements of the design (e.g. SDD and IDD) and that the design does not introduce capability that is not required. 5.2 Ensure that the design provides the required capability (meeting software architecture and software requirements), is able to reliably meet user needs, and is sufficiently stable to proceed with implementation. 5.3 Ensure that the proposed software architecture satisfies the needs of the system, and that it is a feasible solution (i.e. will successfully satisfy the needs of the system, while still being practical). 5.4 Ensure that the internal and external software interface designs are provided for all (in-scope) interfaces with hardware, user, operator, software, and other systems and that they provide sufficient detail to enable the development of software components that implement the interfaces. 5.5: Ensure that complex algorithms have been correctly derived, provide the needed behavior under

	off nominal conditions and assumed conditions, and that the derivation approach is known and understood to support future maintenance. 5.6: Partial: Ensure that the design provides the dependability and fault tolerance required by the system and that the design is capable of controlling identified hazards and does not create hazardous conditions.
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev 0
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	SC FSW - Build 3.0 DEU FSW - Build 1.1
Inputs (includes Technical Reference):	Applicable Technical Reference (i.e. use case scenarios)
Prerequisites:	FSW is not developed using behavior models (uml activity, state, or similar), the models are not of high enough fidelity to analyze code behavior, or the models are not sufficient to provide system level understanding. Note: The process of generating the diagrams can be used to gain system level understanding even when the diagrams duplicate developer products.
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	JWST has been under IV&V for many years, so the analysis methods have evolved over time and the rigor has had a tendency to grow as capability is added. The following is a breakdown of the analysis methods and assurance levels by phase consistent with the current methods in Compass and consistent with the current and past practices in IV&V analyses. The assurance increases by adding additional layers of assurance, and in some cases, the assurance in one level covers lower levels. 1. M-41: Verify Software Interface Design by Inspection Against Interface Requirements a. Assurance Objective: Necessary Interfaces are designed. b. TF: 5.1 (P), 5.4 (F) 2. (New) M-39: Verify Software Design by Inspecting Traces to Requirements and Software Architecture

	a. Assurance Objective: Each Requirement is captured in the design and the design is of high quality (complete, correct, unambiguous, etc) b. TF: 5.1 (F), 5.2 (F), 5.3 (P), 5.4 (P), 5.5 (P), 5.6 (F) c. Alternate Method for MBD i. M-8: Verify Autogenerated Software Implementation by Inspecting Traces from Requirements to Rational Rose Design Model ii. TF: 5.1 (F) 3. M-103: Verify and Validate Requirement Implementation using Flow Diagrams to Uncover Missing, Conflicting, or Unnecessary Behavior a. Assurance Objective: Requirement (and Design Element) Set is implemented correctly and completely. b. Note: Method encompasses Assurance Level 2 mentioned above c. TF: 5.1 (F), 5.2 (F), 5.3 (F), 5.4 (F), 5.5 (F), 5.6 (P) 4. M-42: Validate algorithm design through algorithm qualitative attribute assessment a. Assurance Objective: Complex algorithm is complete and correct. b. Note: SME to assess algorithms for correctness and completeness c. TF: 5.5 (P)
Concerns:	None
Method Application Notes: This method applies to multiple phases of development; however, the mapping here is to Design phase artifacts thus the Design TF Goals are mapped.	
Required Tools:	Drawing Package (i.e. Visio, Together, etc) Code Viewing and analysis tool (i.e. Understand)
Empirical Evidence:	1. Control Flow diagrams and corresponding notes capture IV&V analysis, questions, and concerns. - "Analysis/Findings Report" 2. Diagrams show/trace the behaviors and their source - "Coverage Reports." 2) Test/Analysis Scenarios (nominal and off-nominal) derived from flow paths - "Scenario Report" Example Assurance Claims/Statements: The FSW does not contain any unnecessary functions or "features" The FSW implements all required behaviors with appropriate response to off-nominal conditions

	The FSW verification suite completely covers the required behaviors as well as the nominal, off-nominal, and non-required (i.e. feature) paths in the FSW
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	None

Activity 8: Verify Software Design via Inspection of Interface Requirements

Method:	M-41, Version 1.1 (Current Status: Approved)
Method Title:	Verify Software Interface Design by Inspection Against Interface Requirements
Method Synopsis	Method supports manual evaluation of the integrity of the software requirements to interface design transformation, and detects defects in hardware/user/operator/software/other systems interface coverage completeness/correctness/accuracy and capability for implementation in software.
Subsystem/Entity	GND.FOS; GND.WSS.MMS; GND.OSS; OBS.SCE.FM; OBS.SCE.CTP.SC; OBS.SCE.DEU
Required Method Revisions (if any)	None
Technical Goal:	5.1 Ensure that all (in-scope) requirements (e.g. SRS and IRS) are represented in the appropriate elements of the design (e.g. SDD and IDD) and that the design does not introduce capability that is not required. 5.4 Provide Evidence that the assurance goals related to the internal and external software interface designs are adequately achieved for all interfaces with hardware, user, operator, software, and other systems and that they provide sufficient detail to enable the development of software components that implement the interfaces.
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev O
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	FOS - Build 6 WSS-MMS - Build 5 (Change Impact Analysis (CIA) Only) FM Branch Terminal Analysis – Build 2.2 SC FSW - Build 3.0 DEU FSW - Build 1.1
Inputs (includes Technical Reference):	List of validated interface requirements and identified issues and risks, Interface Control Document (ICD), Interface Requirements Document (IRD), Intended assurance goals/statements, Identified evidence needed to support intended assurance goals/statements, Technical Reference (applicable to

	interface), Adverse conditions, System Capabilities list and description.
Prerequisites:	Validation of the interface requirements
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	JWST has been under IV&V for many years, so the analysis methods have evolved over time and the rigor has had a tendency to grow as capability is added. The following is a breakdown of the analysis methods and assurance levels by phase consistent with the current methods in Compass and consistent with the current and past practices in IV&V analyses. The assurance increases by adding additional layers of assurance, and in some cases, the assurance in one level covers lower levels. 1. M-41: Verify Software Interface Design by Inspection Against Interface Requirements a. Assurance Objective: Necessary Interfaces are designed. b. TF: 5.1 (P), 5.4 (F) 2. (New) M-39: Verify Software Design by Inspecting Traces to Requirements and Software Architecture a. Assurance Objective: Each Requirement is captured in the design and the design is of high quality (complete, correct, unambiguous, etc) b. TF: 5.1 (F), 5.2 (F), 5.3 (P), 5.4 (P), 5.5 (P), 5.6 (F) c. Alternate Method for MBD i. M-8: Verify Autogenerated Software Implementation by Inspecting Traces from Requirements to Rational Rose Design Model ii. TF: 5.1 (F) 3. M-103: Verify and Validate Requirement Implementation using Flow Diagrams to Uncover Missing, Conflicting, or Unnecessary Behavior a. Assurance Objective: Requirement (and Design Element) Set is implemented correctly and completely. b. Note: Method encompasses Assurance Level 2 mentioned above c. TF: 5.1 (F), 5.2 (F), 5.3 (F), 5.4 (F), 5.5 (F), 5.6 (P) 4. M-42: Validate algorithm design through algorithm qualitative attribute assessment

	a. Assurance Objective: Complex algorithm is complete and correct. b. Note: SME to assess algorithms for correctness and completeness c. TF: 5.5 (P)
Concerns:	None
Method Application Notes: None	
Required Tools:	Excel Worksheets (or other data documentation system), ORBIT
Empirical Evidence:	• Engineering worksheets, databases, etc. that contain the results and comments of the requirements to design trace and the design to requirements trace, used to support the intended assurance goals/statements. • TIMs • Documented risks and findings
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	None

Activity 9: Verify Software Design via Inspecting Traces to Requirements and Architecture

Method:	M-39, Version 1.3 (Current Status: Approved)
Method Title:	Verify Software Design by Inspecting Traces to Requirements and Software Architecture
Method Synopsis	Method supports manual evaluation of the integrity of the software design to ensure that all requirements are represented in the appropriate elements of the design and that the design does not introduce capability that is not required, and to identify defects in its satisfaction of the software architecture and validated software requirements. Software design documentation is also evaluated to ensure that the design provides the required capability (meeting software architecture and software requirements), is able to reliably meet user needs, and is sufficiently stable to proceed with implementation, and to identify defects in consistency, ambiguity, correctness, completeness, and testability.
Subsystem/Entity	GND.FOS; GND.WSS.MMS; OBS.SCE.FM; OBS.SCE.DEU
Required Method Revisions (if any)	None
Technical Goal:	5.1 Ensure that all requirements (e.g. SRS and IRS) are represented in the appropriate elements of the design (e.g. SDD and IDD) and that the design does not introduce capability that is not required. 5.2 Ensure that the design provides the required capability (meeting software architecture and software requirements), is able to reliably meet user needs, and is sufficiently stable to proceed with implementation. 5.3 Partial: Ensure that the proposed software architecture satisfies the needs of the system, and that it is a feasible solution (i.e. will successfully satisfy the needs of the system, while still being practical). 5.4 Partial: Ensure that the internal and external software interface designs are provided for all (in-scope) interfaces with hardware, user, operator, software, and other systems and that they provide sufficient detail to enable the development of software components that implement the interfaces. 5.5: Partial: Ensure that complex algorithms have been correctly derived, provide the needed behavior under off nominal conditions and assumed conditions, and that the derivation approach is known and understood to support future maintenance.

	5.6: Ensure that the design provides the dependability and fault tolerance required by the system and that the design is capable of controlling identified hazards and does not create hazardous conditions.
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev O
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	FOS - Build 6 WSS-MMS - Build 5 (Change Impact Analysis (CIA) Only) FM Branch Terminal Analysis – Build 2.2 DEU FSW - Build 1.1
Inputs (includes Technical Reference):	1. Validated software requirements and identified issues and risks. 2. Software Interface Control Documents (ICDs) 3. System Preventative/Responsive Behaviors from project's Technical Reference 4. Adverse Conditions from project's Technical Reference 5. Project-specific evaluation criteria from project's Technical Reference (if applicable) 6. Technical Reference resultant from the 3.5 requirements validation for dependability /fault tolerance 7. Hazard Analysis Artifacts a. Failure Modes Effects Analysis b. Preliminary Hazards Analysis
Prerequisites:	Validation of system and software requirements not including integration requirements
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	JWST has been under IV&V for many years, so the analysis methods have evolved over time and the rigor has had a tendency to grow as capability is added. The following is a breakdown of the analysis methods and assurance levels by phase consistent with the current methods in Compass and consistent with the current and past practices in IV&V analyses. The assurance increases by adding additional layers of assurance, and

	in some cases, the assurance in one level covers lower levels. 1. M-41: Verify Software Interface Design by Inspection Against Interface Requirements a. Assurance Objective: Necessary Interfaces are designed. b. TF: 5.1 (P), 5.4 (F) 2. (New) M-39: Verify Software Design by Inspecting Traces to Requirements and Software Architecture a. Assurance Objective: Each Requirement is captured in the design and the design is of high quality (complete, correct, unambiguous, etc) b. TF: 5.1 (F), 5.2 (F), 5.3 (P), 5.4 (P), 5.5 (P), 5.6 (F) c. Alternate Method for MBD i. M-8: Verify Autogenerated Software Implementation by Inspecting Traces from Requirements to Rational Rose Design Model ii. TF: 5.1 (F) 3. M-103: Verify and Validate Requirement Implementation using Flow Diagrams to Uncover Missing, Conflicting, or Unnecessary Behavior a. Assurance Objective: Requirement (and Design Element) Set is implemented correctly and completely. b. Note: Method encompasses Assurance Level 2 mentioned above c. TF: 5.1 (F), 5.2 (F), 5.3 (F), 5.4 (F), 5.5 (F), 5.6 (P) 4. M-42: Validate algorithm design through algorithm qualitative attribute assessment a. Assurance Objective: Complex algorithm is complete and correct. b. Note: SME to assess algorithms for correctness and completeness c. TF: 5.5 (P)
Concerns:	None
Method Application Notes: None	
Required Tools:	ORBIT, Excel (engineering worksheets)
Empirical Evidence:	• Engineering worksheets documenting results. The worksheets should include: - the requirements (document, section title, number, description) - traces to design artifacts and identified behaviors (including specific Adverse Conditions considered

	during the analysis), software dependability and identified hazards. - assessment of the software architecture, software design, and software algorithms with respect to the requirement sets and identified behaviors (including specific Adverse Conditions considered during the analysis) - assessment of the software design with respect to each individual requirement (analyzed across documentation) - assessment of the software design with respect to each identified hazard control (analyzed across documentation) - additional analyst comments as needed to support assessment.
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	None

Activity 10: Verify Software Design via Inspecting Traces from Requirements to Rational Rose Design Model

Method:	M-8, Version 1.0 (Current Status: Approved)
Method Title:	Verify Autogenerated Software Implementation by Inspecting Traces from Requirements to Rational Rose Design Model
Method Synopsis	Method of using Rational Rose models to verify implementation of requirements and ICDs. Method traces requirements to design model and code implementation. Faults detected via this Method include defects in implementation of required behaviors and required interfaces, and applicable design and coding standards violations.
Subsystem/Entity	OBS.SCE.CTP.SC
Required Method Revisions (if any)	None
Technical Goal:	Ensure that all validated Level 5 requirements for the build have been incorporated into the build design correctly, and that they implement all desired behaviors. Review the model to confirm that no undesired or undocumented behaviors have been implemented. Verify that the internal/external software interfaces, as specified by the requirements and relevant ICDs, have been properly implemented. Evaluate the model and source code to confirm compliance with applicable design and coding standards. 5.1 Ensure that all requirements (e.g. SRS and IRS) are represented in the appropriate elements of the design (e.g. SDD and IDD) and that the design does not introduce capability that is not required.
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev O
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	SC FSW - Build 3.0
Inputs (includes Technical Reference):	Current build Rational model; RequisitePro DB or VDD; FSW Requirements Spec; Design & Coding Standards; Applicable Interface Requirements
Prerequisites:	Corresponding FSW requirements have been validated.
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.

Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	JWST has been under IV&V for many years, so the analysis methods have evolved over time and the rigor has had a tendency to grow as capability is added. The following is a breakdown of the analysis methods and assurance levels by phase consistent with the current methods in Compass and consistent with the current and past practices in IV&V analyses. The assurance increases by adding additional layers of assurance, and in some cases, the assurance in one level covers lower levels. 1. M-41: Verify Software Interface Design by Inspection Against Interface Requirements a. Assurance Objective: Necessary Interfaces are designed. b. TF: 5.1 (P), 5.4 (F) 2. (New) M-39: Verify Software Design by Inspecting Traces to Requirements and Software Architecture a. Assurance Objective: Each Requirement is captured in the design and the design is of high quality (complete, correct, unambiguous, etc) b. TF: 5.1 (F), 5.2 (F), 5.3 (P), 5.4 (P), 5.5 (P), 5.6 (F) c. Alternate Method for MBD i. M-8: Verify Autogenerated Software Implementation by Inspecting Traces from Requirements to Rational Rose Design Model ii. TF: 5.1 (F) 3. M-103: Verify and Validate Requirement Implementation using Flow Diagrams to Uncover Missing, Conflicting, or Unnecessary Behavior a. Assurance Objective: Requirement (and Design Element) Set is implemented correctly and completely. b. Note: Method encompasses Assurance Level 2 mentioned above c. TF: 5.1 (F), 5.2 (F), 5.3 (F), 5.4 (F), 5.5 (F), 5.6 (P) 4. M-42: Validate algorithm design through algorithm qualitative attribute assessment a. Assurance Objective: Complex algorithm is complete and correct. b. Note: SME to assess algorithms for correctness and completeness c. TF: 5.5 (P)
Concerns:	None

Method Application Notes: This method applies to both Design and Implementation Analysis, however, the intent is to perform Implementation analysis with other means and only the Design TF Goals are mapped.	
Required Tools:	Rational Toolset, Klocwork
Empirical Evidence:	Engineering spreadsheet containing the following data: • Requirements in scope • Link to specific design model attribute implementing the requirement (such as class, transition, or operation) • Portion of code in question, if applicable, showing correct implementation of requirement • Analyst comments • Identification of any undocumented functionality or behavior Analysis spreadsheet containing potential errors generated by Klocwork tool.
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	Note: Since a Rational build includes requirements, design and code, the process of Design and Implementation analysis are coupled and performed simultaneously.

Activity 11: Verify Software Code Quality via Static Analysis Tools

Method:	M-9, Version 1.3 (Current Status: Approved)
Method Title:	Verify Software Code Quality using Static Analysis Tools
Method Synopsis	This method applies one or more static code analysis tools to ensure the source code is free of impacting code defects, syntax errors and other code deficiencies, including (but not restricted to): - buffer overflows - security vulnerabilities - null pointer dereferences - infinite loops - unused code - coding standard violations The static code analysis tool(s), configured to exclude undesired warning types, generate candidate issues from a build of code, which are then to be manually reviewed to determine the final set of reportable flaws in the build.
Subsystem/Entity	GND.FOS; GND.WSS.WEx; GND.OSS; OBS.SCE.CTP.SC; OBS.SCE.DEU; OBS.ISIM.ISIM; OBS.ISIM.FGS
Required Method Revisions (if any)	None
Technical Goal:	The goal is to ensure the source code is free of syntax and other code errors, including buffer overflows, use of uninitialized variables, multiple definitions of functions or constants, and unused code. 6.2 (Partial) Ensure that the source code components can reliably perform required capabilities under nominal and off-nominal conditions, perform no undesired functions, and that the documentation (both embedded and stand-alone) can facilitate code maintenance at a later time.
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev O
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	FOS - Build 5 WSS-WEx - Build 5 OSS - Build 5 OSS - Build 6 SC FSW - Build 3.0 DEU FSW - Build 1.1

	ISIM IC15.0 (Change Impact Analysis (CIA) Only)
Inputs (includes Technical Reference):	Optional: Project-specific coding standards. Required: Code Quality Characteristics (see process asset on ECM: Enterprise=>IV&V OFFICE=>TQ&E=>Knowledge Sharing=>IV&V Catalog of Methods=>ProcessAssetLibrary=>Code=>Prioritizing_Static_Tool_Findings.pptx)
Prerequisites:	none identified
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	JWST has been under IV&V for many years, so the analysis methods have evolved over time and the rigor has had a tendency to grow as capability is added. The following is a breakdown of the analysis methods and assurance levels by phase consistent with the current methods in Compass and consistent with the current and past practices in IV&V analyses. The assurance increases by adding additional layers of assurance, and in some cases, the assurance in one level covers lower levels. Code 1. M-9: Verify Software Code Quality using Static Analysis Tools a. Assurance Objective: Code is of high quality and free of syntactic defects and standard violations b. TF: 6.2 (P) 2. (New) M-4: Verify Implementation of Requirements or Design in Source Code or Scripts through Manual Inspection a. Assurance Objective: Every Requirement (and Design Element) is implemented correctly and completely b. TF: 6.1 (F), 6.2 (P), 6.6 (F) 3. M-103: Verify and Validate Requirement Implementation using Flow Diagrams to Uncover Missing, Conflicting, or Unnecessary Behavior a. Assurance Objective: Requirement (and Design Element) Set is implemented correctly and completely. b. Note: Method encompasses Assurance Level 2 mentioned above c. TF: 6.1 (F), 6.2 (F), 6.3 (F), 6.4 (F), 6.5 (F), 6.6 (F)
Concerns:	None
Method Application Notes: None	
Required Tools:	Static code analysis tool(s) appropriate for the implementation language

Empirical Evidence:	- Tool capabilities - tool settings - filtering methods/algorithms - analysis of filtered results.
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	The SWAT Confluence Portal has in-depth detail on usage of each static analysis tool: http://confluence.ivv.nasa.gov:8090/display/SWAT/Available+Tools

Activity 12: Verify and Validate Software Implementation via Flow Diagrams

Method:	M-103, Version 1.0 (Current Status: Approved)
Method Title:	Verify and Validate Requirement Implementation using Flow Diagrams to Uncover Missing, Conflicting, or Unnecessary Behavior
Method Synopsis	Method uses Flow Diagrams to analyze software implementation of requirements to ensure the correct and complete implementation of requirements on a system level as well as an atomic level. (Level is dependent upon the abstraction in the modeling chosen by the analyst as well as the available level of artifacts being targeted). Further, the method is applied to the source code that is not specified by requirements or not specified directly.
Subsystem/Entity	OBS.SCE.CTP.SC; OBS.SCE.DEU
Required Method Revisions (if any)	None
Technical Goal:	1. Gain system/software level understanding 2. Uncover ambiguous or missing behaviors 3. Uncover conflicting or undesired behaviors 4. Uncover failure scenarios (identify whether the FSW is protected against off-nominal/adverse conditions/inputs) 6.1 Ensure that all (in-scope) elements of the design (e.g. SDD and IDD) are represented in the appropriate source code components and that the source code does not introduce capability that is not required. 6.2 Ensure that the source code components can reliably perform required capabilities under nominal and off-nominal conditions, perform no undesired behaviors, and that the documentation (both embedded and stand-alone) can facilitate code maintenance. 6.3 Ensure that the source code that interfaces with hardware, user, operator, software, and other systems reliably provides the right services and data and receives data for internal use. 6.4 Ensure that test results are as expected (per the corresponding plans, cases, procedures, design) and the impacts of any discrepancies are understood. 6.5 Ensure that the source code components provide the dependability and fault tolerance required by the system and that the source code is capable of

	controlling identified hazards and does not create hazardous conditions. 6.6 Ensure that all (in-scope) requirements (e.g. SRS and IRS) are represented in the appropriate source code components and that the source code does not introduce capability that is not required.
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev 0
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	SC FSW - Build 3.0 DEU FSW - Build 1.1
Inputs (includes Technical Reference):	Applicable Technical Reference (i.e. use case scenarios)
Prerequisites:	FSW is not developed using behavior models (uml activity, state, or similar), the models are not of high enough fidelity to analyze code behavior, or the models are not sufficient to provide system level understanding. Note: The process of generating the diagrams can be used to gain system level understanding even when the diagrams duplicate developer products.
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	JWST has been under IV&V for many years, so the analysis methods have evolved over time and the rigor has had a tendency to grow as capability is added. The following is a breakdown of the analysis methods and assurance levels by phase consistent with the current methods in Compass and consistent with the current and past practices in IV&V analyses. The assurance increases by adding additional layers of assurance, and in some cases, the assurance in one level covers lower levels. Code 1. M-9: Verify Software Code Quality using Static Analysis Tools a. Assurance Objective: Code is of high quality and free of syntactic defects and standard violations b. TF: 6.2 (P) 2. (New) M-4: Verify Implementation of Requirements or Design in Source Code or Scripts through Manual Inspection

	a. Assurance Objective: Every Requirement (and Design Element) is implemented correctly and completely b. TF: 6.1 (F), 6.2 (P), 6.6 (F) 3. M-103: Verify and Validate Requirement Implementation using Flow Diagrams to Uncover Missing, Conflicting, or Unnecessary Behavior a. Assurance Objective: Requirement (and Design Element) Set is implemented correctly and completely. b. Note: Method encompasses Assurance Level 2 mentioned above c. TF: 6.1 (F), 6.2 (F), 6.3 (F), 6.4 (F), 6.5 (F), 6.6 (F)
Concerns:	None
Method Application Notes: This method applies to multiple phases of development; however, the mapping here is to Implementation phase artifacts thus the Implementation TF Goals are mapped.	
Required Tools:	Drawing Package (i.e. Visio, Together, etc) Code Viewing and analysis tool (i.e. Understand)
Empirical Evidence:	1. Control Flow diagrams and corresponding notes capture IV&V analysis, questions, and concerns. - "Analysis/Findings Report" 2. Diagrams show/trace the behaviors and their source - "Coverage Reports." 2) Test/Analysis Scenarios (nominal and off-nominal) derived from flow paths - "Scenario Report" Example Assurance Claims/Statements: The FSW does not contain any unnecessary functions or "features" The FSW implements all required behaviors with appropriate response to off-nominal conditions The FSW verification suite completely covers the required behaviors as well as the nominal, off-nominal, and non-required (i.e. feature) paths in the FSW
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	None

Activity 13: Verify Software Implementation via Traces to Requirements

Method:	M-38, Version 1.2 (Current Status: Deprecated)
Method Title:	Verify Software Implementation by Inspecting Traces to Requirements (Nominal, Off-Nominal and Hazards Scenarios)
Method Synopsis	Method performs a manual inspection of requirements traces to software artifacts to confirm correctness, consistency, completeness, accuracy, and verifiability of the implementation. Defects detected include requirements not implemented (or incorrectly/insufficiently implemented), missing requirements for implemented code/script components, improper/inadequate flow of data and control, improper sequences of states and state changes, improper data usage/format (particularly violations of units of measure consistency requirements), and inappropriate coding methods or standards applied. Source code documentation is also evaluated for compliance to configuration management procedures.
Subsystem/Entity	OBS.ISIM.ISIM; OBS.ISIM.FGS
Required Method Revisions (if any)	None
Technical Goal:	6.2 (Full) Ensure that the source code/scripting components can reliably perform required capabilities under nominal and off-nominal conditions, perform no undesired behaviors, and that the documentation (both embedded and stand-alone) can facilitate code/script maintenance 6.4(partial-low-level testing) Ensure that test results are as expected (per the corresponding plans, cases, procedures, design) and the impacts of any discrepancies are understood. 6.5 (Full) Ensure that the source code/script components provide the dependability and fault tolerance required by the system and that the source code/script is capable of controlling identified hazards and does not create hazardous conditions
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev O
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.

Target Artifacts:	1. ISIM Software for Build IC 14.2 2. ISIM FGS Software for Builds 4.97, 5.0, and 5.0+
Inputs (includes Technical Reference):	Software Requirements Information from Technical Framework 2.5 (Ensure that known software based hazard causes, contributors, and controls are identified and documented) Hazard documentation such as: FMEA, Fault Tree Analysis, etc… SW Characterization System Assurance Statements System Preventative Behaviors/Adverse Conditions/Responsive Actions Unit Level Test
Prerequisites:	Completed Technical Framework 2.5 (Ensure that known software based hazard causes, contributors, and controls are identified and documented)
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	JWST has been under IV&V for many years, so the analysis methods have evolved over time and the rigor has had a tendency to grow as capability is added. The following is a breakdown of the analysis methods and assurance levels by phase consistent with the current methods in Compass and consistent with the current and past practices in IV&V analyses. The assurance increases by adding additional layers of assurance, and in some cases, the assurance in one level covers lower levels. Code 1. M-9: Verify Software Code Quality using Static Analysis Tools a. Assurance Objective: Code is of high quality and free of syntactic defects and standard violations b. TF: 6.2 (P) 2. (New) M-4: Verify Implementation of Requirements or Design in Source Code or Scripts

	through Manual Inspection a. Assurance Objective: Every Requirement (and Design Element) is implemented correctly and completely b. TF: 6.1 (F), 6.2 (P), 6.6 (F) 3. M-103: Verify and Validate Requirement Implementation using Flow Diagrams to Uncover Missing, Conflicting, or Unnecessary Behavior a. Assurance Objective: Requirement (and Design Element) Set is implemented correctly and completely. b. Note: Method encompasses Assurance Level 2 mentioned above c. TF: 6.1 (F), 6.2 (F), 6.3 (F), 6.4 (F), 6.5 (F), 6.6 (F)
Concerns:	None
Method Application Notes: This method applies to multiple phases of development; however, the mapping here is to Implementation phase artifacts thus the Implementation TF Goals are mapped.	
Required Tools:	Code parser desired, but not required
Empirical Evidence:	• Engineering worksheets, database, etc. documenting results of tracing requirements to code/script elements and comments confirming the correct implementation of the requirements • Should include requirements traced, module or function traced to, the file, code/script element, and line number(s),and assessment of trace (e.g., does the implementation fulfill the requirements) • Off-nominal conditions will be assessed when verifying the requirement implementation–trace/association either through requirements or directly tracing to the off-nominal conditions • Observations and Issues documenting any discrepancies • Risks or findings in technical reports documenting systemic concerns
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	None

Activity 14: Verify Software Implementation via Manual Inspection

Method:	M-4, Version 2.0 (Current Status: Approved)
Method Title:	Verify Implementation of Requirements or Design in Source Code or Scripts through Manual Inspection
Method Synopsis	This method uses manual bi-directional tracing of in-scope requirements and design to developer-supplied software artifacts to detect defects in requirements, design, and/or code/script artifacts.
Subsystem/Entity	GND.WSS.MMS; GND.OSS; OBS.SCE.CTP.SC; OBS.SCE.DEU; OBS.ISIM.ISIM; OBS.ISIM.FGS
Required Method Revisions (if any)	None
Technical Goal:	6.1: Ensure that all (in-scope) elements of the design (e.g. SDD and IDD) are represented in the appropriate source code components and that the source code does not introduce capability that is not required. Note: Specific assumptions particular to the target design artifact must be considered beforehand as not everything might be directly traceable to code. 6.2: Ensure that the source code components can reliably perform required capabilities under nominal and off-nominal conditions, perform no undesired behaviors, and that the documentation (both embedded and stand-alone) can facilitate code maintenance. 6.6: Ensure that all (in-scope) requirements (e.g. SRS and IRS) are represented in the appropriate source code components and that the source code does not introduce capability that is not required. -Correct - Verify that the implementation accurately represents the requirements and design, complies with applicable coding standards, and that the outputs and behaviors are as expected. -Consistent - Verify that the implementation is consistent with respect to relative requirements and design and consistent within itself. Consistency may include use of terminology, parameter names, type casting, etc. -Complete - Verify that the implementation covers all elements, functions, features, behaviors, and/or

	constraints as specified in the requirements and design. Verify that all in-scope software requirements are traceable to the source code or script components responsible for the implementation; and verify that any functionality/behavior of significant importance is traceable back to the requirements level.
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev O
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	WSS-MMS - Build 5 (Change Impact Analysis (CIA) Only) OSS - Build 5 OSS - Build 6 SC FSW - Build 3.0 DEU FSW - Build 1.1 ISIM IC14.21+ (CIA Only) ISIM IC15.0 (CIA Only) ISIM IC15.0+ (CIA Only) FGS - Build 4.97 (CIA Only) FGS - Build 5.0 FGS - Build 5.0+ (CIA Only)
Inputs (includes Technical Reference):	In-scope requirements or design elements to be implemented in the source code or script (where appropriate)
Prerequisites:	Acquisition of project delivered artifacts such as requirements, design specifications, and other related documentation (e.g. the relevant portion of the IV&V Technical Reference that will support understanding of the target artifacts). Internal activity to determine the set of in-scope candidates for analysis.
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	JWST has been under IV&V for many years, so the analysis methods have evolved over time and the rigor has had a tendency to grow as capability is added. The following is a breakdown of the analysis methods and assurance levels by phase consistent with the current methods in Compass and consistent with the current and past practices in IV&V analyses. The assurance increases by adding additional layers of assurance, and

	in some cases, the assurance in one level covers lower levels. Code 1. M-9: Verify Software Code Quality using Static Analysis Tools a. Assurance Objective: Code is of high quality and free of syntactic defects and standard violations b. TF: 6.2 (P) 2. (New) M-4: Verify Implementation of Requirements or Design in Source Code or Scripts through Manual Inspection a. Assurance Objective: Every Requirement (and Design Element) is implemented correctly and completely b. TF: 6.1 (F), 6.2 (P), 6.6 (F) 3. M-103: Verify and Validate Requirement Implementation using Flow Diagrams to Uncover Missing, Conflicting, or Unnecessary Behavior a. Assurance Objective: Requirement (and Design Element) Set is implemented correctly and completely. b. Note: Method encompasses Assurance Level 2 mentioned above c. TF: 6.1 (F), 6.2 (F), 6.3 (F), 6.4 (F), 6.5 (F), 6.6 (F)
Concerns:	None
Method Application Notes: None	
Required Tools:	Tools applicable for viewing the requirement and design documentation acquired from the project (including any COTS tools that the developer might use, such as Matlab/Simulink, Rational, etc). Code browsing tools providing the ability to trace those artifacts into the code/script. Tools for capturing analysis assessments (e.g. Excel, SMART, AWB, etc) Issue tracking system such as ORBIT, JIRA, Confluence, etc.
Empirical Evidence:	Engineering worksheets or database assessments documenting the results of the Requirements/Design Implementation analysis should include evidence such as: code or script file name(s), line number(s), requirements or design elements traced, description or explanation of code flow, any applicable documents or TIMs referenced, analyst comments, IV&V executed test results and logs.

	Based on analysis results, the Technical Reference for the project may require updates.
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	None

Activity 15: Verify Software Implementation via Inspection Against Interface Design

Method:	M-13, Version 1.1 (Current Status: Approved)
Method Title:	Verify SW Interface Implementation by Inspection Against Interface Design
Method Synopsis	Method supports manual evaluation of interface design to implementation integrity to identify defects in interface data structure and element definitions, design robustness and coupling, implementation of interface pre/post conditions and invariants, and capability to provide required services (data) across interfaces.
Subsystem/Entity	OBS.SCE.DEU
Required Method Revisions (if any)	None
Technical Goal:	6.1 (Partial) Ensure that all elements of the interface design (design descriptions, structures, and data) as provided in the SDDs, ICDs, IDD, and PDR/CDR (interface details) are represented in the appropriate source code components and that the source code does not introduce capability that is not required. 6.3 (Full) Ensure that the source code that interfaces with hardware, user, operator, software, and other systems reliably provides the right services and data and receives data for internal use.
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev O
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	DEU FSW Build 1.1 and 1.2
Inputs (includes Technical Reference):	Verified Interface Design
Prerequisites:	N/A
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	JWST has been under IV&V for many years, so the analysis methods have evolved over time and the rigor has had a tendency to grow as capability is added. The following is a breakdown of the analysis methods and assurance levels by phase consistent with the current methods in Compass and consistent with the current and past practices in IV&V analyses. The assurance

	increases by adding additional layers of assurance, and in some cases, the assurance in one level covers lower levels. Code 1. M-9: Verify Software Code Quality using Static Analysis Tools a. Assurance Objective: Code is of high quality and free of syntactic defects and standard violations b. TF: 6.2 (P) 2. (New) M-4: Verify Implementation of Requirements or Design in Source Code or Script through Manual Inspection a. Assurance Objective: Every Requirement (and Design Element) is implemented correctly and completely b. TF: 6.1 (F), 6.2 (P), 6.6 (F) 3. M-103: Verify and Validate Requirement Implementation using Flow Diagrams to Uncover Missing, Conflicting, or Unnecessary Behavior a. Assurance Objective: Requirement (and Design Element) Set is implemented correctly and completely. b. Note: Method encompasses Assurance Level 2 mentioned above c. TF: 6.1 (F), 6.2 (F), 6.3 (F), 6.4 (F), 6.5 (F), 6.6 (F)
Concerns:	None
Method Application Notes: This method applies to multiple phases of development; however, the mapping here is to Implementation phase artifacts thus the Implementation TF Goals are mapped.	
Required Tools:	Excel Worksheets (or other data documentation system), ORBIT
Empirical Evidence:	• Engineering worksheets, databases, etc. documenting results of tracing I/F requirements design to code and comments confirming the correct implementation of the I/F requirements o Should include I/F requirements traced, module or function traced to, and an assessment of the trace

	(e.g., does the implementation fulfill the requirements) • Off-nominal conditions will be assessed when verifying the I/F requirement implementation – trace/association either through requirements or directly tracing to the off-nominal conditions • Data to support assessment of the implementation against the verified design • Observations and Issues documenting any discrepancies • Risks or findings in technical reports documenting systemic concerns • Evidence that the interfaces are properly implemented per the ICDs as well as any lower level interface requirements (L5 Software I/O requirements)
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	N/A

Activity 16: Verify Critical Software Changes via Change Request Inspection

Method:	M-20, Version 1.0 (Current Status: Approved)
Method Title:	Verify Critical Software Changes By Inspecting Change Requests
Method Synopsis	Method uses manual analysis to evaluate a process for evaluating proposed software changes to assess the technical merit, potential side effects, and the overall impact on other project configuration items.
Subsystem/Entity	GND.WSS.MMS
Required Method Revisions (if any)	None
Technical Goal:	Verify that the intended changes have been correctly implemented and that no unintended changes occurred. This includes software changes, if applicable.
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev O
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	WSS MMS Build 5
Inputs (includes Technical Reference):	Software Change Records
Prerequisites:	Pre-verified system
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	JWST has been under IV&V for many years, so the analysis methods have evolved over time and the rigor has had a tendency to grow as capability is added. The following is a breakdown of the analysis methods and assurance levels by phase consistent with the current methods in Compass and consistent with the current and past practices in IV&V analyses. The assurance increases by adding additional layers of assurance, and in some cases, the assurance in one level covers lower levels. Code 1. M-9: Verify Software Code Quality using Static Analysis Tools a. Assurance Objective: Code is of high quality and free of syntactic defects and standard

	violations b. TF: 6.2 (P) 2. (New) M-4: Verify Implementation of Requirements or Design in Source Code or Scripts through Manual Inspection a. Assurance Objective: Every Requirement (and Design Element) is implemented correctly and completely b. TF: 6.1 (F), 6.2 (P), 6.6 (F) 3. M-103: Verify and Validate Requirement Implementation using Flow Diagrams to Uncover Missing, Conflicting, or Unnecessary Behavior a. Assurance Objective: Requirement (and Design Element) Set is implemented correctly and completely. b. Note: Method encompasses Assurance Level 2 mentioned above c. TF: 6.1 (F), 6.2 (F), 6.3 (F), 6.4 (F), 6.5 (F), 6.6 (F)
Concerns:	None
Method Application Notes: This method applies to multiple phases of development; however, the mapping here is to Test phase artifacts thus the Test TF Goals are mapped.	
Required Tools:	None
Empirical Evidence:	Engineering worksheets (or database) documenting the results of any issues and/or risks for each requirement and conclusions about the completeness and correctness of the set(s) of analyzed requirements
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	None

Activity 17: Validate Test Design via Test Plan Inspection

Method:	M-10, Version 1.2 (Current Status: Approved)
Method Title:	Validate Test Plan by Inspection
Method Synopsis	Method performs manual inspection of a test plan artifact to detect defects in test plan integrity, compliance with applicable test plan standards and key testing principles, capability for requirements and behaviors to be verified under nominal and adverse conditions, documentation of limitations in test plan verification capability, test environment fidelity appropriateness, test operational procedure integrity, test scheduling and risk assessments, and planned regression testing.
Subsystem/Entity	GND.FOS; GND.WSS.WEx; GND.WSS.MMS; GND.OSS; OBS.SCE.SCS; OBS.SCE.EMTB; OBS.SCE.CTP.SC; OBS.SCE.DEU; OBS.ISIM.ISIM; OBS.ISIM.NIRSpec.ICE; OBS.ISIM.NIRSpec.MS; OBS.ISIM.NIRSpec.DS; OBS.ISIM.FGS
Required Method Revisions (if any)	None
Technical Goal:	4.1 Ensure that the planned tests are sufficient to: 4.1.1 Ensure that the software correctly implements system and software requirements in an operational environment under nominal and off-nominal conditions. 4.1.2 Ensure that the complete, integrated system complies with its specified system requirements allocated to software and to validate whether the system meets its original objectives. 4.1.3 Ensure that the software meets all of the software requirements and is ready to be integrated with system hardware. 4.1.4 Ensure that the software correctly implements the software requirements and design as each software component (e.g., units or modules) is incrementally integrated with each other. 4.1.5 Ensure that the software components (e.g., units, source code modules) correctly implement software component requirements. 4.3 Ensure that the planned regression testing to be performed when changes are made to any previously examined software products is sufficient to identify any unintended side effects or impacts of the change on other aspects of the system

	4.8 Ensure that the test environment is sufficiently complete, correct, and accurate to perform the intended testing.
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev O
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	FOS - Build 5 WSS-WEx - Build 5 WSS-MMS - Build 5 (Change Impact Analysis (CIA) Only) OSS - Build 5 OSS - Build 6 Stored Command Sequences (SCS's) – B2.2 EMTB Build 2.2 Suitability EMTB Build 2.2 SC FSW - Build 3.0 DEU FSW - Build 1.1 ISIM IC14.21+ (CIA Only) ISIM IC15.0 (CIA Only) ISIM IC15.0+ (CIA Only) NIRSpec.ICE - Build 3.1.12 (CIA Only) NIRSpec.MS - Build 6.2 (CIA Only) NIRSpec.DS - Build 6.2 (CIA Only) FGS - Build 4.97 FGS - Build 5.0 FGS - Build 5.0+
Inputs (includes Technical Reference):	In Scope developer requirements
Prerequisites:	Any previous internal or external activity needed before this method can be applied
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	JWST has been under IV&V for many years, so the analysis methods have evolved over time and the rigor has had a tendency to grow as capability is added. The following is a breakdown of the analysis methods and assurance levels by phase consistent with the current methods in Compass and consistent with the current and past practices in IV&V analyses. The assurance increases by adding additional layers of assurance, and

	in some cases, the assurance in one level covers lower levels. Test 1. M-10: Validate Test Plan by Inspection a. Assurance Objective: The test approach (i.e. verification methods: Analysis, Inspection, Test) is conclusive and appropriate b. TF: 4.1.1 (F), 4.1.2 (F), 4.1.3 (F), 4.1.4 (F), 4.1.5 (F), 4.3 (P), 4.8 (P) 2. M-35: Validate Test Procedures by Inspection and Traces to Requirements a. Assurance Objective: Planned Test Procedures completely exercise and provide conclusive evidence to fully verify the Requirement set b. TF: 4.2 (P), 4.6 (F) 3. M-25: Validate Test Cases by Inspection and Traces to Requirements a. Assurance Objective: Test Cases/Scripts correctly execute the planned test procedures, and produce conclusive evidence verifying the Requirement set b. TF: 4.2 (F), 4.5 (F), 4.7 (F) 4. (New) M-103: M-103: Verify and Validate Requirement Implementation using Flow Diagrams to Uncover Missing, Conflicting, or Unnecessary Behavior a. Assurance Objective: Software Behaviors (Requirement, Design, or Implementation) are correctly and completely exercised and conclusive evidence is produced. b. Note: Requirement Set is expanded to include the behaviors implied by the requirement set, additional to the design, or logical paths implemented in the code. c. Note: More encompassing than requirement set, less encompassing than code coverage. d. TF: (To be Updated to reflect) 4.1.1 (F), 4.1.2 (F), 4.1.3 (F), 4.1.4 (F), 4.1.5 (F), 4.5 (F), 4.6 (F), 4.7 (F) 5. (New) M-62: Verify Suitability of Developer's Test Environment by Simulation of Developer Test Operations Against Test Environment Validation Criteria a. Assurance Objective: Developer Test Environment (Simulators) are of appropriate fidelity for intended use. b. TF: 4.4 (F), 4.8 (F)
Concerns:	None

Method Application Notes: None	
Required Tools:	None identified
Empirical Evidence:	• Engineering worksheets documenting that test plans, scenarios, etc. will provide adequate verification of the requirements. • Risks or findings in technical reports documenting systemic concerns
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	NASA Missions may use different terminology in defining their test content. Each IV&V Project needs to understand how its Mission's test content is being developed and plan accordingly. Also, because test cases, procedures, and designs may be developed iteratively and at multiple levels, IV&V task iteration may be necessary. (per IVV 09-1 Rev M)

Activity 18: Validate Test Design via Test Procedure Inspection and Tracing to Requirements

Method:	M-35, Version 1.1 (Current Status: Approved)
Method Title:	Validate Test Procedures by Inspection and Traces to Requirements
Method Synopsis	Method for manual evaluation of developer test procedures against requirements and test plan criteria to confirm that they are compliant with applicable project and NASA standards, complete and sufficient to create test cases that will fully verify the requirements of interest. Detects test procedure defects including inadequate coverage of in-scope requirements, inadequately defined inputs, and insufficient evaluation criteria.
Subsystem/Entity	GND.FOS; GND.WSS.WEx; GND.WSS.MMS; GND.OSS; OBS.SCE.SCS; OBS.SCE.EMTB; OBS.SCE.CTP.SC; OBS.SCE.DEU; OBS.ISIM.ISIM; OBS.ISIM.NIRSpec.ICE; OBS.ISIM.NIRSpec.MS; OBS.ISIM.NIRSpec.DS; OBS.ISIM.FGS
Required Method Revisions (if any)	None
Technical Goal:	4.6 Ensure that the Test Procedures under analysis specify the correct sequence of actions necessary for the execution of the tests to satisfy their intended test objectives. • Verify that the Test Procedures under analysis comply with project defined test document purpose, format, and content. 4.2 Ensure that valid relationships are defined between the Test Plans, Designs, Cases, and Procedures for test types and documents subject to IV&V test analysis. • Validate that the Test Procedures under analysis satisfy the criteria in the associated Test Plan, Test Design, and Test Cases.
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev O
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	FOS - Build 5 WSS-WEx - Build 5 WSS-MMS - Build 5 (Change Impact Analysis (CIA) Only) OSS - Build 5 OSS - Build 6

	Stored Command Sequences (SCS's) – B2.2 EMTB Build 2.2 Suitability EMTB Build 2.2 SC FSW - Build 3.0 DEU FSW - Build 1.1 ISIM IC14.21+ (CIA Only) ISIM IC15.0 (CIA Only) ISIM IC15.0+ (CIA Only) NIRSpec.ICE - Build 3.1.12 (CIA Only) NIRSpec.MS - Build 6.2 (CIA Only) NIRSpec.DS - Build 6.2 (CIA Only) FGS - Build 4.97 FGS - Build 5.0 FGS - Build 5.0+
Inputs (includes Technical Reference):	Test Plan for the applicable level of test Test Design/Cases for the associated test plan Validated requirements to validate the Test Procedures against (relevant set of validated requirements will be iteration/instantiation specific). Technical Reference including IV&V-generated list of off-nominal conditions IV&V Test Design/Case analysis results and submitted/TBV Issues, and (if available) planned resolution
Prerequisites:	Validated Requirements
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	JWST has been under IV&V for many years, so the analysis methods have evolved over time and the rigor has had a tendency to grow as capability is added. The following is a breakdown of the analysis methods and assurance levels by phase consistent with the current methods in Compass and consistent with the current and past practices in IV&V analyses. The assurance increases by adding additional layers of assurance, and in some cases, the assurance in one level covers lower levels. Test 1. M-10: Validate Test Plan by Inspection

	a. Assurance Objective: The test approach (i.e. verification methods: Analysis, Inspection, Test) is conclusive and appropriate b. TF: 4.1.1 (F), 4.1.2 (F), 4.1.3 (F), 4.1.4 (F), 4.1.5 (F), 4.3 (P), 4.8 (P) 2. M-35: Validate Test Procedures by Inspection and Traces to Requirements a. Assurance Objective: Planned Test Procedures completely exercise and provide conclusive evidence to fully verify the Requirement set b. TF: 4.2 (P), 4.6 (F) 3. M-25: Validate Test Cases by Inspection and Traces to Requirements a. Assurance Objective: Test Cases/Scripts correctly execute the planned test procedures, and produce conclusive evidence verifying the Requirement set b. TF: 4.2 (F), 4.5 (F), 4.7 (F) 4. (New) M-103: M-103: Verify and Validate Requirement Implementation using Flow Diagrams to Uncover Missing, Conflicting, or Unnecessary Behavior a. Assurance Objective: Software Behaviors (Requirement, Design, or Implementation) are correctly and completely exercised and conclusive evidence is produced. b. Note: Requirement Set is expanded to include the behaviors implied by the requirement set, additional to the design, or logical paths implemented in the code. c. Note: More encompassing than requirement set, less encompassing than code coverage. d. TF: (To be Updated to reflect) 4.1.1 (F), 4.1.2 (F), 4.1.3 (F), 4.1.4 (F), 4.1.5 (F), 4.5 (F), 4.6 (F), 4.7 (F) 5. (New) M-62: Verify Suitability of Developer's Test Environment by Simulation of Developer Test Operations Against Test Environment Validation Criteria a. Assurance Objective: Developer Test Environment (Simulators) are of appropriate fidelity for intended use. b. TF: 4.4 (F), 4.8 (F)
Concerns:	None
Method Application Notes: None	
Required Tools:	None identified
Empirical Evidence:	- Engineering worksheets documenting results of tracing test procedures to the requirements and

	comments confirming the adequate verification of the requirements. - Risks or findings in technical reports documenting systemic concerns
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	May be performed simultaneously "Validate Test Cases by Inspection and Traces to Requirements"

Activity 19: Validate Test Design via Test Case Inspection and Tracing to Requirements

Method:	M-25, Version 1.4 (Current Status: Approved)
Method Title:	Validate Test Cases by Inspection and Traces to Requirements
Method Synopsis	Method for manual evaluation of developer test case artifacts against requirements to confirm presence of defined inputs, expected results and evaluation criteria that comply with test plans and objectives and ensure that all requirements implemented are verified by the appropriate test case. Detects test case defects including inadequate coverage of in-scope requirements, logic errors, inadequate (or missing) defined inputs, results expectations, traceability, or evaluation criteria.
Subsystem/Entity	GND.FOS; GND.WSS.WEx; GND.WSS.MMS; GND.OSS; OBS.SCE.SCS; OBS.SCE.EMTB; OBS.SCE.CTP.SC; OBS.SCE.DEU; OBS.ISIM.ISIM; OBS.ISIM.NIRSpec.ICE; OBS.ISIM.NIRSpec.MS; OBS.ISIM.NIRSpec.DS; OBS.ISIM.FGS
Required Method Revisions (if any)	None
Technical Goal:	Determination that the test cases address the requirements expected to be implemented in the applicable iteration/instantiation under both nominal and off-nominal conditions. Includes verifying that the mapped/traced test cases verify the relevant requirements and providing an assessment of the coverage of the requirements at the applicable level of test Verify that the Test Cases under analysis comply with project defined test document purpose, format, and content. Ensure that valid relationships are defined between the Test Plans, Designs, Cases, and Procedures for test types and documents subject to IV&V test analysis. Ensure that the Test Cases under analysis specify the correct test inputs, predicted results, and sets of execution conditions necessary to satisfy their intended test objectives (covering both nominal and off-nominal conditions).

	Ensure that the Test Designs under analysis correctly specify the details of the test approach for the covered software feature or combination of software features and identify the associated tests. 4.1 Ensure that the planned tests are sufficient to: 4.1.1 Ensure that the software correctly implements system and software requirements in an operational environment under nominal and off-nominal conditions. 4.1.2 Ensure that the complete, integrated system complies with its specified system requirements allocated to software and to validate whether the system meets its original objectives. 4.1.3 Ensure that the software meets all of the software requirements and is ready to be integrated with system hardware. 4.1.4 Ensure that the software correctly implements the software requirements and design as each software component (e.g., units or modules) is incrementally integrated with each other. 4.1.5 Ensure that the software components (e.g., units, source code modules) correctly implement software component requirements. 4.2 Ensure that valid relationships are defined between the Test Plans, Designs, Cases, and Procedures for test types and documents subject to IV&V test analysis. 4.5 Ensure that the Test Cases under analysis specify the correct test inputs, predicted results, and sets of execution conditions necessary to satisfy their intended test objectives (covering both nominal and off-nominal conditions). 4.7 Ensure that the Test Designs under analysis correctly specify the details of the test approach for the covered software feature or combination of software features and identify the associated tests.
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev O
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	FOS - Build 5 WSS-WEx - Build 5 WSS-MMS - Build 5 (Change Impact Analysis (CIA) Only) OSS - Build 5

	OSS - Build 6 Stored Command Sequences (SCS's) – B2.2 EMTB Build 2.2 Suitability EMTB Build 2.2 SC FSW - Build 3.0 DEU FSW - Build 1.1 ISIM IC14.21+ (CIA Only) ISIM IC15.0 (CIA Only) ISIM IC15.0+ (CIA Only) NIRSpec.ICE - Build 3.1.12 (CIA Only) NIRSpec.MS - Build 6.2 (CIA Only) NIRSpec.DS - Build 6.2 (CIA Only) FGS - Build 4.97 FGS - Build 5.0 FGS - Build 5.0+
Inputs (includes Technical Reference):	Test Plan for the applicable level of test Test cases and developer defined scenarios for the test Test scripts Validated requirements to validate the Test Cases against IV&V-generated list of off-nominal conditions Requirement to Test Case Traceability (RTVMs) Software Requirements and Design Specifications (SRDSs) Project Specific Technical Reference material
Prerequisites:	Validated Requirements
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	JWST has been under IV&V for many years, so the analysis methods have evolved over time and the rigor has had a tendency to grow as capability is added. The following is a breakdown of the analysis methods and assurance levels by phase consistent with the current methods in Compass and consistent with the current and past practices in IV&V analyses. The assurance increases by adding additional layers of assurance, and in some cases, the assurance in one level covers lower levels. Test 1. M-10: Validate Test Plan by Inspection

	a. Assurance Objective: The test approach (i.e. verification methods: Analysis, Inspection, Test) is conclusive and appropriate b. TF: 4.1.1 (F), 4.1.2 (F), 4.1.3 (F), 4.1.4 (F), 4.1.5 (F), 4.3 (P), 4.8 (P) 2. M-35: Validate Test Procedures by Inspection and Traces to Requirements a. Assurance Objective: Planned Test Procedures completely exercise and provide conclusive evidence to fully verify the Requirement set b. TF: 4.2 (P), 4.6 (F) 3. M-25: Validate Test Cases by Inspection and Traces to Requirements a. Assurance Objective: Test Cases/Scripts correctly execute the planned test procedures, and produce conclusive evidence verifying the Requirement set b. TF: 4.2 (F), 4.5 (F), 4.7 (F) 4. (New) M-103: M-103: Verify and Validate Requirement Implementation using Flow Diagrams to Uncover Missing, Conflicting, or Unnecessary Behavior a. Assurance Objective: Software Behaviors (Requirement, Design, or Implementation) are correctly and completely exercised and conclusive evidence is produced. b. Note: Requirement Set is expanded to include the behaviors implied by the requirement set, additional to the design, or logical paths implemented in the code. c. Note: More encompassing than requirement set, less encompassing than code coverage. d. TF: (To be Updated to reflect) 4.1.1 (F), 4.1.2 (F), 4.1.3 (F), 4.1.4 (F), 4.1.5 (F), 4.5 (F), 4.6 (F), 4.7 (F) 5. (New) M-62: Verify Suitability of Developer's Test Environment by Simulation of Developer Test Operations Against Test Environment Validation Criteria a. Assurance Objective: Developer Test Environment (Simulators) are of appropriate fidelity for intended use. b. TF: 4.4 (F), 4.8 (F)
Concerns:	None
Method Application Notes: None	
Required Tools:	ORBIT, Excel spreadsheets
Empirical Evidence:	Engineering worksheets documenting results of tracing test procedures to the requirements and

	comments confirming the adequate verification of the requirements.
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	May be performed simultaneously with "Validate Test Procedures by Inspection and Traces to Requirements"

Activity 20: Validate Test Design via Test Cases by Inspection Against Requirements and System Reference Model

Method:	M-26, Version 1.0 (Current Status: Approved)
Method Title:	Validate Test Cases by Inspection Against Requirements and System Reference Model
Method Synopsis	Leverage scenarios from a System Reference Model (SRM) to manually determine that the Test Cases address the requirements expected to be implemented in the applicable iteration/instantiation under both nominal and off-nominal conditions. The analysis ensures that the test cases under analysis specify the correct test inputs, predicted results, and sets of execution conditions necessary to satisfy their intended test objectives. Detects test case defects including inadequate coverage of in-scope requirements, inadequate (or missing) defined inputs, results expectations or evaluation criteria.
Subsystem/Entity	OBS.SCE.EMTB
Required Method Revisions (if any)	None
Technical Goal:	• Determination that the Test Cases address the requirements expected to be implemented in the applicable iteration/instantiation under both nominal and off-nominal conditions – Includes mapping the test cases to the relevant requirements and providing an assessment of the coverage of the requirements at the applicable level of test • Verify that the Test Cases under analysis comply with project defined test document purpose, format, and content. • Validate that the Test Cases under analysis satisfy the criteria in the associated Test Plan. From 09-1 Rev N: 4.1 Ensure that the planned tests are sufficient to: 4.1.1 Ensure that the software correctly implements system and software requirements in an operational environment under nominal and off-nominal conditions. 4.1.2 Ensure that the complete, integrated system complies with its specified system requirements allocated to software and to validate whether the system meets its original objectives.

	4.1.3 Ensure that the software meets all of the (in-scope) software requirements and is ready to be integrated with system hardware. 4.1.4 Ensure that the software correctly implements the software requirements and design as each software component (e.g., units or modules) is incrementally integrated with each other. 4.1.5 Ensure that the software components (e.g., units, source code modules) correctly implement software component requirements. 4.2 Ensure that valid relationships are defined between the Test Plans, Test Cases, and documents subject to IV&V test analysis. 4.5 Ensure that the Test Cases under analysis specify the correct test inputs, predicted results, and sets of execution conditions necessary to satisfy their intended test objectives (covering both nominal and off-nominal conditions).
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev 0
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	Spacecraft EMTB Test Plans
Inputs (includes Technical Reference):	Applicable Technical Reference (i.e. use case scenarios).
Prerequisites:	Documentation of scenarios in the SRM at various levels to serve as added basis for validation analysis. Documentation of criteria to be assessed for both Q2 and Q3.
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	JWST has been under IV&V for many years, so the analysis methods have evolved over time and the rigor has had a tendency to grow as capability is added. The following is a breakdown of the analysis methods and assurance levels by phase consistent with the current methods in Compass and consistent with the current and past practices in IV&V analyses. The assurance increases by adding additional layers of assurance, and in some cases, the assurance in one level covers lower levels. Test

	1. M-10: Validate Test Plan by Inspection a. Assurance Objective: The test approach (i.e. verification methods: Analysis, Inspection, Test) is conclusive and appropriate b. TF: 4.1.1 (F), 4.1.2 (F), 4.1.3 (F), 4.1.4 (F), 4.1.5 (F), 4.3 (P), 4.8 (P) 2. M-35: Validate Test Procedures by Inspection and Traces to Requirements a. Assurance Objective: Planned Test Procedures completely exercise and provide conclusive evidence to fully verify the Requirement set b. TF: 4.2 (P), 4.6 (F) 3. M-25: Validate Test Cases by Inspection and Traces to Requirements a. Assurance Objective: Test Cases/Scripts correctly execute the planned test procedures, and produce conclusive evidence verifying the Requirement set b. TF: 4.2 (F), 4.5 (F), 4.7 (F) 4. (New) M-103: M-103: Verify and Validate Requirement Implementation using Flow Diagrams to Uncover Missing, Conflicting, or Unnecessary Behavior a. Assurance Objective: Software Behaviors (Requirement, Design, or Implementation) are correctly and completely exercised and conclusive evidence is produced. b. Note: Requirement Set is expanded to include the behaviors implied by the requirement set, additional to the design, or logical paths implemented in the code. c. Note: More encompassing than requirement set, less encompassing than code coverage. d. TF: (To be Updated to reflect) 4.1.1 (F), 4.1.2 (F), 4.1.3 (F), 4.1.4 (F), 4.1.5 (F), 4.5 (F), 4.6 (F), 4.7 (F) 5. (New) M-62: Verify Suitability of Developer's Test Environment by Simulation of Developer Test Operations Against Test Environment Validation Criteria

	a. Assurance Objective: Developer Test Environment (Simulators) are of appropriate fidelity for intended use. b. TF: 4.4 (F), 4.8 (F)
Concerns:	None
Method Application Notes: This method applies to multiple phases of development; however, the mapping here is to Test phase artifacts thus the Test TF Goals are mapped.	
Required Tools:	Modeling tool used to navigate and display the SRM
Empirical Evidence:	Engineering worksheets documenting results of tracing Test Cases/Scenarios to the Requirements and comments confirming the adequate verification of the requirements sd well as the assessment of the test implementation
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	None.

Activity 21: Validate Test Design via Inspection of Test Scripts and Validated Requirements

Method:	M-16, Version 1.0 (Current Status: Approved)
Method Title:	Validate Test Design by Inspecting Test Scripts and Validated Requirements
Method Synopsis	Method uses UML modeling of the test design via test use case and test activity diagrams that capture the steps required to achieve the objectives of the test design artifacts under study to evaluate the extent to which source requirements are covered in the test design. Method detects defects in the test design's ability to satisfy the test objective and completely cover the target source requirements. Method supports regression testing as the developer provides updated test design artifacts.
Subsystem/Entity	OBS.ISIM.ISIM
Required Method Revisions (if any)	None
Technical Goal:	4.1 Ensure that the planned tests are sufficient to: 4.1.1 Ensure that the software correctly implements system and software requirements in an operational environment under nominal and off-nominal conditions. 4.1.2 Ensure that the complete, integrated system complies with its specified system requirements allocated to software and to validate whether the system meets its original objectives. 4.1.3 Ensure that the software meets all of the software requirements and is ready to be integrated with system hardware. 4.1.4 Ensure that the software correctly implements the software requirements and design as each software component (e.g., units or modules) is incrementally integrated with each other. 4.1.5 Ensure that the software components (e.g., units, source code modules) correctly implement software component requirements. 4.7 Ensure that the Test Designs under analysis correctly specify the details of the test approach for the covered software feature or combination of software features and identify the associated tests
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev O
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	ISIM MIRI Test Scripts for Build 2.3 & 2.x

Inputs (includes Technical Reference):	Associated Requirements, test plan, test design, test scripts
Prerequisites:	Validated Requirements (for the test level being analyzed)
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	JWST has been under IV&V for many years, so the analysis methods have evolved over time and the rigor has had a tendency to grow as capability is added. The following is a breakdown of the analysis methods and assurance levels by phase consistent with the current methods in Compass and consistent with the current and past practices in IV&V analyses. The assurance increases by adding additional layers of assurance, and in some cases, the assurance in one level covers lower levels. Test 1. M-10: Validate Test Plan by Inspection a. Assurance Objective: The test approach (i.e. verification methods: Analysis, Inspection, Test) is conclusive and appropriate b. TF: 4.1.1 (F), 4.1.2 (F), 4.1.3 (F), 4.1.4 (F), 4.1.5 (F), 4.3 (P), 4.8 (P) 2. M-35: Validate Test Procedures by Inspection and Traces to Requirements a. Assurance Objective: Planned Test Procedures completely exercise and provide conclusive evidence to fully verify the Requirement set b. TF: 4.2 (P), 4.6 (F) 3. M-25: Validate Test Cases by Inspection and Traces to Requirements a. Assurance Objective: Test Cases/Scripts correctly execute the planned test procedures, and produce conclusive evidence verifying the Requirement set b. TF: 4.2 (F), 4.5 (F), 4.7 (F) 4. (New) M-103: M-103: Verify and Validate Requirement Implementation using Flow

	Diagrams to Uncover Missing, Conflicting, or Unnecessary Behavior a. Assurance Objective: Software Behaviors (Requirement, Design, or Implementation) are correctly and completely exercised and conclusive evidence is produced. b. Note: Requirement Set is expanded to include the behaviors implied by the requirement set, additional to the design, or logical paths implemented in the code. c. Note: More encompassing than requirement set, less encompassing than code coverage. d. TF: (To be Updated to reflect) 4.1.1 (F), 4.1.2 (F), 4.1.3 (F), 4.1.4 (F), 4.1.5 (F), 4.5 (F), 4.6 (F), 4.7 (F) 5. (New) M-62: Verify Suitability of Developer's Test Environment by Simulation of Developer Test Operations Against Test Environment Validation Criteria a. Assurance Objective: Developer Test Environment (Simulators) are of appropriate fidelity for intended use. b. TF: 4.4 (F), 4.8 (F)
Concerns:	None
Method Application Notes: This method applies to multiple phases of development; however, the mapping here is to Test phase artifacts thus the Test TF Goals are mapped.	
Required Tools:	• Excel sheet • Model browser (if models are used) • Code reviewer
Empirical Evidence:	Engineering worksheets documenting results of tracing behaviors to test design sections and comments confirming the valid testing of the rqmts associated with specific behaviors.
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	None

Activity 22: Validate Test Design via Inspecting Traces from Scenarios

Method:	M-21, Version 1.0 (Current Status: Approved)
Method Title:	Validate Test Design by Inspecting Traces from Scenarios
Method Synopsis	Method develops and validates a set of scenarios based upon critical behaviors defined at a level of detail that matches test artifacts under evaluation, traces those scenarios to requirements, software structures and source code, and test design and test environment components to detect defects in the test design.
Subsystem/Entity	OBS.ISIM.ISIM; OBS.ISIM.NIRSpec.MS; OBS.ISIM.NIRSpec.DS
Required Method Revisions (if any)	None
Technical Goal:	4.7 Ensure that the Test Designs under analysis correctly specify the details of the test approach for the covered software feature or combination of software features and identify the associated tests. 4.8 Ensure that the test environment is sufficiently complete, correct, and accurate to perform the intended testing
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev O
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	1. ISIM MIRI Build 2.3 & 2.x Test Documentation 2. ISIM NIRSpec MS Build 6.1 & 6.2 Test Documentation 3. ISIM NIRSpec DS Build 6.1 & 6.2 Test Documentation
Inputs (includes Technical Reference):	In-scope requirements (functional, interface) Software design Concept of Operations
Prerequisites:	None
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	JWST has been under IV&V for many years, so the analysis methods have evolved over time and the rigor has had a tendency to grow as capability is added. The following is a breakdown of the analysis methods and assurance levels by phase consistent with the current methods in Compass and consistent with the current

	and past practices in IV&V analyses. The assurance increases by adding additional layers of assurance, and in some cases, the assurance in one level covers lower levels. Test 1. M-10: Validate Test Plan by Inspection a. Assurance Objective: The test approach (i.e. verification methods: Analysis, Inspection, Test) is conclusive and appropriate b. TF: 4.1.1 (F), 4.1.2 (F), 4.1.3 (F), 4.1.4 (F), 4.1.5 (F), 4.3 (P), 4.8 (P) 2. M-35: Validate Test Procedures by Inspection and Traces to Requirements a. Assurance Objective: Planned Test Procedures completely exercise and provide conclusive evidence to fully verify the Requirement set b. TF: 4.2 (P), 4.6 (F) 3. M-25: Validate Test Cases by Inspection and Traces to Requirements a. Assurance Objective: Test Cases/Scripts correctly execute the planned test procedures, and produce conclusive evidence verifying the Requirement set b. TF: 4.2 (F), 4.5 (F), 4.7 (F) 4. (New) M-103: M-103: Verify and Validate Requirement Implementation using Flow Diagrams to Uncover Missing, Conflicting, or Unnecessary Behavior a. Assurance Objective: Software Behaviors (Requirement, Design, or Implementation) are correctly and completely exercised and conclusive evidence is produced. b. Note: Requirement Set is expanded to include the behaviors implied by the requirement set, additional to the design, or logical paths implemented in the code. c. Note: More encompassing than requirement set, less encompassing than code coverage.

	d. TF: (To be Updated to reflect) 4.1.1 (F), 4.1.2 (F), 4.1.3 (F), 4.1.4 (F), 4.1.5 (F), 4.5 (F), 4.6 (F), 4.7 (F) 5. (New) M-62: Verify Suitability of Developer's Test Environment by Simulation of Developer Test Operations Against Test Environment Validation Criteria a. Assurance Objective: Developer Test Environment (Simulators) are of appropriate fidelity for intended use. b. TF: 4.4 (F), 4.8 (F)
Concerns:	None
Method Application Notes: This method applies to multiple phases of development; however, the mapping here is to Test phase artifacts thus the Test TF Goals are mapped.	
Required Tools:	None
Empirical Evidence:	Scenarios Traces
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	Issues may also be identified during scenario development and tracing

Activity 23: Verify and Validate Test Design via Flow Diagrams

Method:	M-103, Version 1.0 (Current Status: Approved)
Method Title:	Verify and Validate Requirement Implementation using Flow Diagrams to Uncover Missing, Conflicting, or Unnecessary Behavior
Method Synopsis	Method uses Flow Diagrams to analyze software implementation of requirements to ensure the correct and complete implementation of requirements on a system level as well as an atomic level. (Level is dependent upon the abstraction in the modeling chosen by the analyst as well as the available level of artifacts being targeted). Further, the method is applied to the source code that is not specified by requirements or not specified directly.
Subsystem/Entity	OBS.SCE.CTP.SC
Required Method Revisions (if any)	(Method needs to be updated to include the applicability to test analysis)
Technical Goal:	1. Gain system/software level understanding 2. Uncover ambiguous or missing behaviors 3. Uncover conflicting or undesired behaviors 4. Uncover failure scenarios (identify whether the FSW is protected against off-nominal/adverse conditions/inputs) 4.1 Ensure that the planned tests are sufficient to: 4.1.1 Ensure that the software correctly implements system and software requirements in an operational environment under nominal and off-nominal conditions. 4.1.2 Ensure that the complete, integrated system complies with its specified system requirements allocated to software and to validate whether the system meets its original objectives. 4.1.3 Ensure that the software meets all of the (in-scope) software requirements and is ready to be integrated with system hardware. 4.1.4 Ensure that the software correctly implements the software requirements and design as each software component (e.g., units or modules) is incrementally integrated with each other. 4.1.5 Ensure that the software components (e.g., units, source code modules) correctly implement software component requirements. 4.5 Ensure that the Test Cases under analysis specify the correct test inputs, predicted results, and sets of

	execution conditions necessary to satisfy their intended test objectives (covering both nominal and off-nominal conditions). 4.6 Ensure that the Test Procedures under analysis specify the correct sequence of actions necessary for the execution of the tests to satisfy their intended test objectives. 4.7 Ensure that the Test Designs under analysis correctly specify the details of the test approach for the covered software feature or combination of software features and identify the associated tests.
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev O 4.1.1 (F), 4.1.2 (F), 4.1.3 (F), 4.1.4 (F), 4.1.5 (F), 4.5 (F), 4.6 (F), 4.7 (F)
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	SC FSW - Build 3.0
Inputs (includes Technical Reference):	Applicable Technical Reference (i.e. use case scenarios)
Prerequisites:	FSW is not developed using behavior models (uml activity, state, or similar), the models are not of high enough fidelity to analyze code behavior, or the models are not sufficient to provide system level understanding. Note: The process of generating the diagrams can be used to gain system level understanding even when the diagrams duplicate developer products.
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	JWST has been under IV&V for many years, so the analysis methods have evolved over time and the rigor has had a tendency to grow as capability is added. The following is a breakdown of the analysis methods and assurance levels by phase consistent with the current methods in Compass and consistent with the current and past practices in IV&V analyses. The assurance increases by adding additional layers of assurance, and in some cases, the assurance in one level covers lower levels. Test 1. M-10: Validate Test Plan by Inspection

	a. Assurance Objective: The test approach (i.e. verification methods: Analysis, Inspection, Test) is conclusive and appropriate b. TF: 4.1.1 (F), 4.1.2 (F), 4.1.3 (F), 4.1.4 (F), 4.1.5 (F), 4.3 (P), 4.8 (P) 2. M-35: Validate Test Procedures by Inspection and Traces to Requirements a. Assurance Objective: Planned Test Procedures completely exercise and provide conclusive evidence to fully verify the Requirement set b. TF: 4.2 (P), 4.6 (F) 3. M-25: Validate Test Cases by Inspection and Traces to Requirements a. Assurance Objective: Test Cases/Scripts correctly execute the planned test procedures, and produce conclusive evidence verifying the Requirement set b. TF: 4.2 (F), 4.5 (F), 4.7 (F) 4. (New) M-103: M-103: Verify and Validate Requirement Implementation using Flow Diagrams to Uncover Missing, Conflicting, or Unnecessary Behavior a. Assurance Objective: Software Behaviors (Requirement, Design, or Implementation) are correctly and completely exercised and conclusive evidence is produced. b. Note: Requirement Set is expanded to include the behaviors implied by the requirement set, additional to the design, or logical paths implemented in the code. c. Note: More encompassing than requirement set, less encompassing than code coverage. d. TF: (To be Updated to reflect) 4.1.1 (F), 4.1.2 (F), 4.1.3 (F), 4.1.4 (F), 4.1.5 (F), 4.5 (F), 4.6 (F), 4.7 (F) 5. (New) M-62: Verify Suitability of Developer's Test Environment by Simulation of Developer Test Operations Against Test Environment Validation Criteria a. Assurance Objective: Developer Test Environment (Simulators) are of appropriate fidelity for intended use. b. TF: 4.4 (F), 4.8 (F)
Concerns:	None
Method Application Notes: This method applies to multiple phases of development; however, the mapping here is to Test phase artifacts thus the Test TF Goals are mapped.	
Required Tools:	Drawing Package (i.e. Visio, Together, etc) Code Viewing and analysis tool (i.e. Understand)

Empirical Evidence:	1. Control Flow diagrams and corresponding notes capture IV&V analysis, questions, and concerns. - "Analysis/Findings Report" 2. Diagrams show/trace the behaviors and their source - "Coverage Reports." 2) Test/Analysis Scenarios (nominal and off-nominal) derived from flow paths - "Scenario Report" Example Assurance Claims/Statements: The FSW does not contain any unnecessary functions or "features" The FSW implements all required behaviors with appropriate response to off-nominal conditions The FSW verification suite completely covers the required behaviors as well as the nominal, off-nominal, and non-required (i.e. feature) paths in the FSW
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	None

Activity 24: Verify Test Design via Simulation of Developer Test Operations Against Test Environment Validation

Method:	M-62, Version 1.0 (Current Status: Approved)
Method Title:	Verify Suitability of Developer's Test Environment by Simulation of Developer Test Operations Against Test Environment Validation Criteria
Method Synopsis	In some instances the IV&V Program will work with the development organization to replicate the development organization's V&V environment. IV&V analysts can utilize the provided environment to ensure the developer's test environment is sufficiently complete, correct, and accurate to perform the intended testing. This method benefits early testing and provides assurance in that the Developer's test environment is capable of executing the planned test suite.
Subsystem/Entity	OBS.SCE.EMTB; OBS.SCE.CTP.SC
Required Method Revisions (if any)	None
Technical Goal:	4.4 Ensure that any incorporated simulations from the developer into the IV&V Test Environment are sufficiently complete, correct, and accurate to perform the intended testing. Test Environments can be evaluated for correctness, accuracy, reliability, flexibility, testability, portability, reusability, and interoperability. 4.8 Ensure that the test environment is sufficiently complete, correct, and accurate to perform the intended testing.
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev O
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	EMTB Build 2.2 Suitability SC FSW - Build 3.0
Inputs (includes Technical Reference):	• Developer Test Artifacts (Test Plan, Test Scenarios, Test Procedures) • Requirements trace to test cases • Test Results (logs, output, etc.) • Test Configurations (where applicable)
Prerequisites:	Software under test (source and binary preferred), IV&V Test Environment (mirror of Developer Test Environment)

Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	JWST has been under IV&V for many years, so the analysis methods have evolved over time and the rigor has had a tendency to grow as capability is added. The following is a breakdown of the analysis methods and assurance levels by phase consistent with the current methods in Compass and consistent with the current and past practices in IV&V analyses. The assurance increases by adding additional layers of assurance, and in some cases, the assurance in one level covers lower levels. Test 1. M-10: Validate Test Plan by Inspection a. Assurance Objective: The test approach (i.e. verification methods: Analysis, Inspection, Test) is conclusive and appropriate b. TF: 4.1.1 (F), 4.1.2 (F), 4.1.3 (F), 4.1.4 (F), 4.1.5 (F), 4.3 (P), 4.8 (P) 2. M-35: Validate Test Procedures by Inspection and Traces to Requirements a. Assurance Objective: Planned Test Procedures completely exercise and provide conclusive evidence to fully verify the Requirement set b. TF: 4.2 (P), 4.6 (F) 3. M-25: Validate Test Cases by Inspection and Traces to Requirements a. Assurance Objective: Test Cases/Scripts correctly execute the planned test procedures, and produce conclusive evidence verifying the Requirement set b. TF: 4.2 (F), 4.5 (F), 4.7 (F) 4. (New) M-103: M-103: Verify and Validate Requirement Implementation using Flow Diagrams to Uncover Missing, Conflicting, or Unnecessary Behavior a. Assurance Objective: Software Behaviors (Requirement, Design, or Implementation) are correctly and completely exercised and conclusive evidence is produced. b. Note: Requirement Set is expanded to include the behaviors implied by the requirement set, additional to the design, or logical paths implemented in the code.

	c. Note: More encompassing than requirement set, less encompassing than code coverage. d. TF: (To be Updated to reflect) 4.1.1 (F), 4.1.2 (F), 4.1.3 (F), 4.1.4 (F), 4.1.5 (F), 4.5 (F), 4.6 (F), 4.7 (F) 5. (New) M-62: Verify Suitability of Developer's Test Environment by Simulation of Developer Test Operations Against Test Environment Validation Criteria a. Assurance Objective: Developer Test Environment (Simulators) are of appropriate fidelity for intended use. b. TF: 4.4 (F), 4.8 (F)
Concerns:	None
Method Application Notes: None	
Required Tools:	IV&V Test Environment that mirrors Developer Test Environment
Empirical Evidence:	• Test Procedures, Results and Logs provide evidence and coverage on the Development Organization's test environment.
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	None

Activity 25: Verify Test Execution via Test Case, Test Inputs and Results

Method:	M-11, Version 1.2 (Current Status: Approved)
Method Title:	Verify Test Execution by Inspection of Test Cases, Inputs and Results
Method Synopsis	Method performs manual inspection of test results to detect defects in test execution integrity, regression testing, requirements to test coverage, and test result documentation and software defect resolution handling.
Subsystem/Entity	GND.WSS.MMS; OBS.ISIM.ISIM
Required Method Revisions (if any)	None
Technical Goal:	NOTE: Coordination between the requirements, test validation, and implementation verification (including assessment of test results) efforts is essential to assess the entire picture. 6.4 (Full) Ensure that test results are as expected (per the corresponding plans, cases, procedures, design) and the impacts of any discrepancies are understood. 6.2 (Partial) Ensure that the source code components can reliably perform required capabilities under nominal and off-nominal conditions, perform no undesired functions, and that the documentation (both embedded and stand-alone) can facilitate code maintenance at a later time.
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev O
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	1. ISIM Test Cases, Inputs and Results for Build 14.22, 14.3, 15.0, 15.0+ 2. MMS Test Cases, Inputs and Results for Build 5.0
Inputs (includes Technical Reference):	1) Test execution results, 2) expected test results, 3) results from objective 4.5 (should include RTTM, test cases, assessment of test coverage of requirements); Verification Cross reference Matrix indicating links from requirements to tests, and Version Description Documents describing the test status for all requirements, known problems for the Build tested.
Prerequisites:	Objective 4.5 (validation of Test Cases) has been met utilizing the most current test cases (i.e. no changes to

	the Test Cases have been made since objective 4.5 was met).
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	JWST has been under IV&V for many years, so the analysis methods have evolved over time and the rigor has had a tendency to grow as capability is added. The following is a breakdown of the analysis methods and assurance levels by phase consistent with the current methods in Compass and consistent with the current and past practices in IV&V analyses. The assurance increases by adding additional layers of assurance, and in some cases, the assurance in one level covers lower levels. Test 1. M-10: Validate Test Plan by Inspection a. Assurance Objective: The test approach (i.e. verification methods: Analysis, Inspection, Test) is conclusive and appropriate b. TF: 4.1.1 (F), 4.1.2 (F), 4.1.3 (F), 4.1.4 (F), 4.1.5 (F), 4.3 (P), 4.8 (P) 2. M-35: Validate Test Procedures by Inspection and Traces to Requirements a. Assurance Objective: Planned Test Procedures completely exercise and provide conclusive evidence to fully verify the Requirement set b. TF: 4.2 (P), 4.6 (F) 3. M-25: Validate Test Cases by Inspection and Traces to Requirements a. Assurance Objective: Test Cases/Scripts correctly execute the planned test procedures, and produce conclusive evidence verifying the Requirement set b. TF: 4.2 (F), 4.5 (F), 4.7 (F) 4. (New) M-103: M-103: Verify and Validate Requirement Implementation using Flow Diagrams to Uncover Missing, Conflicting, or Unnecessary Behavior

	a. Assurance Objective: Software Behaviors (Requirement, Design, or Implementation) are correctly and completely exercised and conclusive evidence is produced. b. Note: Requirement Set is expanded to include the behaviors implied by the requirement set, additional to the design, or logical paths implemented in the code. c. Note: More encompassing than requirement set, less encompassing than code coverage. d. TF: (To be Updated to reflect) 4.1.1 (F), 4.1.2 (F), 4.1.3 (F), 4.1.4 (F), 4.1.5 (F), 4.5 (F), 4.6 (F), 4.7 (F) 5. (New) M-62: Verify Suitability of Developer's Test Environment by Simulation of Developer Test Operations Against Test Environment Validation Criteria a. Assurance Objective: Developer Test Environment (Simulators) are of appropriate fidelity for intended use. b. TF: 4.4 (F), 4.8 (F)
Concerns:	None
Method Application Notes: This method applies to multiple phases of development; however, the mapping here is to Test phase artifacts thus the Test TF Goals are mapped.	
Required Tools:	none
Empirical Evidence:	- Engineering worksheets documenting that test results were verified to be as expected. Worksheet should include, for each test, assessment of steps A, B, C, D, and E. Results of steps F and G may also be included in the worksheet or subsequent analysis report.
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	Because test cases, procedures, and designs may be developed iteratively and at multiple levels, IV&V task iteration may be necessary.

Activity 26: Verify Software Integration by Test Plan Inspection

Method:	M-10, Version 1.2 (Current Status: Approved)
Method Title:	Validate Test Plan by Inspection
Method Synopsis	Method performs manual inspection of a test plan artifact to detect defects in test plan integrity, compliance with applicable test plan standards and key testing principles, capability for requirements and behaviors to be verified under nominal and adverse conditions, documentation of limitations in test plan verification capability, test environment fidelity appropriateness, test operational procedure integrity, test scheduling and risk assessments, and planned regression testing.
Subsystem/Entity	GND.OSS; OBS.SCE.SCS; OBS.SCE.EMTB; OBS.SCE.CTP.SC; OBS.ISIM.ISIM
Required Method Revisions (if any)	None
Technical Goal:	4.1 Ensure that the planned tests are sufficient to: 4.1.1 Ensure that the software correctly implements system and software requirements in an operational environment under nominal and off-nominal conditions. 4.1.2 Ensure that the complete, integrated system complies with its specified system requirements allocated to software and to validate whether the system meets its original objectives. 4.1.3 Ensure that the software meets all of the software requirements and is ready to be integrated with system hardware. 4.1.4 Ensure that the software correctly implements the software requirements and design as each software component (e.g., units or modules) is incrementally integrated with each other. 4.1.5 Ensure that the software components (e.g., units, source code modules) correctly implement software component requirements. 4.3 Ensure that the planned regression testing to be performed when changes are made to any previously examined software products is sufficient to identify any unintended side effects or impacts of the change on other aspects of the system 4.8 Ensure that the test environment is sufficiently complete, correct, and accurate to perform the intended testing.
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev O

Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	OSS - Build 5 OSS - Build 6 Stored Command Sequences (SCS's) – B2.2 EMTB Build 2.2 SC FSW - Build 2.2 ISIM IC14.21+ ISIM IC15.0 ISIM IC15.0+
Inputs (includes Technical Reference):	In Scope developer requirements
Prerequisites:	Any previous internal or external activity needed before this method can be applied
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	Integration analysis relies on numerous test analysis methods for interface verification and validation level testing. End-to-End Analysis method is also used to analyze the flowdown of design choices throughout the systems, interfaces, and the software units. Integration 1. M-10: Validate Test Plan by Inspection a. Assurance Objective: The test approach (i.e. verification methods: Analysis, Inspection, Test) is conclusive and appropriate b. TF: 4.1.1 (F), 4.1.2 (F), 4.1.3 (F), 4.1.4 (F), 4.1.5 (F), 4.3 (P), 4.8 (P) 2. M-35: Validate Test Procedures by Inspection and Traces to Requirements a. Assurance Objective: Planned Test Procedures completely exercise and provide conclusive evidence to fully verify the Requirement set b. TF: 4.2 (P), 4.6 (F) 3. M-25: Validate Test Cases by Inspection and Traces to Requirements c. Assurance Objective: Test Cases/Scripts correctly execute the planned test procedures, and produce conclusive evidence verifying the Requirement set d. TF: 4.2 (F), 4.5 (F), 4.7 (F)

	4. (New) M-62: Verify Suitability of Developer's Test Environment by Simulation of Developer Test Operations Against Test Environment Validation Criteria e. Assurance Objective: Developer Test Environment (Simulators) are of appropriate fidelity for intended use. f. TF: 4.4 (F), 4.8 (F) 5. M-37: End-to-End Fault Management Verification through Database Development and Analysis a. Assurance Objective: Collection of Software/System behaves as expected under off-nominal conditions. b. Note: Scoped to the full system. c. Note: generally used to feed scenarios to Method M-68 d. TF: 3.1 (P), 3.2 (F), 3.4 (F), 3.5 (F), 4.1.1 (P), 4.1.2 (P), 4.1.3 (P), 4.1.4 (P), 4.1.5 (P), 5.1 (P), 5.2 (F), 5.3 (F), 5.4 (F), 5.5 (P), 5.6 (F), 6.1 (P), 6.3 (F), 6.5 (F), 6.6 (P), 7.8 (F)
Concerns:	None
Method Application Notes: None	
Required Tools:	None identified
Empirical Evidence:	• Engineering worksheets documenting that test plans, scenarios, etc. will provide adequate verification of the requirements. • Risks or findings in technical reports documenting systemic concerns
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	NASA Missions may use different terminology in defining their test content. Each IV&V Project needs to understand how its Mission's test content is being developed and plan accordingly. Also, because test cases, procedures, and designs may be developed iteratively and at multiple levels, IV&V task iteration may be necessary. (per IVV 09-1 Rev M)

Activity 27: Verify Software Integration via Test Procedure Inspection and Tracing to Requirements

Method:	M-35, Version 1.1 (Current Status: Approved)
Method Title:	Validate Test Procedures by Inspection and Traces to Requirements
Method Synopsis	Method for manual evaluation of developer test procedures against requirements and test plan criteria to confirm that they are compliant with applicable project and NASA standards, complete and sufficient to create test cases that will fully verify the requirements of interest. Detects test procedure defects including inadequate coverage of in-scope requirements, inadequately defined inputs, and insufficient evaluation criteria.
Subsystem/Entity	GND.OSS; OBS.SCE.SCS; OBS.SCE.EMTB; OBS.SCE.CTP.SC; OBS.ISIM.ISIM
Required Method Revisions (if any)	None
Technical Goal:	4.6 Ensure that the Test Procedures under analysis specify the correct sequence of actions necessary for the execution of the tests to satisfy their intended test objectives. • Verify that the Test Procedures under analysis comply with project defined test document purpose, format, and content. 4.2 Ensure that valid relationships are defined between the Test Plans, Designs, Cases, and Procedures for test types and documents subject to IV&V test analysis. • Validate that the Test Procedures under analysis satisfy the criteria in the associated Test Plan, Test Design, and Test Cases.
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev O
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	OSS - Build 5 OSS - Build 6 Stored Command Sequences (SCS's) – B2.2 EMTB Build 2.2 SC FSW - Build 2.2 ISIM IC14.21+ ISIM IC15.0

	ISIM IC15.0+
Inputs (includes Technical Reference):	Test Plan for the applicable level of test Test Design/Cases for the associated test plan Validated requirements to validate the Test Procedures against (relevant set of validated requirements will be iteration/instantiation specific). Technical Reference including IV&V-generated list of off-nominal conditions IV&V Test Design/Case analysis results and submitted/TBV Issues, and (if available) planned resolution
Prerequisites:	Validated Requirements
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	Integration analysis relies on numerous test analysis methods for interface verification and validation level testing. End-to-End Analysis method is also used to analyze the flowdown of design choices throughout the systems, interfaces, and the software units. Integration 1. M-10: Validate Test Plan by Inspection a. Assurance Objective: The test approach (i.e. verification methods: Analysis, Inspection, Test) is conclusive and appropriate b. TF: 4.1.1 (F), 4.1.2 (F), 4.1.3 (F), 4.1.4 (F), 4.1.5 (F), 4.3 (P), 4.8 (P) 2. M-35: Validate Test Procedures by Inspection and Traces to Requirements a. Assurance Objective: Planned Test Procedures completely exercise and provide conclusive evidence to fully verify the Requirement set b. TF: 4.2 (P), 4.6 (F) 3. M-25: Validate Test Cases by Inspection and Traces to Requirements c. Assurance Objective: Test Cases/Scripts correctly execute the planned test procedures, and produce conclusive evidence verifying the Requirement set d. TF: 4.2 (F), 4.5 (F), 4.7 (F) 4. (New) M-62: Verify Suitability of Developer's Test Environment by Simulation of Developer Test Operations Against Test Environment Validation Criteria

	e. Assurance Objective: Developer Test Environment (Simulators) are of appropriate fidelity for intended use. f. TF: 4.4 (F), 4.8 (F) 5. M-37: End-to-End Fault Management Verification through Database Development and Analysis a. Assurance Objective: Collection of Software/System behaves as expected under off-nominal conditions. b. Note: Scoped to the full system. c. Note: generally used to feed scenarios to Method M-68 d. TF: 3.1 (P), 3.2 (F), 3.4 (F), 3.5 (F), 4.1.1 (P), 4.1.2 (P), 4.1.3 (P), 4.1.4 (P), 4.1.5 (P), 5.1 (P), 5.2 (F), 5.3 (F), 5.4 (F), 5.5 (P), 5.6 (F), 6.1 (P), 6.3 (F), 6.5 (F), 6.6 (P), 7.8 (F)
Concerns:	None
Method Application Notes: None	
Required Tools:	None identified
Empirical Evidence:	- Engineering worksheets documenting results of tracing test procedures to the requirements and comments confirming the adequate verification of the requirements. - Risks or findings in technical reports documenting systemic concerns
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	May be performed simultaneously "Validate Test Cases by Inspection and Traces to Requirements"

Activity 28: Verify Software Integration via Test Case Inspection and Tracing to Requirements

Method:	M-25, Version 1.4 (Current Status: Approved)
Method Title:	Validate Test Cases by Inspection and Traces to Requirements
Method Synopsis	Method for manual evaluation of developer test case artifacts against requirements to confirm presence of defined inputs, expected results and evaluation criteria that comply with test plans and objectives and ensure that all requirements implemented are verified by the appropriate test case. Detects test case defects including inadequate coverage of in-scope requirements, logic errors, inadequate (or missing) defined inputs, results expectations, traceability, or evaluation criteria.
Subsystem/Entity	GND.OSS; OBS.SCE.SCS; OBS.SCE.EMTB; OBS.SCE.CTP.SC; OBS.ISIM.ISIM
Required Method Revisions (if any)	None
Technical Goal:	4.1 Ensure that the planned tests are sufficient to: 4.1.1 Ensure that the software correctly implements system and software requirements in an operational environment under nominal and off-nominal conditions. 4.1.2 Ensure that the complete, integrated system complies with its specified system requirements allocated to software and to validate whether the system meets its original objectives. 4.1.3 Ensure that the software meets all of the (in-scope) software requirements and is ready to be integrated with system hardware. 4.1.4 Ensure that the software correctly implements the software requirements and design as each software component (e.g., units or modules) is incrementally integrated with each other. 4.1.5 Ensure that the software components (e.g., units, source code modules) correctly implement software component requirements. 4.2 Ensure that valid relationships are defined between the Test Plans, Designs, Cases, and Procedures for test types and documents subject to IV&V test analysis. 4.5 Ensure that the Test Cases under analysis specify the correct test inputs, predicted results, and sets of execution conditions necessary to satisfy their

	intended test objectives (covering both nominal and off-nominal conditions). 4.7 Ensure that the Test Designs under analysis correctly specify the details of the test approach for the covered software feature or combination of software features and identify the associated tests.
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev O
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	OSS - Build 5 OSS - Build 6 Stored Command Sequences (SCS's) – B2.2 EMTB Build 2.2 SC FSW - Build 2.2 ISIM IC14.21+ ISIM IC15.0 ISIM IC15.0+
Inputs (includes Technical Reference):	Test Plan for the applicable level of test Test Design/Cases for the associated test plan Validated requirements to validate the Test Procedures against (relevant set of validated requirements will be iteration/instantiation specific). Technical Reference including IV&V-generated list of off-nominal conditions IV&V Test Design/Case analysis results and submitted/TBV Issues, and (if available) planned resolution
Prerequisites:	Validated Requirements
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	Integration analysis relies on numerous test analysis methods for interface verification and validation level testing. End-to-End Analysis method is also used to analyze the flowdown of design choices throughout the systems, interfaces, and the software units. Integration 1. M-10: Validate Test Plan by Inspection

	a. Assurance Objective: The test approach (i.e. verification methods: Analysis, Inspection, Test) is conclusive and appropriate b. TF: 4.1.1 (F), 4.1.2 (F), 4.1.3 (F), 4.1.4 (F), 4.1.5 (F), 4.3 (P), 4.8 (P) 2. M-35: Validate Test Procedures by Inspection and Traces to Requirements a. Assurance Objective: Planned Test Procedures completely exercise and provide conclusive evidence to fully verify the Requirement set b. TF: 4.2 (P), 4.6 (F) 3. M-25: Validate Test Cases by Inspection and Traces to Requirements c. Assurance Objective: Test Cases/Scripts correctly execute the planned test procedures, and produce conclusive evidence verifying the Requirement set d. TF: 4.2 (F), 4.5 (F), 4.7 (F) 4. (New) M-62: Verify Suitability of Developer's Test Environment by Simulation of Developer Test Operations Against Test Environment Validation Criteria e. Assurance Objective: Developer Test Environment (Simulators) are of appropriate fidelity for intended use. f. TF: 4.4 (F), 4.8 (F) 5. M-37: End-to-End Fault Management Verification through Database Development and Analysis a. Assurance Objective: Collection of Software/System behaves as expected under off-nominal conditions. b. Note: Scoped to the full system. c. Note: generally used to feed scenarios to Method M-68 d. TF: 3.1 (P), 3.2 (F), 3.4 (F), 3.5 (F), 4.1.1 (P), 4.1.2 (P), 4.1.3 (P), 4.1.4 (P), 4.1.5 (P), 5.1 (P), 5.2 (F), 5.3 (F), 5.4 (F), 5.5 (P), 5.6 (F), 6.1 (P), 6.3 (F), 6.5 (F), 6.6 (P), 7.8 (F)
Concerns:	None
Method Application Notes: None	
Required Tools:	None identified
Empirical Evidence:	- Engineering worksheets documenting results of tracing test procedures to the requirements and comments confirming the adequate verification of the requirements.

	- Risks or findings in technical reports documenting systemic concerns
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	May be performed simultaneously "Validate Test Cases by Inspection and Traces to Requirements"

Activity 29: Verify Software Integration via Simulation of Developer Test Operations Against Test Environment Validation

Method:	M-62, Version 1.0 (Current Status: Approved)
Method Title:	Verify Suitability of Developer's Test Environment by Simulation of Developer Test Operations Against Test Environment Validation Criteria
Method Synopsis	In some instances the IV&V Program will work with the development organization to replicate the development organization's V&V environment. IV&V analysts can utilize the provided environment to ensure the developer's test environment is sufficiently complete, correct, and accurate to perform the intended testing. This method benefits early testing and provides assurance in that the Developer's test environment is capable of executing the planned test suite.
Subsystem/Entity	OBS.SCE.EMTB; OBS.SCE.CTP.SC
Required Method Revisions (if any)	None
Technical Goal:	4.4 Ensure that any incorporated simulations from the developer into the IV&V Test Environment are sufficiently complete, correct, and accurate to perform the intended testing. Test Environments can be evaluated for correctness, accuracy, reliability, flexibility, testability, portability, reusability, and interoperability. 4.8 Ensure that the test environment is sufficiently complete, correct, and accurate to perform the intended testing.
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev 0
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	EMTB Build 2.2

Inputs (includes Technical Reference):	• Developer Test Artifacts (Test Plan, Test Scenarios, Test Procedures) • Requirements trace to test cases • Test Results (logs, output, etc.) • Test Configurations (where applicable)
Prerequisites:	Software under test (source and binary preferred), IV&V Test Environment (mirror of Developer Test Environment)
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	Integration analysis relies on numerous test analysis methods for interface verification and validation level testing. End-to-End Analysis method is also used to analyze the flowdown of design choices throughout the systems, interfaces, and the software units. Integration 1. M-10: Validate Test Plan by Inspection a. Assurance Objective: The test approach (i.e. verification methods: Analysis, Inspection, Test) is conclusive and appropriate b. TF: 4.1.1 (F), 4.1.2 (F), 4.1.3 (F), 4.1.4 (F), 4.1.5 (F), 4.3 (P), 4.8 (P) 2. M-35: Validate Test Procedures by Inspection and Traces to Requirements a. Assurance Objective: Planned Test Procedures completely exercise and provide conclusive evidence to fully verify the Requirement set b. TF: 4.2 (P), 4.6 (F) 3. M-25: Validate Test Cases by Inspection and Traces to Requirements c. Assurance Objective: Test Cases/Scripts correctly execute the planned test procedures, and produce conclusive evidence verifying the Requirement set d. TF: 4.2 (F), 4.5 (F), 4.7 (F) 4. (New) M-62: Verify Suitability of Developer's Test Environment by Simulation of Developer Test Operations Against Test Environment Validation Criteria e. Assurance Objective: Developer Test Environment (Simulators) are of appropriate fidelity for intended use. f. TF: 4.4 (F), 4.8 (F)

	5. M-37: End-to-End Fault Management Verification through Database Development and Analysis a. Assurance Objective: Collection of Software/System behaves as expected under off-nominal conditions. b. Note: Scoped to the full system. c. Note: generally used to feed scenarios to Method M-68 d. TF: 3.1 (P), 3.2 (F), 3.4 (F), 3.5 (F), 4.1.1 (P), 4.1.2 (P), 4.1.3 (P), 4.1.4 (P), 4.1.5 (P), 5.1 (P), 5.2 (F), 5.3 (F), 5.4 (F), 5.5 (P), 5.6 (F), 6.1 (P), 6.3 (F), 6.5 (F), 6.6 (P), 7.8 (F)
Concerns:	None
Method Application Notes: None	
Required Tools:	IV&V Test Environment that mirrors Developer Test Environment
Empirical Evidence:	• Test Procedures, Results and Logs provide evidence and coverage on the Development Organization's test environment.
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	None

Activity 30: Verify Software Integration via Construct of Fault Management Database

Method:	M-37, Version 1.1 (Current Status: Approved)
Method Title:	End-to-End Fault Management Verification through Database Development and Analysis
Method Synopsis	Method applies to the development of a System Fault Management Database that captures relationships and behaviors to aid in the analysis of Fault Management Systems in large distributed systems. A series of incremental databases are built, maintained, and integrated to derive a total system perspective. The database is an extension of the SMART/AWB traceability database where Fault Management Requirements are identiifed along with their drivers (i.e. parent requirements, Failure Mode and Effect Analysis, and Fault Tree Analysis). In addition, the database is further developed to include an "Event Network" that provides a quasi-dynamic (i.e. executable) abstraction of the system. Queries are used to produce scenarios where interactions are more complex, and potential resource conflicts are likely. Manual analysis is then used to determine the validity of such scenarios and uncover defects. For scenarios that are too complex or time-dependent to analyze manually, test procedures are developed for execution by either the developer or IV&V. Primarily, the method is intended to reduce the set of large or infinite scenarios for analysis/test to a reduced set that either has errors identified, or has a higher likelihood of error.
Subsystem/Entity	OBS.SCE.FM; OBS.SCE.CTP.SC
Required Method Revisions (if any)	None
Technical Goal:	1. Gain system level understanding 2. Uncover ambiguous or missing behaviors 3. Uncover conflicting or undesired behaviors 4. Uncover failure scenarios Ensure the system is capable of identifying, controlling, preventing, or properly responding to any credible fault scenario. Ensure every fault is properly controlled by a requirement.

	Uncover credible Failure Scenarios to target analysis and independent tests Maintain dependencies and conflicts between system entities to aid in change impact analysis 3.1 (P) Ensure that the system requirements are of high quality and are consistent with acquirer needs as they relate to the system's software. 3.2 Ensure that all (in-scope) parent requirements are represented in the appropriate child requirements and that the child requirements do not introduce capability that is not required. 3.4 Ensure that the requirements for software interfaces with hardware, user, operator, and other systems are adequate to meet the needs of the system with respect to expectations of its customer and users, operational environment, dependability and fault tolerance, and both functional and non-functional perspectives. 3.5 Ensure that software requirements meet the dependability and fault tolerance required by the system and provide the capability of controlling identified hazards and do not create hazardous conditions. 4.1 Ensure that the planned tests are sufficient to: 4.1.1 (P) Ensure that the software correctly implements system and software requirements in an operational environment under nominal and off-nominal conditions. 4.1.2 (P) Ensure that the complete, integrated system complies with its specified system requirements allocated to software and to validate whether the system meets its original objectives. 4.1.3 (P) Ensure that the software meets all of the (in-scope) software requirements and is ready to be integrated with system hardware. 4.1.4 (P) Ensure that the software correctly implements the software requirements and design as each software component (e.g., units or modules) is incrementally integrated with each other. 4.1.5 (P) Ensure that the software components (e.g., units, source code modules) correctly implement software component requirements.

	5.1 (P) Ensure that all (in-scope) requirements (e.g. SRS and IRS) are represented in the appropriate elements of the design (e.g. SDD and IDD) and that the design does not introduce capability that is not required. 5.2 Ensure that the design provides the required capability (meeting software architecture and software requirements), is able to reliably meet user needs, and is sufficiently stable to proceed with implementation. 5.3 Ensure that the proposed software architecture satisfies the needs of the system, and that it is a feasible solution (i.e. will successfully satisfy the needs of the system, while still being practical). 5.4 Ensure that the internal and external software interface designs are provided for all (in-scope) interfaces with hardware, user, operator, software, and other systems and that they provide sufficient detail to enable the development of software components that implement the interfaces. 5.5 (P) Ensure that complex algorithms have been correctly derived, provide the needed behavior under off nominal conditions and assumed conditions, and that the derivation approach is known and understood to support future maintenance. 5.6 Ensure that the design provides the dependability and fault tolerance required by the system and that the design is capable of controlling identified hazards and does not create hazardous conditions. 6.1 (P) Ensure that all (in-scope) elements of the design (e.g. SDD and IDD) are represented in the appropriate source code components and that the source code does not introduce capability that is not required. 6.3 Ensure that the source code that interfaces with hardware, user, operator, software, and other systems reliably provides the right services and data and receives data for internal use. 6.5 Ensure that the source code components provide the dependability and fault tolerance required by the system and that the source code is capable of controlling identified hazards and does not create hazardous conditions. 6.6 (P) Ensure that all (in-scope) requirements (e.g. SRS and IRS) are represented in the appropriate source

	code components and that the source code does not introduce capability that is not required. 7.8 Ensure that user documentation is consistent with the implementation and capable of communicating the use of user-accessible system functions.
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev O
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	FM Branch Terminal Analysis – Build 2.2 SC FSW - Build 2.2
Inputs (includes Technical Reference):	1. Failure Modes and Effects Analysis (FMEA) 2. Fault Management Algorithm Document (FMAD) 3. FSW Requirements 4. FSW Algorithms Requirement Documents 5. FSW Source Code 6. Fault Management Control Flows (or equivalent system and scenario designs) 7. Master Command And Telemetry Database (MCTDB) 8. Stored Command Sequence (SCS) Database (Relative Time or Absolute Time sequence of actions)
Prerequisites:	Database Engineer (Design, Maintain, Administer) Subject Matter Expert (Fault Management, Sub-systems)
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	Integration analysis relies on numerous test analysis methods for interface verification and validation level testing. End-to-End Analysis method is also used to analyze the flowdown of design choices throughout the systems, interfaces, and the software units. Integration 1. M-10: Validate Test Plan by Inspection a. Assurance Objective: The test approach (i.e. verification methods: Analysis, Inspection, Test) is conclusive and appropriate b. TF: 4.1.1 (F), 4.1.2 (F), 4.1.3 (F), 4.1.4 (F), 4.1.5 (F), 4.3 (P), 4.8 (P) 2. M-35: Validate Test Procedures by Inspection and Traces to Requirements

	a. Assurance Objective: Planned Test Procedures completely exercise and provide conclusive evidence to fully verify the Requirement set b. TF: 4.2 (P), 4.6 (F) 3. M-25: Validate Test Cases by Inspection and Traces to Requirements c. Assurance Objective: Test Cases/Scripts correctly execute the planned test procedures, and produce conclusive evidence verifying the Requirement set d. TF: 4.2 (F), 4.5 (F), 4.7 (F) 4. (New) M-62: Verify Suitability of Developer's Test Environment by Simulation of Developer Test Operations Against Test Environment Validation Criteria e. Assurance Objective: Developer Test Environment (Simulators) are of appropriate fidelity for intended use. f. TF: 4.4 (F), 4.8 (F) 5. M-37: End-to-End Fault Management Verification through Database Development and Analysis a. Assurance Objective: Collection of Software/System behaves as expected under off-nominal conditions. b. Note: Scoped to the full system. c. Note: generally used to feed scenarios to Method M-68 d. TF: 3.1 (P), 3.2 (F), 3.4 (F), 3.5 (F), 4.1.1 (P), 4.1.2 (P), 4.1.3 (P), 4.1.4 (P), 4.1.5 (P), 5.1 (P), 5.2 (F), 5.3 (F), 5.4 (F), 5.5 (P), 5.6 (F), 6.1 (P), 6.3 (F), 6.5 (F), 6.6 (P), 7.8 (F)
Concerns:	None
Method Application Notes: None	
Required Tools:	Database Tool (Access, MySQL, SQL Server, etc) (Database Conversion Tools) MS Excel (to capture results, ingest inputs, etc)
Empirical Evidence:	1. Requirement/Behavior deficiencies (incomplete, missing, conflicting) uncovered during database development and analysis 2. Database entry deficiencies (incomplete, missing, conflicting) uncovered during database development and analysis 3. Independent Test Scenarios and Results
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.

Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	JWST_WBS_Coverage_Rationale.xlsx (on ECM)

Activity 31: Validate Software Security Requirements via Quality Requirements Analysis (QRA)

Method:	M-2, Version 1.3 (Current Status: Approved)
Method Title:	Validate Requirements by Inspecting Against Quality Criteria and System/Software Background Artifacts
Method Synopsis	Method for tool-supported manual inspection of a set of requirements to assess and document the degree to which they individually and collectively exhibit desired quality attributes (Unambiguous, Verifiable, Consistent, Correct, Complete, Design Independent, Feasible). Use documents that inform the validation target to insure that the requirements are complete and correct.
Subsystem/Entity	GND.FOS
Required Method Revisions (if any)	None
Technical Goal:	3.1 Ensure that the system requirements are of high quality and are consistent with acquirer needs as they relate to the system's software. (IVV 09-1 Rev N) 3.3 Ensure that the software requirements are of high quality and adequately meet the needs of the system with respect to expectations of its customer and users, operational environment, and both functional and non-functional perspectives. (IVV 09-1 Rev N) 3.4 Ensure that the requirements for software interfaces with hardware, user, operator, and other systems are adequate to meet the needs of the system with respect to expectations of its customer and users, operational environment, dependability and fault tolerance, and both functional and non-functional perspectives. (IVV 09-1 Rev N)
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev O
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	FOS - Build 5 FOS - Build 6
Inputs (includes Technical Reference):	Needs, Goals and Objectives document, opsCon, trades, higher level requirements, and any other additional background materials to understand the requirements to be assessed
Prerequisites:	none
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.

Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	JWST Security analysis will rely on the standard approach to Requirements, Design, and Test analysis to ensure that requirements are specified with regards to security aspects and standards and that the design is capable of achieving the necessary security features. Test analysis is also performed to ensure that software and system level tests are planned which will exercise the security features. Code analysis as it pertains specifically to Security is not explicitly performed or planned.
Concerns:	None
Method Application Notes: None	
Required Tools:	Engineering worksheets (or database) document the assessment of the quality attributes
Empirical Evidence:	Engineering worksheets (or database) documenting the results of the assessment of the quality attributes for each requirement and conclusions about the completeness and correctness of the set(s) of analyzed requirements. Evidence must include an indication that each requirement was examined for every qualitative attribute (i.e. correctness, completeness, etc.) and the version of the requirements that was assessed.
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	While this method's effectiveness is largely a function of the analyst(s) performing it, it can nevertheless be applied in a relatively short time period to provide valuable feedback to a mission project Other methods may need to be applied to garner additional rigor and confidence in the correctness, completeness, and overall consistency of the requirements

Activity 32: Validate Software Security Requirements via Bidirectional Tracing

Method:	M-3, Version 1.3 (Current Status: Approved)
Method Title:	Validate Requirements by Inspecting Bidirectional Traces
Method Synopsis	Method for tool-supported manual inspection of a set of requirements to assess and document the degree to which they adequately specify a logical decomposition of the parent requirements, and any functional allocations identified by the developer. This method addresses the integrity of the requirements structure, and identifies faults in correctness, completeness, consistency, and bi-directional tracing of parent to child requirements.
Subsystem/Entity	GND.FOS
Required Method Revisions (if any)	None
Technical Goal:	Assess the quality of the requirements (set) and the degree to which they adequately specify a logical decomposition of the parent requirements 3.1: Ensure that the system requirements are of high quality and are consistent with acquirer needs as they relate to the system's software. (Partial) 3.2: Ensure that all (in-scope) parent requirements are represented in the appropriate child requirements and that the child requirements do not introduce capability that is not required. 3.3: Ensure that the software requirements are of high quality and adequately meet the needs of the system with respect to expectations of its customer and users, operational environment, and both functional and non-functional perspectives. (Partial)
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev O
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution
Target Artifacts:	FOS - Build 5 FOS - Build 6
Inputs (includes Technical Reference):	1. Requirement traces developed by the Mission Project 2. Additional Reference Artifacts to understand the requirements to be assessed, including IV&V Project Technical Reference

	3. Capabilities defined to level of analysis (PBRA, RBA) [scope]
Prerequisites:	Requirements and developer provided traces loaded into traceability tool (spreadsheet / analysis tool)
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	JWST Security analysis will rely on the standard approach to Requirements, Design, and Test analysis to ensure that requirements are specified with regards to security aspects and standards and that the design is capable of achieving the necessary security features. Test analysis is also performed to ensure that software and system level tests are planned which will exercise the security features. Code analysis as it pertains specifically to Security is not explicitly performed or planned
Concerns:	None
Method Application Notes: None	
Required Tools:	Engineering worksheets or analysis tool to document results of tracing analysis
Empirical Evidence:	Completeness/correctness/consistency status in engineering worksheets (or analysis tools) for each requirement, list of orphans, list of childless parents
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	None

Activity 33: Validate Software Security Design via Inspection of Interface Requirements

Method:	M-41, Version 1.1 (Current Status: Approved)
Method Title:	Verify Software Interface Design by Inspection Against Interface Requirements
Method Synopsis	Method supports manual evaluation of the integrity of the software requirements to interface design transformation, and detects defects in hardware/user/operator/software/other systems interface coverage completeness/correctness/accuracy and capability for implementation in software.
Subsystem/Entity	GND.FOS
Required Method Revisions (if any)	None
Technical Goal:	5.1 Ensure that all (in-scope) requirements (e.g. SRS and IRS) are represented in the appropriate elements of the design (e.g. SDD and IDD) and that the design does not introduce capability that is not required. 5.4 Provide Evidence that the assurance goals related to the internal and external software interface designs are adequately achieved for all interfaces with hardware, user, operator, software, and other systems and that they provide sufficient detail to enable the development of software components that implement the interfaces.
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev O
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	FOS - Build 5 FOS - Build 6
Inputs (includes Technical Reference):	List of validated interface requirements and identified issues and risks, Interface Control Document (ICD), Interface Requirements Document (IRD), Intended assurance goals/statements, Identified evidence needed to support intended assurance goals/statements, Technical Reference (applicable to interface), Adverse conditions, System Capabilities list and description.
Prerequisites:	Validation of the interface requirements
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.

Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	JWST Security analysis will rely on the standard approach to Requirements, Design, and Test analysis to ensure that requirements are specified with regards to security aspects and standards and that the design is capable of achieving the necessary security features. Test analysis is also performed to ensure that software and system level tests are planned which will exercise the security features. Code analysis as it pertains specifically to Security is not explicitly performed or planned.
Concerns:	None
Method Application Notes: None	
Required Tools:	Excel Worksheets (or other data documentation system), ORBIT
Empirical Evidence:	• Engineering worksheets, databases, etc. that contain the results and comments of the requirements to design trace and the design to requirements trace, used to support the intended assurance goals/statements. • TIMs • Documented risks and findings
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	None

Activity 34: Validate Software Security Test Design via Test Plan Inspection

Method:	M-10, Version 1.2 (Current Status: Approved)
Method Title:	Validate Test Plan by Inspection
Method Synopsis	Method performs manual inspection of a test plan artifact to detect defects in test plan integrity, compliance with applicable test plan standards and key testing principles, capability for requirements and behaviors to be verified under nominal and adverse conditions, documentation of limitations in test plan verification capability, test environment fidelity appropriateness, test operational procedure integrity, test scheduling and risk assessments, and planned regression testing.
Subsystem/Entity	GND.FOS
Required Method Revisions (if any)	None
Technical Goal:	4.1 Ensure that the planned tests are sufficient to: 4.1.1 Ensure that the software correctly implements system and software requirements in an operational environment under nominal and off-nominal conditions. 4.1.2 Ensure that the complete, integrated system complies with its specified system requirements allocated to software and to validate whether the system meets its original objectives. 4.1.3 Ensure that the software meets all of the software requirements and is ready to be integrated with system hardware. 4.1.4 Ensure that the software correctly implements the software requirements and design as each software component (e.g., units or modules) is incrementally integrated with each other. 4.1.5 Ensure that the software components (e.g., units, source code modules) correctly implement software component requirements. 4.3 Ensure that the planned regression testing to be performed when changes are made to any previously examined software products is sufficient to identify any unintended side effects or impacts of the change on other aspects of the system 4.8 Ensure that the test environment is sufficiently complete, correct, and accurate to perform the intended testing.
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev 0

Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	FOS - Build 5 FOS - Build 6
Inputs (includes Technical Reference):	In Scope developer requirements
Prerequisites:	Any previous internal or external activity needed before this method can be applied
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	JWST Security analysis will rely on the standard approach to Requirements, Design, and Test analysis to ensure that requirements are specified with regards to security aspects and standards and that the design is capable of achieving the necessary security features. Test analysis is also performed to ensure that software and system level tests are planned which will exercise the security features. Code analysis as it pertains specifically to Security is not explicitly performed or planned.
Concerns:	None
Method Application Notes: None	
Required Tools:	None identified
Empirical Evidence:	• Engineering worksheets documenting that test plans, scenarios, etc. will provide adequate verification of the requirements. • Risks or findings in technical reports documenting systemic concerns
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule
Other:	NASA Missions may use different terminology in defining their test content. Each IV&V Project needs to understand how its Mission's test content is being developed and plan accordingly. Also, because test cases, procedures, and designs may be developed iteratively and at multiple levels, IV&V task iteration may be necessary. (per IVV 09-1 Rev M)

Activity 35: Validate Software Security Test Design via Test Procedure Inspection and Tracing to Requirements

Method:	M-35, Version 1.1 (Current Status: Approved)
Method Title:	Validate Test Procedures by Inspection and Traces to Requirements
Method Synopsis	Method for manual evaluation of developer test procedures against requirements and test plan criteria to confirm that they are compliant with applicable project and NASA standards, complete and sufficient to create test cases that will fully verify the requirements of interest. Detects test procedure defects including inadequate coverage of in-scope requirements, inadequately defined inputs, and insufficient evaluation criteria.
Subsystem/Entity	GND.FOS
Required Method Revisions (if any)	None
Technical Goal:	4.6 Ensure that the Test Procedures under analysis specify the correct sequence of actions necessary for the execution of the tests to satisfy their intended test objectives. • Verify that the Test Procedures under analysis comply with project defined test document purpose, format, and content. 4.2 Ensure that valid relationships are defined between the Test Plans, Designs, Cases, and Procedures for test types and documents subject to IV&V test analysis. • Validate that the Test Procedures under analysis satisfy the criteria in the associated Test Plan, Test Design, and Test Cases.
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev O
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	FOS - Build 5 FOS - Build 6
Inputs (includes Technical Reference):	Test Plan for the applicable level of test Test Design/Cases for the associated test plan Validated requirements to validate the Test Procedures against (relevant set of validated requirements will be iteration/instantiation specific). Technical Reference including IV&V-generated list of off-nominal conditions

	IV&V Test Design/Case analysis results and submitted/TBV Issues, and (if available) planned resolution
Prerequisites:	Validated Requirements
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	JWST Security analysis will rely on the standard approach to Requirements, Design, and Test analysis to ensure that requirements are specified with regards to security aspects and standards and that the design is capable of achieving the necessary security features. Test analysis is also performed to ensure that software and system level tests are planned which will exercise the security features. Code analysis as it pertains specifically to Security is not explicitly performed or planned.
Concerns:	None
Method Application Notes: None	
Required Tools:	None identified
Empirical Evidence:	- Engineering worksheets documenting results of tracing test procedures to the requirements and comments confirming the adequate verification of the requirements. - Risks or findings in technical reports documenting systemic concerns
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	May be performed simultaneously "Validate Test Cases by Inspection and Traces to Requirements"

Activity 36: Validate Software Behaviors Dynamically via Independent Test by Executing Simulations/Models

Method:	M-40, Version 1.1 (Current Status: Approved)
Method Title:	Validate System Behaviors Dynamically by Executing Simulations/Models
Method Synopsis	Method applies MATLAB/Simulink (or similar continuous/discrete event modeling tool) to assist analysts in gaining system level understanding of component behaviors, uncovering ambiguous or missing behaviors, uncovering conflicting or undesired behaviors, and uncovering failure scenarios.
Subsystem/Entity	OBS.SCE.SCS; OBS.SCE.CTP.SC
Required Method Revisions (if any)	None
Technical Goal:	1. Gain system level understanding 2. Uncover ambiguous or missing behaviors 3. Uncover conflicting or undesired behaviors 4. Uncover failure scenarios 2.2 (P) Ensure that the system architecture contains the necessary computing related items (subsystems, components, etc.) to carry out the mission of the system and satisfy user needs and operational scenarios or use cases. 3.1 (P) Ensure that the system requirements are of high quality and are consistent with acquirer needs as they relate to the system's software. 3.3 (P) Ensure that the software requirements are of high quality and adequately meet the needs of the system with respect to expectations of its customer and users, operational environment, and both functional and non-functional perspectives. 3.4 (P) Ensure that the requirements for software interfaces with hardware, user, operator, and other systems are adequate to meet the needs of the system with respect to expectations of its customer and users, operational environment, reliability and fault tolerance, and both functional and non-functional perspectives. 3.5 (P) Ensure that software requirements meet the reliability and fault tolerance required by the system and provide the capability of controlling identified hazards and do not create hazardous conditions.

	5.2 (P) Ensure that the design provides the required capability (meeting software architecture and software requirements), is able to reliably meet user needs, and is sufficiently stable to proceed with implementation. 5.3 (P) Ensure that the proposed software architecture satisfies the needs of the system, and that it is a feasible solution (i.e. will successfully satisfy the needs of the system, while still being practical). 5.4 (P) Ensure that the internal and external software interface designs are provided for all (in-scope) interfaces with hardware, user, operator, software, and other systems and that they provide sufficient detail to enable the development of software components that implement the interfaces. 5.5 (P) Ensure that complex algorithms have been correctly derived, provide the needed behavior under off nominal conditions and assumed conditions, and that the derivation approach is known and understood to support future maintenance. 5.6 (P) Ensure that the design provides the dependability and fault tolerance required by the system and that the design is capable of controlling identified hazards and does not create hazardous conditions. 6.2 (P) Ensure that the source code components can reliably perform required capabilities under nominal and off-nominal conditions, perform no undesired behaviors, and that the documentation (both embedded and stand-alone) can facilitate code maintenance. 6.3 (P) Ensure that the source code that interfaces with hardware, user, operator, software, and other systems reliably provides the right services and data and receives data for internal use. 6.5 (P) Ensure that the source code components provide the dependability and fault tolerance required by the system and that the source code is capable of controlling identified hazards and does not create hazardous conditions.
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev O
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.

Target Artifacts:	Stored Command Sequences (SCS's) – B2.2 SC FSW - Build 2.2 SC FSW - Build 3.0
Inputs (includes Technical Reference):	1. System Level Specifications 2. System or Software Artifacts to be analyzed 3. Detailed System Schematic (optional - increases model fidelity) 4. As-Run Test Results (optional - increases model fidelity) 5. Source Code (optional - increases model cohesion with actual code)
Prerequisites:	Subject matter expert is available or system understanding is sufficient for modeling.
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	Independent Testing utilizes the JIST to perform scenario based testing as well as execution of developer's verification and validation procedures. Testing is prioritized based on risk assessment of the scenarios and the capability of the JIST.
Concerns:	None
Method Application Notes: None	
Required Tools:	MATLAB/Simulink (or similar continuous/discrete event modeling tool)
Empirical Evidence:	Evidence Based Assurance covered by the following empirical evidence: 1. Behavior deficiencies (ambiguous, incomplete, missing, conflicting) uncovered during modeling and analysis 2. Simulation inputs and outputs (describing scenarios/paths of execution)
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	1. Additional work can be performed to implement the developer Source Code (or portions of the developer code) in place of Software Model (or

	individual model components) to be used in interactive SIM / Test bed environment. 2. This approach is specific to a single component or subsystem. The approach can be performed iteratively to produce and incorporate multiple components.

Activity 37: Validate Software Key Capabilities via Independent Test of High Risk Scenarios

Method:	M-68, Version 1.0 (Current Status: Approved)
Method Title:	Validate Key Capabilities via Dynamic Testing against High Risk Scenarios to Reduce Risk of Operations
Method Synopsis	Develop high risk scenarios, especially those that include off-nominal and fault scenarios and those that require cooperation of multiple CSCIs, CSCs, and otherwise cross integration boundaries. Provide assurance that capabilities needed to correctly operate as expected are complete and correct by executing the scenarios in a validated test bed.
Subsystem/Entity	OBS.SCE.SCS; OBS.SCE.CTP.SC
Required Method Revisions (if any)	None
Technical Goal:	6.2 Ensure that the source code components can reliably perform required capabilities under nominal and off-nominal conditions, perform no undesired behaviors, and that the documentation (both embedded and stand-alone) can facilitate code maintenance. 6.3 Ensure that the source code that interfaces with hardware, user, operator, software, and other systems reliably provides the right services and data and receives data for internal use. 6.5. Utilize the test environment and fault conditions to ensure that source code can identify and handle (if appropriate) hazardous conditions. 6.6 Ensure that all (in-scope) requirements (e.g. SRS and IRS) are represented in the appropriate source code components and that the source code does not introduce capability that is not required.
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev 0
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	Stored Command Sequences (SCS's) – B2.2 SC FSW - Build 2.2

	SC FSW - Build 3.0
Inputs (includes Technical Reference):	Concept of Operations (CONOPS) Document (developer), Fault Management artifacts (developer), test plans (developer), test cases (developer), test procedures (developer), test scripts (developer), test results (developer), flight software source code (developer), Design Documents, and TIMs (IV&V, especially Project Accepts Risk), list of risks to be explored (IV&V Technical Reference). Optional material may also include: - Hazard Analysis documentation (if not already part of the IV&V Technical Reference), - Design Models as provided by the developer, - Adverse condition list (part of the IV&V Technical Reference)
Prerequisites:	Some System Integration testing has been performed by the developer prior to IV&V performing risk reduction analysis.
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	Independent Testing utilizes the JIST to perform scenario based testing as well as execution of developer's verification and validation procedures. Testing is prioritized based on risk assessment of the scenarios and the capability of the JIST.
Concerns:	None
Method Application Notes: None	
Required Tools:	Validated ITC Test Bed environment. Supporting tools can be used as needed.
Empirical Evidence:	IV&V Test Cases, IV&V Test Procedures, IV&V Test Scripts, and IV&V Test Results
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	Not all Risk Reduction Scenarios may be able to be converted to an executable test suitable for the test bed. In those cases, alternative Methods may be

	required to fully assure that the risks of interest are mitigated.

Activity 38: Validate Software Interfaces via Independent Test by Simulated Dynamic Testing

Method:	M-59, Version 1.0 (Current Status: Approved)
Method Title:	Verify Interface Implementation in Software by Simulated Dynamic Testing to Demonstrate Successful Software Component Integration
Method Synopsis	Method supports analysis of interface implementation by exercising interface components in a test environment engineered to verify/validate that software components integrate properly with hardware elements (physical or simulated) of the system under study. Defects discovered by this Method include code flaws, mismatches between expected hardware function and software implementation of the function, and timing or other performance constraints.
Subsystem/Entity	OBS.SCE.SCS; OBS.SCE.CTP.SC
Required Method Revisions (if any)	None
Technical Goal:	6.3. Perform interface analysis using the test environment on the available simulated or actual interfaces that are provided via the test environment
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev O
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	Stored Command Sequences (SCS's) – B2.2 SC FSW - Build 2.2 SC FSW - Build 3.0
Inputs (includes Technical Reference):	• Test Artifacts (Test Plan, Test Scenarios, Test Procedures, Test Cases) • Requirements • Technical Reference • Interface Description Language (IDL) • Interface Control Documents (ICDs) • Test Results (logs, output, etc.) • User's Manuals (if applicable)
Prerequisites:	Software under test (source and binary preferred), Validated IV&V Test Environment, any applicable simulators from Development Organization
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.

Rationale for Approach:	Independent Testing utilizes the JIST to perform scenario based testing as well as execution of developer's verification and validation procedures. Testing is prioritized based on risk assessment of the scenarios and the capability of the JIST.
Concerns:	None
Method Application Notes: None	
Required Tools:	Mechanism to execute software via validated simulation and/or test environment - called the IV&V Test Environment
Empirical Evidence:	Test Procedures, Test Results, Log Files, and Issues. This method provides measurable software assurance that the interfaces meet the requirements and is operational.
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	None

Activity 39: Verify Software Behavior for Off-Nominal Conditions via Independent Test

Method:	M-14, Version 1.2 (Current Status: Approved)
Method Title:	Verify Software Behavior for Off-Nominal Conditions using Independent Testing
Method Synopsis	This method provides an approach for testing software behavior for IV&V Q2 (software will not do what it is not supposed to do) and Q3 (software behaves adequately under adverse conditions). Test scripts are independently created and executed within the IV&V Test environment.
Subsystem/Entity	OBS.SCE.SCS; OBS.SCE.CTP.SC
Required Method Revisions (if any)	None
Technical Goal:	6.2 Ensure that the source code components can reliably perform required capabilities under nominal and off-nominal conditions, perform no undesired behaviors, and that the documentation (both embedded and stand-alone) can facilitate code maintenance. (Partial) 6.3 Ensure that the source code that interfaces with hardware, user, operator, software, and other systems reliably provides the right services and data and receives data for internal use. (Partial) 6.5 Ensure that the source code components provide the dependability and fault tolerance required by the system and that the source code is capable of controlling identified hazards and does not create hazardous conditions. (Partial) Notes: • The method is designed to verify the TF goal 6.2 partially, i.e., it verifies that the software can perform reliably under off-nominal conditions (IV&V Q3) and does not produce undesired behavior (IV&V Q2) • Depending on the software behaviors tested, this method provides partial coverage of TF 6.3 and 6.5. If interfaces are involved, then TF 6.3 could receive coverage and to a limited extent testing the Q2/Q3 aspects could ensure the software implements proper fault tolerance (TF 6.5).
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev 0

Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	Stored Command Sequences (SCS's) – B2.2 SC FSW - Build 2.2 SC FSW - Build 3.0
Inputs (includes Technical Reference):	• Technical Reference • Developer Test Artifacts (Test Plan, Test Scenarios, Test Procedures) • Requirements • Requirements Traceability Matrix • Interface Control Documents (ICDs) • Source Code
Prerequisites:	Software under test (source preferred but binary required), validated IV&V Test Environment
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	Independent Testing utilizes the JIST to perform scenario based testing as well as execution of developer's verification and validation procedures. Testing is prioritized based on risk assessment of the scenarios and the capability of the JIST.
Concerns:	None
Method Application Notes: None	
Required Tools:	Mechanism to execute software via validated simulation and/or test environment
Empirical Evidence:	• Analysis of IV&V's Test Results/Log Files captured in worksheet or database which objectively shows the software will operate correctly under the off-nominal conditions
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	None

Activity 40: Verify Software Capabilities via Independent Test of Operational Scenarios

Method:	M-60, Version 1.0 (Current Status: Approved)
Method Title:	Verify Software Capabilities through Independent Testing of Operational Scenarios
Method Synopsis	Having test environments available aid in determining the operational readiness of software. If the test environment has the proper fidelity, operational day-in-the-life scenarios can be executed which can increase confidence in the operational readiness.
Subsystem/Entity	OBS.SCE.SCS; OBS.SCE.CTP.SC
Required Method Revisions (if any)	None
Technical Goal:	6.2 Ensure that the source code components can reliably perform required capabilities under nominal and off-nominal conditions, perform no undesired behaviors, and that the documentation (both embedded and stand-alone) can facilitate code maintenance. 7.2 Ensure deployment readiness and operational readiness of the software.
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev 0
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	Stored Command Sequences (SCS's) – B2.2 SC FSW - Build 2.2 SC FSW - Build 3.0
Inputs (includes Technical Reference):	• ConOPS • Mission Ops Plan • Command and Telemetry Databases • User's Guides • System level test artifacts (Test Plan, Test Scenarios, Test Procedures) -- if available, system level tests generally address "operational" type scenarios. • Performance requirements • Test Results (logs, output, etc.) • Technical Reference
Prerequisites:	Software under test (source and binary preferred), IV&V Test Environment, any applicable simulators from Development Organization
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.

Rationale for Approach:	Independent Testing utilizes the JIST to perform scenario based testing as well as execution of developer's verification and validation procedures. Testing is prioritized based on risk assessment of the scenarios and the capability of the JIST.
Concerns:	None
Method Application Notes: None	
Required Tools:	Mechanism to execute software via validated simulation and/or test environment - called the IV&V Test Environment.
Empirical Evidence:	• Analysis of IV&V's Test Results/Log Files captured in worksheet or database which objectively shows the software will operate as expected. This can be compared to the mission documentation and/or IV&V Technical Reference to ensure the software is operating as expected. • This method provides measurable software assurance that the software is ready for operations.
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.
Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	None

Activity 41: Verify Software Performance Requirements via Independent Test

Method:	M-64, Version 1.0 (Current Status: Approved)
Method Title:	Verify Performance Requirements Implementation via Simulated Dynamic Testing to Stress Software Boundaries and Limitations
Method Synopsis	This method uses performance requirements to generate test cases (aka scenarios) that will stress the system and its interfaces under test by exercising the boundaries and limits described by the performance requirements. Performance requirements are any requirements which are described in terms of quantity, quality, coverage, timeliness or readiness. These requirements are not limited to those listed under "Performance Requirements" in requirements specifications for the system under test. Additionally, stress tests should exercise the system by creating "stressful" conditions by exceeding operating constraints and design margins such as: • CPU load is greater than or equal to 90% • CPU load is maximized at 100% if possible • Use maximum I/O data rates • Maximum data bus usage • Utilize all available memory • Overflow buffers/queues - Note: Overflow buffers may be more of a 'consequence' than a condition. If a buffer overflow occurs then this is something due to improper coding or because of memory being completely utilized, and sometimes the results can be random. Potential consequences of the above “stressful” conditions are: • Dropped commands • Tasks not running as scheduled • Tasks not prioritized correctly • Unexpected processor reset • Unexpected rejection of commands • Degraded system performance • Undesired side effects in one software area or task • Corrupt data/memory • Unexpected telemetry being received/Data being received on the wrong communications channel • Hazardous conditions, actions, or undesired events Stress testing should demonstrate the robustness and recoverability of the system under test. The execution

	of stress tests should take place upon or near the final release of software, after IV&V has been performed on all phases of the system. This will ensure that the tests are being executed on the most mature, complete and error free revision of the software. The goal of stress testing is to identify errors in software that can remain hidden from standard or traditional unit and acceptance testing, including any fault conditions that can cause hazards. It is not until the system is subjected to out-of-the-ordinary conditions that errors arise or system performance degrades below operational levels. By analyzing the results of stress testing the IV&V team will be able to identify the root cause in these undesired software system behaviors.
Subsystem/Entity	OBS.SCE.SCS; OBS.SCE.CTP.SC
Required Method Revisions (if any)	None
Technical Goal:	4.0 Ensure that the collection of test related content will serve as a sufficient means to verify and validate that the implementation meets the requirements and operational need under nominal and off-nominal conditions. (Partial) 4.1.1 Ensure that the software correctly implements system and software requirements in an operational environment under nominal and off-nominal conditions. (Partial) 4.1.2 Ensure that the complete, integrated system complies with its specified system requirements allocated to software and to validate whether the system meets its original objectives. 4.1.3 Ensure that the software meets all of the (in-scope) software requirements and is ready to be integrated with system hardware. (Partial) 4.1.4 Ensure that the software correctly implements the software requirements and design as each software component (e.g., units or modules) is incrementally integrated with each other. (Partial) 4.1.5 Ensure that the software components (e.g., units, source code modules) correctly implement software component requirements. (Partial) 4.5. Ensure developer and IV&V test cases provide correct inputs, predicted results, and sets of execution conditions (in developer and IV&V test environments) satisfy the intended test objectives. (Full)

	4.6. Execute developer and IV&V test procedures/scripts to ensure that the correct sequence of actions occurs to satisfy test objectives. (Full) 4.7. Validate test design and associated tests via setting up and executing tests in a similar or exact environment as the developer. (Full) 4.8. For cases where developer simulation and test environments are replicated, the IV&V analyst can use the in-house test environment to perform IV&V against, to satisfy 4.7. (Full) 6.0. Ensure that the software system correctly and completely implements the requirements and meets the operational need under nominal and off-nominal conditions by exhausting the system via executing developer and independent test procedures. (Partial) 6.1. Utilize design documents and test environment to ensure that design elements are represented within the source code. (Partial) 6.2. Utilize requirements, RTTMs, and test environment to ensure that the required capabilities are represented within the source code. (Partial) 6.3. Perform interface analysis using the test environment on the available simulated or actual interfaces that are provided via the test environment. (Partial) 6.4. Compare expected independent test results to actual test results and impacts of discrepancies are understood. (Full) 6.5. Utilize the test environment and fault conditions to ensure that source code can identify and handle (if appropriate) hazardous conditions. (Partial) 6.6. Execute desired test procedures, scripts to ensure that all requirements are represented in the source code and that the source code does not introduce capability that is not required. Note: The activity of tracing requirements to source code may not be performed as it is currently under consideration to be eliminated. (Partial)
WBS Coverage:	IVV 09-1 IV&V Technical Framework, Rev O
Scope:	Scope will be described in the applicable Analysis Activity in JIRA prior to execution.
Target Artifacts:	Stored Command Sequences (SCS's) – B2.2 SC FSW - Build 2.2 SC FSW - Build 3.0

Inputs (includes Technical Reference):	In-scope performance requirements (developer), design documents (developer), test plans (developer and IV&V), test procedures/scripts (developer and IV&V), test Results (developer and IV&V), source code (developer), design models (developer, if available), TIMs (IV&V), target assurance statements, and the Technical Reference. Optional material may also include: - Hazard Analysis documentation (if not already part of the IV&V Technical Reference), - Design Models as provided by the developer, - Adverse condition list (part of the IV&V Technical Reference)
Prerequisites:	This method should be executed on mature software systems near the end of the software development lifecycle, after IV&V has been performed on all phases of the system including implementation and test. Additionally, the test bed should be validated by executing a subset of developer test procedures/scripts and achieving identical results. Any discrepancies between IV&V test results and developer test results during validation must be resolved prior to the beginning of independent testing.
Success Criteria:	Success Criteria will be described in the applicable Analysis Activity in JIRA prior to execution.
Activity Assumptions:	Assumptions will be described in the applicable Analysis Activity in JIRA prior to execution.
Rationale for Approach:	Independent Testing utilizes the JIST to perform scenario based testing as well as execution of developer's verification and validation procedures. Testing is prioritized based on risk assessment of the scenarios and the capability of the JIST.
Concerns:	None
Method Application Notes: None	
Required Tools:	Validated ITC Test Bed environment. Supporting tools can be used as needed.
Empirical Evidence:	IV&V Test Plan, IV&V Test Cases, IV&V Test Procedures, IV&V Test Scripts, gaps identified in developer testing, and IV&V Stress Test Results
Output (include updates to Project Technical Reference):	Outputs will be described in the applicable Analysis Activity in JIRA prior to execution.

Basis of Estimate:	Basis of Estimate will be described in the applicable Analysis Activity in JIRA prior to execution and will be captured in the IV&V Schedule.
Other:	None

Appendix A: Acronyms-Example

ACS	Attitude Control Subsystem
ADU	Actuator Deployment Unit
ARD	Algorithm Requirements Document
CC	Cryo Cooler
CCE	Cryocooler Control Electronics
C&DH	Command and Data Handling
CDR	Critical Design Review
CSCI	Computer Software Configuration Item
CSO	Chief Safety and Mission Assurance Officer
DEU	Deployment Electronics Unit
DI	Dictionary Interface
DMS	Data Management System
DS	Detector Subsystem
DSS	Detector Subsystem Software
EPS	Electrical Power Subsystem
FGS	Fine Guidance Sensor
FMAD	Fault Management Algorithms Document
FMEA	Failure Modes Effect Analysis
FMECA	Failure Mode Effects and Critical Analysis
FOS	Flight Operations System
FSW	Flight Software
FTA	Fault Tree Analysis
FY	Fiscal Year
GS	Ground Segment
IBA	IV&V Board of Advisors
ICD	Interface Control Document
IPEP	IV&V Project Execution Plan
IRCD	Interface Requirements Control Document
IRD	Interface Requirements Document
ISIM	Integrated Science Instrument Module
ISS	International Space Station
ITC	Independent Test Capability
IV&V	Independent Verification and Validation
JPIM	JWST Payload Interface Module
JWST	James Webb Space Telescope
MAP	Microwave Anisotropy Probe
MCFSWA	MIRI Cooler Flight Software Application
MCS	Mirror Control Software
MDL	Mission Directorate Lead

MIRI	Mid-Infrared Instrument
MIRI OS	MIRI Optical System
MMS	Mirror Management Software
MRR	Mission Readiness Review
MSR	Monthly Status Review
MSS	Micro Shutter Subsystem
NASA	National Aeronautics and Space Administration
NC	Near-Infrared Camera
NGAS	Northrop Grumman Aerospace Systems
NIRCam	Near-Infrared Camera
NIRSpec	Near-Infrared Spectrograph
NIRISS	Near-Infrared Imager and Slitless Spectrograph
NLT	No Later Than
NODIS	NASA Online Directives Information System
NS	Near-Infrared Spectrograph
OBFM	On Board Fault Management
OSS	Operations Script Subsystem
OTE	Optical Telescope Element
PBRA	Portfolio Based Risk Assessment
PHA	Preliminary Hazard Analysis
PDR	Preliminary Design Review
PL	Project Lead
PM	Project Manager
PMC	Program Management Council
POC	Point of Contact
PPP	Project Protection Plan
PPS	Proposal Planning System
PRDS	Project Reference Database System
PTS	Project Threat Summary
RBA	Risk Based Assessment
RMO	Resource Management Office
RMS	Risk Management System
SBU	Sensitive But Unclassified
S/C	Spacecraft
SC	Spacecraft
SCE	Spacecraft Element
SCS	Stored Command Sequence
SDO	Solar Dynamics Observatory
SMA	Safety and Mission Assurance
SMSR	Safety and Mission Success Review
S&OC	Science and Operations Center
SQA	Software Quality Assurance
SRM	System Reference Model
SRS	Software Requirements Specification
SSP	System Security Plan

SUROM	Start Up Read Only Memory
TBD	To Be Determined
TCS	Thermal Control Subsystem
TF	Technical Framework
TIM	Technical Issue Memorandum
TQ&E	Technical Quality and Excellence
TS&R	Technical Scope and Rigor
V&V	Verification and Validation
WAS	WFS&C Analysis Software
WBS	Work Breakdown Structure
WEx	WFS&C Executive
WFS&C	Wave Front Sensing and Control
WSS	WFS&C Software Sub-System