National Aeronautics and Space Administration

Independent Verification and Validation (IV&V)

IV&V Technical Scope and Rigor

Multi-Purpose Crew Vehicle EM-1 and Subsequent Flights

Version 1.8

07/29/2015

NASA Independent Verification and Validation Facility

100 University Drive, Fairmont WV 26554

Revision History

Date	Contributor	Version	Notes
06/24/2013	Various	0.1	Initial draft for EM-1 and EM-2 flights
09/30/2013	Various	1.1	RBA update and management recommended changes for October Fy14 activity restructuring.
10/25/13	Various	1.2	Provided clearer representation of priorities within Table 7 "TF Element Coverage Guidance"
01/06/14	Various	1.3	Provided clearer representation of priorities within Table 7 "TF Element Coverage Guidance"
05/14/2014	Various	1.4	Updated as part of FY14 mid year planning activities.
08/20/2014	Various	1.5	Updated as part of FY15 annual planning. Updated to address COMPASS tool results.
01/05/2015	Various	1.6	Updated to address materials delivered by the MPCV Program after the FY15 Baseline plan was developed. Updated for changes in M-3 and M-35.
03/02/2015	Various	1.7	Updated as part of FY15 mid year planning activities.
07/29/2015	Various	1.8	Updated as part of FY16 annual planning.

Table of Contents

1. Purpose 4
2. Assessment of Technical Scope 4
3. Technical Rigor 5
3.1. TF 2.0 Verify and Validate Concept Documentation 8
3.2. TF 3.0 Verify and Validate Requirements 26
3.3. TF 4.0 Verify and Validate Test Documentation 36
3.4. TF 5.0 Verify and Validate Design 50
3.5. TF 6 Verify and Validate Implementation 64
3.6. TF 7 Verify and Validate Operations and Maintenance Content 79

Table of Tables

Table 1 Integrated Mission Analysis Report Derived Capabilities 5
Table 2 HEO PBRA Results 5
Table 3 MPCV RBA Scores 5
Table 4 TF Element Mapping to Selected Method 6
Table 5 TF Element Coverage Guidance 6
Table 6 IV&V 09-1 TF Full and Partial Coverage for IV&V Project per RBA over the full Project Lifecycle 7
Table 7 IV&V 09-1 Technical Framework for IV&V Project FY16 Planning 7
Table 8 TF Achieved by IV&V Project-to-date 8
Table 9 TF 2 Element Mapping to Selected Method 9
Table 10 Activity 1: Determine Reuse Applicability by Manually Comparing Operational Environments 10
Table 11 Activity 2: Validate Software Architecture by Inspecting Traces to Essential Properties 12
Table 12 Activity 3: Validate Mission Project Operational Concepts by Generating Use Cases from Concept Documentation 14
Table 13 Activity 4: Validate Feasibility Study Conclusions by Inspection 16
Table 14 Activity 6: Validate System Security Categorization and Regulatory Security Requirements by Inspection using Security Risk Management Framework (NIST-SP-800-37, Step 1) 18
Table 15 Activity 7: Verify Security Control Selection and Threats/Risks Identification by Inspection using Security Risk Management Framework (NIST-SP-800-37, Step 2)20
Table 16 Activity 5: Verify System Software Safety by Comparing Concept Documentation, Requirements, Testing, Design and Code with Hazard Analysis Documentation to Establish a Safety Case, Across the Software Development Life Cycle 23
Table 17 TF 3 Element Mapping to Selected Method 26

Table 18 Activity 8: Validate Requirements by Inspecting Against Quality Criteria and System/Software Background Artifacts 27
Table 19 Activity 9: Validate Requirements by Inspecting Bidirectional Traces 29
Table 20 Activity 10: Validate Safety Requirements by Inspection of Traces to Fault Trees and FMEA 32
Table 21 Activity 11: Validate Interface Requirements by Inspection Against Component Interfaces 34
Table 22 TF 4 Element Mapping to Selected Method 36
Table 23 Activity 12: Validate Test Plan by Inspection 38
Table 24 Activity 13: Validate Test Environment by Inspecting Against Scenarios and NASA Guidelines 42
Table 25 Activity 14: Validate Test Cases by Inspection and Traces to Requirements 44
Table 26 Activity 16: Validate Test Design by Inspecting Traces from Scenarios 46
Table 27 Activity 15: Validate Test Procedures by Inspection and Traces to Requirements 48
Table 28 TF 5 Element Mapping to Selected Method 50
Table 29 Activity 17: Verify Software Design by Inspecting Traces to Requirements and Software Architecture 52
Table 30 Activity 18: Verify Software Interface Design by Inspection Against Interface Requirements 55
Table 31 Activity 24: Verify Scripted Timeline Via Manual Multi-Directional Tracing 57
Table 32 Activity 19: Verify System/Software Architecture Using a Discrete Model of Performance Requirements in Stressing Scenarios 60
Table 33 Activity 27: Verify GN&C Software Design by Inspecting Traces to Requirements and Software Architecture 62
Table 34 TF 6 Element Mapping to Selected Methods 64
Table 35 Activity 20: Verify Critical Software Changes By Inspecting Change Requests 65
Table 36 Activity 21: Verify Test Execution by Inspection of Test Cases, Inputs and Results 68
Table 37 Activity 25: Verify SW Interface Implementation by Inspection Against Interface Design 70
Table 38 Activity 22: Verify Software Implementation by Inspecting Traces to Requirements 72
Table 39 Activity 23: Verify Software Code Quality using Static Analysis Tools 75
Table 40 Activity 26: Verify Software Behavior for Off-Nominal Conditions using Independent Testing 77
Table 41 TF 7 Element Mapping to Selected Methods 79
Table 42 Activity 28: Assurance that the Software Operations, Anomalies and Contingencies are known 80

1. Purpose

The Multi-Purpose Crew Vehicle IV&V Technical Scope and Rigor document describes (a) where IV&V's effort will be applied on the Multi-Purpose Crew Vehicle (MPCV) IV&V Project for EM-1 and subsequent flights, (b) the approaches selected for performing technical tasks that will start in the next planning period, (c) the basis for deciding on an approach or approaches, and (d) each selected analysis approach.

2. Assessment of Technical Scope

IV&V Technical Scope discussion requires understanding the MPCV vehicle and MPCV Flight Software (FSW). The MPCV is based on the Orion design requirements for traveling Beyond low Earth Orbit (BEO). The MPCV will serve as the exploration vehicle that will carry the crew to space, provide emergency abort capability, sustain the crew during the space travel, and provide safe re-entry from deep space return velocities.

The MPCV spacecraft includes crew and service modules, a spacecraft adaptor, and a Launch Abort System (LAS) intended to significantly increase crew safety. The MPCV's Crew Module (CM) is the transportation capsule that provides a safe habitat for the crew, provides storage for consumables and research instruments, and serves as the docking port for crew transfers. The crew module is much larger than Apollo's and can support more crew members for short or long-duration spaceflight missions. The CM is the only part of the MPCV that returns to Earth after each mission. The Service Module (SM) supports the CM from launch through separation prior to reentry. It provides in-space propulsion capability for orbital transfer, attitude control, and high altitude ascent aborts. When mated with the crew module, it provides the water, oxygen, and nitrogen needed for a habitable environment, generates and stores electrical power while on-orbit, and maintains the temperature of the vehicle's systems and components. The service module can also transport unpressurized cargo and scientific payloads by mounting them on its structure. The LAS, positioned above the crew module, activates within milliseconds to propel the crew module to safety in the event of an emergency during launch or ascent to orbit. The system also protects the crew module from dangerous atmospheric loads and heating, then jettisons after the MPCV is through the initial mission phase of ascent to orbit.

The MPCV avionics will incorporate approximately 827k[1] Source Lines of Code (SLOC) equivalent, using a table driven architecture in an Aeronautical, Inc (ARINC) 653 time and space partitioned environment, with seventeen high level Computer Software Configuration Items (CSCIs), with all of the Flight Software (FSW) operating in a single primary flight computer, which is duplicated for physical redundancy.

The HEO ESD PBRA was a clean start from prior IV&V PBRAs. The PBRA/RBA was completed using the process defined in S3106 Revision D; "PBRA and RBA Process".

[1] The SLOC total includes an estimated 231k SLOC of nondevelopmental "Off The Shelf" code.

The activity applied capabilities from the HEO ESD Joint Program/ESD Integrated Mission Analysis (IMA) Report ESD 10014. Table 1, below, was used to guide and document the EM-1/2 capabilities' definitions.

Table 1 Integrated Mission Analysis Report Derived Capabilities

< Table redacted>

The MPCV capabilities were scored along with an associated rationale. These results were then peer reviewed.

Table 2 HEO PBRA Results

< Table redacted>

The RBA activity began with a definition of the entities. The CSCIs define how the requirements and design are packaged in artifacts so it is the rational place to start for effort estimations. MPCV EFT-1 experimentation with using the "Domain" level for entities demonstrated no significant advantage and a dramatically increased analyst effort when using Domains for the RBA entities. The RBA scoring and rationale for each CSCI or CSCI-like entity was then performed and reviewed. The result is shown in Table 3.

Table 3 MPCV RBA Scores

< Table redacted>

3. Technical Rigor

This section defines the technical tasks that will be performed in Fiscal Year 2016. The analysis methods were chosen from the Catalog of Methods consistent with IV&V practices, catalog of methods content, and the TS&R template. The tables required by the TS&R template for each selected method are shown below. The tables are organized by Technical Framework (TF) within the IV&V Technical Framework.

TF 1.0, discussing management tasks, is not included in this report because it does not deal with the technical aspects of project planning. It is included in the IPEP and effort estimate.

Table 4 TF Element Mapping to Selected Method

< Table redacted>

Table 5 Lifecycle TF Element Coverage Guidance

< Table redacted>

Table 6 IV&V 09-1 TF Full and Partial Coverage for IV&V Project per RBA over the full Project Lifecycle

IV&V 09-1 Rev O. Technical Framework Full and Partial Coverage												
M&P	1.1	1.2	1.3	1.4	1.5	1.6						
Conc.	2.1	2.2	2.3	2.4	2.5	2.6						
Reqts.	3.1	3.2	3.3	3.4	3.5							
Test	4.1.1	4.1.2	4.1.3	4.1.4	4.1.5	4.2	4.3	4.4	4.5	4.6	4.7	4.8
Design	5.1	5.2	5.3	5.4	5.5	5.6						
Impl.	6.1	6.2	6.3	6.4	6.5	6.6						
O&M	7.1	7.2	7.3	7.4	7.5	7.6	7.7	7.8				

TF Coverage Key:

= Full Coverage of TF element

= Partial Coverage of TF element

= TF element not being performed on IV&V Project

Table 7 IV&V 09-1 Technical Framework for IV&V Project FY16 Planning

IV&V 09-1 Rev O. Technical Framework Full and Partial Coverage												
M&P	1.1	1.2	1.3	1.4	1.5	1.6						
Conc.	2.1	2.2	2.3	2.4	2.5	2.6[2]						
Reqts.	3.1	3.2	3.3	3.4	3.5							
Test	4.1.1	4.1.2	4.1.3	4.1.4	4.1.5	4.2	4.3	4.4	4.5	4.6	4.7	4.8
Design	5.1	5.2	5.3	5.4	5.5	5.6						
Impl.	6.1	6.2	6.3	6.4	6.5	6.6						

[2] TF2.6 analyses are performed by the Information Assurance core team working with the IV&V HEO ESD Integration team and MPCV IV&V team.

O&M	**7.1**	**7.2**	**7.3**	**7.4**	**7.5**	**7.6**	**7.7**	**7.8**

Table 8 TF Achieved by IV&V Project-to-date

IV&V 09-1 Rev O. TF Project-to-date Full and Partial Coverage												
M&P	**1.1**	**1.2**	**1.3**	**1.4**	**1.5**	**1.6**						
Conc.	**2.1**	**2.2**	**2.3**	**2.4**	**2.5**	**2.6**						
Reqts.	**3.1**	**3.2**	**3.3**	**3.4**	**3.5**							
Test	**4.1.1**	**4.1.2**	**4.1.3**	**4.1.4**	**4.1.5**	**4.2**	**4.3**	**4.4**	**4.5**	**4.6**	**4.7**	**4.8**
Design	**5.1**	**5.2**	**5.3**	**5.4**	**5.5**	**5.6**						
Impl.	**6.1**	**6.2**	**6.3**	**6.4**	**6.5**	**6.6**						
O&M	**7.1**	**7.2**	**7.3**	**7.4**	**7.5**	**7.6**	**7.7**	**7.8**				

3.1. TF 2.0 Verify and Validate Concept Documentation

The methods used for each element item are shown in Table 9. The specific methods are described in Table 10 through Table 16

Table 9 TF 2 Element Mapping to Selected Method

TF	Element Narrative	Coverage Comment	**Method**
2	The objective of Concept IV&V is to validate the selected solution and ensure that no false assumptions have been incorporated in the solution.	The Concept stage of the MPCV program is largely complete. IV&V analyses in this area will be performed out of phase.	There is no unique method necessary for TF 2. Completion of all subsequent 2.X sub elements address 2 in aggregate.
2.1	Ensure that software planned for reuse meets the fit, form, and function as a component within the new application.	"Reuse" will include consideration of EFT-1 FSW being applied to EM-1/2. This is not truly reuse. EFT-1 and EM-1/2 are the same program and largely the same vehicle; however • the CSCIs are being changed, • the mission profile is changed, and • the software classification is changing	M-5 v1.0 Determine Reuse Applicability by Manually Comparing Operational Environments[3]
2.2	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.	The system architecture is a modification of the architecture used for "Pad Abort – 1" and Exploration Flight Test – 1. There will be significant experience in the basic computing architecture. Covered as indicated above in Table 5 Lifecycle TF Element Coverage Guidance	M-17 v1.0 Validate Software Architecture by Inspecting Traces to Essential Properties
2.3	Ensure that the concepts for the operations, mission objectives (including mission retirement), and the system are sufficiently defined as a basis for the engineering and planning of computing related functions.	Covered as indicated above in Table 5 Lifecycle TF Element Coverage Guidance	M-36 v1.0 Validate Mission Project Operational Concepts by Generating Use Cases from Concept Documentation
2.4	Ensure that feasibility studies provide the results necessary to confidently support the key decisions that drove the need for the study.	EM-1 execution (November and December 2014) has identified current Program activities that necessitate IV&V coverage. This pattern is expected to continue during FY16; the MPCV Program's Release Plans and JIRA database will be used to identify feasibility studies for review.	M-34 v1.0 Validate Feasibility Study Conclusions by Inspection

[3] The addition of the Goddard Space Flight Center (GSFC) Core Flight Software (CFS) into the FSW architecture as part of CSCI will require additional TF 2.1 analyses after the MPCV Program reaches the post-PDR stage.

TF	Element Narrative	Coverage Comment	Method
2.5	Ensure that known software based hazard causes, contributors, and controls are identified and documented.	Covered as indicated above in Table 5 Lifecycle TF Element Coverage Guidance	M-57 v1.0 Verify System Software Safety by Comparing Concept Documentation, Requirements, Testing, Design and Code with Hazard Analysis Documentation to Establish a Safety Case, Across the Software Development Life Cycle
2.6	Ensure that security threats and risks are known and documented and that relevant regulatory requirements are identified.	Covered as indicated above in Table 5 Lifecycle TF Element Coverage Guidance	M-52 v1.0 Validate System Security Categorization and Regulatory Security Requirements by Inspection using Security Risk Management Framework (NIST-SP-800-37, Step 1) M-53 v1.0 Verify Security Control Selection and Threats/Risks Identification by Inspection using Security Risk Management Framework (NIST-SP-800-37, Step 2)

Table 10 Activity 1: Determine Reuse Applicability by Manually Comparing Operational Environments

Method:	M-5, Version 1.0 (Current Status: Approved)
Method Title:	Determine Reuse Applicability by Manually Comparing Operational Environments
Method Synopsis	Method that uses manual inspection to determine differences between legacy and intended operational environments, establishes lines of inquiry to assess the impacts of those differences, and evaluates the reuse proposal on the basis of those impacts. When necessary, detailed inspection of the proposed reuse artifacts may also be performed, to address critical reuse suitability concerns.
Required Method Revisions (if any)	None identified
Technical Goal:	2.1 Ensure that software planned for reuse meets the fit, form, and function as a component within the new application.

WBS Coverage:	IVV 9-1 IV&V Technical Framework, Rev N: 2.1
Scope:	Activity will include both EFT-1 and EM-1/2 concepts of operations
Target Artifacts:	EFT-1 ConOps EM-1/2 ConOps
Inputs (includes Technical Reference):	See Analysis Target
Prerequisites:	None Identified
Success Criteria:	Determination of differences between EFT-1 and EM-1/2 operational environments. Determination of impacts of differences between the two operational environments.
Activity Assumptions:	Necessary artifacts are documented with sufficient detail to merit analysis.
Rationale for Approach:	This is an approved method that has been shown to be effective.
Concerns:	None
Method Application Notes: None	
Required Tools:	None Identified
Empirical Evidence:	Answers to reuse questions for each module along with assessment and rationale for assessment For input/output analysis, evidence is lists of inputs and outputs or interface contracts with assessment of each item
Output (include updates to Project Technical Reference):	- Issues and risks - Sections in MPSRs - Sections in post analysis reports (e.g. “Capstone” reports)
Basis of Estimate:	
Other:	None Identified

Table 11 Activity 2: Validate Software Architecture by Inspecting Traces to Essential Properties

Method:	M-17, Version 1.0 (Current Status: Approved)
Method Title:	Validate Software Architecture by Inspecting Traces to Essential Properties
Method Synopsis	Method supports a manual evaluation of a software architecture by developing a set of essential properties against which elements of the software architecture may be compared. Method detects defects in the software architecture (inadequate coverage of essential properties, missing capabilities, inessential capabilities), and supports regression testing as developer-provided software architecture artifacts mature.
Required Method Revisions (if any)	None identified
Technical Goal:	2.2 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 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).
WBS Coverage:	IVV 9-1 IV&V Technical Framework, Rev N: 2.2 (fully), 5.3 (fully)
Scope:	Activity will include EM-1/2 software architecture artifacts
Target Artifacts:	EFT-1system architecture documents EM-1/2 system architecture documents
Inputs (includes Technical Reference):	Essential properties of the software identified by the IV&V Team
Prerequisites:	Identify essential properties of the software
Success Criteria:	Confirmation that EM-1/2 system architecture contains the necessary computing related items (subsystems, components, etc.) to carry out the mission of the system. Confirmation that EM-1/2 system architecture satisfies user needs and operational scenarios or use cases.
Activity Assumptions:	The system architecture is a modification of the architecture used for “Pad Abort – 1” and Exploration Flight Test – 1.
Rationale for Approach:	This is an approved method that has been shown to be effective.
Concerns:	None
Method Application Notes: TF 2.2 will be the only Technical Goal performed for this analysis. TF 5.3 will not be performed for this analysis.	

Required Tools:	A database or spreadsheet is needed to store and manage the empirical evidence that is expected to be produced by this activity.
Empirical Evidence:	A) Trace of architectural element and essential properties B) Assessment of each essential property in terms of the architecture description element traced to that property
Output (include updates to Project Technical Reference):	- Issues and risks - Sections in MPSRs - Sections in post analysis reports (e.g. "Capstone" reports)
Basis of Estimate:	
Other:	It is highly recommended that outputs from this method as executed on other projects be used when determining the necessary properties of interest; this will insure completeness and efficiency of execution.

Table 12 Activity 3: Validate Mission Project Operational Concepts by Generating Use Cases from Concept Documentation

Method:	M-36, Version 1.0 (Current Status: Approved)
Method Title:	Validate Mission Project Operational Concepts by Generating Use Cases from Concept Documentation
Method Synopsis	Manual Method to evaluate Mission Project prepared Operational Concepts. Faults detected through this Method include a) poor Operational Concepts quality (fails to answer one or more of the key Operational Concepts quality attribute questions), b) failure to represent the needs of users and stakeholders of the system (inability to trace Operational Concepts to mission objectives, output from stakeholder interviews, etc.).
Required Method Revisions (if any)	Rather than generate use cases, the Use Cases from the MPCV Program's own work will be evaluated/reviewed and then applied to all subsequent steps.
Technical Goal:	2.3 Ensure that the concepts for the operations, mission objectives (including mission retirement), and the system are sufficiently defined as a basis for the engineering and planning of computing related functions. This analysis should a) Ensure the Program Concept of Operations and Mission Objectives adequately represent the needs of the users and stakeholders of the system. b) Ensure the Program Concept of Operations and Mission Objectives are high quality as defined in the analysis steps
WBS Coverage:	IVV 9-1 IV&V Technical Framework, Rev N: 2.3 (full)
Scope:	Activity will include EM-1/2 concepts of operations
Target Artifacts:	Concept of operations Program Use Cases
Inputs (includes Technical Reference):	1. Concept of Operation/Design Reference Missions/Needs-Goals-Objectives documentation 2. Integrated Master Timeline (if available) 3. Portfolio-Based Risk Assessment 4. Program Use Cases
Prerequisites:	None
Success Criteria:	Confirmation that the concepts for the operations, mission objectives, and the system are sufficiently defined as a basis for the engineering and planning of computing related functions. Confirmation that the universal set of Use Cases (desired system capabilities) adequately represent the needs of the

	users and stakeholders of the system. Determination that Use Cases are high quality as defined in the analysis steps
Activity Assumptions:	Necessary artifacts are available for IV&V timely analysis
Rationale for Approach:	Reuse of Program provided use cases will return Use Case comments to the MPCV Program at the earliest possible stage. This is an approved method. Duplication of the effort to produce independent Use Cases would be valuable, but prohibitive given the current Project plan requires IV&V to omit critical "red" CSCIs from IV&V coverage. The indicated modification saves effort allowing other work to be performed.
Concerns:	None
Method Application Notes: None	
Required Tools:	Engineering worksheets (or database) to document empirical evidence.
Empirical Evidence:	IV&V answer for each question in Step B for each use case analyzed. Rationale for IV&V answer to each question, including appropriate references/links to Mission Project content that supports IV&V's conclusion. For Step C, IV&V conclusion and rationale, including appropriate references/links to Mission Project content that supports IV&V's conclusion. Issues identified in Mission Project Operational Concept documentation.
Output (include updates to Project Technical Reference):	Issues and risks Sections in MPSRs Sections in post analysis reports (e.g. "Capstone" reports)
Basis of Estimate:	
Other:	

Table 13 Activity 4: Validate Feasibility Study Conclusions by Inspection

Method:	M-34, Version 1.0 (Current Status: Approved)
Method Title:	Validate Feasibility Study Conclusions by Inspection
Method Synopsis	Validate the conclusions of a Feasibility Study by assessing the accuracy and completeness of the study by addressing a set of evaluation questions. These questions are composed of a standard set that is expanded to include evaluation questions specific to the feasibility study itself. This Method is designed to identify logic flaws in the chain from assumptions to conclusions; missing, inconsistent or extraneous constraints/assumptions; and general weakness of conclusions reached by the feasibility study under evaluation.
Required Method Revisions (if any)	None identified
Technical Goal:	2.4 Ensure that feasibility studies provide the results necessary to confidently support the key decisions that drove the need for the study. Ensure that feasibility study assumptions, methods, processes utilized and results give confidence that the feasibility study appropriately addresses the question(s) that drove the need for the study.
WBS Coverage:	IVV 9-1 IV&V Technical Framework, Rev N: 2.4 (full)
Scope:	EM-1 execution (November and December 2014) has identified current Program activities that necessitate IV&V coverage. This pattern is expected to continue during FY16. The delivery of specific analyses will be driven by the Program deliveries of completed studies.
Target Artifacts:	EM-1/2 feasibility studies
Inputs (includes Technical Reference):	1. Question(s) that drove the need for the feasibility study. 2. Access to all input data, algorithms and methods used in the performance of the feasibility study.
Prerequisites:	None
Success Criteria:	Confirmation that the conclusions of a Feasibility Study are accurate and complete. Identification of any logic flaws in the chain from assumptions to conclusions. Identification of missing, inconsistent or extraneous constraints/assumptions, and general weakness of conclusions reached by the feasibility study under evaluation.
Activity Assumptions:	Identification of need based on identification of weaknesses from other activities.
Rationale for Approach:	This is an approved method that has been shown to be effective.

Concerns:	None
Method Application Notes: None	
Required Tools:	None
Empirical Evidence:	Completed IV&V question checklist and documented justification to support the answers
Output (include updates to Project Technical Reference):	- Issues and risks - Sections in MPSRs - Sections in post analysis reports (e.g. “Capstone” reports)
Basis of Estimate:	
Other:	None

Table 14 Activity 6: Validate System Security Categorization and Regulatory Security Requirements by Inspection using Security Risk Management Framework (NIST-SP-800-37, Step 1)

Method:	M-52, Version 1.0 (Current Status: Approved)
Method Title:	Validate System Security Categorization and Regulatory Security Requirements by Inspection using Security Risk Management Framework (NIST-SP-800-37, Step 1)
Method Synopsis	Verify system categorization is appropriate for selection of Security Controls and validate security requirements meet system needs
Required Method Revisions (if any)	None
Technical Goal:	2.6: Ensure that security threats and risks are known and documented and that relevant regulatory requirements are identified. **< redacted>**
WBS Coverage:	IVV 9-1 IV&V Technical Framework, Rev O: WBS 2.6 is partially addressed with this method and wholly addressed when combined with the "Verify Security Controls" Method addressing NIST SP 800-37 RMF Step 2.
Scope:	**< redacted>**
Target Artifacts:	**< redacted>**
Inputs (includes Technical Reference):	Apply checklists provided in NIST SP 800-37 RMF Task 1-2 to verify completeness of the Information System description
Prerequisites:	Analyst should be familiar with: NIST SP 800-37, Risk Mgt Framework (RMF) Step 1, Categorize Information System FIPS Pub 199 , Standards for Security Categorization of Federal Information and information systems NIST SP 800-59, Guideline for Identifying an Information System as a National Security System NIST SP 800-60, Guide for mapping types of Information and Information Systems to Security Categories NASA ITS-HBK 2810.x series
Success Criteria:	Determination if the organization completed a security categorization of the information system including the information to be processed, stored, and transmitted by the system. Determination if the results of the security categorization process for the information system consistent with the organization's enterprise

	architecture and commitment to protecting organizational mission/business processes. Determination if the results of the security categorization process reflect the organization's risk management strategy. Determination if the organization adequately described the characteristics of the information system. Determination if the organization registered the information system for purposes of management, accountability, coordination, and oversight
Activity Assumptions:	Materials are available and provide suitable sources of necessary elements and information.
Rationale for Approach:	Provides reasonable coverage of TF element with an approved method
Concerns:	None
Method Application Notes: A secret level security clearance is required to perform IV&V Security analysis for MPCV.	
Required Tools:	Analyst should Apply the guidance described in the following sources to determine whether the system is correctly categorized: NIST SP 800-37, Risk Mgt Framework (RMF) Step 1, Categorize Information System FIPS Pub 199 , Standards for Security Categorization of Federal Information and information systems NIST SP 800-59, Guideline for Identifying an Information System as a National Security System NIST SP 800-60, Guide for mapping types of Information and Information Systems to Security Categories
Empirical Evidence:	The IV&V Team will generate independent assessments for the security categorization of the systems as supported by the system descriptions. The IV&V Team will demonstrate traceability of applicable federal and organizational regulatory security requirements to the system security plan.
Output (include updates to Project Technical Reference):	IV&V Issues IV&V Risks Material for Architecture Closeout Reports
Basis of Estimate:	
Other:	None

Table 15 Activity 7: Verify Security Control Selection and Threats/Risks Identification by Inspection using Security Risk Management Framework (NIST-SP-800-37, Step 2)

Method:	M-53, Version 1.0 (Current Status: Approved)
Method Title:	Verify Security Control Selection and Threats/Risks Identification by Inspection using Security Risk Management Framework (NIST-SP-800-37, Step 2)
Method Synopsis	Verify that selected security controls are appropriate for the system and its security categorization
Required Method Revisions (if any)	None
Technical Goal:	2.6 Ensure that security threats and risks are known and documented and that relevant regulatory requirements are identified. **< redacted>**
WBS Coverage:	IVV 9-1 IV&V Technical Framework, Rev O: WBS 2.6 is partially addressed with this method and wholly addressed when combined with the "Verify System Security Categorization" Method addressing NIST SP 800-37 RMF Step 1.
Scope:	**< redacted>**
Target Artifacts:	**< redacted>**
Inputs (includes Technical Reference):	1) Technical Reference information about System Security Categorization and regulatory security requirements from RMF Step 1. 2) Organizational security requirements, enterprise architecture, and common controls
Prerequisites:	1) Analyst should have completed the Validate System Categorization and Regulatory Security Requirements method. 2) Analyst should be familiar with: NIST SP 800-37, Risk Mgt Framework (RMF) Step 2 Task 2-2, Security Control Selection NIST SP 800-53, Recommended Security Controls for Federal Information Systems and Organizations FIPS - 200, Minimum Security Requirements for Federal Information and Information Systems CNSS Instruction 1253, Security Categorization and Control Selection for National Security Systems NASA ITS-HBK 2810.x series
Success Criteria:	Determine if the organization allocated all security controls to the information system as system-specific, hybrid, or common controls. Determine if the organization used its risk assessment (either formal or informal) to inform and guide the

	security control selection process. Determine if the organization identified all common controls inherited by the system. Determine if the organization tailored and supplemented the baseline security controls to ensure that the controls, if implemented, adequately mitigate risks to organizational operations and assets, individuals, other organizations, and the Nation Determine if the organization addressed minimum assurance requirements for the security controls employed within and inherited by the information system Determine if the organization consulted information system owners when identifying common controls to ensure that the security capability provided by the inherited controls is sufficient to deliver adequate protection Determine if the organization supplemented the common controls with system-specific or hybrid controls when the security control baselines of the common controls are less than those of the information system inheriting the controls? Determine if the organization documented the common controls inherited from external providers
Activity Assumptions:	Materials are available and provide suitable sources of necessary elements and information.
Rationale for Approach:	Only available method
Concerns:	None
Method Application Notes: A secret level security clearance is required to perform IV&V Security analysis for MPCV.	
Required Tools:	Analyst should apply the guidance described in the following sources to determine whether the selected security controls meet requirements and are appropriate for the information system: NIST SP 800-37, Risk Mgt Framework (RMF) Step 2, Task 2-2, Security Control Selection NIST SP 800-53, Recommended Security Controls for Federal Information Systems and Organizations FIPS - 200, Minimum Security Requirements for Federal Information and Information Systems CNSS Instruction 1253, Security Categorization and Control Selection for National Security Systems Analyst should apply the guidance in NIST SP 800-30, Risk Management Guide for Information Technology

	Systems, to evaluate the information technology security risk.
Empirical Evidence:	Using NIST SP 800-53, create a checklist of security controls for the system categorization and provide rationale for deviation from NIST SP 800-53 (controls that exceed guidance or controls that are recommended but not provided).
Output (include updates to Project Technical Reference):	IV&V Issues IV&V Risks Material for Architecture Closeout Reports
Basis of Estimate:	
Other:	None

Table 16 Activity 5: Verify System Software Safety by Comparing Concept Documentation, Requirements, Testing, Design and Code with Hazard Analysis Documentation to Establish a Safety Case, Across the Software Development Life Cycle

Method:	M-57, Version 1.0 (Current Status: Approved)
Method Title:	Verify System Software Safety by Comparing Concept Documentation, Requirements, Testing, Design and Code with Hazard Analysis Documentation to Establish a Safety Case, Across the Software Development Life Cycle
Method Synopsis	A safety-case is a structured argument which provides evidence sufficient to make a conclusion about the "safety of a system" or about the focus of the safety case. The intent of this analysis is to provide a well-reasoned evaluation of the factors that contribute to the safety of the system as it relates to software. This method provides a systematic approach to analyze (and document) the software safety of a system. This analysis can be done on the system as a whole, or on a targeted (focused) part of the system (as determined by risk/criticality assessment). A Safety Case may also be performed on a specific behavior or function.
Required Method Revisions (if any)	None
Technical Goal:	2.5 (Full) Ensure that known software based hazard causes, contributors, and controls are identified and documented. 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. 4.1.1 Ensure that the software correctly implements system and software requirements in an operational environment under nominal and off-nominal conditions. 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. 4.4 Ensure that any simulations are sufficiently complete, correct, and accurate to perform the intended testing. 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). 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. 6.2 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. 6.5 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 9-1 IV&V Technical Framework, Rev N: 2.5 (full)
Scope:	Activity will include all EM-1/2 safety documentation
Target Artifacts:	PHA, SRS Requirements, FMEA, FMECA
Inputs (includes Technical Reference):	Fault Tree Analysis (FTA - produced by the project), System Hazard Analysis (SHA - produced by the project), Preliminary Hazard Analysis (PHA - produced by the project), Failure Modes Effects Analysis (FMEA - produced by the project), Fault Management Effects and Criticality Analysis (FMECA - produced by the project), Subsystem System Hazard Analysis (SSHA - produced by the project)
Prerequisites:	IV&V has performed a PBRA/RBA to identify safety-critical areas of the system based on existing mission documentation IV&V has produced a list of IV&V independent failures (see process assets for safety case for list produced by GPM) FMEA is available PHA is available
Success Criteria:	Confirmation that software based hazard causes, contributors, and controls are identified and documented.
Activity Assumptions:	None
Rationale for Approach:	Provides coverage of TF element.
Concerns:	This method requires as input, the FTA/FMEA/PHA documents. In the case that software requirements/design have not been identified as safety-critical by the project (due to the delay in the above docs), this should become an IV&V

	risk for this project.
Method Application Notes: TF 2.5, 5.6, 6.2, and 6.5 will be the only Technical Goals performed for this analysis. TF 3.4, 4.1.1, 4.3, 4.4, 4.5, and 6.6 will not be performed for this analysis.	
Required Tools:	None
Empirical Evidence:	Safety case analysis results produces a report that includes the following: - Traceability between validated SRS requirements with FMEA/PHA/SHA/SSHA documents - Traceability between FMEA/PHA/SHA/SSHA and test/code. - Uncovered hazards (identified by issues/risks) from requirements/code/test
Output (include updates to Project Technical Reference):	- Issues and risks - Sections in MPSRs - Sections in post analysis reports (e.g. “Capstone” reports)
Basis of Estimate:	
Other:	None.

3.2. TF 3.0 Verify and Validate Requirements

The methods used for each element item are shown in Table 17. The specific methods are described in Table 18 through Table 21.

Table 17 TF 3 Element Mapping to Selected Method

TF	Element Narrative	Coverage Comment	Method
3	The objective of Requirements IV&V is to ensure the system's software requirements are high quality (correct, consistent, complete, accurate, unambiguous , and verifiable), and will adequately meet the needs of the system and expectations of its customers and users, considering its operational environment under nominal and off-nominal conditions, and that no unintended features are introduced.		No method required at the 3 (integer) level. 3 is addressed by addressing all of the sub elements of 3 listed below.
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.	Covered as indicated above in Table 5 Lifecycle TF Element Coverage Guidance	M-2 v1.3 Validate Requirements by Inspecting Against Quality Criteria and System/Software Background Artifacts
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.	Covered as indicated above in Table 5 Lifecycle TF Element Coverage Guidance	M-3 v1.3 Validate Requirements by Inspecting Bidirectional Traces
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.	Covered as indicated above in Table 5 Lifecycle TF Element Coverage Guidance	M-2 v1.3 Validate Requirements by Inspecting Against Quality Criteria and System/Software Background Artifacts M-20 V1.0 Verify Critical Software Changes By Inspecting Change Requests"
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.	Covered as indicated above in Table 5 Lifecycle TF Element Coverage Guidance	M-1 V1.0 "Validate Interface Requirements by Inspection Against Component Interfaces"
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.	Covered as indicated above in Table 5 Lifecycle TF Element Coverage Guidance	M-6 v1.0 Validate Safety Requirements by Inspection of Traces to Fault Trees and FMEA

Table 18 Activity 8: Validate Requirements by Inspecting Against Quality Criteria and System/Software Background Artifacts

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.
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 9-1 IV&V Technical Framework, Rev N: 3.1, 3.3, 3.4
Scope:	Activity will include all EM-1/2 MPCV software requirements.
Target Artifacts:	All EM-1/2 software requirements.
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:	• Requirements artifacts parsed, loaded into assessment tool (spreadsheet or Access database), and requirements suitable for implementation in software identified with associated rationale. • Each requirement inspected with respect to the quality attribute criteria.

	• The set of requirements inspected to ensure the set addresses the Functionality, Process definition and scheduling, Hardware, software, and user interface descriptions, performance criteria, critical configuration data; and system, device, and software control. • The qualitative status for each requirement documented.
Activity Assumptions:	Necessary artifacts are documented with sufficient detail to merit analysis.
Rationale for Approach:	The approach is generally applicable to the MPCV's current state and does not require highly developed completion of prior activities. Specifically this approach does not require the separate development of simulations or models. This method, when combined with method "Validate Requirements by Inspection Against Component Interfaces" provides a reasonable coverage of TF 3.4.
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):	- Issues and risks - Sections in MPSRs - Sections in post analysis reports (e.g. "Capstone" reports)
Basis of Estimate:	
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

Table 19 Activity 9: Validate Requirements by Inspecting Bidirectional Traces

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.
Required Method Revisions (if any)	
Technical Goal:	Assess the quality of the requirements (set) and the degree to which they adequately specify: 1. A logical decomposition of the parent requirements 2. Any functional allocations identified by the developer It may be possible to consider 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). 3.2 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. [Planning on pulling this out - Traceability Analysis] 3.3 Ensure the software requirements are of high quality and adequately meet the needs of the system with respect to expectations of its stakeholder and users, operational environment, reliability and fault tolerance, and both functional and non-functional perspectives.
WBS Coverage:	IVV 9-1 IV&V Technical Framework, Rev N: 3.1 (partial); 3.2, 3.3
Scope:	Activity will include all EM-1/2 MPCV software requirements.
Target Artifacts:	Software Requirements Specifications Requirements Traceability Matrix
Inputs (includes Technical Reference):	Requirements Trace developed by the Mission Project Additional Reference Artifacts to understand the requirements to be assessed Capabilities defined to level of analysis (PBRA, RBA)

Prerequisites:	Requirements and developer provided traces loaded into traceability tool (spreadsheet/Access database)
Success Criteria:	• Identify requirements that could be implemented by software. Provide rationale. • Evaluate the completeness, correctness and consistency of the collection of software-related requirements by performing a Bi-directional trace that leverages the project's trace documentation
Activity Assumptions:	Necessary artifacts are documented with sufficient detail to merit analysis.
Rationale for Approach:	The approach is generally applicable to the MPCV's current state and does not require highly developed completion of prior activities. Specifically this approach does not require the separate development of simulations or models. This method provides a reasonable coverage of TF 3.2. This method when combined with "Validate Requirements by Inspecting Against Quality Criteria" and "Verify Critical Software Changes By Inspecting Change Requests" provides reasonable coverage of TF 3.1 and TF 3.3.
Concerns:	None
Method Application Notes: None	
Required Tools:	Engineering worksheets (or Requirements Validation Assessment Management System [MS Access] database) to document results of tracing analysis Observations, Issues, and Risk Management System (ORBIT)
Empirical Evidence:	Completeness/correctness/consistency status in Engineering worksheets (or Requirements Validation Assessment Management System database) for each requirement.
Output (include updates to Project Technical Reference):	- Issues and risks - Sections in MPSRs - Sections in post analysis reports (e.g. "Capstone" reports)
Basis of Estimate:	
Other:	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 the collection has been evaluated for completeness, correctness and consistency • All issues have been synthesized into concerns • All requirement assessments with no issues have been synthesized into confirmations

Table 20 Activity 10: Validate Safety Requirements by Inspection of Traces to Fault Trees and FMEA

Method:	M-6, Version 1.0 (Current Status: Approved)
Method Title:	Validate Safety Requirements by Inspection of Traces to Fault Trees and FMEA
Method Synopsis	Method uses manual tracing of in-scope safety requirements to developer-supplied fault tree and FMEA artifacts (to assess quality of developer's fault tree/FMEA engineering) and to safety-critical entities. This method identifies defects in safety requirements, fault tree/FMEA artifacts, and safety requirements traces.
Required Method Revisions (if any)	None
Technical Goal:	Provide evidence that the specified safety requirements fulfill the quality criteria of complete and correct. Determination and evidence that: - No missing safety requirements to address fault - No missing fault scenarios exist - No inadequate responses to faults 3.5 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.
WBS Coverage:	IVV 9-1 IV&V Technical Framework, Rev M: 3.5 (partial)
Scope:	Activity will include all EM-1/2 MPCV software requirements.
Target Artifacts:	Software Requirements Specifications FMEA FTA
Inputs (includes Technical Reference):	Fault trees and FMEA artifacts. Will include hazards caused by S/W and Hazards where S/W is the control. List of safety-critical entities, and list of non safety-critical entities.
Prerequisites:	1. A qualitative assessment of the safety requirements has been performed 2. Requirements have been traced to safety-critical entities
Success Criteria:	• Safety requirements traced to fault trees and FMEA • Safety requirements that do not map to fault trees/FMEA analyzed as potentially unnecessary or indicative of incomplete fault trees/FMEA • Fault trees/FMEA with no requirements traced to them analyzed as potentially unnecessary or indicative of missing requirements • Safety requirements that do not trace to safety critical entities analyzed as potentially unnecessary, indicative of incomplete identification of safety critical entities, or a

	missing trace • Safety critical entities with no requirements mapped to it analyzed as potentially incorrect identification of the software as safety critical, missing safety requirements, or a missing trace
Activity Assumptions:	Necessary artifacts are documented with sufficient detail to merit analysis.
Rationale for Approach:	The approach is generally applicable to the MPCV's current state and does not require highly developed completion of prior activities. Specifically this approach does not require the separate development of simulations or models.
Concerns:	This method only provides partial coverage of the TF 3.5 goals.
Method Application Notes: None	
Required Tools:	Engineering worksheets of traceability and engineering assessment of traceability.
Empirical Evidence:	Trace of safety requirements to fault trees/FMEA Trace analysis in engineering worksheet
Output (include updates to Project Technical Reference):	- Issues and risks - Sections in MPSRs - Sections in post analysis reports (e.g. "Capstone" reports)
Basis of Estimate:	
Other:	None identified

Table 21 Activity 11: Validate Interface Requirements by Inspection Against 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.
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 9-1 IV&V Technical Framework, Rev M: 3.4
Scope:	In scope CSCI interfaces
Target Artifacts:	Interface Requirements
Inputs (includes Technical Reference):	None identified
Prerequisites:	None identified
Success Criteria:	Determination 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
Activity Assumptions:	Necessary artifacts are documented with sufficient detail to merit analysis of targets and merit use as a reference for supporting documents.
Rationale for	Provides coverage of the TF element.

Approach:	
Concerns:	None
Method Application Notes:	
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):	- Issues and risks - Sections in MPSRs - Sections in post analysis reports (e.g. “Capstone” reports) - Engineering worksheets used to document assessment as objective evidence.
Basis of Estimate:	
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.

3.3. TF 4.0 Verify and Validate Test Documentation

The methods used for each element item are shown in Table 22. The specific methods are described in

Table 23 through Table 27.

Table 22 TF 4 Element Mapping to Selected Method

TF	Element Narrative	Coverage Comment	Method
4	The objective is to 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.		No method required at the 4 (integer) level. TF 4 is addressed by addressing all of the sub elements of TF 4 listed below.
4.1.1	Ensure that the software correctly implements system and software requirements in an operational environment under nominal and off-nominal conditions.	Covered as indicated above in Table 5 Lifecycle TF Element Coverage Guidance	M-10 v1.2 Validate Test Plan by Inspection
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.	Covered as indicated above in Table 5 Lifecycle TF Element Coverage Guidance	M-10 v1.2 Validate Test Plan by Inspection
4.1.3	Ensure that the software meets all of the (in-scope) software requirements and is ready to be integrated with system hardware.	Covered as indicated above in Table 5 Lifecycle TF Element Coverage Guidance	M-10 v1.2 Validate Test Plan by Inspection
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.	Covered as indicated above in Table 5 Lifecycle TF Element Coverage Guidance	M-10 v1.2 Validate Test Plan by Inspection
4.1.5	Ensure that the software components (e.g., units, source code modules) correctly implement software component requirements.	Covered as indicated above in Table 5 Lifecycle TF Element Coverage Guidance	M-10 v1.2 Validate Test Plan by Inspection
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.	Covered as indicated above in Table 5 Lifecycle TF Element Coverage Guidance	M-25 v1.4 Validate Test Cases by Inspection and Traces to 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.	Covered as indicated above in Table 5 Lifecycle TF Element Coverage Guidance	M-10 v1.2 Validate Test Plan by Inspection
4.4	Ensure that any simulations are	It is not planned to	M-22 v1.0 Validate Test

TF	Element Narrative	Coverage Comment	Method
	sufficiently complete, correct, and accurate to perform the intended testing.	address this TF element unless other activities show this to be an issue that threatens program FSW success. See Table 5 Lifecycle TF Element Coverage Guidance	Environment by Inspecting Against Scenarios and NASA Guidelines
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).	Covered as indicated above in Table 5 Lifecycle TF Element Coverage Guidance	M-25 v1.4 Validate Test Cases by Inspection and Traces to Requirements
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.	Covered as indicated above in Table 5 Lifecycle TF Element Coverage Guidance	M-35 v1.1 Validate Test Procedures by Inspection and Traces to 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.	See Table 5 Lifecycle TF Element Coverage Guidance	M-21 v1.0"Validate Test Design by Inspecting Traces from Scenarios"
4.8	Ensure that the test environment is sufficiently complete, correct, and accurate to perform the intended testing.	It is not planned to address this TF element unless other activities show this to be an issue that threatens program FSW success. See Table 5 Lifecycle TF Element Coverage Guidance	M-21 v1.0"Validate Test Design by Inspecting Traces from Scenarios"

Table 23 Activity 12: Validate Test Plan by 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.
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:	4.1.1-4.1.5, 4.3 (partial), and 4.8 (partial)
Scope:	Activity applies to all MPCV EM-1/2 software deliveries
Target Artifacts:	Software Test Plan Software Test Description Software Test Reports Each CSCI Software Requirements Specification (SRS)

Inputs (includes Technical Reference):	In Scope developer requirements
Prerequisites:	Released and baselined documentation of EM-1/2 FSW requirements, system requirements, and flight test objectives
Success Criteria:	- All requirements mapped to test case - All cases of partially mapped requirements are verified to be fully mapped with full set of test cases - "Test Like You Fly" / "Fly Like You Test" and key testing principles are upheld. For example, a non-compressible test list is adequately defined and not compromised. - The test plan adequately describes the plans for each level of testing, per NPR 7150.2A (e.g., software component (or Unit) level testing, software integration testing, software qualification (or Build Verification, including Acceptance) testing, and system qualification testing (or System or Mission Operations Testing) of software systems. Note there may be separate test plans for each level of test. - The appropriate types of testing (black box, white box (or functional), external interface tests (to HW, ground, SW from other organizations), regression, end-to-end, day-in-the life, performance, endurance/stress tests) as well as contingency scenarios are planned to verify the target requirements and behaviors under nominal and adverse conditions. - The planned test levels and test types are adequate to verify in scope requirements and system behaviors under nominal and adverse conditions (refer to requirements traceability information if available). - The test plan clearly spells out what the test program will, will not, and cannot verify. For areas that will not/cannot be verified by test, consider if proposed alternate methods for verifying these areas / obtaining waivers are adequate, rationale is provided and associated risk is identified and deemed acceptable. - The fidelity of the test environment (hw/sw/simulators/data logging (for test results, issues, and risks)/special test equipment) is appropriate for accurately exercising the desired requirements and behaviors under test. - The test plan contains appropriate description of test team responsibilities, the test process, including test progression (from unit to integration to build to acceptance to system testing) and grouping of test scenarios/procedures as applicable. If applicable consider data recording/reduction/analysis (per NPR 7150.2A). - The test schedule is feasible and considers prerequisite tests

	that must be successfully performed. Schedule risks are identified (e.g., hardware, emulators, simulations, etc. availability is considered). - Regression tests are planned for each level of testing (e.g., unit, integration, build, acceptance, and post acceptance testing). - The test plan is consistent throughout. - There are no systemic concerns with the test plan, test levels or test types, scope. - The test plan includes information as described in NPR 7150.2A, NASA Software Engineering Standards, Section 5.1.3 and Section 3.4, and as described in NASA-STD-8719.13B, NASA Software Safety Standard, and Section 6.4. - For each requirement/behavior to be tested: - A. The set of inputs used during testing are a fair representative sample from the set of all possible inputs to the software, and test inputs include boundary condition inputs, rarely encountered inputs, invalid inputs, and inputs related to identified hazards. Consider: Test initial conditions do not negate needed functionality or values preset to prevent failure detection and/or reporting without notation. - B. Test scenarios/procedures contain explicit Pass/Fail criteria sufficient to assure assessment of the requirement/behavior. - C. Test scenarios/procedures are clear, unambiguous and repeatable so that regression tests and re-test of failed requirements can be performed exactly as in the original test. - D. The regression test set includes all major FSW functional and performance requirements. - E. The verification method (inspection, analysis, test, demonstration) chosen is appropriate to cover the requirement (e.g., for unit, integration, Build Verification and Acceptance Testing) and/or desired behaviors (for System Testing) defined by the requirements. - Confirmation that the full set of requirements are completely and correctly verified or the appropriate issues and risks are written.
Activity Assumptions:	Artifacts will be available as scheduled
Rationale for Approach:	This is an approved method that has been shown to be effective.
Concerns:	Artifacts may have insufficient content for adequate evaluation
Method Application Notes: None identified	

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):	- Issues and risks - Sections in MPSRs - Sections in post analysis reports (e.g. "Capstone" reports)
Basis of Estimate:	
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)

Table 24 Activity 13: Validate Test Environment by Inspecting Against Scenarios and NASA Guidelines

Method:	M-22, Version 1.0 (Current Status: Approved)
Method Title:	Validate Test Environment by Inspecting Against Scenarios and NASA Guidelines
Method Synopsis	Develop and apply a set of operational scenarios to manually validate that the Test Systems includes the information as described in NPR 7150.2A, NASA Software Engineering Standards, Section 3.4, and ensure the test systems will adequately support the test objectives, via their ability to drive appropriate inputs and capture the appropriate outputs. Method detects defects in compliance to the relevant standard, and defects in test environment and simulations that degrade performance of the intended testing.
Required Method Revisions (if any)	None identified
Technical Goal:	4.4 Ensure that any simulations are sufficiently complete, correct, and accurate to perform the intended testing. 4.8 Ensure that the test environment is sufficiently complete, correct, and accurate to perform the intended testing.
WBS Coverage:	IVV 91- IV&V Technical Framework, Rev N: 4.4, 4.8
Scope:	Activity applies to all MPCV EM-1/2 software deliveries
Target Artifacts:	Software Test Design Software Test Procedures
Inputs (includes Technical Reference):	In-scope developer requirements Test design Test Cases Validated requirements to be verified via testing NPR 7150.2A: NASA Software Engineering Standard Other artifacts that provide insight into how the software is organized and built, such as: Software Design Code Hardware-Software interface spec
Prerequisites:	Released and baselined documentation of EM-1/2 FSW requirements, system requirements, and flight test objectives
Success Criteria:	Each test support system can support the test objectives via their ability to drive appropriate inputs and capture appropriate outputs. The test facility provides an environment sufficiently detailed and complete to exercise all behaviors of interest in a manner that accurately simulates their operational environment. The IV&V analyst assures that, * It is feasible to perform the defined test procedures in the specified test environment;

	* The specified test environment is sufficient to verify feasibility of operations and maintenance of the software in its intended environment; * The specified test environment is sufficient to verify any nonfunctional requirements mapped to the test cases, including those requiring a contended environment for their verification; * The specified test environment provides the capabilities needed for verification of the system safety requirements, such as the abilities to simulate hardware failures and operator errors under system load, stress, and task contention situations. * When validating test cases for a software validation test, the IV&V analyst assures that test cases comply with software engineering process requirement SWE-073 from NASA-NPR-7150.2A, NASA Software Engineering Requirements, which requires that software validation be performed on the target platform of high-fidelity simulation. A high-fidelity simulation typically uses the same processor type, processor performance and timing, memory size, and interfaces as the flight unit.
Activity Assumptions:	Artifacts will be available as scheduled
Rationale for Approach:	This is an approved method that has been shown to be effective.
Concerns:	Artifacts may have insufficient content for adequate evaluation
Method Application Notes: None identified	
Required Tools:	None
Empirical Evidence:	Scenarios Traces Inspection of Test Support System capability
Output (include updates to Project Technical Reference):	- Issues and risks - Sections in MPSRs - Sections in post analysis reports (e.g. “Capstone” reports)
Basis of Estimate:	
Other:	As a side effect, inconsistencies and deficiencies found in inputs may be found, and documented as issues or observations

Table 25 Activity 14: Validate Test Cases by Inspection and Traces 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.
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 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 9-1 IV&V Technical Framework, Rev M: 4.2 and 4.5
Scope:	Activity applies to all MPCV EM-1/2 software deliveries
Target Artifacts:	Software Test Plan Software Test Description Software Test Reports Each CSCI Software Requirements
Inputs (includes Technical Reference):	Test Plan for the applicable level of test (per TRR) Test cases and developer defined scenarios for the test Test scripts Implemented Source Code for applicable iteration/instantiation Validated requirements to validate the Test Cases against IV&V-generated list of off-nominal conditions Command and telemetry database Parameter database Test Logs (e.g. Integrated Test Report, Datamark report) Requirement to Test Case Traceability

Prerequisites:	Validated Requirements
Success Criteria:	- 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 - Confirmation that the Test Cases under analysis comply with project defined test document purpose, format, and content. - Confirmation that the Test Cases under analysis satisfy the criteria in the associated Test Plan. - 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).
Activity Assumptions:	Artifacts will be available as scheduled
Rationale for Approach:	This is an approved method that has been shown to be effective.
Concerns:	Artifacts may have insufficient content for adequate evaluation
Method Application Notes: None identified	
Required Tools:	Code browsing capability (e.g. Understand C++, Eclipse, etc.)
Empirical Evidence:	- Completion of analysis results spreadsheet - Risks documenting systemic concerns - TIMs documented in ORBIT
Output (include updates to Project Technical Reference):	- Issues and risks - Sections in MPSRs - Sections in post analysis reports (e.g. “Capstone” reports)
Basis of Estimate:	
Other:	May be performed simultaneously with "Validate Test Procedures by Inspection and Traces to Requirements" 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).

Table 26 Activity 16: Validate Test Design by 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.
Required Method Revisions (if any)	None identified
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 9-1 IV&V Technical Framework, Rev N: 4.7, 4.8
Scope:	Activity applies to all MPCV EM-1/2 software deliveries
Target Artifacts:	Software Test Plan Software Test Description Software Test Reports Each CSCI Software Requirements Specification (SRS) Software Requirements Report (SRR)
Inputs (includes Technical Reference):	In-scope requirements (functional, interface) Software design Concept of Operations
Prerequisites:	None
Success Criteria:	A. A set of scenarios has been developed via application of the Three Questions for the critical behaviors that thread together low level software functions (sufficient depth to match test artifacts) B. The operational scenarios to ensure correctness (e.g., scenario walk-through) have been validated C. Requirements trace to scenarios D. Scenarios trace to software structure (package/function) and source lines of code E. Scenarios trace to Test Design and Test Environment; scenarios that do not map to the test design indicate incompleteness of the test design
Activity Assumptions:	Artifacts will be available as scheduled
Rationale for Approach:	This is an approved method that has been shown to be effective.
Concerns:	Artifacts may have insufficient content for adequate

	evaluation
Method Application Notes: Considerable domain knowledge and understanding of software architecture is needed for scenario development.	
Required Tools:	None
Empirical Evidence:	Scenarios Traces
Output (include updates to Project Technical Reference):	- Issues and risks - Sections in MPSRs - Sections in post analysis reports (e.g. “Capstone” reports)
Basis of Estimate:	
Other:	Issues may also be identified during scenario development and tracing

Table 27 Activity 15: Validate Test Procedures by Inspection and Traces 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.
Required Method Revisions (if any)	None identified
Technical Goal:	• Verify that the Test Procedures under analysis comply with project defined test document purpose, format, and content. • Validate that the Test Procedures under analysis satisfy the criteria in the associated Test Plan. • 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.
WBS Coverage:	IVV 9-1 IV&V Technical Framework, Rev N: 4.6
Scope:	Activity applies to all MPCV EM-1/2 software deliveries
Target Artifacts:	Software Test Plan Software Test Description Software Test Reports Each CSCI Software Requirements
Inputs (includes Technical Reference):	Test Plan for the applicable level of test Test Procedures for the associated test plan Implemented or draft test cases, if available Validated requirements to validate the Test Procedures against (relevant set of validated requirements will be iteration/instantiation specific). IV&V-generated list of off-nominal conditions
Prerequisites:	Validated Requirements
Success Criteria:	- Determination that the Test Procedures under analysis comply with project defined test document purpose, format, and content. - Determination that the Test Procedures under analysis satisfy the criteria in the associated Test Plan. -Confirmation 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.
Activity Assumptions:	Artifacts will be available as scheduled
Rationale for Approach:	This is an approved method that has been shown to be effective.
Concerns:	Artifacts may have insufficient content for adequate evaluation
Method Application Notes: None identified	
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):	- Issues and risks - Sections in MPSRs - Sections in post analysis reports (e.g. “Capstone” reports)
Basis of Estimate:	
Other:	May be performed simultaneously with "Validate Test Cases by Inspection and Traces to Requirements"

3.4. TF 5.0 Verify and Validate Design

The methods used for each element item are shown in Table 28. The specific methods are described in

Table 29 through Table 32.

Table 28 TF 5 Element Mapping to Selected Method

TF	Element Narrative	Coverage Comment	Method
5	The objective of Design IV&V is to ensure that the design is a correct, accurate, and complete transformation of the software requirements that will meet the operational need under nominal and off-nominal conditions and that no unintended features are introduced		No method required at the 5 (integer) level. TF 5 is addressed by addressing all of the sub elements of TF 5 listed below.
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.	Covered as indicated above in Table 5 Lifecycle TF Element Coverage Guidance	M-39 v1.3 Verify Software Design by Inspecting Traces to Requirements and Software Architecture M-32 v1.0 Verify Scripted Timeline Via Manual Multi-Directional Tracing M-103 Verify and Validate Requirement Implementation using Flow Diagrams to Uncover Missing, Conflicting, or Unnecessary Behavior
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.	Covered as indicated above in Table 5 Lifecycle TF Element Coverage Guidance	M-39 v1.3 Verify Software Design by Inspecting Traces to Requirements and Software Architecture M-103 Verify and Validate Requirement Implementation using Flow Diagrams to Uncover Missing, Conflicting, or Unnecessary Behavior
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).	Covered as indicated above in Table 5 Lifecycle TF Element Coverage Guidance	M-39 v1.3 Verify Software Design by Inspecting Traces to Requirements and Software Architecture M-28 v1.0 Verify System/Software Architecture Using a Discrete Model of Performance Requirements in Stressing Scenarios M-103 Verify and Validate

TF	Element Narrative	Coverage Comment	Method
			Requirement Implementation using Flow Diagrams to Uncover Missing, Conflicting, or Unnecessary Behavior
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.	Covered as indicated above in Table 5 Lifecycle TF Element Coverage Guidance	M-41 v1.1 Verify Software Interface Design by Inspection Against Interface Requirements
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.	Covered as indicated above in Table 5 Lifecycle TF Element Coverage Guidance	M-39 v1.3 Verify Software Design by Inspecting Traces to Requirements and Software Architecture M-103 Verify and Validate Requirement Implementation using Flow Diagrams to Uncover Missing, Conflicting, or Unnecessary Behavior
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.	Covered as indicated above in Table 5 Lifecycle TF Element Coverage Guidance This method has not been used for TF 5.6 before. See the concerns section of Table 23 for further information on this topic.	M-39 v1.2 Verify Software Design by Inspecting Traces to Requirements and Software Architecture M-57 v1.0 Verify System Software Safety by Comparing Concept Documentation, Requirements, Testing, Design and Code with Hazard Analysis Documentation to Establish a Safety Case, Across the Software Development Life Cycle M-103 Verify and Validate Requirement Implementation using Flow Diagrams to Uncover Missing, Conflicting, or Unnecessary Behavior

Table 29 Activity 17: Verify Software Design by Inspecting Traces to Requirements and Software 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.
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). 5.5 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 Partial: (The tracing of hazards to FSW requirements allows this method to address 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 9-1 IV&V Technical Framework, Rev N: 5.1 (Full), 5.2 (Full), 5.3 (Partial), 5.5 (Partial), 5.6 (Partial)
Scope:	Activity applies to all MPCV EM-1/2 software deliveries

Target Artifacts:	Software Requirements for each CSCI Software Design Document for each CSCI
Inputs (includes Technical Reference):	Validated software requirements and identified issues and risks. Validated system requirements as reference only (as needed).
Prerequisites:	Validation of system and software requirements not including integration requirements
Success Criteria:	- The software design adequately satisfies the software architecture and validated software requirements - The design is correct, complete, and consistent with its set of requirements - Each requirement is correct, complete, and consistent with the design - The software design is consistent across documentation
Activity Assumptions:	If there is software architecture heritage from other missions, this heritage is documented as part of the design (PDR, CDR slides). The S/W architecture is documented as part of the design Artifacts and will be available as scheduled
Rationale for Approach:	This is an approved method that has been shown to be effective.
Concerns:	Artifacts may have insufficient content for adequate evaluation. This method is being used for TF 5.6, but it has never been used for TF 5.6 before. The CoM methods that are available for TF 5.6 involve use of an executable UML model or FM database. The IV&V MPCV project does not have such a UML or database. Attempting to develop executable UML models or database that are proven to represent the software being built would be prohibitively expensive. A method proven effective on other similar TF elements is recommended. A method approved for TFS 5.1 and 5.2 has been selected for this TF. TF 5.1, 5.2 and 5.6 include similar language and engineering review of the method suggests it can be applied to TF 5.6.
Method Application Notes: None identified	
Required Tools:	Excel Worksheets (or other data documentation system), ORBIT DOORS access to requirements
Empirical Evidence:	• Engineering worksheets documenting results. The worksheets should include:

	- the requirements (document, section title, number, description) - traces to design artifacts and identified behaviors - assessment of the software architecture, software design, and software algorithms with respect to the requirement sets and identified behaviors - assessment of the software design with respect to each individual requirement (analyzed across documentation) - additional analyst comments as needed to support assessment. • Issues • Risks and Issues documented in the Design Verification Report
Output (include updates to Project Technical Reference):	- Issues and risks - Sections in MPSRs - Sections in post analysis reports (e.g. “Capstone” reports)
Basis of Estimate:	
Other:	N/A

Table 30 Activity 18: Verify Software Interface Design by Inspection Against 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.
Required Method Revisions (if any)	None identified
Technical Goal:	5.4 Ensure that the internal and external software interface designs are provided 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. 5.0 Ensure that the design is a correct, accurate, and complete transformation of the software requirements that will meet the operational need under nominal and off-nominal conditions and that no unintended features/functionality are introduced, the relevant behaviors meet the criteria of the 3Qs, and the higher-level and module level design represent the L5 software requirements as expected.
WBS Coverage:	IVV 9-1 IV&V Technical Framework, Rev N: 5.0 Verify and Validate Design 5.4 - Interface Analysis – Design
Scope:	Activity applies to all MPCV EM-1/2 software deliveries
Target Artifacts:	Software Requirements for each CSCI Software Design Document for each CSCI Software Interface Control Document
Inputs (includes Technical Reference):	List of validated interface requirements and identified issues and risks
Prerequisites:	Validation of the interface requirements
Success Criteria:	The interface design is correct, complete, and consistent with architecture and requirements
Activity Assumptions:	Artifacts will be available as scheduled and will be consistent
Rationale for Approach:	This method was chosen because it is operational and proven and meets the needs of the associated TF.
Concerns:	None identified
Method Application Notes:	

None identified	
Required Tools:	Excel Worksheets (or other data documentation system), ORBIT
Empirical Evidence:	• Engineering worksheets, databases, etc. documenting the results and comments of the requirements to design trace and the design to requirements trace. • TIMs and Observations • Risks and findings documented in Interface Design Verification Report
Output (include updates to Project Technical Reference):	- Issues and risks - Sections in MPSRs - Sections in post analysis reports (e.g. “Capstone” reports)
Basis of Estimate:	
Other:	N/A

Table 31 Activity 24: Verify Scripted Timeline Via Manual Multi-Directional Tracing

Method:	M-32, Version 1.0 (Current Status: Approved)
Method Title:	Verify Scripted Timeline Via Manual Multi-Directional Tracing
Method Synopsis	Method for analysis of timeline engine and functions implemented using a timeline engine by multi-way trace and consistency check of requirement and design artifacts. Defects discovered by this Method include in-scope requirements with invalid or missing traces to corresponding Timeline Action Definitions, "orphaned" or invalid Timeline Action Definitions (those with no valid traceable link to in-scope requirements), undefined Timeline Events, and unscheduled Timeline Action Definitions.
Required Method Revisions (if any)	None identified
Technical Goal:	Analysis provides assurance the design, relative to the actions/timeline/commands (design), correctly and completely meets the validated reqts 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. 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.
WBS Coverage:	IVV 09-1 Rev N : 5.1 & 5.2 with respect to scripted timeline
Scope:	Applies to data (and executable as necessary) planned for the control of vehicle sequences. This is one of a small number of methods aimed at data tables driving FSW execution.
Target Artifacts:	Avionics Design Description Volume 7 (and Volumes 1 to 6 if necessary) FSW Data FSW Design Documents FSW Requirements
Inputs (includes Technical Reference):	Validated FSW requirements
Prerequisites:	Validation of FSW requirements
Success Criteria:	Demonstration that the FSW can reliably perform required capabilities under nominal (and more importantly) off nominal conditions.

	Demonstration that no undesired behaviors are manifested via omitted steps or unplanned steps inserted into the timeline.
Activity Assumptions:	Necessary materials are available.
Rationale for Approach:	The approach is approved and provides the necessary Technical Framework coverage.
Concerns:	None
Method Application Notes: None	
Required Tools:	None
Empirical Evidence:	Engineering worksheets documenting results of verifying and validating design adequately meets reqts. Limitations identified in the form of risks, issues, or observations.
Output (include updates to Project Technical Reference):	Issues and Risks Input for Design and Implementation analysis reports
Basis of Estimate:	
Other:	- If code implementing the timeline is not auto-generated, then this method can be extended to include tracing the code to and verifying consistency with the other artifact types. If so, FSW code artifacts implementing the scripted timeline becomes an additional Analysis Target. Also, the following technical goals are partially accomplished: - 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. - 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. - 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. - If code artifacts are to be analyzed, Analysis Steps 1, 2, and 3 are modified to also trace to and verify consistency with the Timeline Implementation (i.e., code), and Analysis Step 4) is added as follows: 4) "Timeline Implementation Pass" - Timeline Implementation (i.e., code) is traced to and verified for consistency with Timeline Engine Documentation,

	Timeline Action Description, and then with Reqts. - If code artifacts are to be analyzed, Empirical Evidence will also include: Engineering worksheets documenting results of verifying and validating code adequately meets reqts. and design. - Method can be performed against unvalidated requirements, but If so it will need to be re-performed against the validated requirements.

Table 32 Activity 19: Verify System/Software Architecture Using a Discrete Model of Performance Requirements in Stressing Scenarios

Method:	M-28, Version 1.0 (Current Status: Approved)
Method Title:	Verify System/Software Architecture Using a Discrete Model of Performance Requirements in Stressing Scenarios
Method Synopsis	Use stressing scenarios executed in a discrete, event- and time-based model of the performance characteristics, as implemented in an Excel spreadsheet, determine if the architecture will support the design and capacity limits of the system.
Required Method Revisions (if any)	None
Technical Goal:	Verify the system/software architecture (the top-level system/software design) will meet all of the system/software performance requirements, e.g. CPU loading, Bus loading, Message throughput, Uplink and downlink throughput, Storage, Timing, Command processing. 2.2 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. [Performance Requirements only] 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). [Performance Requirements only]
WBS Coverage:	2.2 (partial), 5.3 (partial)
Scope:	Intended to address timing issues at the partition level (or ARINC-653 level) or the movement of data within the system generally.
Target Artifacts:	Software design document Avionics Design Description Volume 7 (or Volumes 1 through 7 as needed) Software Requirements
Inputs (includes Technical Reference):	Operational Scenarios Performance requirements and characteristics of the architecture
Prerequisites:	Development of mission scenarios suitable for elaboration in assessing system or software architecture against defined performance requirements.

	Scenarios should cover Q1/Q2/Q3 perspectives.
Success Criteria:	Architecture and timing allocations are complete and allow for the planned execution of the FSW within the time slices provided.
Activity Assumptions:	Artifacts will be available and scheduled and provide a consistent and accurate description of the timing.
Rationale for Approach:	Method is approved and meets required analysis areas.
Concerns:	None identified
Method Application Notes: TF 5.3 will be the only Technical Goal performed for this analysis. TF 2.2 will not be performed for this analysis.	
Required Tools:	Spreadsheet to model performance requirements and architecture elements to be analyzed, and scenario to be examined, Discrete Performance Model Sheet (DPMS)
Empirical Evidence:	Completed DPMS with scenario, modeled performance requirements, graph of capacity usage for each modeled component indicated where (if) design or capacity limits are exceeded
Output (include updates to Project Technical Reference):	Issues and risks Support to architecture and design analysis reports
Basis of Estimate:	
Other:	Application to OS, I/O, and Telecom is pretty straightforward Application to domains highly dependent on mathematics or physics (e.g. GN&C, thermal) is less straightforward - The method may still be applicable - For such domains, this method will drive out performance peaks, but must be supplemented with other analysis approaches - This approach can still add value even without such detailed supplemental analysis

Table 33 Activity 27: Verify GN&C Software Design by Inspecting Traces to Requirements and Software Architecture

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	
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). 5.5 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 Partial: (The tracing of hazards to FSW requirements allows this method to address 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 9-1 IV&V Technical Framework, Rev N: 5.1 (Full), 5.2 (Full), 5.3 (Partial), 5.5 (Partial), 5.6 (Partial)
Scope:	Activity applies to GN&C MPCV EM-1/2 software deliveries.

Target Artifacts:	Software Requirements for each CSCI Software Design Document for each CSCI
Inputs (includes Technical Reference):	Validated software requirements and identified issues and risks. Validated system requirements as reference only (as needed).
Prerequisites:	Validation of system and software requirements not including integration requirements
Success Criteria:	- The software design adequately satisfies the software architecture and validated software requirements - The design is correct, complete, and consistent with its set of requirements - Each requirement is correct, complete, and consistent with the design - The software design is consistent across documentation
Activity Assumptions:	If there is software architecture heritage from other missions, this heritage is documented as part of the design (PDR, CDR slides). The S/W architecture is documented as part of the design Artifacts will be available as scheduled
Rationale for Approach:	This is an approved method that has been shown to be effective.
Concerns:	None
Method Application Notes: None identified	
Required Tools:	Excel Worksheets (or other data documentation system), ORBIT DOORS access to requirements
Empirical Evidence:	• Engineering worksheets documenting results. The worksheets should include: - the requirements (document, section title, number, description) - traces to design artifacts and identified behaviors - assessment of the software architecture, software design, and software algorithms with respect to the requirement sets and identified behaviors - assessment of the software design with respect to each individual requirement (analyzed across documentation) - additional analyst comments as needed to support assessment. • Issues • Risks and Issues documented in the Design

	Verification Report
Output (include updates to Project Technical Reference):	- Issues and risks - Sections in MPSRs - Sections in post analysis reports (e.g. "Capstone" reports)
Basis of Estimate:	
Other:	N/A

3.5. TF 6 Verify and Validate Implementation

The methods used for each element item are shown in Table 34. The specific methods are described in Table 35 through Table 40.

Table 34 TF 6 Element Mapping to Selected Methods

TF	Element Narrative	Coverage Comment	Method
6	The objective of Implementation IV&V is to verify and validate that these transformations are correct, accurate, and complete, yielding source code that correctly implements requirements, meets the operational need under nominal and off-nominal conditions, and introduces no unintended features.		No method required at the 6 (integer) level. TF 6 is addressed by addressing all of the sub elements of TF 6 listed below.
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.	Covered as indicated above in Table 5 Lifecycle TF Element Coverage Guidance	M-13 v1.1 Verify SW Interface Implementation by Inspection Against Interface Design M-20 v1.0 Verify Critical Software Changes By Inspecting Change Requests
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.	Covered as indicated above in Table 5 Lifecycle TF Element Coverage Guidance	M-38 v1.2 Verify Software Implementation by Inspecting Traces to Requirements M-9 v1.2 Verify Software Code Quality using Static Analysis Tools M-57 v1.0 Verify System Software Safety by Comparing Concept Documentation, Requirements, Testing, Design and Code with Hazard Analysis

TF	Element Narrative	Coverage Comment	Method
			Documentation to Establish a Safety Case, Across the Software Development Life Cycle M-14 v1.2 Verify Software Behavior for Off-Nominal Conditions using Independent Testing
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.	Covered as indicated above in Table 5 Lifecycle TF Element Coverage Guidance	M-13 v1.1 Verify SW Interface Implementation by Inspection Against Interface Design
6.4	Ensure that test results are as expected (per the corresponding plans, cases, procedures, design) and the impacts of any discrepancies are understood.	Covered as indicated above in Table 5 Lifecycle TF Element Coverage Guidance	M-11 v1.2 Verify Test Execution by Inspection of Test Cases, Inputs and Results
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.	Covered as indicated above in Table 5 Lifecycle TF Element Coverage Guidance	M-38 v1.2 Verify Software Implementation by Inspecting Traces to Requirements M-57 v1.0 Verify System Software Safety by Comparing Concept Documentation, Requirements, Testing, Design and Code with Hazard Analysis Documentation to Establish a Safety Case, Across the Software Development Life Cycle
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.	Covered as indicated above in Table 5 Lifecycle TF Element Coverage Guidance	M-38 v1.2 Verify Software Implementation by Inspecting Traces to Requirements

Table 35 Activity 20: Verify Critical Software Changes By Inspecting Change Requests

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.

Required Method Revisions (if any)	None. "Pre-verified system" is applied to this method to mean subjected to earlier (completed) IV&V verification analyses. This is not associated with Program verification status.
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:	3.3 (partial) Software requirements evaluation as applicable to change(s) only 6.1 (partial) changes only
Scope:	Activity applies to all MPCV EM-1/2 software change requests when the requests are not to be addressed as part of larger release reviews.
Target Artifacts:	Development Integration Issue Records (DIIRs), Software Requests (SR), and similar DCR-like software change documentation.
Inputs (includes Technical Reference):	Software Change Records including DIIRs, SRs, or DCRs
Prerequisites:	Pre-verified system
Success Criteria:	• All intended changes have been correctly implemented or the defects have been identified. • No unintended changes occurred. • Analysis results are documented in ORBIT issues and summarized in post analysis reports (if applicable). • Risks
Activity Assumptions:	None identified.
Rationale for Approach:	Selected to cover changes when complete FSW analysis is not appropriate. Method is effective at addressing that niche.
Concerns:	None
Method Application Notes: None	
Required Tools:	No special IV&V Tools IV&V analyses on the MPCV Program require access to "DOORS", "Rhapsody", "Synergy" and "MATLAB" to gain access to the artifacts and see the data in native formats.
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):	- Issues and risks - Sections in MPSRs - Sections in post analysis reports (e.g. "Capstone" reports)

Basis of Estimate:	
Other:	None

Table 36 Activity 21: Verify Test Execution by Inspection of Test Cases, 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.
Required Method Revisions (if any)	None identified
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 9-1 IV&V Technical Framework, Rev M: 6.4, 6.2 (Partial)
Scope:	Activity applies to all MPCV EM-1/2 software deliveries
Target Artifacts:	Test Cases, Test Inputs and Test Results
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:	• Determination that all tests expected to be executed were executed and documentation of the results is complete. • Determination that the test execution utilized the approved inputs and initial conditions. • Determination that the test produced the expected outputs for the requirements and associated behaviors being tested. • All test results match the test output. • All discrepancies are understood

	• All failed tests have a DCR associated with them. • Determination that any open requirements issues/risks are considered. • Determination that all open test documentation issues/risks are considered.
Activity Assumptions:	None
Rationale for Approach:	The approach works as part of strategy to provide full coverage of TF 6.4.
Concerns:	None
Method Application Notes: TF 6.4 will be the only Technical Goal performed for this analysis. TF 6.2 will not be performed for this analysis.	
Required Tools:	No special IV&V Tools IV&V analyses under "Verify Test Execution by Inspection of Test Cases, Inputs and Results" on the MPCV Program will require access to ELAB, Code Collaborator, Lockheed Martin NT, and related tools. Most tools use a web based thin client interface.
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):	- Issues and risks - Sections in MPSRs - Sections in post analysis reports (e.g. "Capstone" reports)
Basis of Estimate:	
Other:	Because test cases, procedures, and designs may be developed iteratively and at multiple levels, IV&V task iteration may be necessary.

Table 37 Activity 25: Verify SW Interface Implementation by 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.
Required Method Revisions (if any)	None Will be performed as part of integrated design and implementation analysis.
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 9-1 IV&V Technical Framework, Rev N: 6.1 - (Partial) Traceability Analysis – Implementation 6.3 - (Full) Interface Analysis - Implementation
Scope:	EM-1/2 interface design for priority CSCIs.
Target Artifacts:	Interface Design Documents Interface Control Documents Rhapsody and Matlab Models FSW source code.
Inputs (includes Technical Reference):	Verified Interface Design
Prerequisites:	N/A
Success Criteria:	• All design elements have been correctly implemented or the defects have been identified. • No unintended changes occurred. • Analysis results are documented in ORBIT issues and summarized in post analysis reports (if applicable). • Risks
Activity Assumptions:	None identified.
Rationale for Approach:	The approach works as part of a strategy to provide full

	coverage of TF 6.1 and 6.3
Concerns:	None
Method Application Notes: None	
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):	- Issues and risks - Sections in MPSRs - Sections in post analysis reports (e.g. “Capstone” reports)
Basis of Estimate:	
Other:	N/A

Table 38 Activity 22: Verify Software Implementation by Inspecting Traces to Requirements

Method:	M-38, Version 1.2 (Current Status: Approved)
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.
Required Method Revisions (if any)	None identified
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 6.5 (Partial) 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 6.6 (Partial) Ensure that all (in-scope) requirements (e.g. SRS and IRS) are represented in the appropriate source code/script components and that the source code/script does not introduce capability that is not required.
WBS Coverage:	IVV 9-1 IV&V Technical Framework, Rev N: 6.2 - (Full) Source Code and Documentation Evaluation 6.5 - (Partial) Software Hazard Tolerance Analysis (partial) 6.6 - (Full) Traceability Analysis – Requirements-&-Implementation

Scope:	In scope CSCIs' implementation.
Target Artifacts:	Software source code Software requirements Traceability Matrix
Inputs (includes Technical Reference):	Validated requirements, Source Code/Script Elements
Prerequisites:	None
Success Criteria:	Appropriate analysis scope is established Trace condition is accurately documented between each requirement verified, and the associated SW task, the targeted IV&V PBRA/RBA SW behavior, Code file, Code line, and Description of Code. Order/sequence dependent requirements are implemented correctly.
Activity Assumptions:	Artifacts are provided with sufficient time to permit meaningful analysis.
Rationale for Approach:	The method complements other analyses to be performed, fills in necessary aspects of code evaluations, and has no significant required prerequisites (such as SRM development). Verify Software Implementation by Inspecting Traces to Requirements will not be the only method applied to perform IV&V Framework Element 6.2. Verify Software Implementation by Inspecting Traces to Requirements will be combined with “Verify Test Execution by Inspection of Test Cases, Inputs and Results” and “Static Analysis” to provide appropriate coverage of TF element 6.2.
Concerns:	None
Method Application Notes: MPCV applies auto code generation. Traces to design within the auto coder environment are functionally identical to traces to source code. MPCV IV&V will apply auto code design in the same fashion as source code where that choice is appropriate.	
Required Tools:	Code parser (such as Eclipse, Understand for C++) IV&V implementation and design analyses on the MPCV Program require access to “DOORS”, “Rhapsody”, “Synergy” and “MATLAB” to gain access to the artifacts and see the data in native formats.
Empirical Evidence:	• Engineering worksheets, database, etc. documenting results of tracing requirements to code/script elements and comments confirming the correct implementation of the requirements o 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 • 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
Output (include updates to Project Technical Reference):	- Issues and risks - Sections in MPSRs - Sections in post analysis reports (e.g. “Capstone” reports)
Basis of Estimate:	
Other:	N/A

Table 39 Activity 23: Verify Software Code Quality using Static Analysis Tools

Method:	M-9, Version 1.2 (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 syntax and other code errors, including (but not restricted to) buffer overflows, use of uninitialized variables, multiple definitions of functions or constants, and unused code. The static code analyzer(s) generate candidate findings from a build of code, which are filtered to ignore undesired types of errors, and the remaining results manually reviewed to determine the final set of reportable flaws in the build. Issues reported from static code analysis of earlier builds are also assessed and dispositioned as necessary.
Required Method Revisions (if any)	None identified
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 9-1 IV&V Technical Framework, Rev M: 6.2 (Partial)
Scope:	In scope CSCIs implementation.
Target Artifacts:	MPCV source code.
Inputs (includes Technical Reference):	Project-specific coding standards. Code Quality Characteristics
Prerequisites:	None
Success Criteria:	Analyzing the code using appropriate tool(s). Filtering the results to identify defects with significant potential to impact mission execution reliability (including safety and code quality). Effective analysis of filtered results to establish actual impacts.
Activity Assumptions:	None
Rationale for Approach:	Static code analysis can provide statistically significant analysis of large segments of flight code using defined standards of source code defects. This automated process provides a unique ability to evaluate large segments of code

	objectively. This objective evidence can provide meaningful objective insight into code development processes. Static code analysis supports the complier and coding tools used by the project. Static code analysis provides limited value for achieving all elements of TF 6.2, and MPCV IV&V will also perform two additional methods "Verify Test Execution by Inspection of Test Cases, Inputs and Results" and "Verify Software Implementation by Inspecting Traces to Requirements" to provide full coverage of TF 6.2. All three methods complement each other in achieving the aims of TF 6.2.
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):	- Issues and risks - Sections in MPSRs - Sections in post analysis reports (e.g. "Capstone" reports)
Basis of Estimate:	
Other:	None

Table 40 Activity 26: Verify Software Behavior for Off-Nominal Conditions using Independent Testing

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.
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. 6.6 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 9-1 IV&V Technical Framework, Rev N: 6.2 (partial), 6.6 (partial)
Scope:	Targeted activities within high priority (e.g. "Red" entities and "Red" capabilities will be subjected to dynamic testing when other analyses suggest that testing is appropriate).
Target Artifacts:	Implementation/FSW code
Inputs (includes Technical Reference):	In-scope requirements
Prerequisites:	None
Success Criteria:	Demonstration that FSW code components reliably perform required capabilities under nominal and off-nominal conditions. Demonstration that no undesired behaviors are manifest during testing, where the undesired behaviors preclude intended test results.
Activity Assumptions:	None
Rationale for Approach:	Complements static analysis and engineering analysis of implementation.
Concerns:	None
Method Application Notes: None	

Required Tools:	Mechanism to execute software via validated simulation and/or test environment **< redacted>** IV&V JSTAR Lab
Empirical Evidence:	Independent Test Scenarios, Test Procedures, Test Scripts, Test Execution Logs, Test-Requirements Traces
Output (include updates to Project Technical Reference):	- Issues and risks - Sections in MPSRs - Sections in post analysis reports (e.g. “Capstone” reports)
Basis of Estimate:	
Other:	This method can be executed at any level, i.e. at the system level, a component, a CSCI, etc. Additional rigor may be realized over manual inspection techniques alone May use IV&V's testing facility, or the customer’s facility.

3.6. TF 7 Verify and Validate Operations and Maintenance Content

There are no identified TF 7.0 activities during FY 2016 in TF 7.1, 7.2, 7.3, 7.5, 7.6, 7.7, and 7.8. These areas will be monitored and new activities' technical scope and rigor will be addressed as quickly as possible after the activities' identification. EFT-1 flight test data will be evaluated as part of EM-1 FSW development as if EFT-1 was a very complex but very limited EM-1 FSW test.

There is no current Catalog of Methods approach to satisfaction of 7.4 that is appropriate to the analysis of test flight data. CoM Method 11 "Verify Test Execution by Inspection of Test Cases, Inputs and Results", where the test flight is considered a FSW test, comes closest to full satisfaction of 7.4.

Table 41 TF 7 Element Mapping to Selected Methods

TF	Element Narrative	Coverage Comment	Method
7.1	Ensure that the disaster recovery plan is adequate to restore critical operation of the system in the case of an extended system outage.	No work scheduled in FY16	None
7.2	Ensure deployment readiness and operational readiness of the software.	No work scheduled in FY16	None
7.3	Ensure that the operating procedures are consistent with the user documentation and conform to the system requirements.	No work scheduled in FY16	None
7.4	Ensure that the effect of software operation anomalies are understood and appropriately addressed.	EFT-1 test flight data will be reviewed as if EFT-1 was a very complex but very limited test of EM-1 FSW. This work could be considered part of 6.4 or 7.4.	Modified M-11 v1.2 Verify Test Execution by Inspection of Test Cases, Inputs and Results
7.5	Ensure that training documentation provides adequate guidance to system operators and users to enable correct use of the system and that this documentation is consistent with the system design and implementation.	No work scheduled in FY16	None
7.6	Ensure that the software requirements and implementation address 1) specific migration requirements, 2) migration tools, 3) conversion of software products and data, 4) software archiving, 5) support for the prior environment, and 6) user notification.	No work scheduled in FY16	None

TF	Element Narrative	Coverage Comment	Method
7.7	For software retirement, ensure that the installation package addresses: 1) software support, 2) impact on existing systems and databases, 3) software archiving, 4) transition to a new software product, and 5) user notification.	No work scheduled in FY16	None
7.8	Ensure that user documentation is consistent with the implementation and capable of communicating the use of user-accessible system functions.	No work scheduled in FY16	None

Table 42 Activity 28: Assurance that the Software Operations, Anomalies and Contingencies are known

Method:	M-11, Version 1.2 (Current Status: Approved) (Needs to be modified to address TF 7.4)
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
Required Method Revisions (if any)	Traces of requirements to on flight behaviors are unlikely to be directly provided by the MPCV program except in cases of substantial anomalies. IV&V will have to draw analogous traces using preflight test cases and flight behaviors performed either successfully or unsuccessfully. Analysis Steps A and B do not apply except to establish the value of performing all other steps. E.g. Failure to perform planned attached flight could establish "inputs" that exceed approved limits and invalidate further analysis.
Technical Goal:	Ensure that the effect of software operation anomalies are understood and appropriately addressed.
WBS Coverage:	6.2 and 6.4. 7.4 coverage is a novel reuse of the method for EFT-1 test flight data analysis. 7.4 coverage of EFT-1 flight test data as part of EM-1 is also novel and could be considered just a complex preflight test for EM-1 just like all other preflight tests. Selection of 7.4 over 6.4 for

	planning is arbitrary.
Scope:	Targeted activities within high priority (e.g. "Red" entities and "Red" capabilities. Additional analyses may be performed as incidental analysis of flight data or MPCV Program flight test reports.
Target Artifacts:	Test Flight Results
Inputs (includes Technical Reference):	Test Flight Data Test Flight Reports Test Flight Anomaly Reports Test Flight Objectives (in lieu of planned test case) Test Flight Timeline (in lieu of the actual test case) Preflight tests (in lieu of traces of requirements to flight behaviors)
Prerequisites:	Preflight tests are complete and traces to requirements are available for use to establish traces to flight nominal and off nominal flight behaviors.
Success Criteria:	Impact of test flight anomalies on EM-1 FSW development is understood, traced to FSW, and documented. Identification of flight anomalies within Program reports or test flight data and establishing a traced impact to EM-1 FSW requirements, design or implementation.
Activity Assumptions:	Necessary artifacts are available within the necessary review period.
Rationale for Approach:	There is no current Catalog of Methods approach to satisfaction of 7.4 that is appropriate to the analysis of test flight data. CoM Method 11 "Verify Test Execution by Inspection of Test Cases, Inputs and Results", where the test flight is considered a FSW test, comes closest to full satisfaction of 7.4 for the use of EFT-1 FSW on the EM-1 flight.
Concerns:	Flight test documentation may not equal STR/STD level of details.
Method Application Notes:	
Required Tools:	None (normal office products are assumed: pdf readers, spreadsheets, word processors, etc.)
Empirical Evidence:	• Engineering worksheets, database, etc. documenting results of tracing requirements to planned flight test behaviors (test objectives) • Engineering worksheets documenting that test flight results were 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):	Issues and risks Test Flight Report
Basis of Estimate:	Engineering flight test reports were used as BOE
Other:	