

MSL IV&V Unified Analysis Process

(Our “*Best Practice*” from a Technical Analyst’s perspective)

Rich Kowalski

Richard.E.Kowalski@ivv.nasa.gov

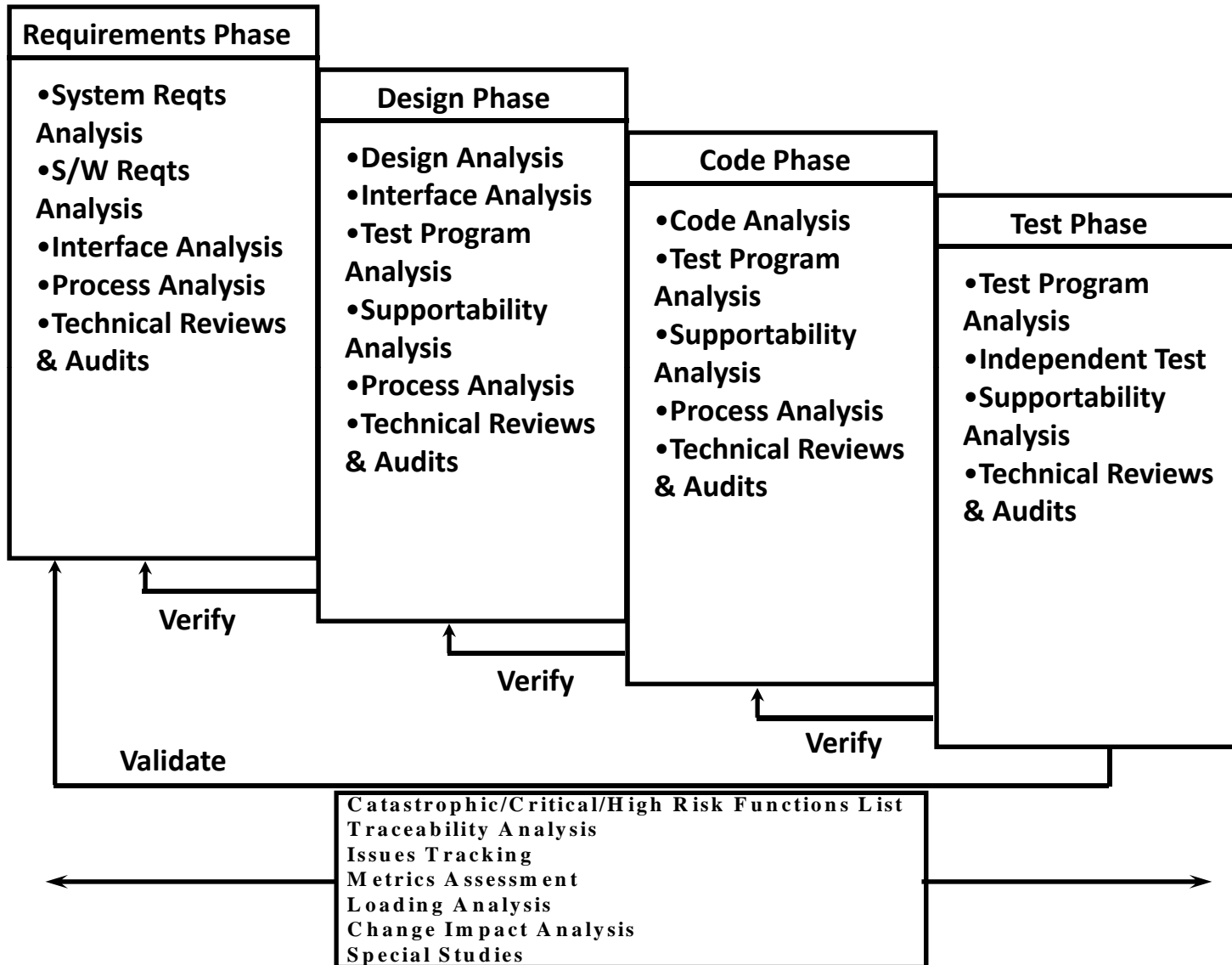
Agenda

- Overview
- MSL IV&V Systems Lifecycle (*4 main phases*)
- MSL IV&V Unified Analysis Goals
- MSL IV&V Unified Analysis Workflow Tasking
- MSL IV&V Unified Analysis Case Studies
- MSL IV&V Unified Analysis Tracking Tool
- Summary
- Questions

Overview

This presentation pertains to a *unified paradigm* for the verification and validation of MSL software and systems engineering artifacts. This paradigm relies on an established *synergy between seven salient workflows*, which are FDD requirements analysis, FDD design analysis, MSL code analysis, MSL test analysis, MSL fault protection analysis, MSL monitor mining, and MSL code mining. To *illustrate the accomplishment* of our results, we have produced *flowcharts illustrating the processing involved in each workflow*. We also provide *eight case studies* in the MSL IV&V Unified Analysis Process workflow's flowchart diagram to demonstrate the benefits of our methodology. The MSL Analysis Tracking Java application is the main *technical tool utilized* to unify the analysis of all the workflows.

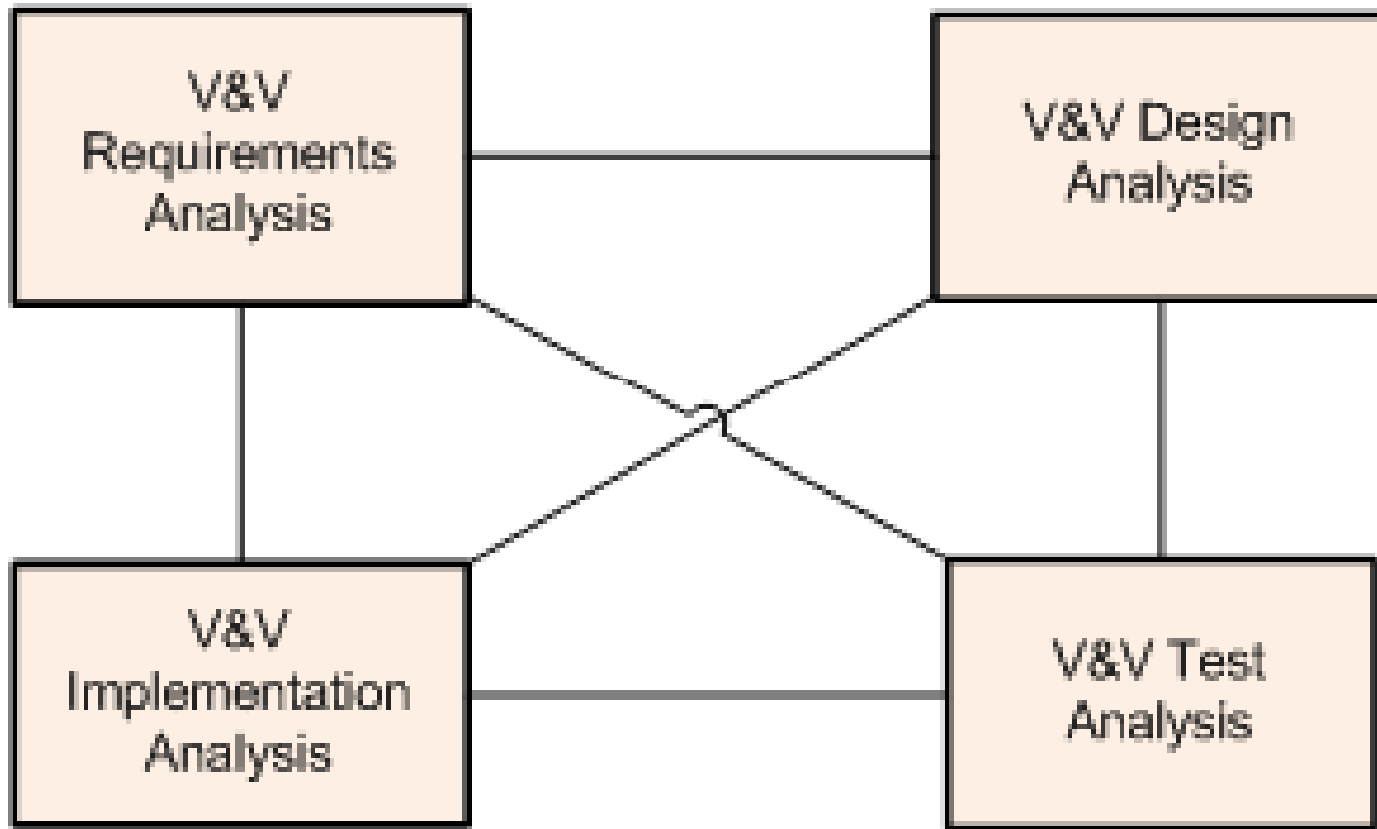
MSL IV&V Systems Lifecycle



What is Unified Analysis Process?

- MSL Analysis Philosophy - MSL is a *complex mission* with *extremely large* (~3M SLOC) and *complex software*.
- A *unified paradigm* for the V&V of MSL software and systems engineering artifacts.
 - *Eliminate isolated phase only analysis!*
- Utilize all phases of the systems lifecycle as a *whole* to *detect and resolve issues*. All phases work together to perform IV&V instead of being isolated.
- This paradigm relies on an established *cooperation between all phase workflows* to facilitate a brisk issue detection and resolution process.
 - *Greater value and less expense; better ROI!*

MSL V&V Unified Analysis Context Diagram



The four analysis phases collaborate with each other to detect and resolve issues.

MSL Unified Analysis Process Goals

- Formulate a **coherent** and **understandable** approach to V&V throughout the product's system/software lifecycle.
- **Simplify** and **unify** V&V methodology for performing V&V analysis throughout the system/software lifecycle.
- Eliminate **isolated** separate-phase-only **procedures** for workflow analysis and produce interaction amongst phase workflows.
- Exploit the **cooperation** among the V&V **procedures** of each lifecycle phase to ensure requirements are correctly mapped in each phase.
- Perform the Unified Analysis process in main **MSL Spacecraft modes** - *Cruise, EDL, Surface Ops*.
- Improve IV&V **performance** and meet time constraints.
- Improve IV&V **process maturity** level and **knowledge gain** for future growth.
- Facilitate a **brisk issue detection and resolution** process.
- Produce better measured product value by **eliminating fault-slip-through**.
- Identify and resolve high-risk issues early in the software life-cycle to **save time and money (better ROI)**.

MSL Unified Analysis Tasking Workflows

- *7 MSL Workflows (per lifecycle phase):*
 - *FDD Requirements Analysis*
 - System and Software Requirements
 - *FDD Design Analysis*
 - System and Software Architecture
 - System and Software Design
 - *Code Analysis*
 - C / C++ & Autocode (*Python script*)
 - *Test Analysis*
 - *System Test Procedures*
 - *Cross-cutting Fault Protection (FP)*
 - *Fault Protection Analysis*
 - *Monitor Mining Analysis*
 - *Code Mining Analysis*

FDD Requirements Analysis

- Validate that system and software requirements are unambiguous, correct, complete, consistent and verifiable.
 - *Pass the high-quality checks*
- Perform requirement goodness checks to determine issues.
- Verify the ‘3 questions’ per requirement.
 - *Is there FDD ‘objective evidence’? If so, verify evidence! Q2 is prime!*
- Check requirement for consistency within phase artifacts:
 - *Same text*
 - *Requirement missing in some artifact versions*
 - *Requirement deleted but still present in some artifact versions*
- No flow down and traceability checks of MSL requirements!
 - *In DOORS, FDDs, SQL Server DB, IVV Test Analysis Database, and Release Plan (official requirements)*

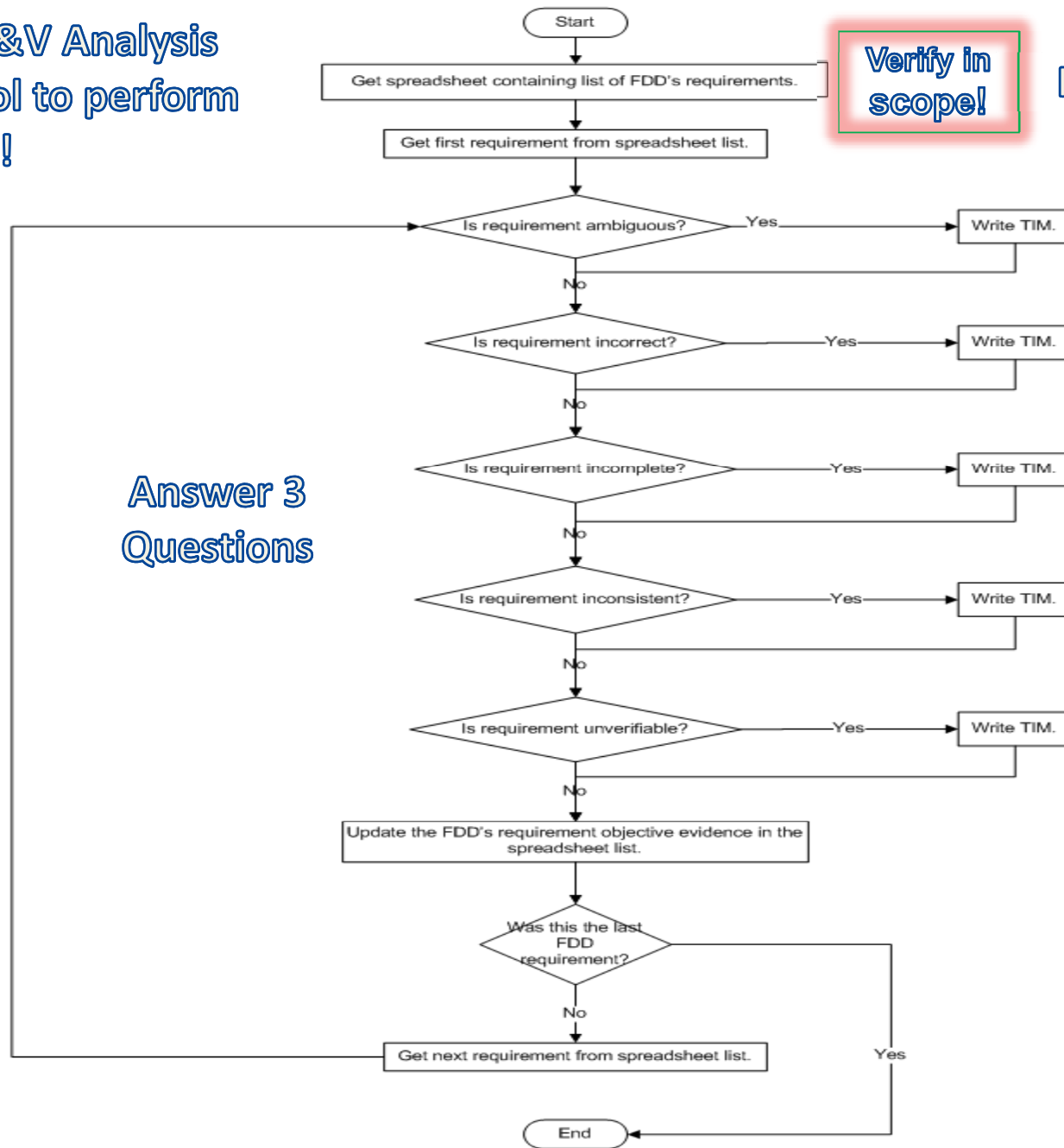
The Three Questions

- Question 1: Will the system's software **do what** it is **supposed to do** under nominal conditions?
- *Question 2:* Will the system's software **not do what** it is **not supposed to do** under off-nominal conditions?
- Question 3: Will the system's software **respond** as expected **under adverse conditions** to introduced unintended features?

FDD Requirements Analysis Workflow

Use MSL IV&V Analysis Tracking tool to perform assessment!

Use Release Plan for Requirements!



Answer 3 Questions

Problem detection and solving loop!

'Quality Attributes'

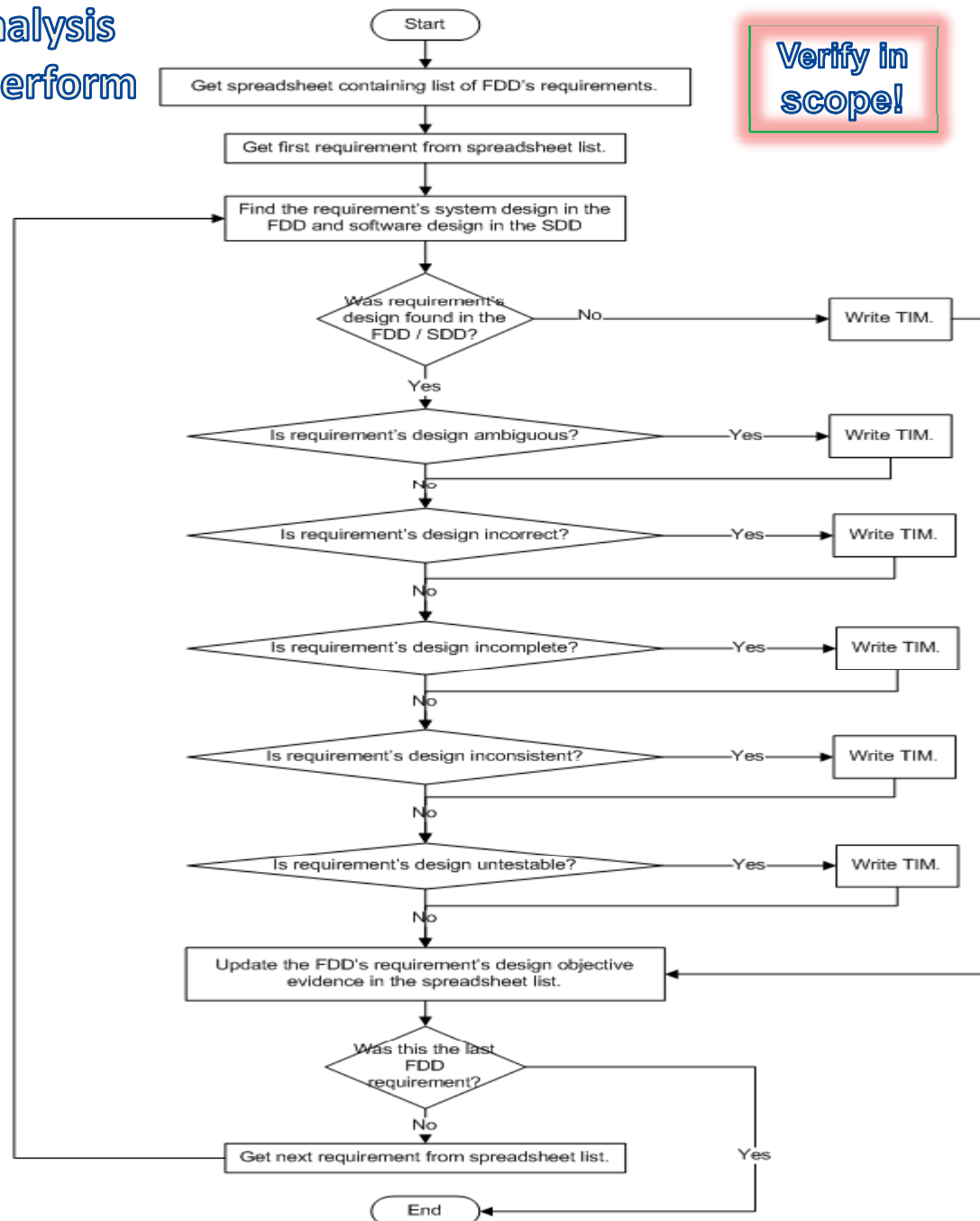
In scope? Testable?

FDD Software Design Analysis

- Does the design support the requirements?
- Verify that system and software design is unambiguous, correct, complete, consistent and verifiable per requirement. (*high-quality checks*)
- Perform requirement's design goodness checks to determine issues (*TIMs*).
- Verify the '3 questions' per design requirement.
- Does the design have any characteristics that will cause it to fail under operational scenarios?
What solutions are appropriate for TIM?

FDD Design Analysis Workflow

Use MSL IV&V Analysis Tracking tool to perform assessment!



Verify in scope!

Use Release Plan for Requirements!

'Quality Attributes'

In scope? Testable?

Answer 3 Questions

Semantic Code Analysis

- Does the *code reflect the design*?
- Is the *code logically correct and abides by coding standards*?
- Verify that the code is correct, complete, maintainable and verifiable. (*high-quality checks*)
- Verify the '3 questions' per requirement implementation. (*Q2 is prime!*)
- Ensure *requirements traceability to code*.
 - *Code must implement requirement as stated*
- *Semantically analyze* selected code, unit test, and system tests and results to verify full coverage of requirement logic paths, range of input (boundary) conditions, error handling, etc.

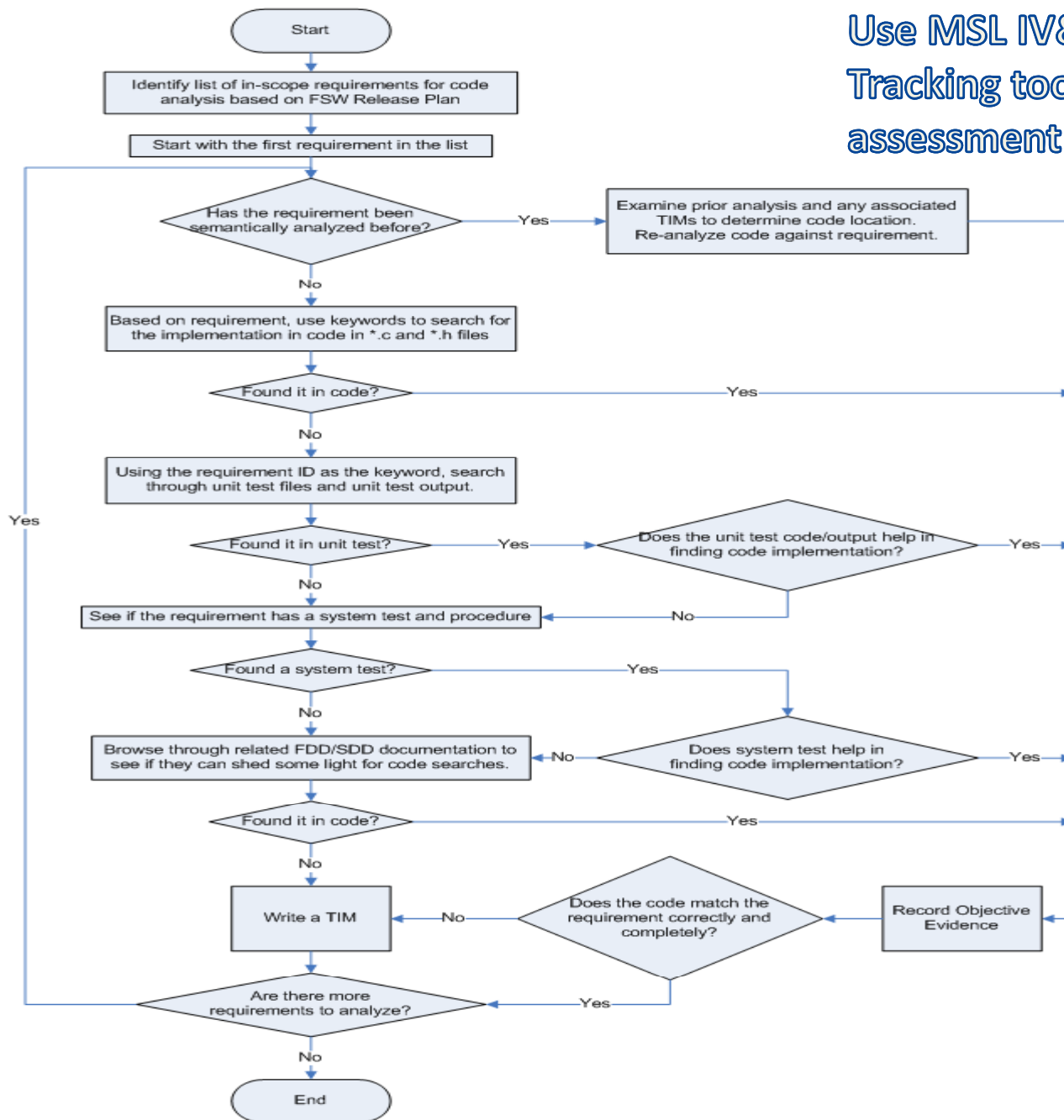
Syntactic Code Analysis

- Use automated code analysis (i.e., *Flexelint* and *Klocwork*) tools to syntactically verify source code:
 - *Ensure language correctness*
 - *Syntactical correctness is the base layer of verification*
 - *Requires strong knowledge of the language*
- Use *Coding Standards* when performing static (syntactic) code analysis. Code should abide by the coding standards.
- Large numbers of *false positives (WARNINGS)* are winnowed manually.
- *Error types:*
 - *Uninitialized variables (when accessed)*
 - *Unused variables (defined)*
 - *Loss of precision on type conversion (double to float)*
 - *Lack of return values (from non-void function calls)*
 - *Dereference of possibly <NULL> pointer (**MAJOR ERROR**)*
 - *No check on buffer or array size (boundary condition)*
 - *Unreachable code*

Semantic Analysis Workflow

Use MSL IV&V Analysis Tracking tool to perform assessment!

Answer 3 Questions



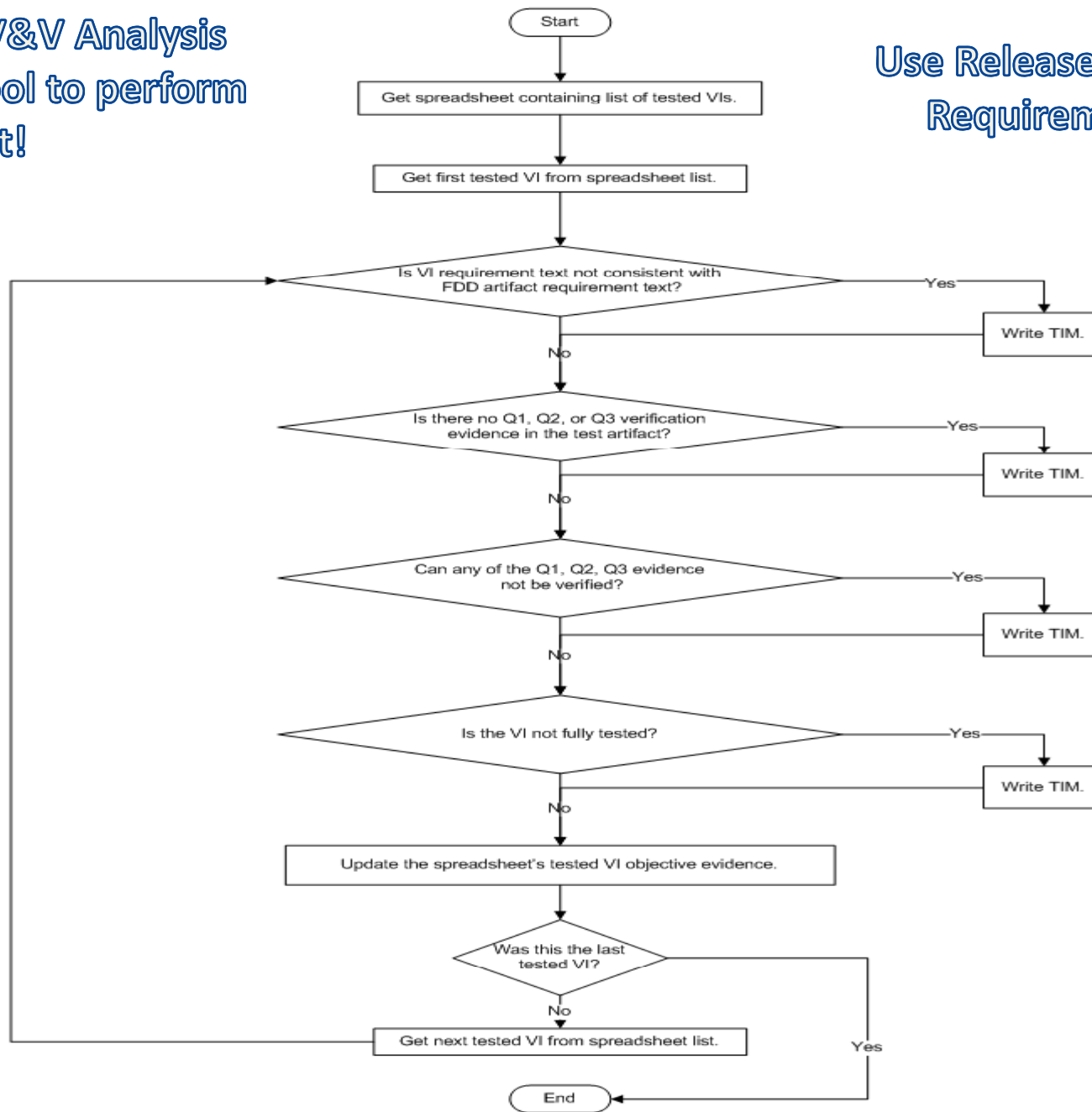
Test Analysis

- Ensure requirements are in the test artifacts?
 - *Examine test artifacts for completeness of requirement coverage.*
- Ensure FDD/Release Plan requirements match those in the test artifacts.
- Check availability of artifacts as specified in Verification Item Database.
 - *IVV Test Analysis Database spreadsheet*
- Verify that all Verification Items (VIs) are fully tested (DATI). VIs are requirements!
- Verify the '3 questions' per test requirement.

Test Analysis Workflow

Use MSL IV&V Analysis Tracking tool to perform assessment!

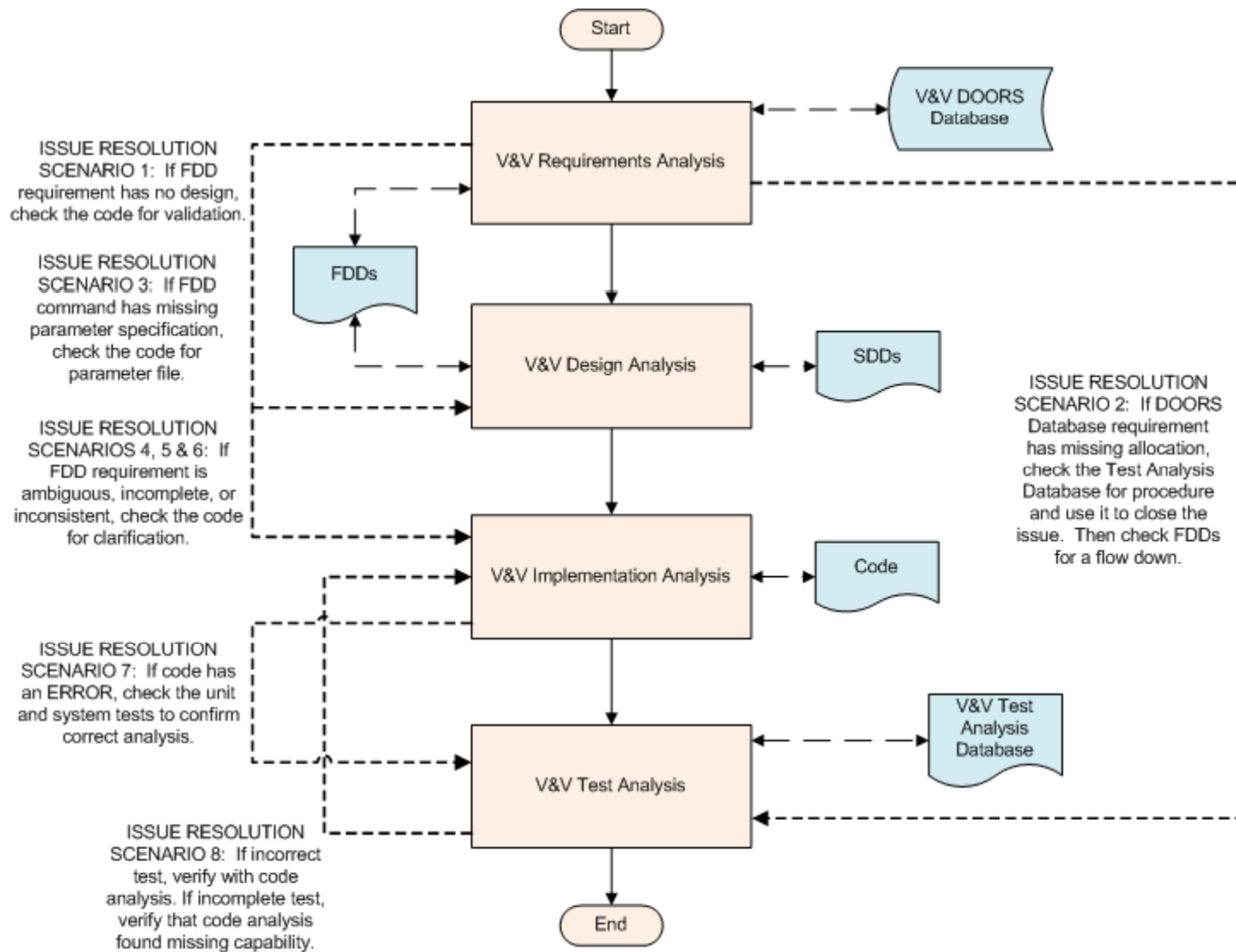
Use Release Plan for Requirements!



MSL Unified Analysis Technical Tools

- In the beginning – only *MS Excel spreadsheets!*
- ***MSL IV&V Analysis Tracking Tool*** (*details later...*)
 - Written in Java for performing V&V analysis assessment
 - MS SQL Server DB centralized data source
 - MS Access DB
- ***Fault Protection IV&V Analysis Tool***
 - Written in Java
 - MS SQL Server/Access DB centralized data source
 - MS Access DB
- ***Enterprise Content Management (ECM); Livelink***
 - Project content/data storage workspace
- **Observation, Risk, Requirement, Backlog and Issue Tracking (ORBIT)**
 - Issue tracking and resolution; Defect Tracking tool

MSL V&V Unified Analysis Process Workflow & Case Studies



MSL Unified Analysis Case Studies

- Case Study 1: If FDD requirement has no design, check the code for validation.
 - *Requirements and Code phases*
- Case Study 2: If DOORS Database requirement has missing allocation, check the Test Analysis Database for test procedure and use it to resolve the issue. Then check FDDs for the requirement flow down and verifiability.
 - *Requirements, (Design, Code,) and Test phases*
- Case Study 3: If FDD command has missing parameter specification, check the code for the parameter file.
 - *Requirements, Design, and Code phases*
- Case Study 4: If FDD requirement is *ambiguous*, check the code for clarification.
 - *Requirements and Code phases*

MSL Unified Analysis Case Studies 2

- Case Study 5: If FDD requirement is *incomplete*, check the code for clarification.
 - *Requirements and Code phases*
- Case Study 6: If FDD requirement is *inconsistent*, check the code for clarification.
 - *Requirements and Code phases*
- Case Study 7: If code has an ERROR, check the unit and system tests to confirm correct analysis.
 - *Code and Test phases*
- Case Study 8: If incorrect test procedure, verify with code analysis. If incomplete test, verify that code analysis found missing capability.
 - *Code and Test phases*

MSL Analysis Tracking Tool – System Information Tab

MSL Analysis Tracking Tool

Nimbus

User: Richard Kowalski
server: THOR
build: 1.3.1
releaseDate: 02/07/2012

System Information | Requirements Validation | Design Verification | Semantic Code Analysis | Test Analysis | Search

Initialization Complete

Old Way - FDD Worksheet

- Requirement Validation Tab:

IV&V System Requirements Analysis								
WBS Requirement Validation Questions							Issue? (Y/N): If answer is yes, then fill out issue form	Peer Reviewed (Date, who)
Unambiguous? (Y/N)	Correct? (Y/N)	Complete? (Y/N)	Consistent? (Y/N)	DATIS Verifiable? (Y/N)	Requirement in scope? (Y/N)	Can be represented in the SRM? (Y/N)		

- Design Verification Tab: (SDD was used)

IV&V Software Design Analysis										
WBS Software Design Verification Questions						FDD Evidence of Architecture Support		Issue? (Y/N): If answer is yes, then fill out issue form	Analysts (Date, who)	Peer Reviewed (Date, who)
In-Scope (Y/N)	Unambiguous (Y/N)	Correct (Y/N)	Complete (Y/N)	Consistent (Y/N)	Testable (Y/N)	Location of implementation in the FDD (section, page, etc.)	Copy associated text/figures referenced in prior column (<-)			

MSL Analysis Tracking Tool – Requirements Validation Tab

MSL Analysis Tracking Tool

Analysis: 5.1.1.2 Rationale: Collecting SOH

CCAM_POWER - Actions (Power ON)

STEP ACTION

1 Check that instrument is marked HEALTHY. If instrument is not HEALTHY, fail command and issue EVR.

2 Obtain CCAM/UHF resource.

3 Power on the instrument and generate an EVR stating that the instrument has been powered on. If the instrument is already on when commanded to power on, reinforce the power state, issue an additional EVR stating that the instrument was powered on when already on, and continue with next steps.

4 Wait 10 seconds.

5 Send i-cmd: NO_OP = 0xFE = 254.

6 Send i-cmd: NO_OP = 0xFE = 254.

7 If CCAM_INIT_DEFAULT = INIT0, INIT1 then pulse the discrete to reboot into the boot bank specified (INIT0 = Bank0, INIT1 = Bank1). Generate an EVR stating that the instrument has been rebooted.

Requirement: FDD:

The FSW shall, upon receipt of the command CCAM_POWER(ON), execute the actions specified in the "CCAM_POWER -Actions (Power ON)" table in the ChemCam FDD.

Requirement ID	Functional Descript...	Status	Issue
FSW-CCAM-001	Chemcam	Complete	No
FSW-CCAM-002	Chemcam	Complete	No
FSW-CCAM-003	Chemcam	In Process	Yes
FSW-CCAM-004	Chemcam	Complete	No
FSW-CCAM-005	Chemcam	In Process	Yes
FSW-CCAM-006	Chemcam	Complete	No
FSW-CCAM-007	Chemcam	Complete	No
FSW-CCAM-009	Chemcam	Complete	No
FSW-CCAM-010	Chemcam	In Process	Yes
FSW-CCAM-011	Chemcam	In Process	Yes
FSW-CCAM-012	Chemcam	In Process	Yes
FSW-CCAM-013	Chemcam	Complete	No
FSW-CCAM-014	Chemcam	Complete	No
FSW-CCAM-015	Chemcam	Complete	No
FSW-CCAM-016	Chemcam	In Process	Yes
FSW-CCAM-017	Chemcam	In Process	Yes
FSW-CCAM-018	Chemcam	In Process	Yes
FSW-CCAM-019	Chemcam	Complete	No

Analysis Complete: Deferred

System Information | **Requirements Validation** | Design Verification | Semantic Code Analysis | Test Analysis | Search

Requirements Validation selected

MSL Analysis Tracking Tool – Design Verification Tab

MSL Analysis Tracking Tool

Validated
 Correlated to SRM
 Question 2
 Architecture Supports
 Architecture is Feasible

Support Justification:
 This system design's requirement is validated, its system architecture is feasible and supports the requirement, and it abides by Q2. This system design is not correlated to any System Reference Model (SRM). MSL does not support an SRM. This requirement's design is traced to section 4.2.1 Behavior Coordination.

Feasibility Justification:
 The Activity Constraint Manager (ACM) ensures that the vehicle is in a safe condition to perform a given activity. For mobility, ACM declares that it is not safe to perform vehicle motion if there is a mobility motion error, a mobility goal error, or if moving the vehicle is explicitly prevented via a ground preclude. Other conditions such as the arm being unstowed and the SAM not being safe for driving are also checked (complete list can be found in the ACM / ARB FDD). If any of these conditions exist, ACM declares that it is not safe to drive. MOM must check with ACM to ensure that it is

Issue TIM ID:

Requirement: FDD:

The FSW shall keep the vehicle from performing any mobility motion if the Activity Constraint Manager (ACM) reports that mobility is not allowed.

Requirement ID	Functional Descript...	Status	Issue
FSW-MOB-101	Mobility	Complete	No
FSW-MOB-102	Mobility	Complete	No
FSW-MOB-105	Mobility	Complete	No
FSW-MOB-106	Mobility	Complete	No
FSW-MOB-107	Mobility	Complete	No
FSW-MOB-108	Mobility	Complete	No
FSW-MOB-109	Mobility	Complete	No
FSW-MOB-110	Mobility	Complete	No
FSW-MOB-111	Mobility	Complete	No
FSW-MOB-112	Mobility	Complete	No
FSW-MOB-113	Mobility	Complete	No
FSW-MOB-115	Mobility	Complete	No
FSW-MOB-116	Mobility	Complete	No
FSW-MOB-301	Mobility	Complete	No
FSW-MOB-302	Mobility	Complete	No
FSW-MOB-303	Mobility	Complete	No
FSW-MOB-304	Mobility	Complete	No
FSW-MOB-305	Mobility	Complete	No

 Analysis Complete:
 Deferred

Design Verification selected

MSL Analysis Tracking Tool – Semantic Code Analysis Tab

The screenshot displays the MSL Analysis Tracking Tool interface. At the top, the title bar reads "MSL Analysis Tracking Tool". The main window is divided into several sections:

- Software Build:** A dropdown menu showing "10.2.0".
- Analysis:** A large empty text area for entering analysis details.
- Objective Evidence:** A large empty text area for entering objective evidence.
- Issue:** A large empty text area for entering issue details.
- TIM ID:** A small text input field.
- Requirement:** A text input field.
- FDD:** A text input field.
- Analysis Complete:** A checkbox and a "Deferred" button.
- Navigation:** Buttons for navigation: "<", "<<", ">>", and ">|".
- Buttons:** "Save" and "Cancel" buttons.
- Bottom Bar:** A series of tabs: "System Information", "Requirements Validation", "Design Verification", "Semantic Code Analysis" (selected), "Test Analysis", and "Search".

On the right side, there is a vertical panel with three sections, each containing a "Validation Analysis" and "Issue" sub-section:

- Validation Analysis Issue
- Verification Analysis Issue
- Test Analysis Issue

At the bottom left, a status bar indicates "Semantic Code Analysis selected".

MSL Analysis Tracking Tool – Test Analysis Tab

MSL Analysis Tracking Tool

Requirement Consistent in Artifact(s)

Passing Results Shown in Artifact(s) Question 2

Artifact Discovery:

Analysis:
Babble was injected for 5 seconds. This made RCE_A reset to isolation and wait for RCE_B to become Prime. The new Prime RCE (RCE_B) created a CrossStringDP for the isolated string. RMCA was set SICK by the Preferred RCE Response as part of isolation recovery.

Issue: TIM ID:

Validation Analysis: Test Data Not Yet Implemented

Verification Analysis: Issue

Semantic Analysis: Randall

Requirement: FDD:

Scope: Regression Analysis Issue?:

VI Owner Status: Verification Method:

Phase/Domain: Phase/Domain Status:

Requirement ID	Functional Descripti...	Status	Issue
FSW-BFP-002	1553 BFP	Complete	No
FSW-BFP-019	1553 BFP	Complete	No
FSW-BFP-023	1553 BFP	Complete	No
FSW-BFP-032	1553 BFP	Complete	Yes

Requirement Text:
FSW shall provide the capability to 'suspend' normal intercomm services to isolate and then recover from 'peripheral interfering' faults.

Test Artifact (In IVV Possession):
<BASE>FDD - 1553 BFPIC-119900
BFP_EDL_Bus__Babble_Recovery_Data_Review_2011_0720.pptx

Artifact Procedure Procedure URL Report URL

Analysis Complete: Deferred

Save Cancel

System Information Requirements Validation Design Verification Semantic Code Analysis **Test Analysis** Search

Test Analysis selected

MSL Analysis Tracking Tool – Search Tab

MSL Analysis Tracking Tool

Search

Requirements Validation Design Verification Semantic Analysis Test Analysis

Rqmnt Number: FSW-CCAM-011 Analyst: Kowalski, Richard

The FSW shall, upon receipt of the command CCAM_ACTV_SPECTRAL_OBS, execute the actions specified in the "CCAM_ACTV_SPECTRAL_OBS -Actions" table in the ChemCam FDD.

Unambiguous Correct Complete Consistent Verifiable

Question 2

Analysis:

This requirement is unambiguous, correct, incomplete, inconsistent, verifiable, and abides by Q2. The Actions Table, containing 21 steps, for this S-command is on page 105 in the design section of the FDD.

CCAM_ACTV_SPECTRAL_OBS - Actions

STEP	ACTION

Issue:

Should step 2 also "skip all remaining steps"?

UNUSED

Analyst: Kowalski, Richard FDD: Unselected

Rqmnt ID	FDD	Analysis Type	Status	Issue
FSW-CCAM-001	Chemcam	Validate Rqmnts	Complete	No
FSW-CCAM-002	Chemcam	Validate Rqmnts	Complete	No
FSW-CCAM-003	Chemcam	Validate Rqmnts	In Process	Yes
FSW-CCAM-004	Chemcam	Validate Rqmnts	Complete	No
FSW-CCAM-005	Chemcam	Validate Rqmnts	In Process	Yes
FSW-CCAM-006	Chemcam	Validate Rqmnts	Complete	No
FSW-CCAM-007	Chemcam	Validate Rqmnts	Complete	No
FSW-CCAM-009	Chemcam	Validate Rqmnts	Complete	No
FSW-CCAM-010	Chemcam	Validate Rqmnts	In Process	Yes
FSW-CCAM-011	Chemcam	Validate Rqmnts	In Process	Yes
FSW-CCAM-012	Chemcam	Validate Rqmnts	In Process	Yes
FSW-CCAM-013	Chemcam	Validate Rqmnts	Complete	No
FSW-CCAM-014	Chemcam	Validate Rqmnts	Complete	No
FSW-CCAM-015	Chemcam	Validate Rqmnts	Complete	No
FSW-CCAM-016	Chemcam	Validate Rqmnts	In Process	Yes
FSW-CCAM-017	Chemcam	Validate Rqmnts	In Process	Yes
FSW-CCAM-018	Chemcam	Validate Rqmnts	In Process	Yes
FSW-CCAM-019	Chemcam	Validate Rqmnts	Complete	No
FSW-CCAM-020	Chemcam	Validate Rqmnts	Complete	No
FSW-CCAM-021	Chemcam	Validate Rqmnts	In Process	Yes
FSW-CCAM-022	Chemcam	Validate Rqmnts	In Process	Yes
FSW-CCAM-023	Chemcam	Validate Rqmnts	In Process	Yes
FSW-CCAM-024	Chemcam	Validate Rqmnts	In Process	Yes
FSW-CCAM-025	Chemcam	Validate Rqmnts	In Process	Yes
FSW-CCAM-026	Chemcam	Validate Rqmnts	In Process	Yes
FSW-CCAM-027	Chemcam	Validate Rqmnts	Complete	No
FSW-CCAM-028	Chemcam	Validate Rqmnts	In Process	Yes
FSW-CCAM-100	Chemcam	Validate Rqmnts	Complete	No
FSW-CCAM-101	Chemcam	Validate Rqmnts	In Process	Yes
FSW-CCAM-102	Chemcam	Validate Rqmnts	Complete	No
FSW-CCAM-103	Chemcam	Validate Rqmnts	Complete	No
FSW-CCAM-104	Chemcam	Validate Rqmnts	Complete	No
FSW-CCAM-105	Chemcam	Validate Rqmnts	Complete	No
FSW-CCAM-106	Chemcam	Validate Rqmnts	Complete	No
FSW-CCAM-202	Chemcam	Validate Rqmnts	In Process	Yes
FSW-CCAM-203	Chemcam	Validate Rqmnts	In Process	Yes
FSW-CCAM-204	Chemcam	Validate Rqmnts	In Process	Yes
FSW-CCAM-300	Chemcam	Validate Rqmnts	Complete	No
FSW-MOB-104	Mobility	Validate Rqmnts	Complete	No
FSW-MOB-105	Mobility	Validate Rqmnts	Complete	No

System Information Requirements Validation Design Verification Semantic Code Analysis Test Analysis Search

Finished

Summary

- Provide the analysis **procedures and tools** appropriate to **have all phases** of the system lifecycle **collaborate** during issue detection and resolution.
- Formulates a **standard, coherent** and **comprehensive approach** to verification and validation of the spacecraft's system lifecycle.
- **Unify** the workflow **procedures** for performing IV&V analysis throughout the system/software lifecycle.
- **Eliminate isolated separate use of the procedures** for workflow analysis!
- **Exploit the interactions among the procedures** of each phase of the lifecycle to meet goals and facilitate a brisk issue detection and resolution process.

Questions?

