MSL IV&V Unified Analysis Process

(Our "*Best Practice*" from a Technical Analyst's perspective)

Rich Kowalski

Richard.E.Kowalski@ivv.nasa.gov

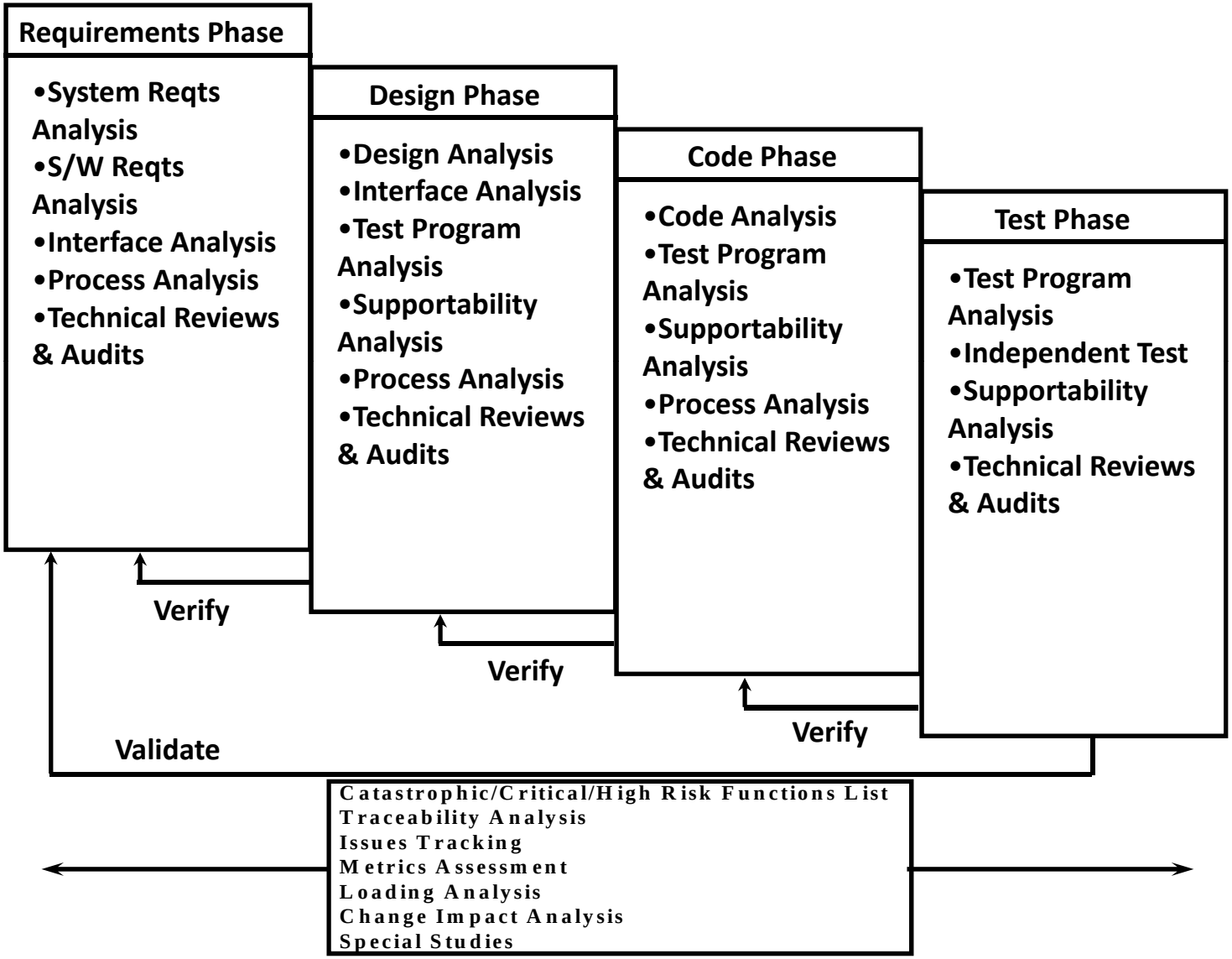
Agenda

- 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?

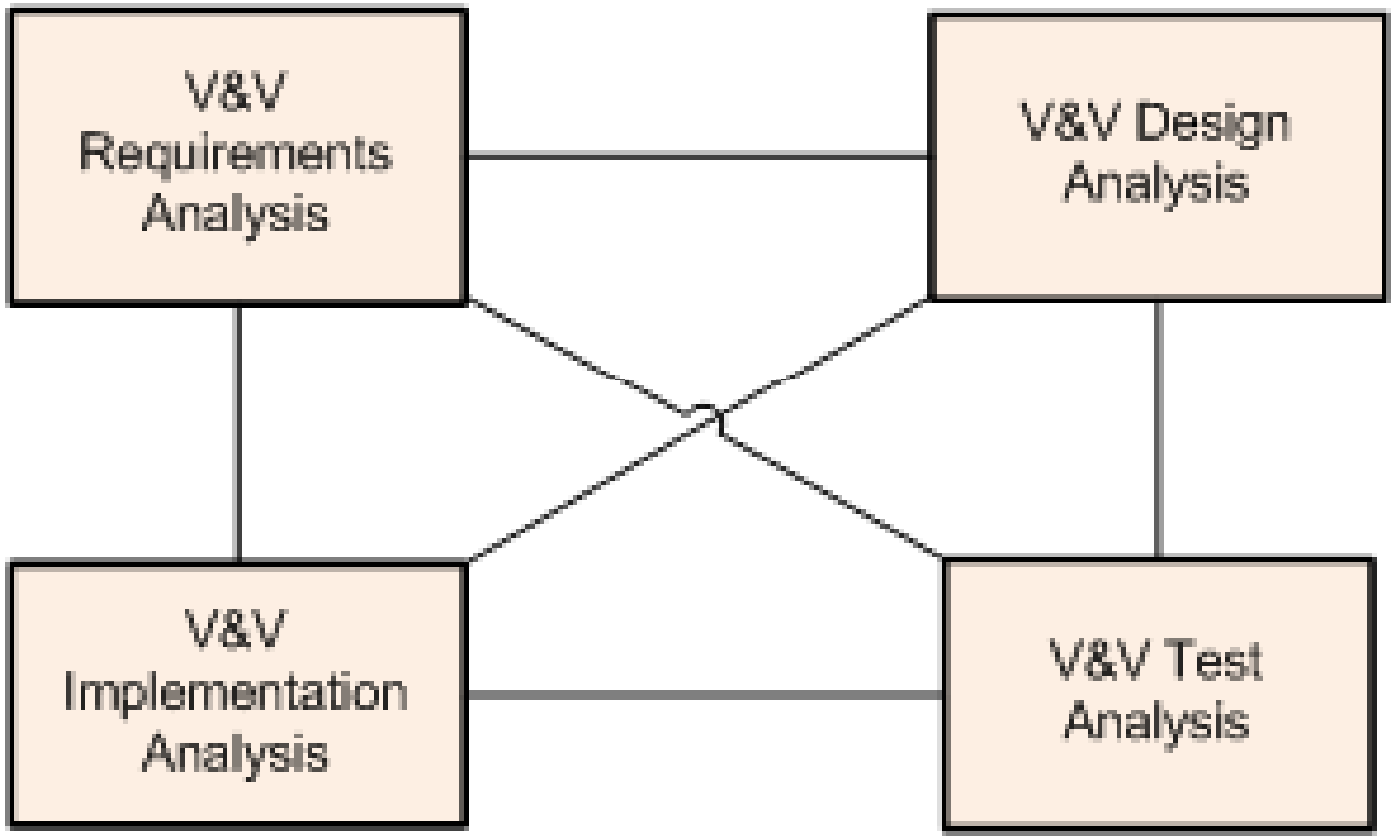
- MSL Analysis Philosophy - MSL is a *complex mission* with *extremely large* (~3M SLOC) and *complex software*.

- 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?

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

Use Release Plan for Requirements!

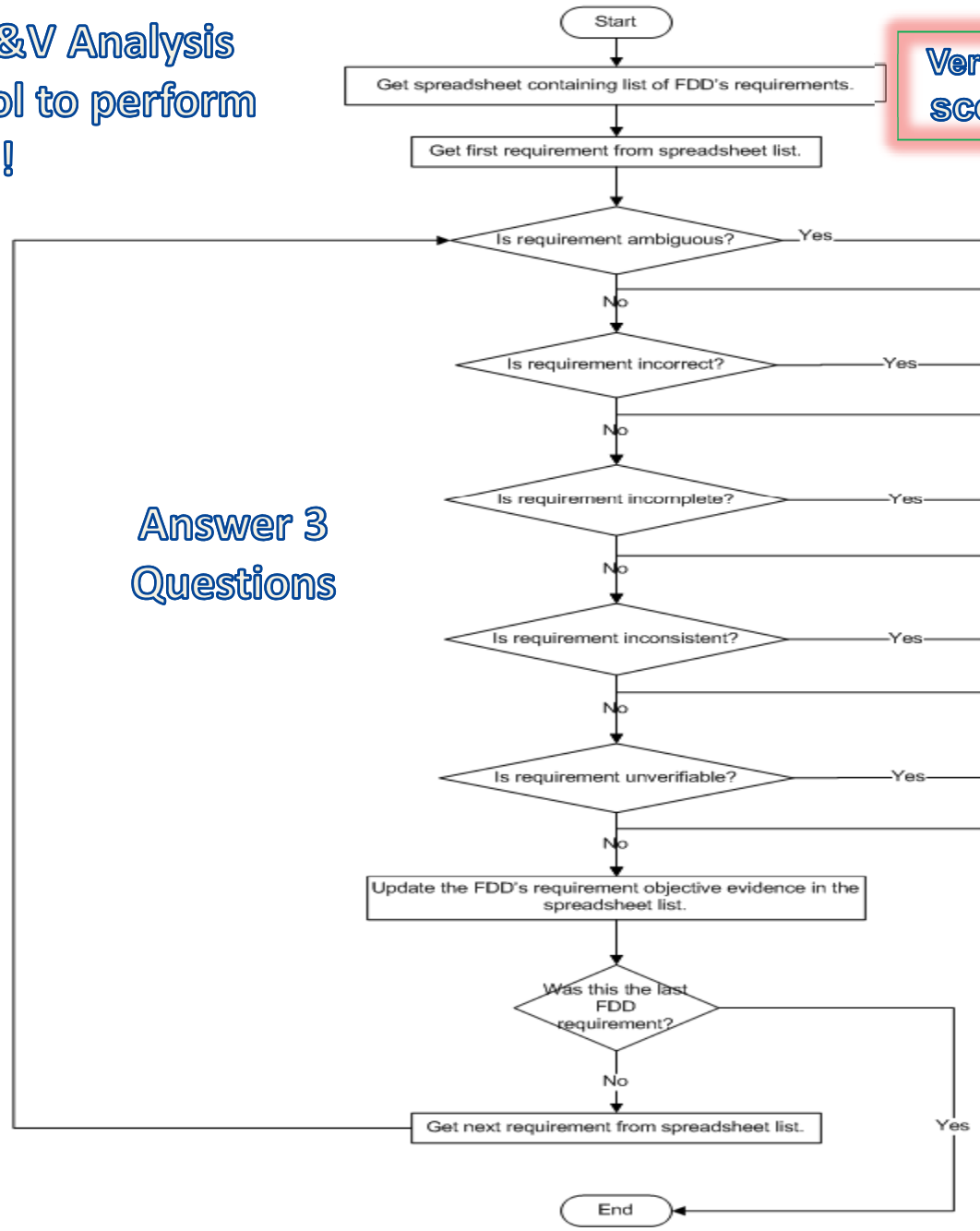
Verify in scope!

'Quality Attributes'

In scope? Testable?

Answer 3 Questions

Problem detection and solving loop!

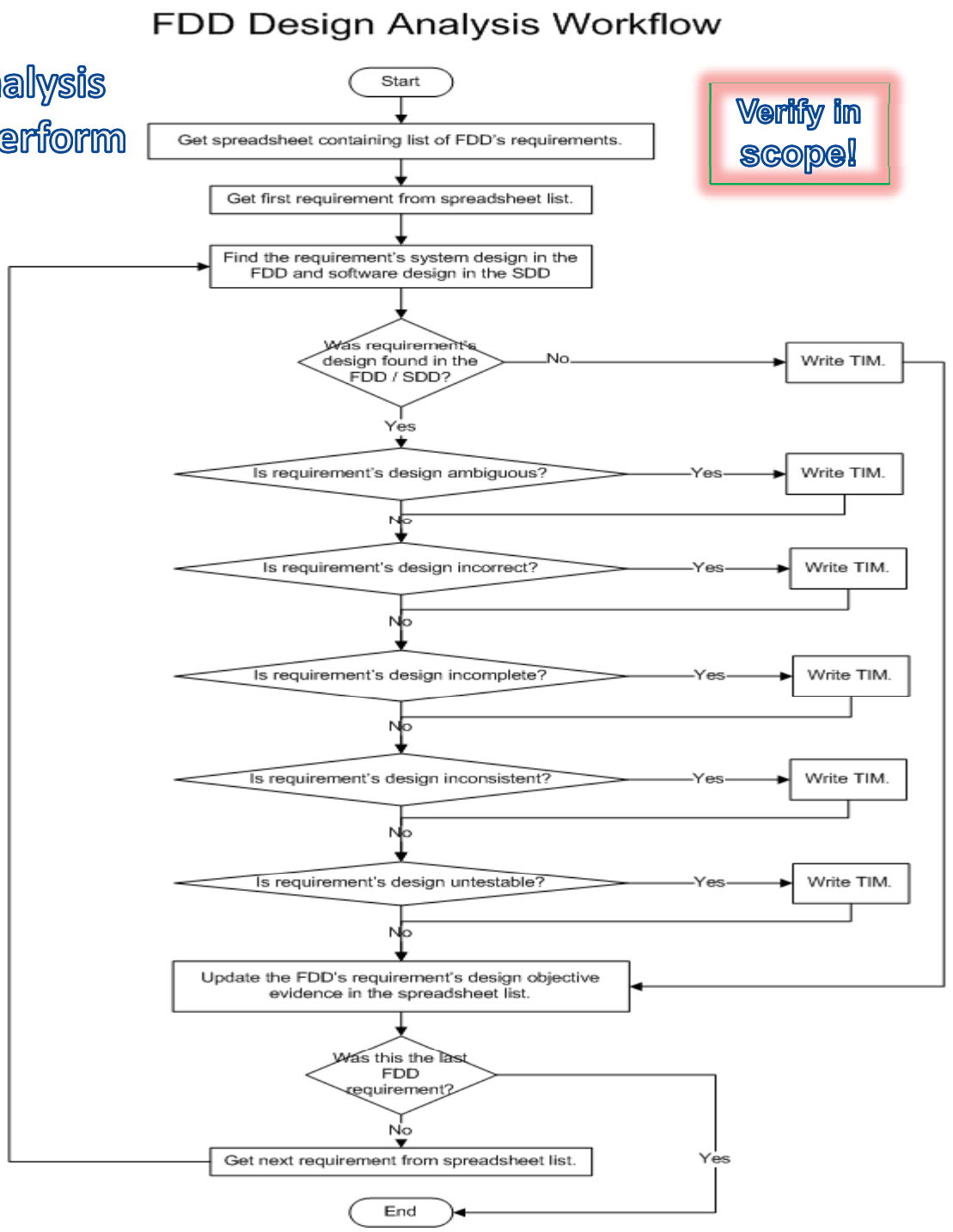


FDD Software Design Analysis

- 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?

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

Answer 3 Questions



Verify in scope!

Use Release Plan for Requirements!

'Quality Attributes'

In scope? Testable?

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.

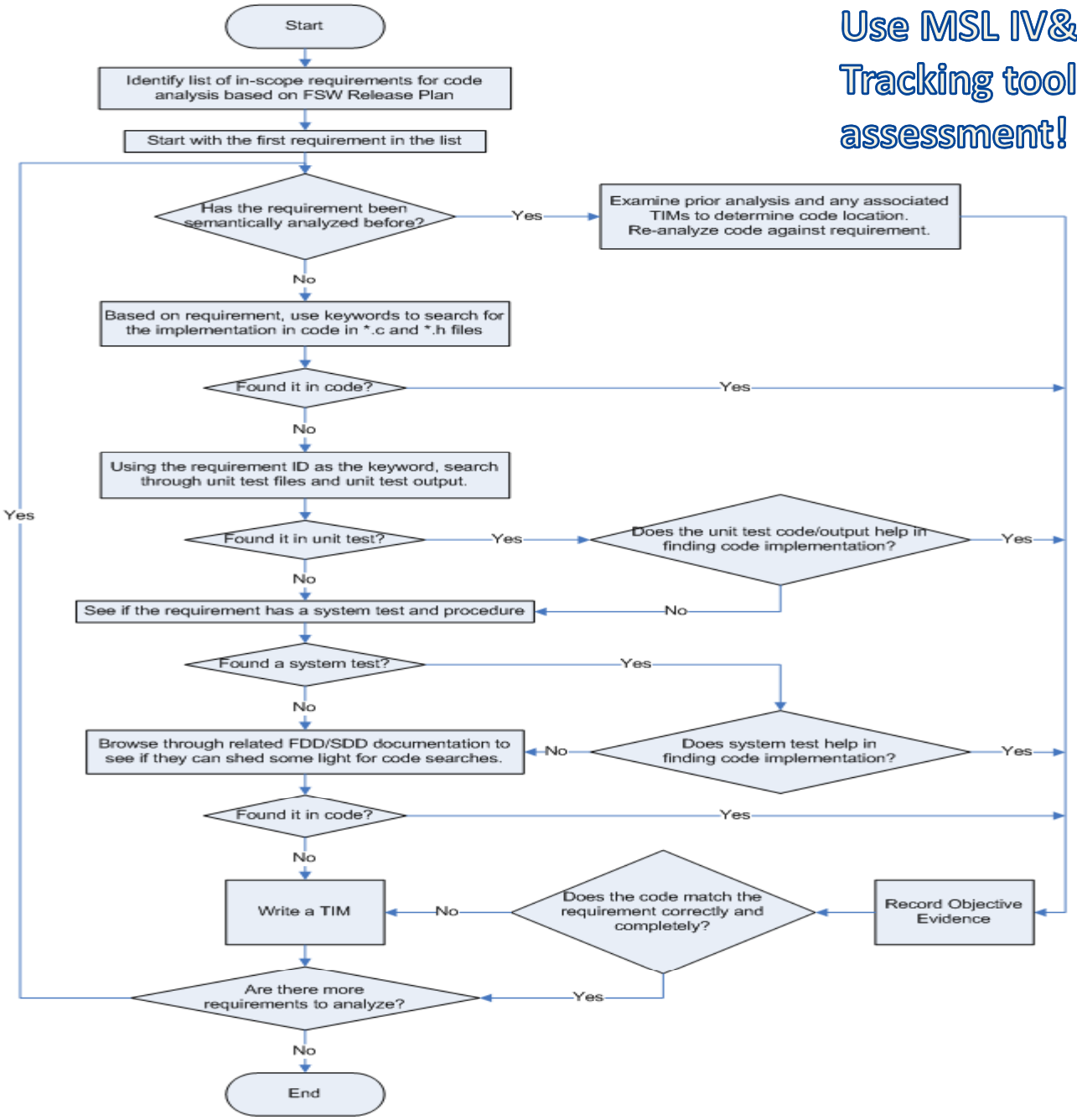
- Large numbers of *false positives (WARNINGs)* are winnowed manually.

- *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

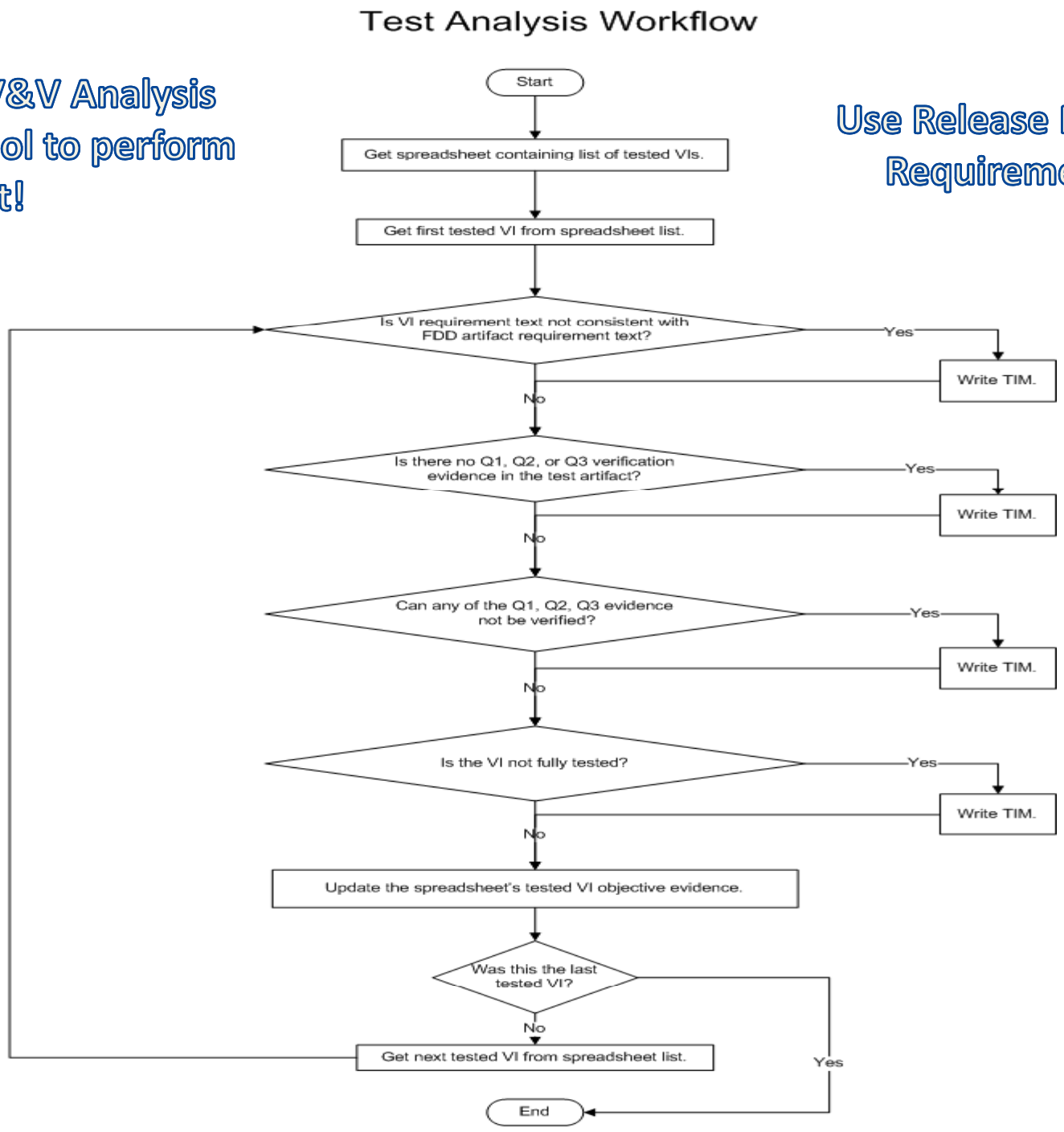


Test Analysis

- 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.

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

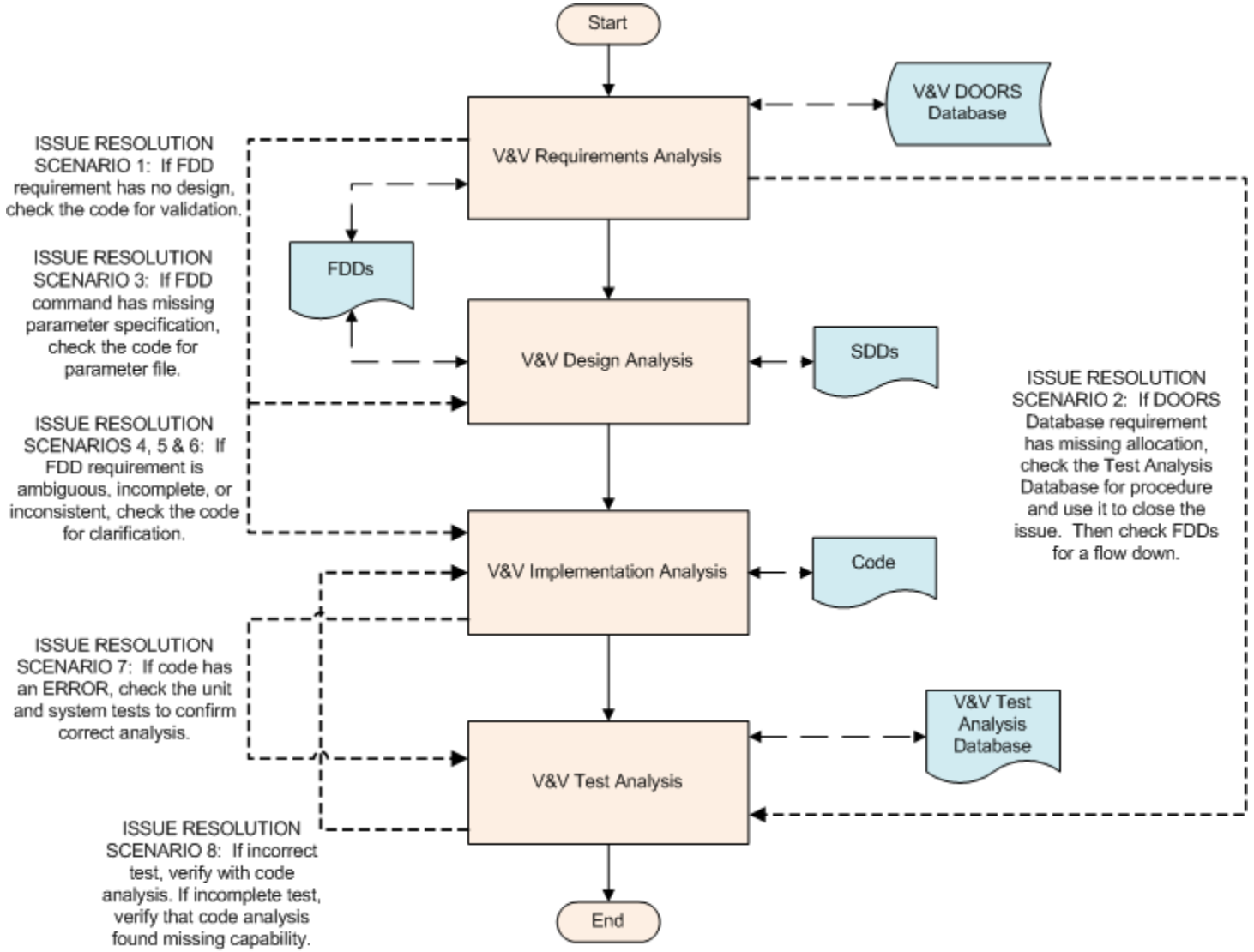
Use Release Plan for Requirements!



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



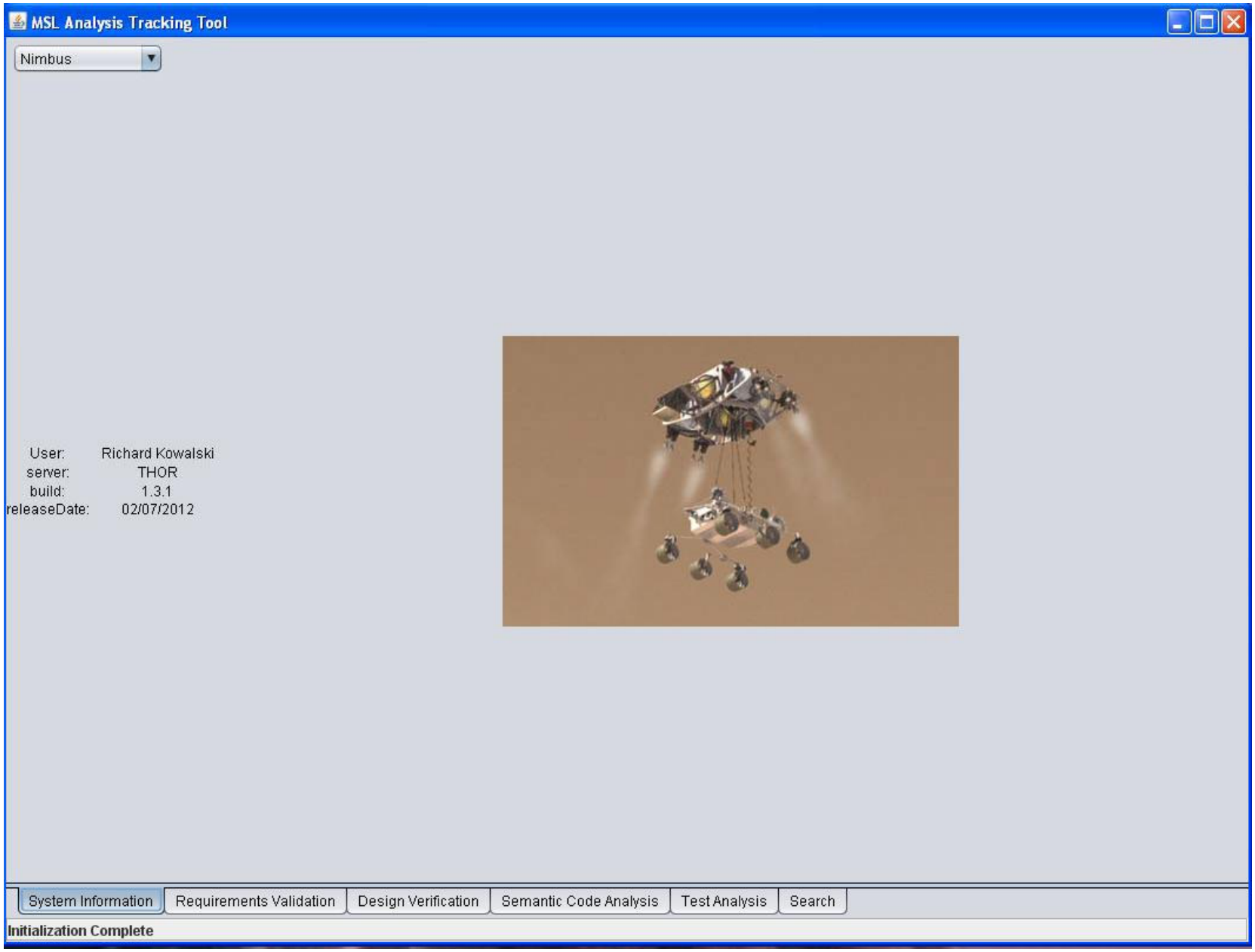
MSL Unified Analysis Case Studies

- 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



Old Way - FDD Worksheet

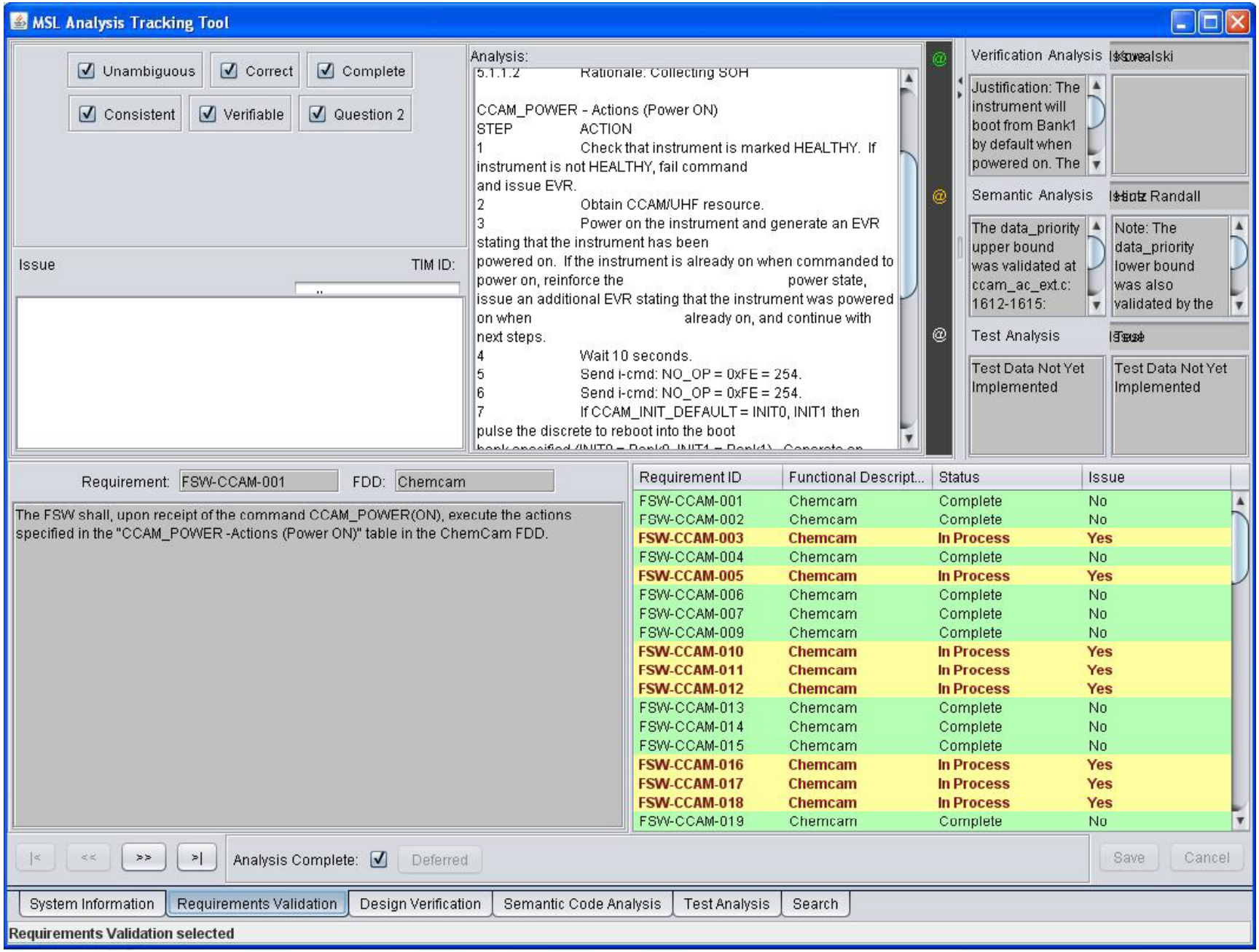
- **Requirement Validation Tab:**

IV&V System Requirements Analysis								
WBS Requirement Validation Questions								
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)	Issue? (Y/N): If answer is yes, then fill out issue form	Peer Reviewed (Date, who)

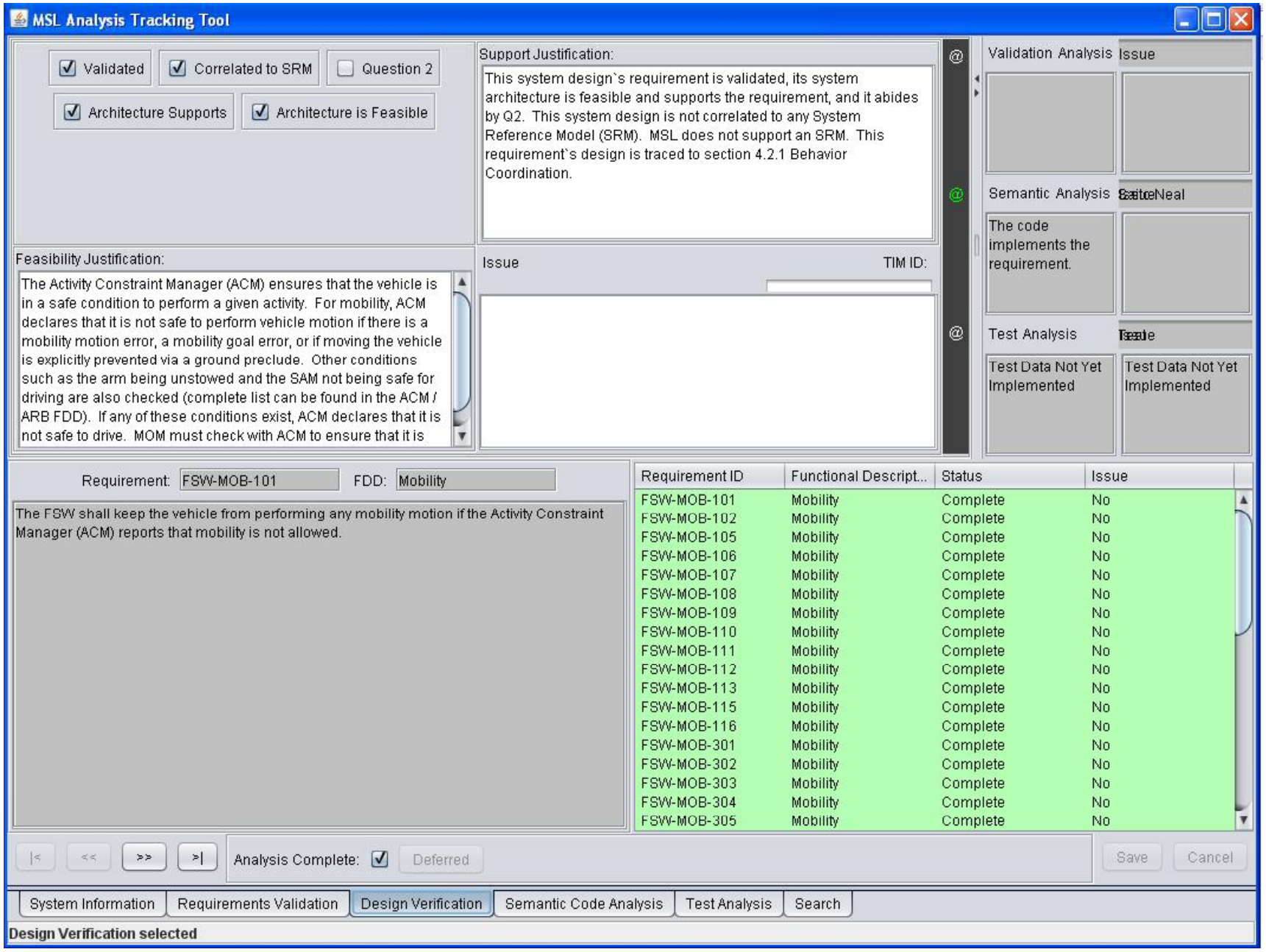
- **Design Verification Tab: (SDD was used)**

IV&V Software Design Analysis										
WBS Software Design Verification Questions						FDD Evidence of Architecture Support				
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 (<--)	Issue? (Y/N): If answer is yes, then fill out issue form	Analysts (Date, who)	Peer Reviewed (Date, who)

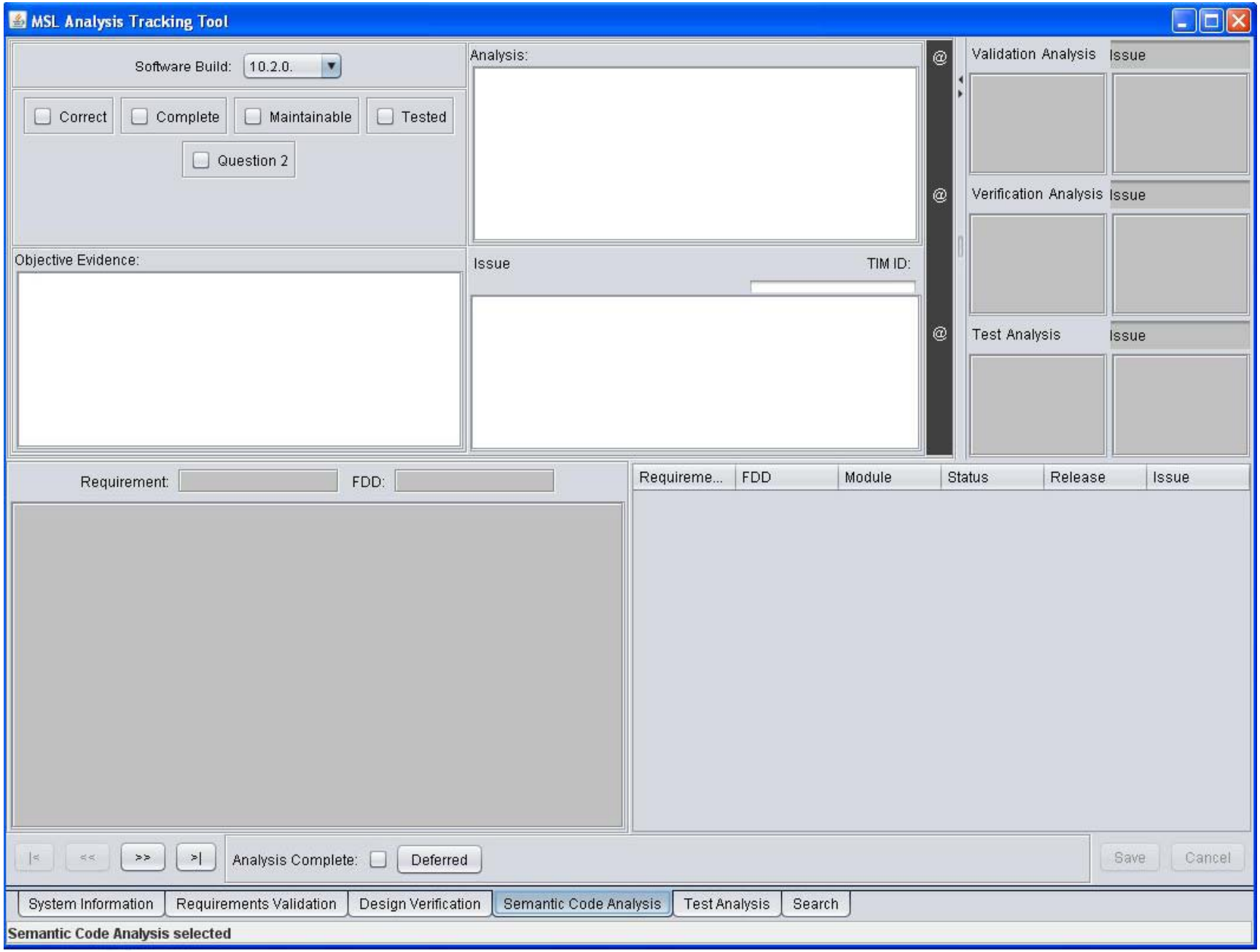
MSL Analysis Tracking Tool – **Requirements Validation** Tab



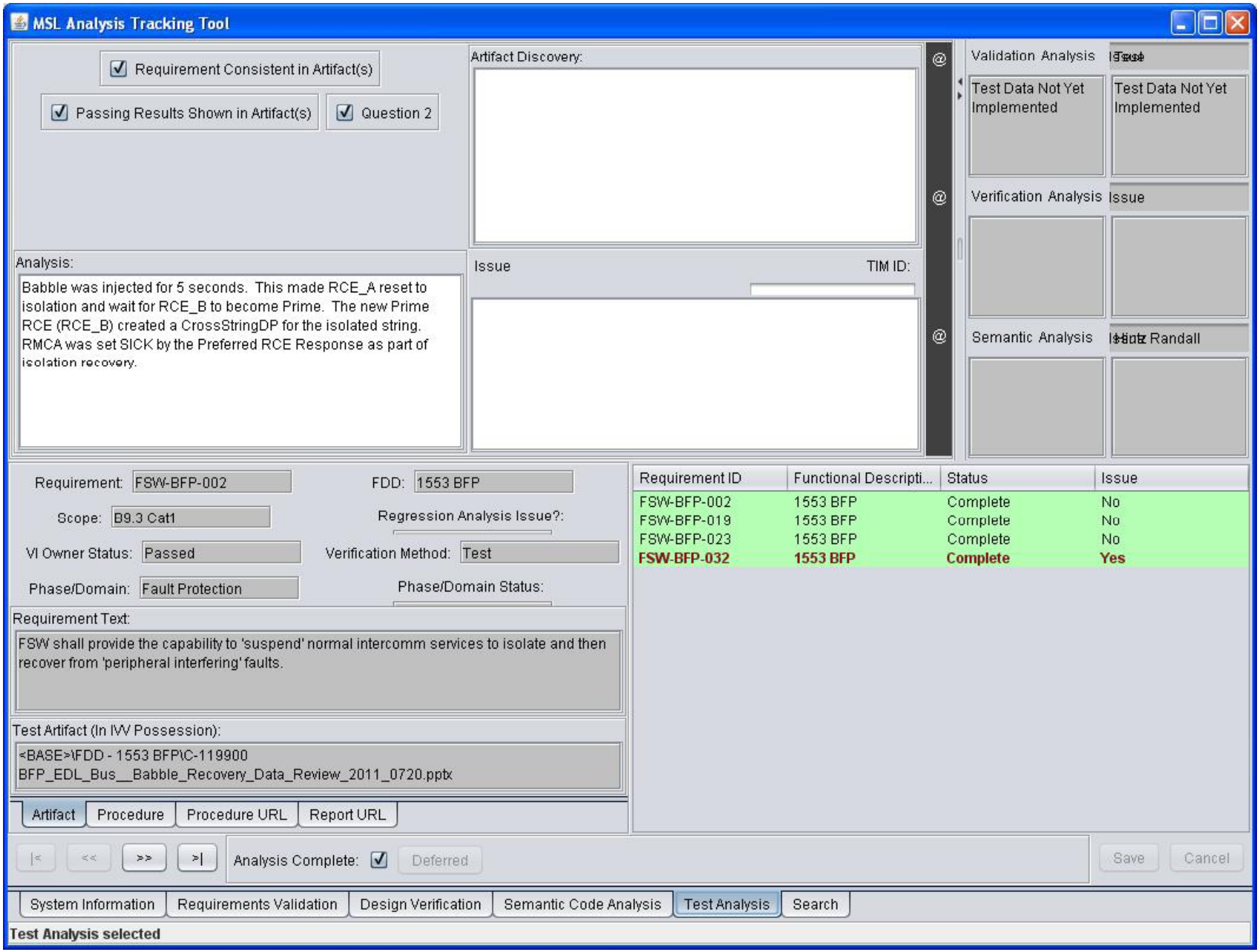
MSL Analysis Tracking Tool – **Design Verification** Tab



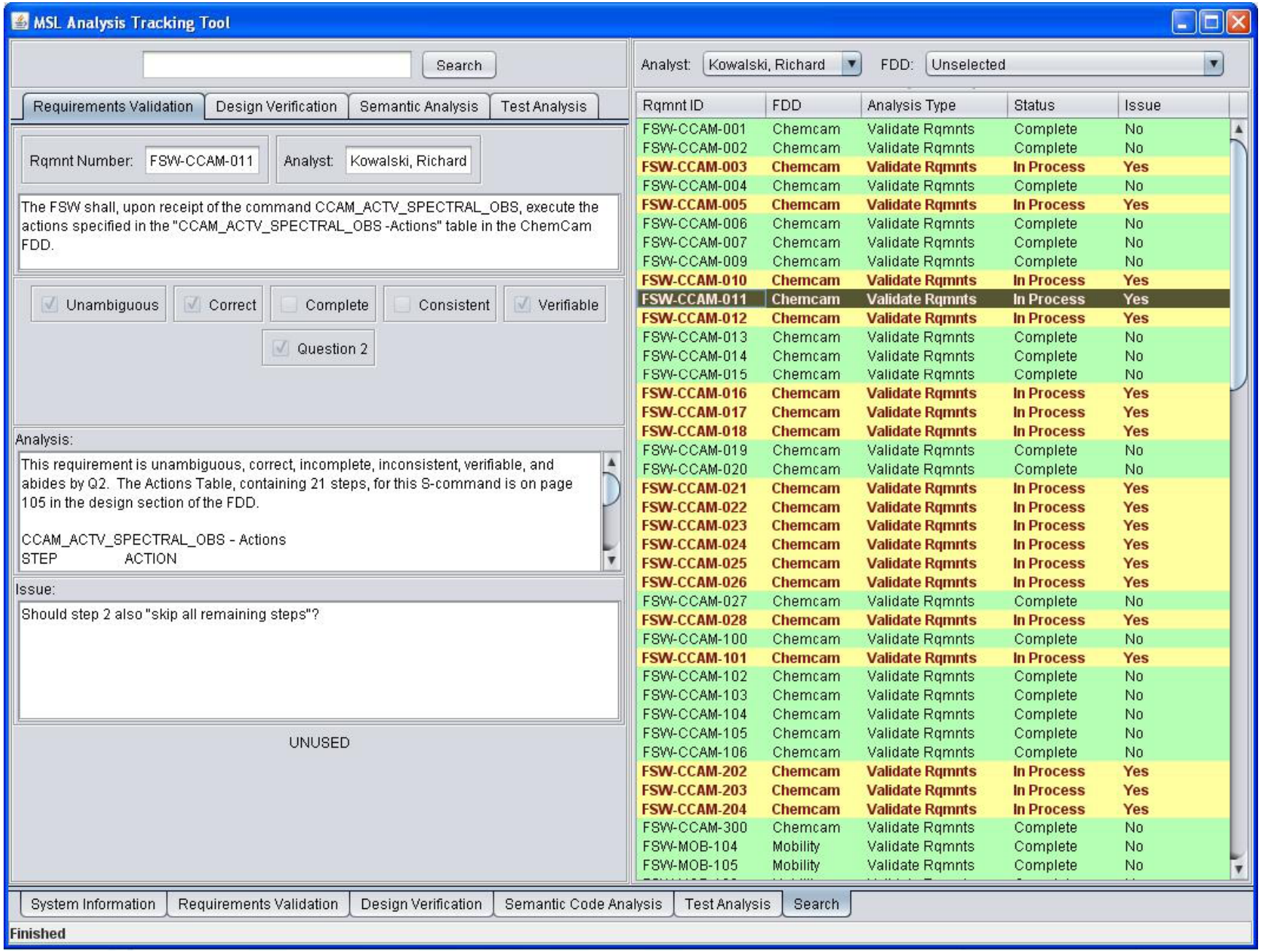
MSL Analysis Tracking Tool – **Semantic Code Analysis** Tab



MSL Analysis Tracking Tool – **Test Analysis** Tab



MSL Analysis Tracking Tool – **Search** Tab



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?