How Can We Use Verification and Validation (V&V) Techniques in Early Systems Engineering?

STEVEN H. DAM, PH.D., ESEP
CHRIS RITTER
SPEC INNOVATIONS
STEVEN.DAM@SPECINNOVATIONS.COM

Outline

1. V&V in the Early Stages of the Lifecycle?
2. Ontology for V&V
3. What's a Verifiable Requirement?
4. Developing a V&V Plan Using MBSE

1. V&V IN THE EARLY STAGES OF THE LIFECYCLE?

What Is Verification?

- Verification: “the process of establishing the truth, accuracy, or validity of something” [Google definition]
- Meets specification-level requirements
- Proves that the solution-dependent requirements are satisfied
- Proves that solution was or is being built according to agreed requirements

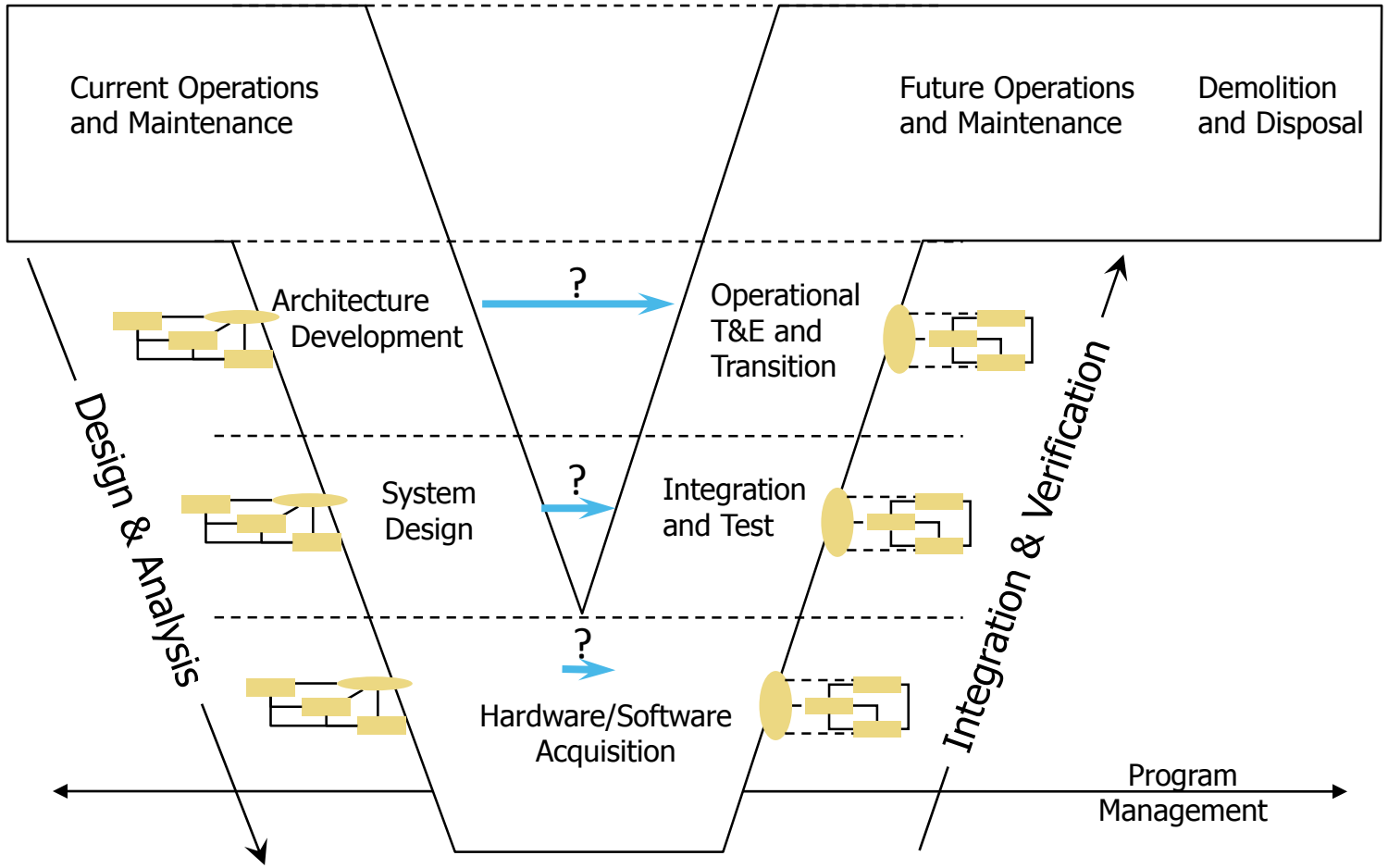
Frequently the terms “Unit Test” and “Developmental Test and Evaluation” apply to verification

What Is Validation?

- Validation: “the action of checking or proving the validity or accuracy of something” [Google definition]
- Relates to customer and other stakeholder satisfaction
- Proves solution-independent requirements are being satisfied
- Usually performed at the enterprise and/or system level
- In process, validation helps ensure system will ultimately be part of the accepted solution in the target environment

Frequently the term “Operational Test and Evaluation” applies to validation

What Do We Need to Do Early?



What Do We Need Early?

- Ontology that includes V&V information
- Verifiable Requirements
- V&V Plan

To do any of these things you need to understand how to prepare for V&V, conduct V&V activities, and report V&V results, so make sure an expert in these areas are part of the early phase

2. ONTOLOGY FOR V&V

Lifecycle Modeling Language's Primary Classes

- Action
- Artifact
- Asset
  - Resource
- Characteristic
  - Measure
- Connection
  - Conduit
  - Logical
- Cost
- Decision
- Input/Output
- Location
  - Physical, Orbital, Virtual
- Risk
- Statement
  - Requirement
- Time

See www.lifecyclemodeling.org for LML specification

Supports capturing information throughout the lifecycle

V&V Extension to LML

- In developing Innoslate's new Test Center capability, we discovered the need to add a subclass to Action: Test Case
- We used the Schema Editor to make these changes, but also we had to add views and reports
- You may discover that you want to add other classes, attributes, and relationships, as well and develop special reports and data entry screens using the SDK

Test Case (Subclass of Action)

- A Test Case entity represents a set of conditions utilized to determine whether a system works as intended

- Properties (Attributes)

Name	Type	Description
Status	ENUMERATION	Status represents the current state of this Test Case.
Actual Result	BIG_TEXT	Actual Results represents the results that are captured after executing a Test Case.
Expected Result	BIG_TEXT	Expected Result represents the result expected when executing this Test Case.
Set Up	BIG_TEXT	Set Up represents the prerequisites that must be fulfilled before executing this Test Case.

3. WHAT'S A VERIFIABLE REQUIREMENT?

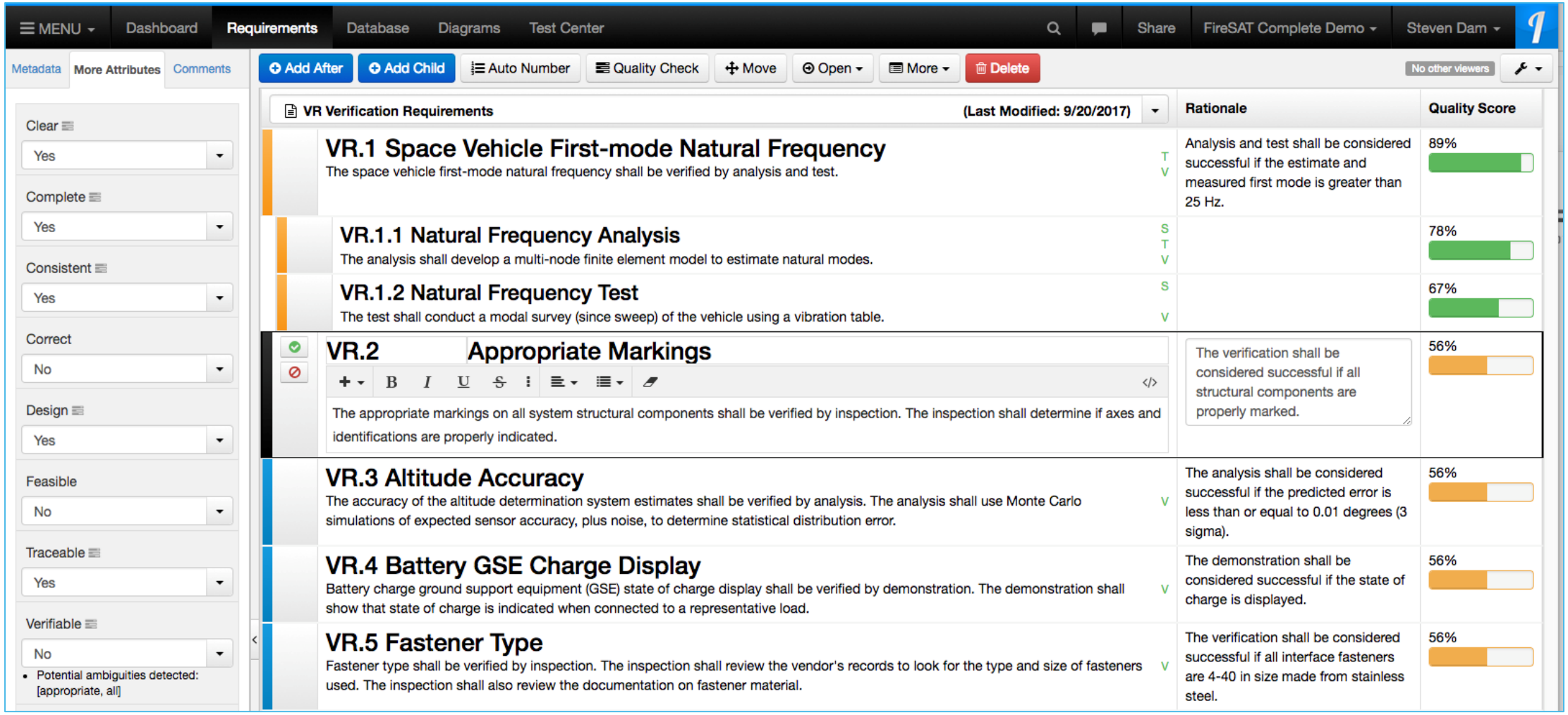
Characteristics of Good Requirements

- Each individual requirement should be:
  - Correct: Describes the user's true intent and is legally possible
  - Complete: Express a whole idea
  - Clear: Unambiguous and not confusing
  - Consistent: Not in conflict with other requirements
  - ***Verifiable: Provable (within realistic cost and schedule) that the system meets the requirement***
  - Traceable: Uniquely identified, and able to be tracked to predecessor and successor lifecycle items/objects
  - Feasible: Able to be implemented with existing technology, and within cost and schedule
  - Modular: Can be changed without excessive impact on other requirements
  - Design: Does not impose a specific solution on design; says "what", Independent not "how"

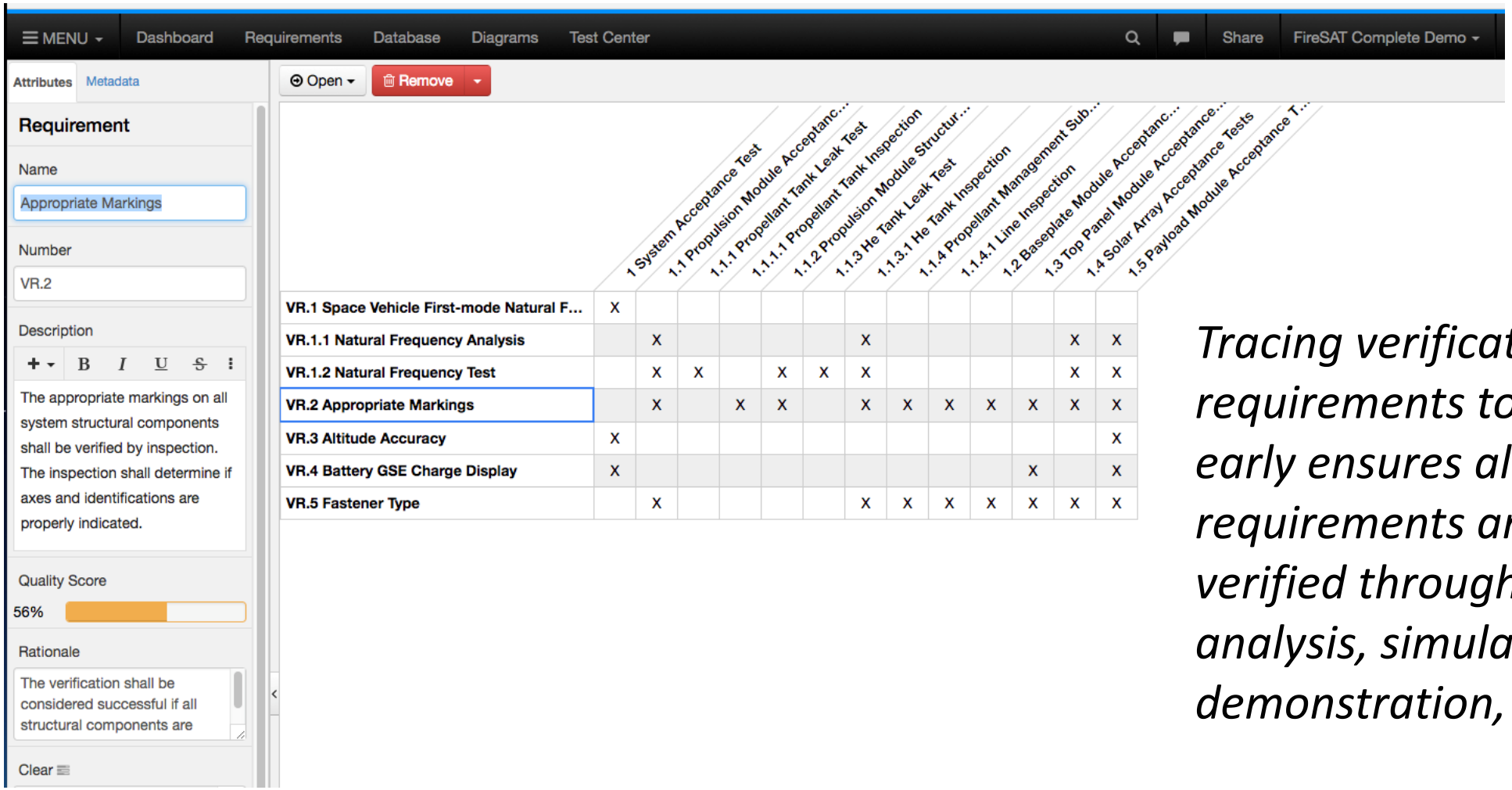
What Makes a Requirement Verifiable?

- You need to answer the following questions:
  - Can the requirement be met?
    - By analysis, demonstration, inspection, modeling & simulation, or test?
  - Will the user accept the results?
- Avoid words such as “excessive,” “sufficient,” “resistant,” etc.
  - Make the requirement quantifiable
- Even “suitability” can be quantified

Example Verification Requirements

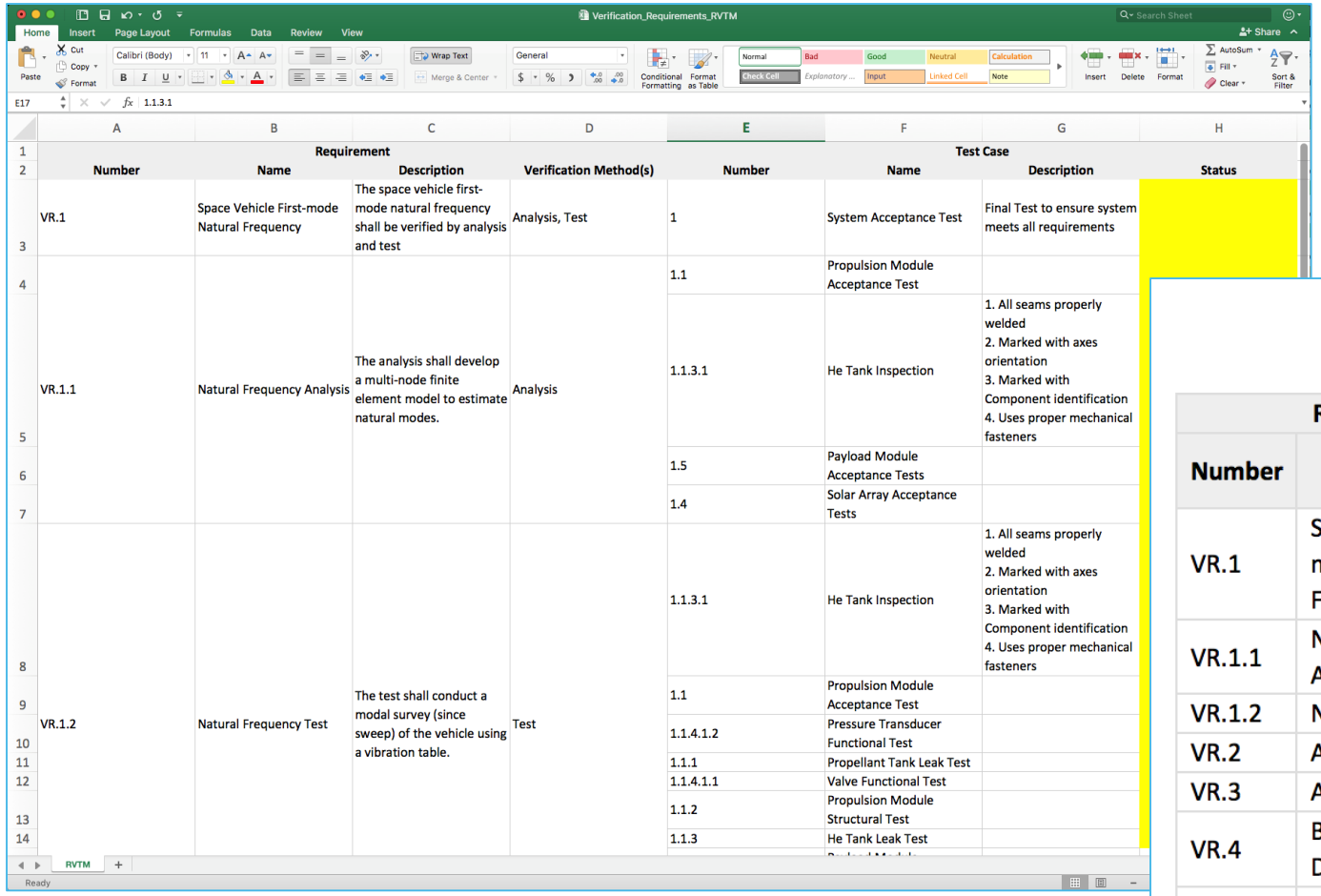


Trace VRs to Test Cases



Tracing verification requirements to test cases early ensures all requirements are properly verified through inspection, analysis, simulation, demonstration, and/or test

Produce Key Reports and Archive Them



Requirement				Test Case			
Number	Name	Description	Verification Method(s)	Number	Name	Description	Status
VR.1	Space Vehicle First-mode Natural Frequency	The space vehicle first-mode natural frequency shall be verified by analysis and test	Analysis, Test	1	System Acceptance Test	Final Test to ensure system meets all requirements	
VR.1.1	Natural Frequency Analysis	The analysis shall develop a multi-node finite element model to estimate natural modes.	Analysis	1.1	Propulsion Module Acceptance Test		
				1.1.3.1	He Tank Inspection	1. All seams properly welded 2. Marked with axes orientation 3. Marked with Component identification 4. Uses proper mechanical fasteners	
				1.5	Payload Module Acceptance Tests		
				1.4	Solar Array Acceptance Tests		
VR.1.2	Natural Frequency Test	The test shall conduct a modal survey (since sweep) of the vehicle using a vibration table.	Test	1.1.3.1	He Tank Inspection	1. All seams properly welded 2. Marked with axes orientation 3. Marked with Component identification 4. Uses proper mechanical fasteners	
				1.1	Propulsion Module Acceptance Test		
				1.1.4.1.2	Pressure Transducer Functional Test		
				1.1.1	Propellant Tank Leak Test		
				1.1.4.1.1	Valve Functional Test		
				1.1.2	Propulsion Module Structural Test		
				1.1.3	He Tank Leak Test		

Add actual results and status as V&V occurs

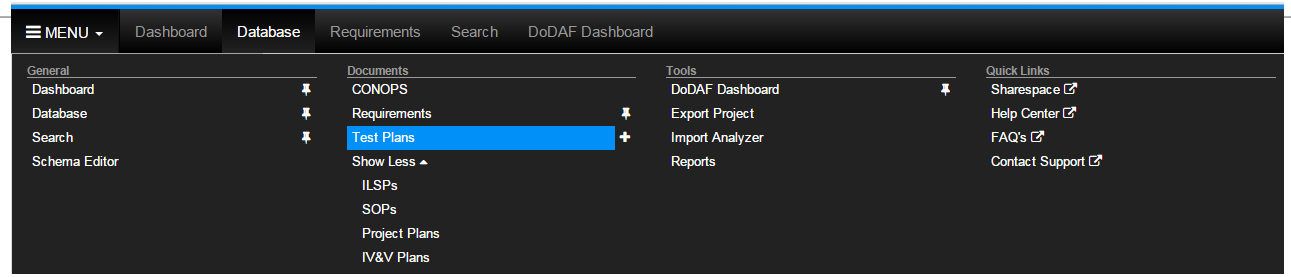
Verification Cross-Reference Matrix

Requirement		Verification Method(s)				
Number	Name	Analysis	Demonstration	Inspection	Modeling & Simulation	Test
VR.1	Space Vehicle First-mode Natural Frequency	X				X
VR.1.1	Natural Frequency Analysis	X				
VR.1.2	Natural Frequency Test					X
VR.2	Appropriate Markings			X		
VR.3	Altitude Accuracy	X				
VR.4	Battery GSE Charge Display		X			
VR.5	Fastener Type			X		

4. DEVELOPING A V&V PLAN USING MBSE

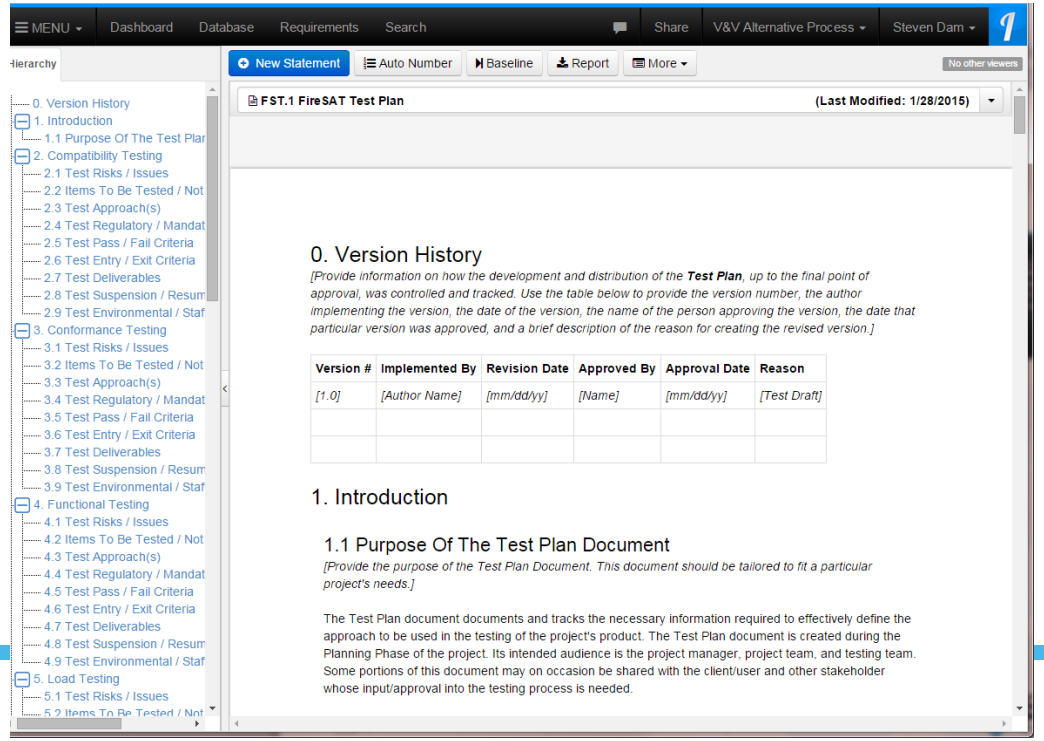
Create a Test Plan

1. Select Test Plans from Menu



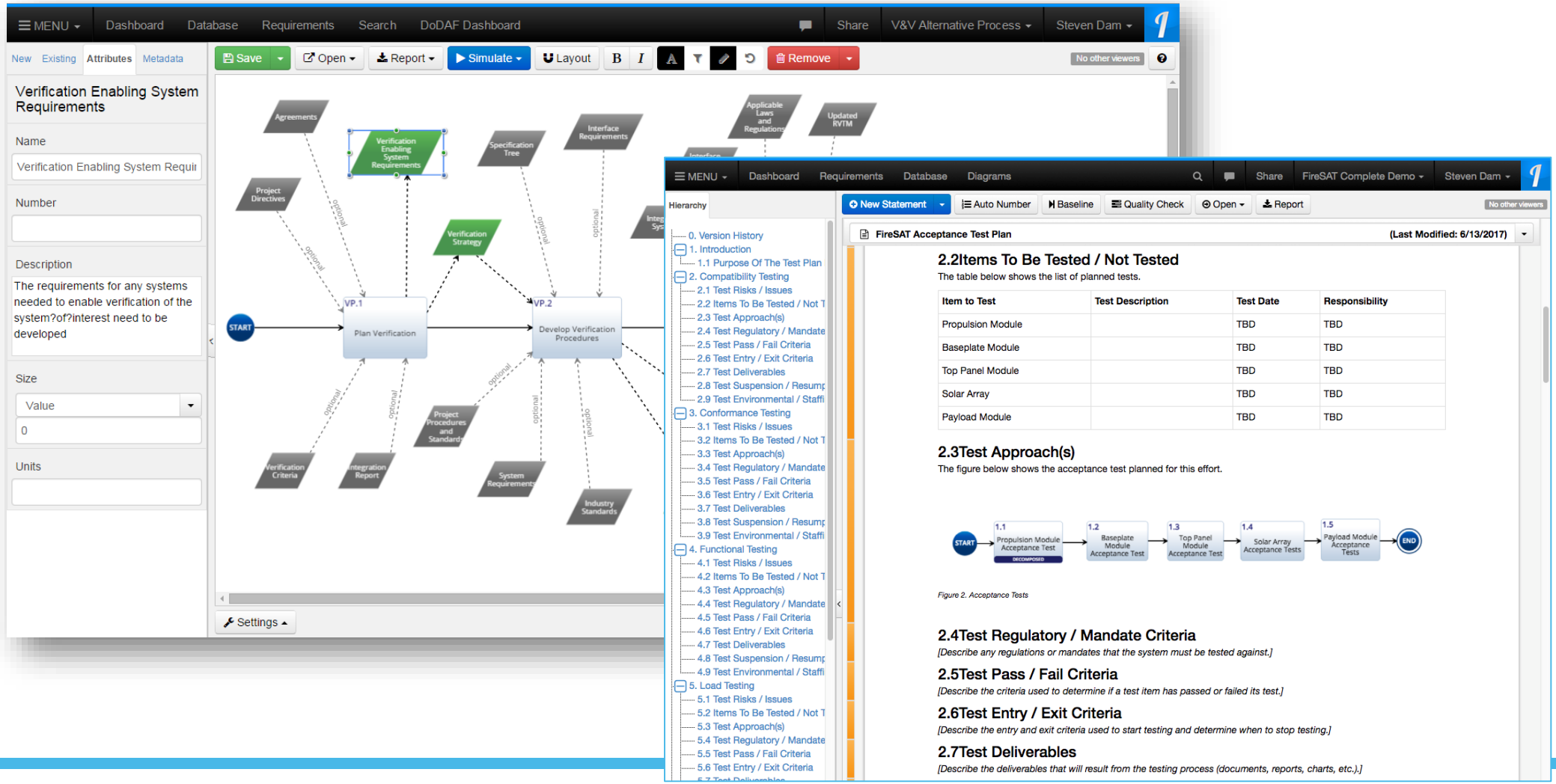
2. Answer questions (use our template or create your own)

3. Fill-in resulting template

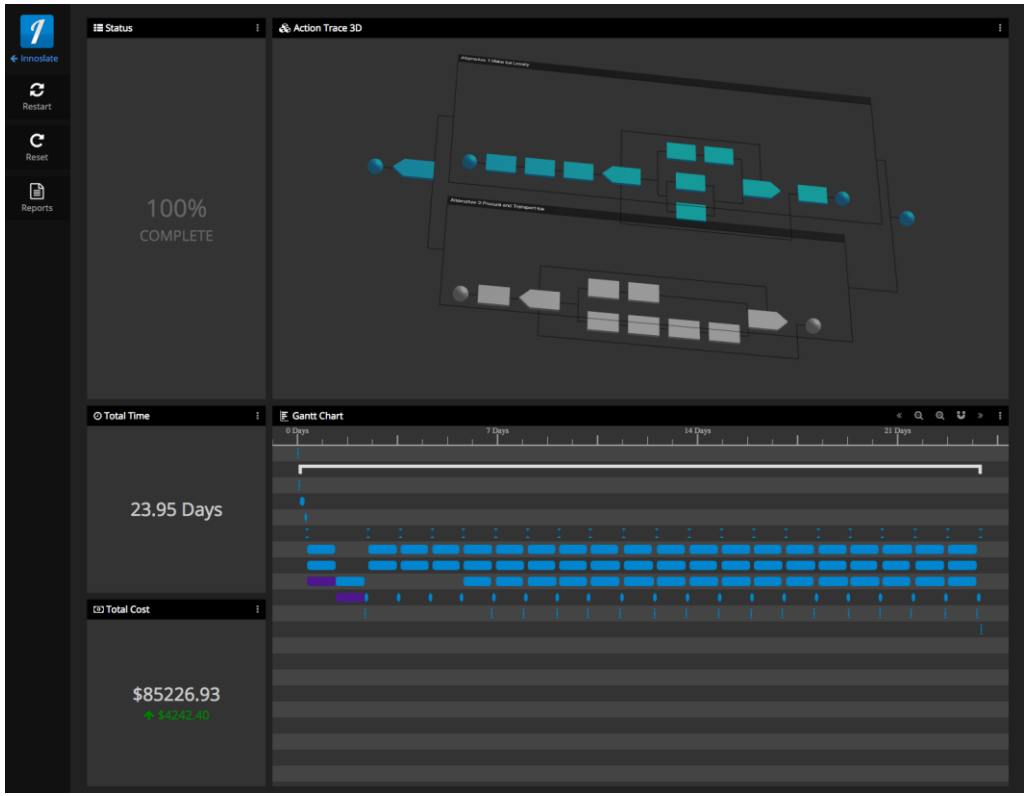


Create Test Processes (Test Cases)

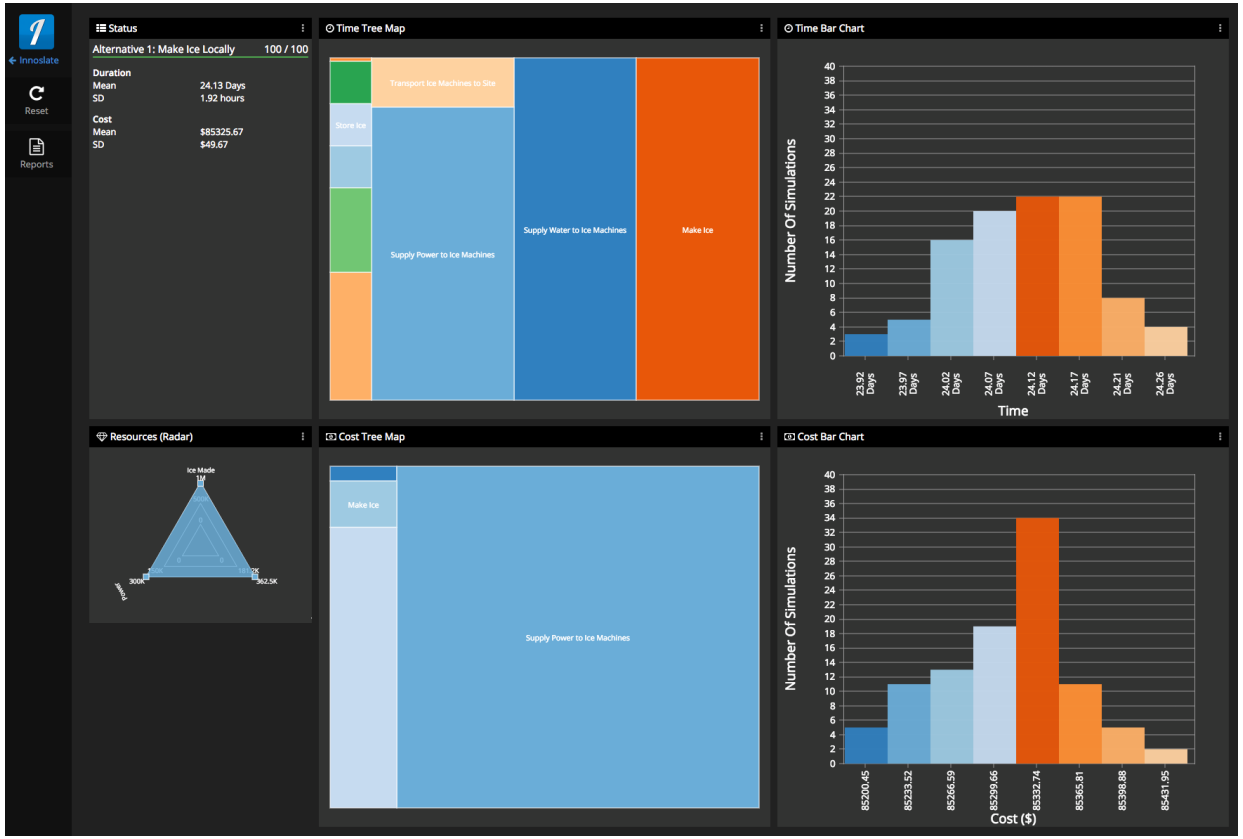
- Model processes
- Embed processes into test plan



Make Sure the Test Cases Work Using Simulation



Watch test case execute at all levels using the Discrete Event Simulator

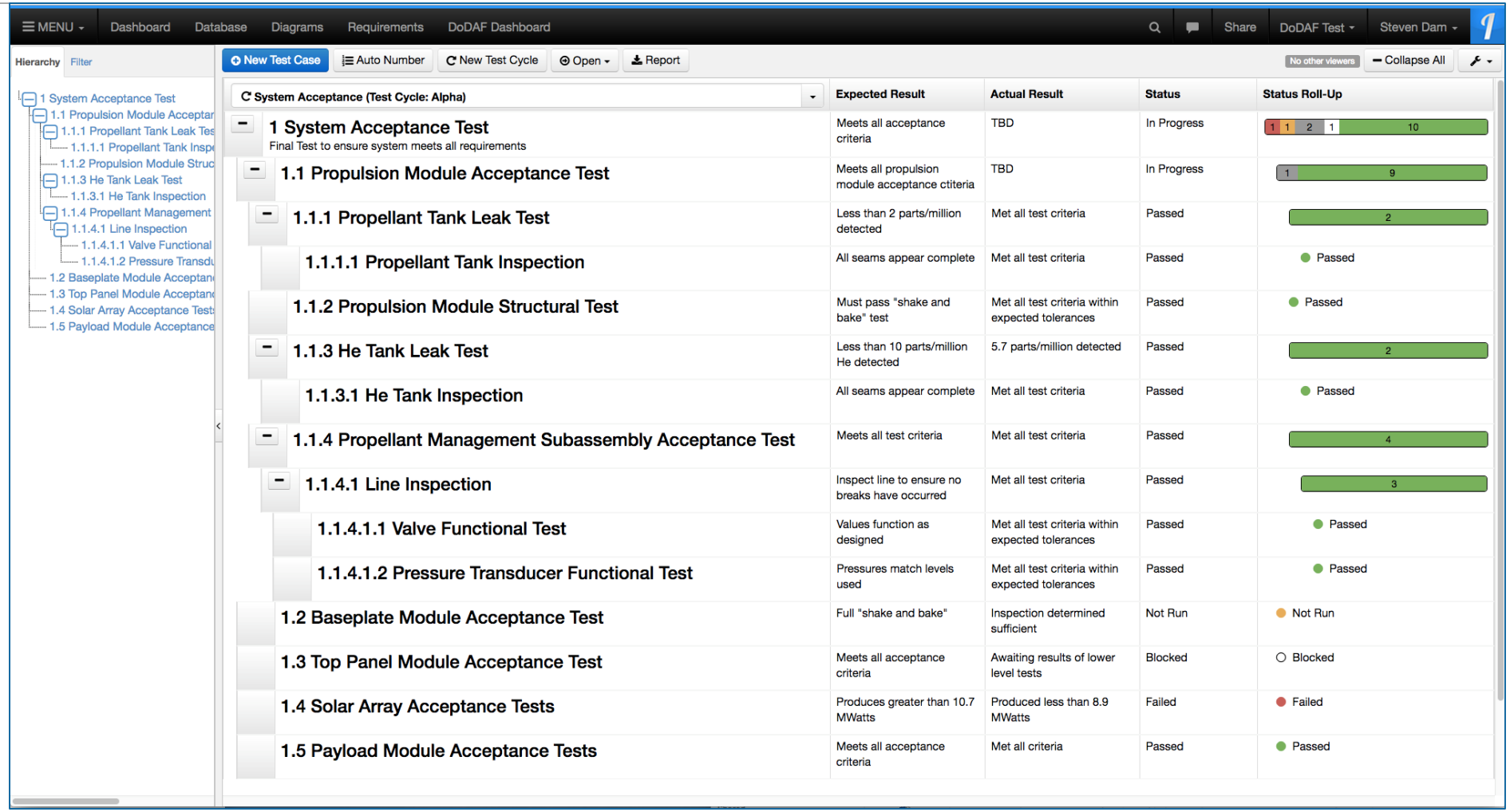


Explore the variation of individual steps to execute the model over many iterations using the Monte Carlo Simulator

You can add resources and costs to enhance your planning

Capture Actual Results and Status During V&V

- Capture actual results as part of test report



Other Information Needed for V&V Activities

- Test Equipment and Facilities (Assets)
- Test Organization (Assets) and Roles/Responsibilities (Statements)
- Risks and mitigations (Risk and many others)
- Criteria (Characteristics/Measures)
- Schedule (Timeline and/or process simulation output)
- Costs (WBS linked to processes)
- Data capture (actual results as Measures and Times)
- Artifacts
- Location(s)
- Decisions (assumptions, issues, etc.)

BENEFITS OF EARLY V&V

Benefits of Early V&V

1. Well defined plan for V&V
   - We know what data we need to capture
   - We know what test facilities we will need
   - We have modeled test scenarios to derive expected results
2. Well crafted verification requirements
   - Opportunity for early customer approval
   - Increase in stakeholder confidence
3. Reduced costs
   - Fewer errors in design
   - Fewer tests may be needed