Are We Doing Enough in Requirements Management?

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

Outline

- Why Do I Need More Than a Spreadsheet?
- What Kinds of Requirements Are We Trying to Capture?
- How Can I Improve My Requirements Management and Analysis Capabilities?

Why Do I Need More Than a Spreadsheet?

What Do Spreadsheets Do?

- Pro
  - Spreadsheets are a wonderful tool for dealing with numbers
    - Excel can perform significant math functions
    - Excel can also plot the numbers very well
- Con
  - Spreadsheets require a schema for collecting information
  - Most Requirements are not pure numbers
    - Functional requirements require context
    - Non-functional requirements are often non-numerical
  - Spreadsheets are not databases (CM, Baselining and other capabilities are difficult)
  - Spreadsheets cannot provide the functional analysis and simulation capabilities needed

So why are we using spreadsheets for requirements management?

Why Are Spreadsheets Used?

- It’s what I have
- I know how to use it
- It’s cheap
- Everyone has MS Office
- My management won’t buy anything else
- The requirements tools are complicated and expensive
- I don’t want to learn a new tool

The end result is poor quality requirements are developed and the cost of fixing them later in the lifecycle grows by orders of magnitude

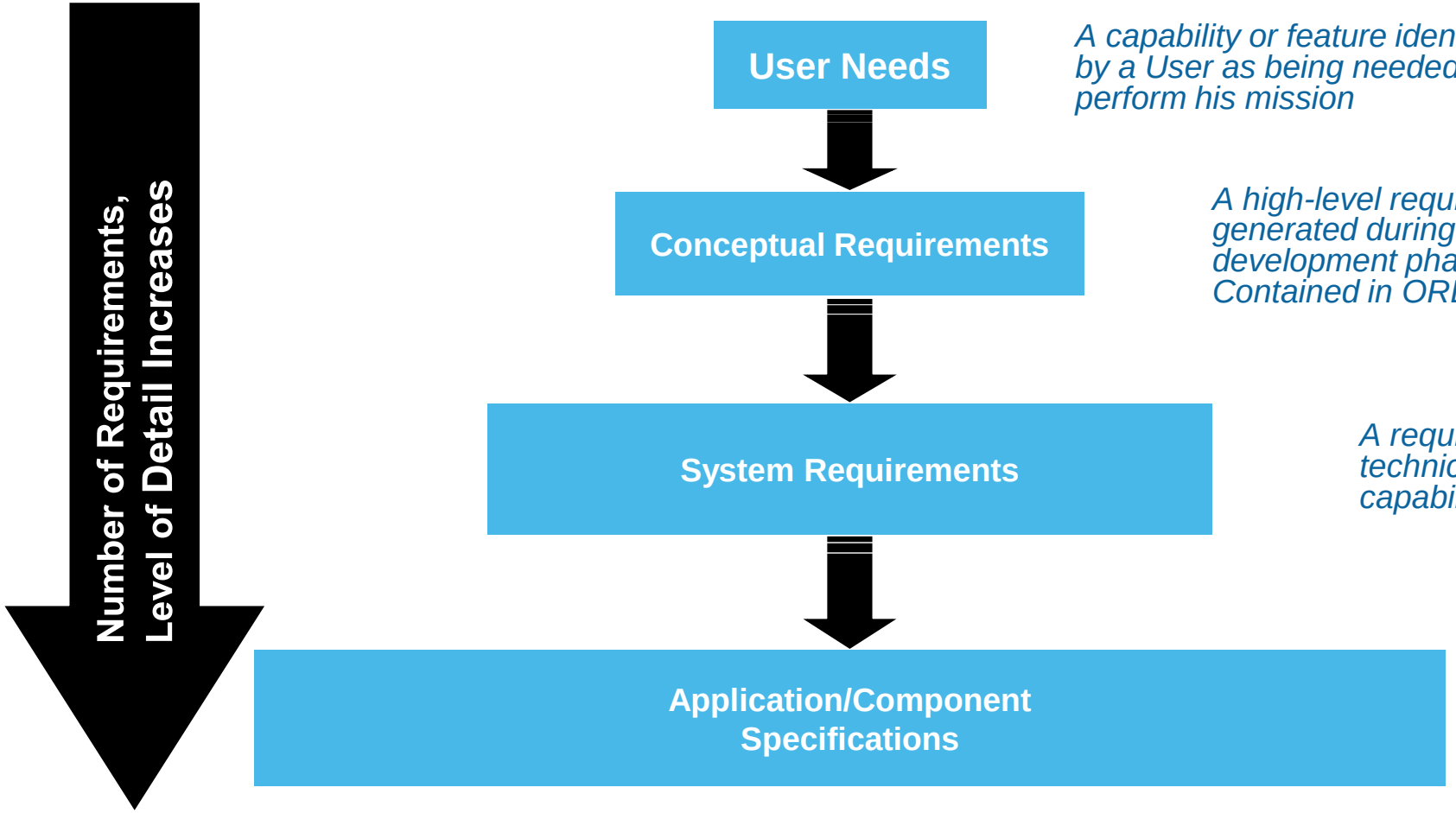
To High Quality Requirements We Need to:

- Support requirements analysis
  - Quality attributes
  - Quality checkers
- Support requirements management
  - Importing capability
  - Configuration Management (i.e. change history, baselining)
- Support functional analysis
  - Includes simulation for verification of models
- Track to Test Results
  - Traceability between test results and requirements
- Be collaborative
  - Commenting capability
- Be scalable
  - Need to store and visualize large number of objects in a database

What Kinds of Requirements Are We Trying to Capture?

What Level Am I Trying to Capture?

The Requirements Hierarchy



A capability or feature identified by a User as being needed to perform his mission

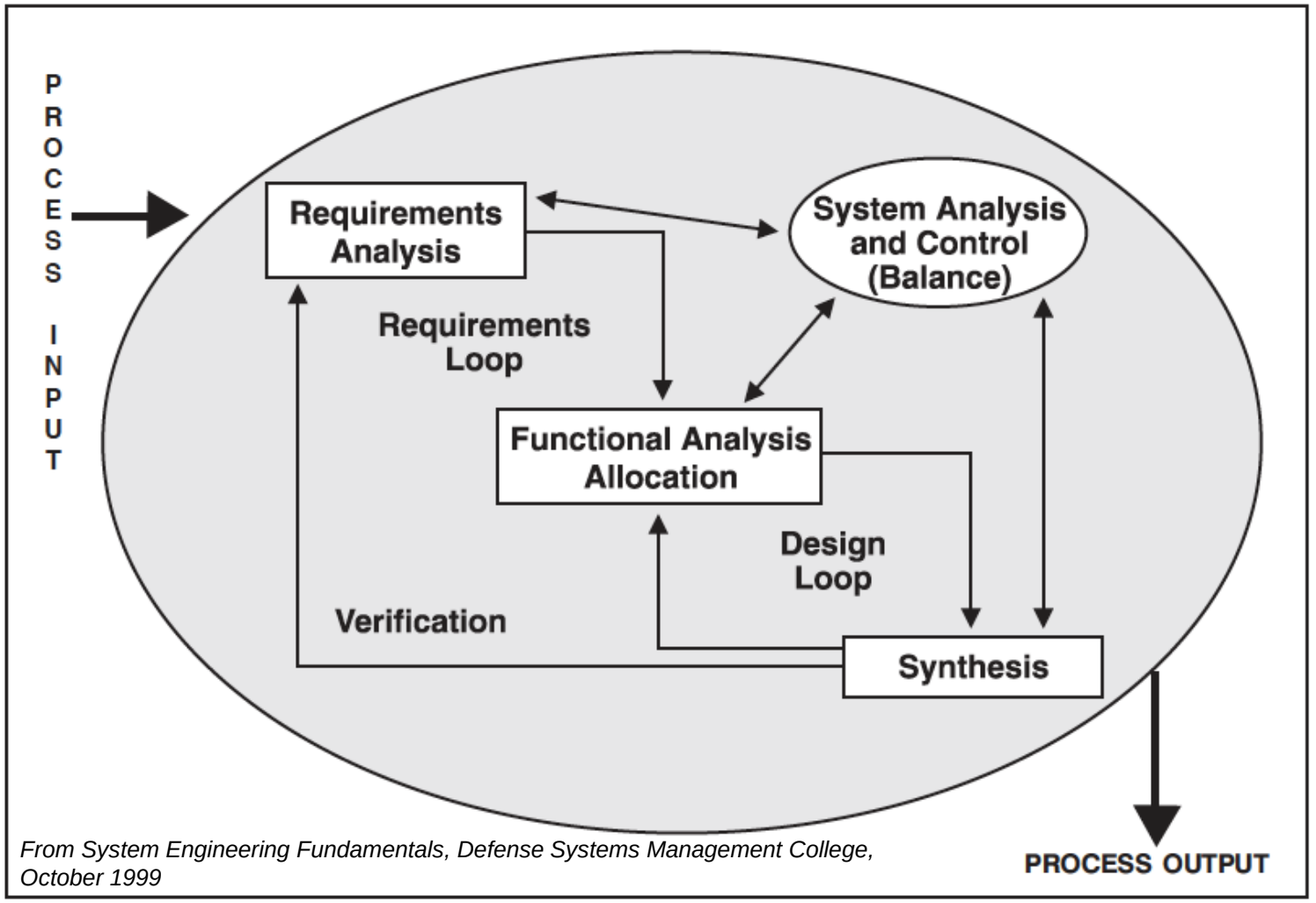
A high-level requirement generated during the concept development phase. Contained in ORD, CONOPS

A requirement that describes in technical language the desired capabilities of a system.

Requirement that is at the level of detail needed for actually designing a new capability. Contained in System Requirements Specification (SRD)

Different kinds of analyses are needed to develop high quality requirements

The SE Process Develops Requirements

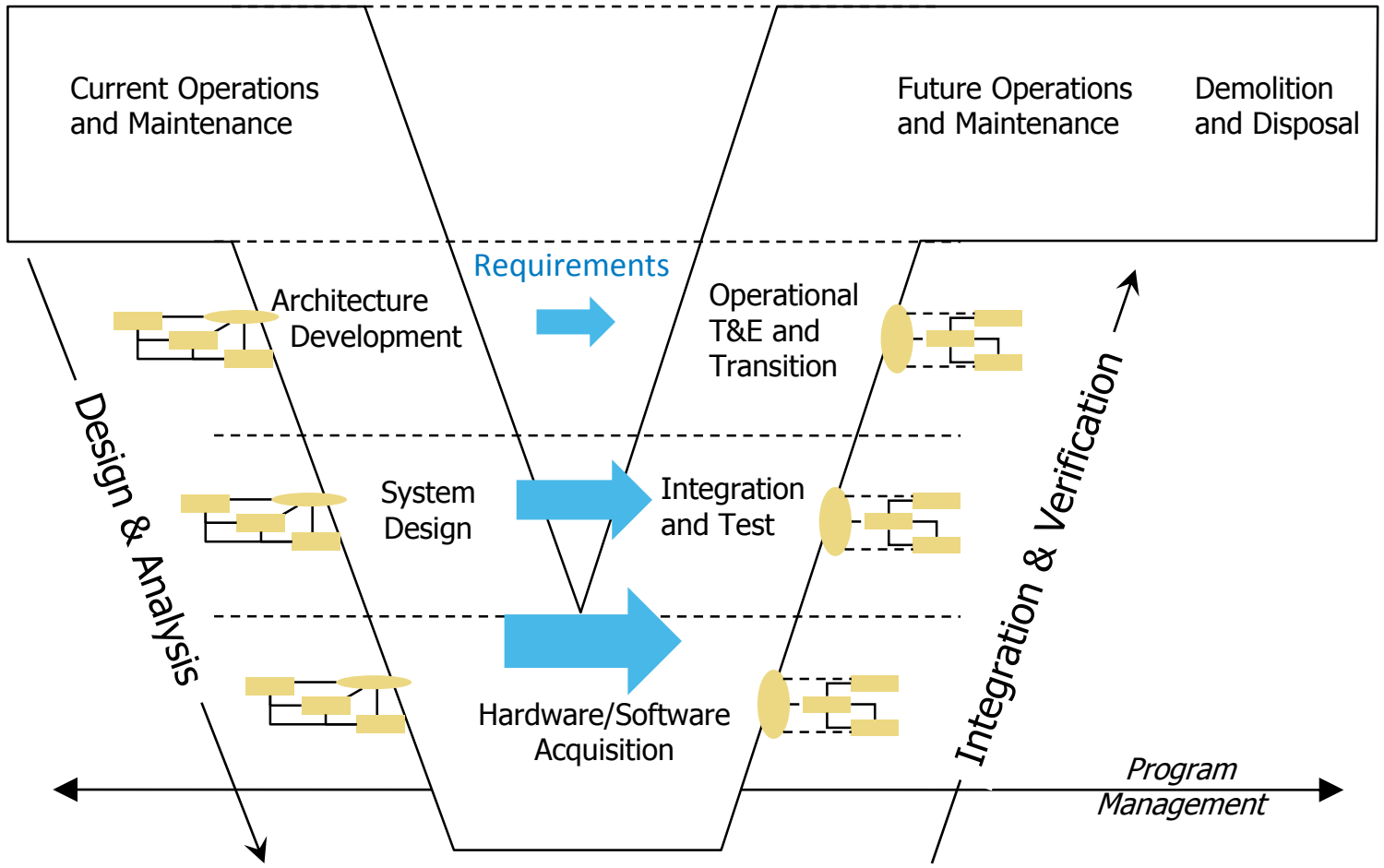


From System Engineering Fundamentals, Defense Systems Management College, October 1999

Figure 1-3. The Systems Engineering Process

- Process input starts with user needs
- Process output results in specifications for the next level of decomposition
- The steps in the process can be executed in any order and simultaneously
- Result is *functional* and *non-functional* requirements for each level

Role of Requirements in the Lifecycle

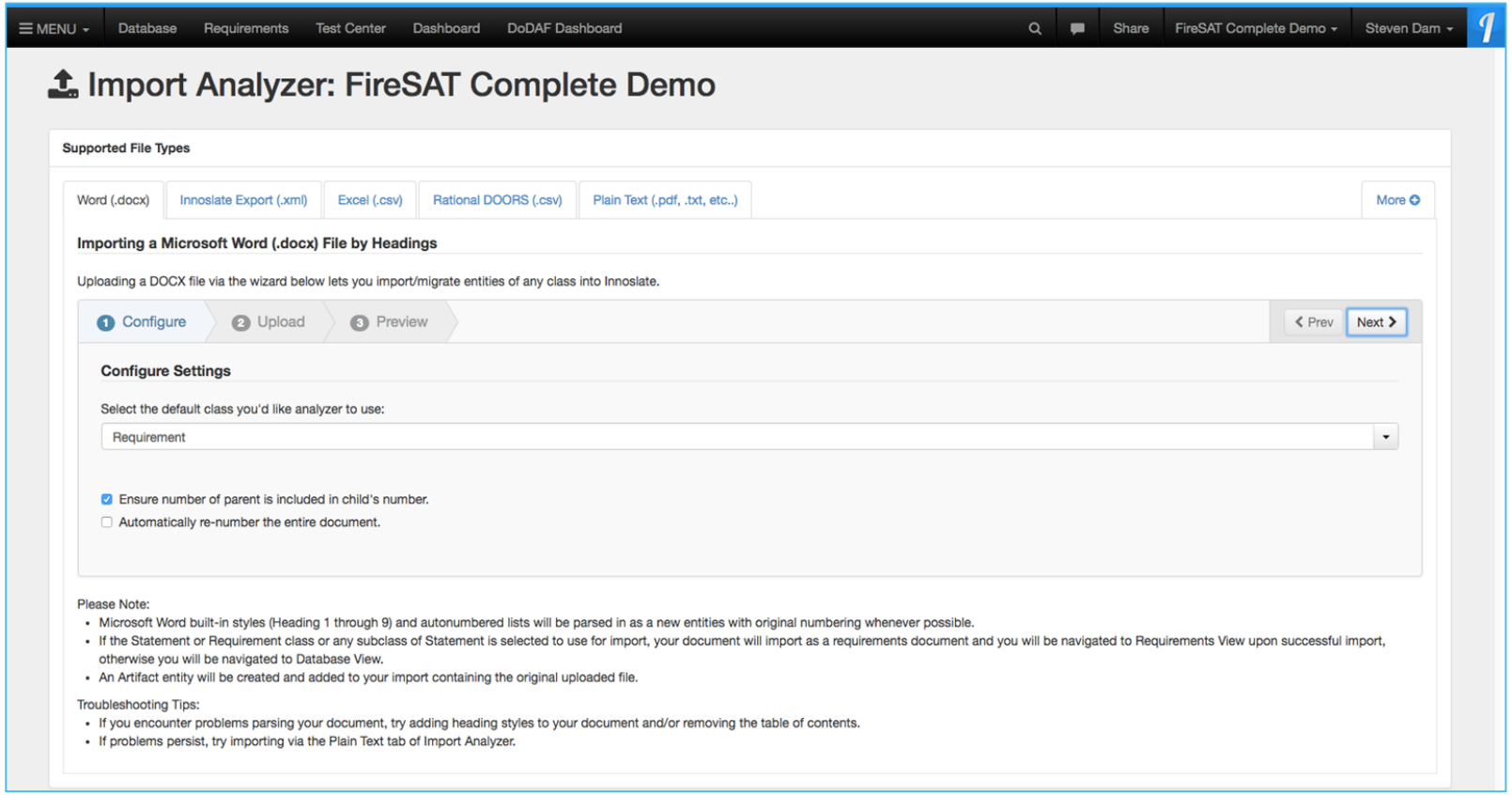


- Requirements are developed at the beginning of the lifecycle
- Resulting components, systems, and complete architectures are validated later in the lifecycle using these requirements
- The number of requirements increases as we decompose the architecture

How Can Improve My Requirements Management and Analysis Capabilities?

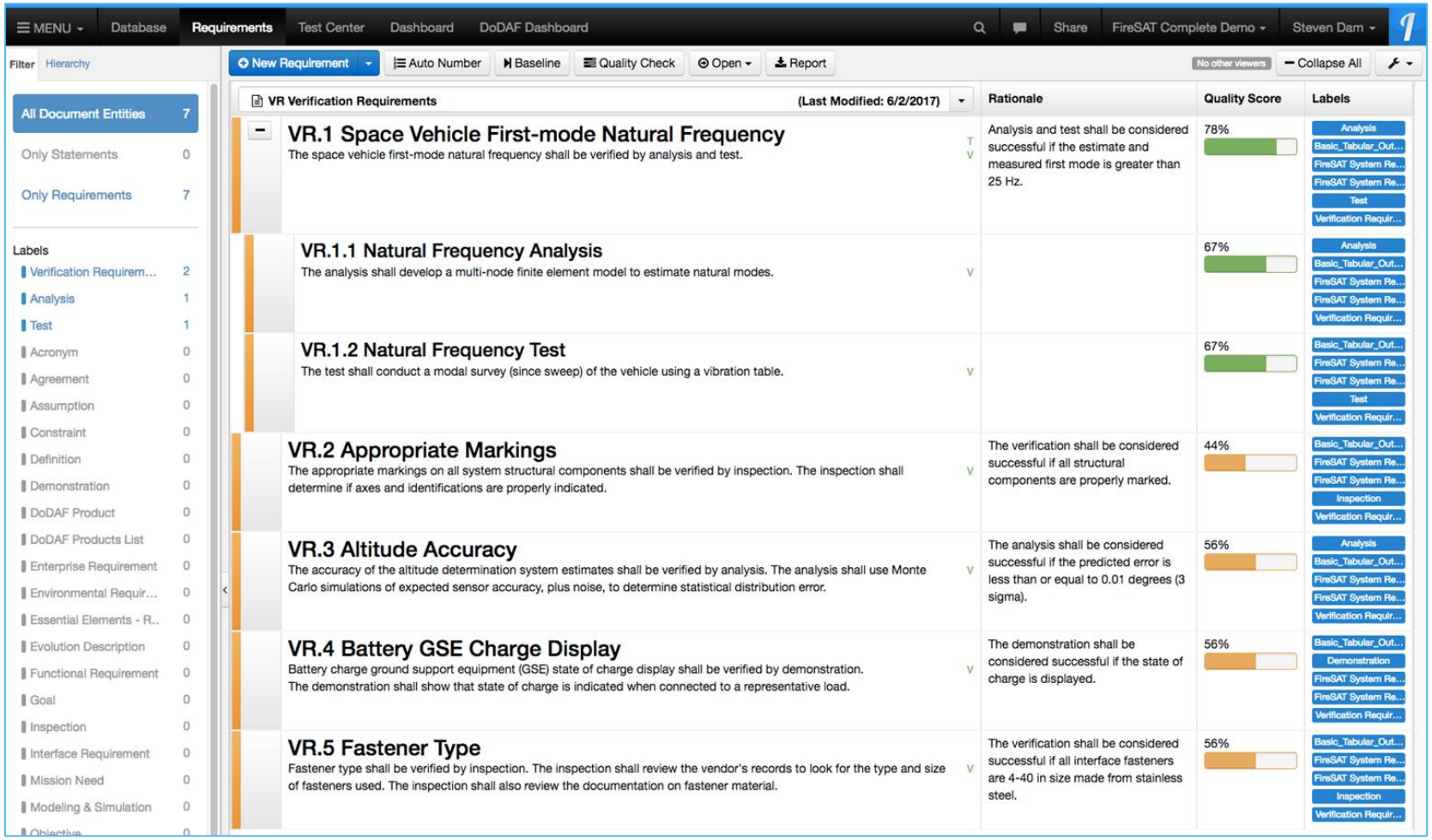
Step 1: Capture Originating Artifacts

- Import directly
  - MS Word files
  - CSV
  - DOORS CVS
  - Plain Text (PDF)
  - XML
- Analyze numbering scheme to create parent-child relationships automatically
- Preview before saving



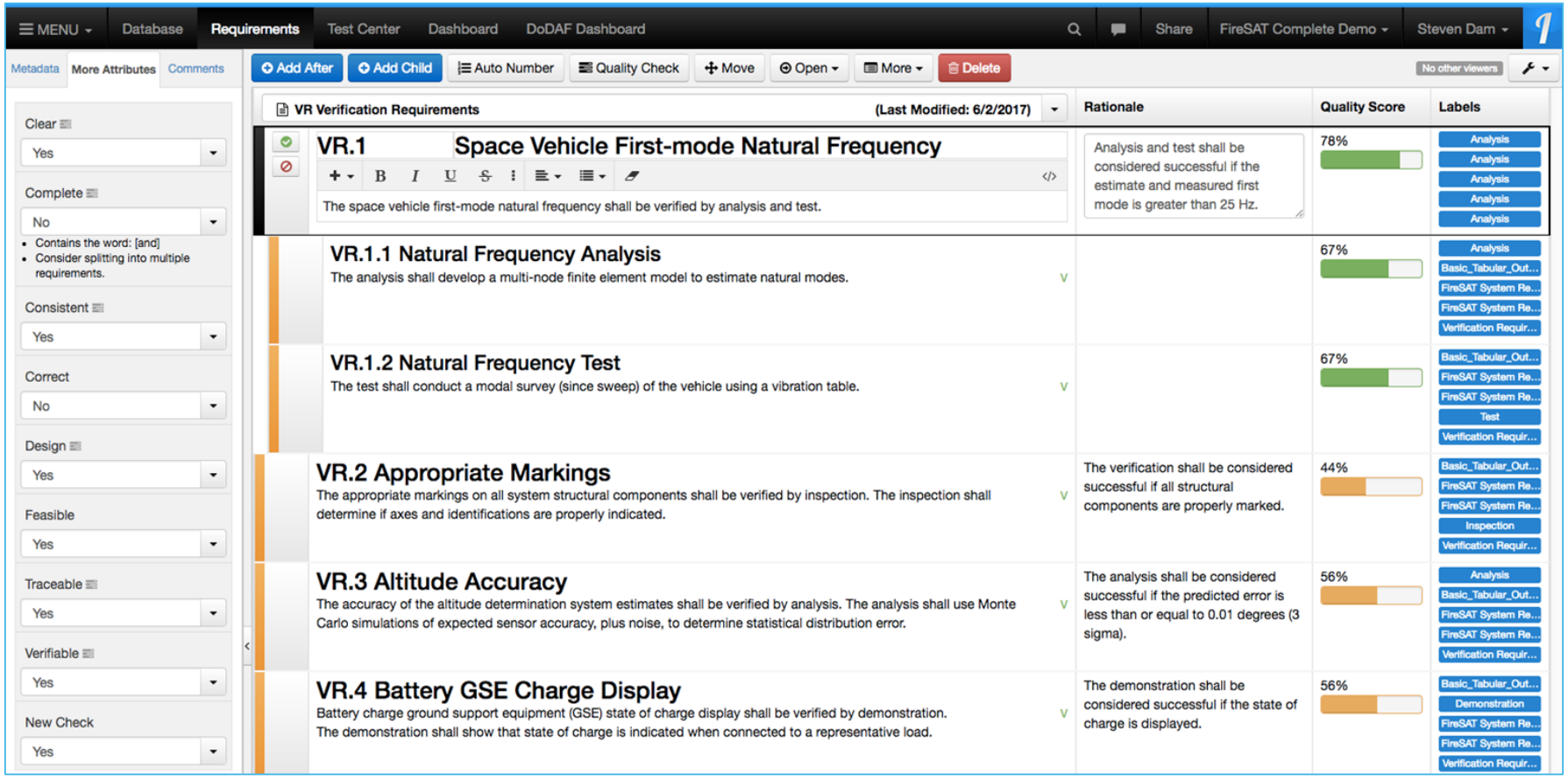
Step 2: Analyze Requirements

- Quality Check each requirement
- Add a Rationale
- Create Reports
- Visualize requirements

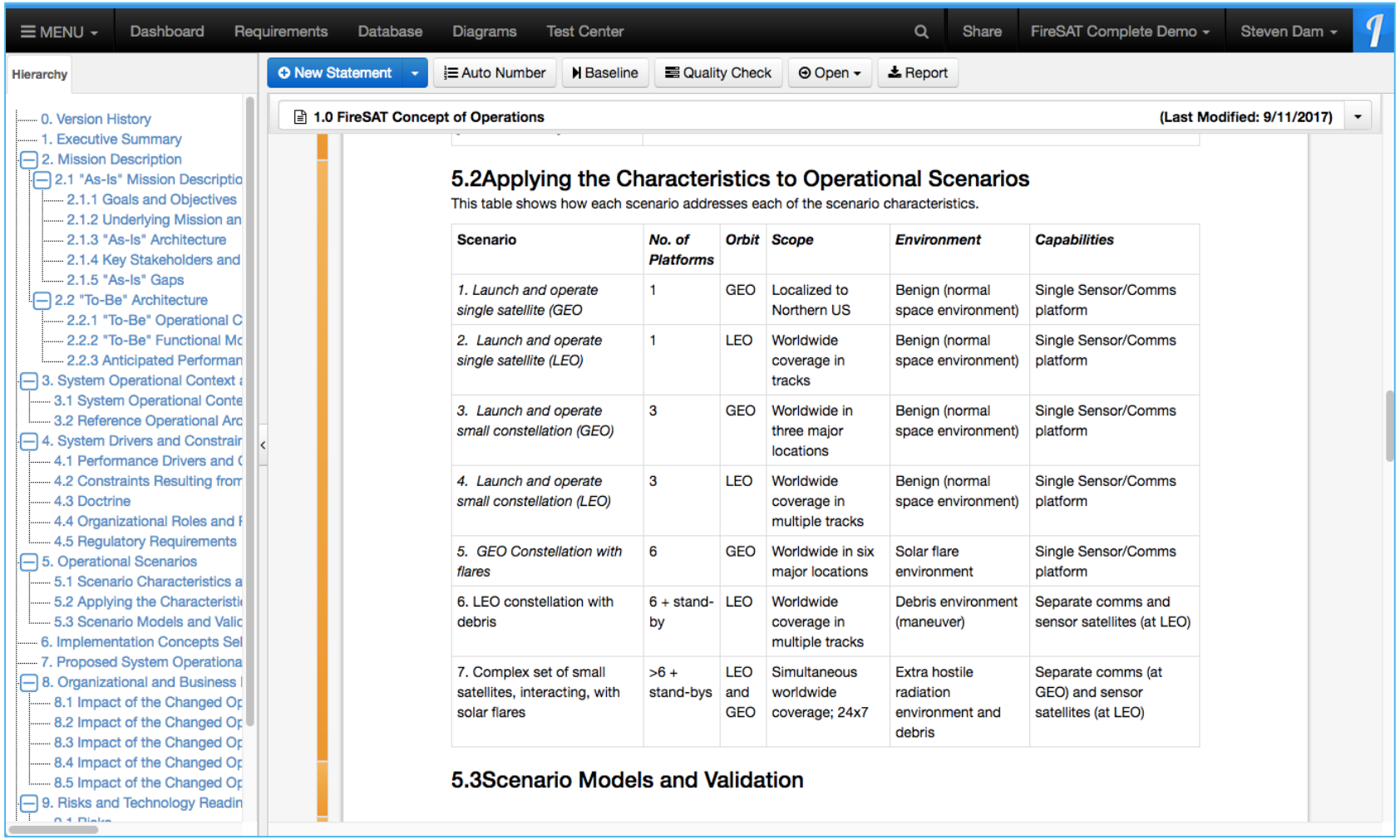


Step 3: Review and Approve Requirements

- Have reviewers provide comments on requirements, but don't let them change the requirement
- If you want reviewers to change requirements create a branch for them to edit
- Baseline requirements when completed



Step 4: Develop Scenarios



5.2Applying the Characteristics to Operational Scenarios

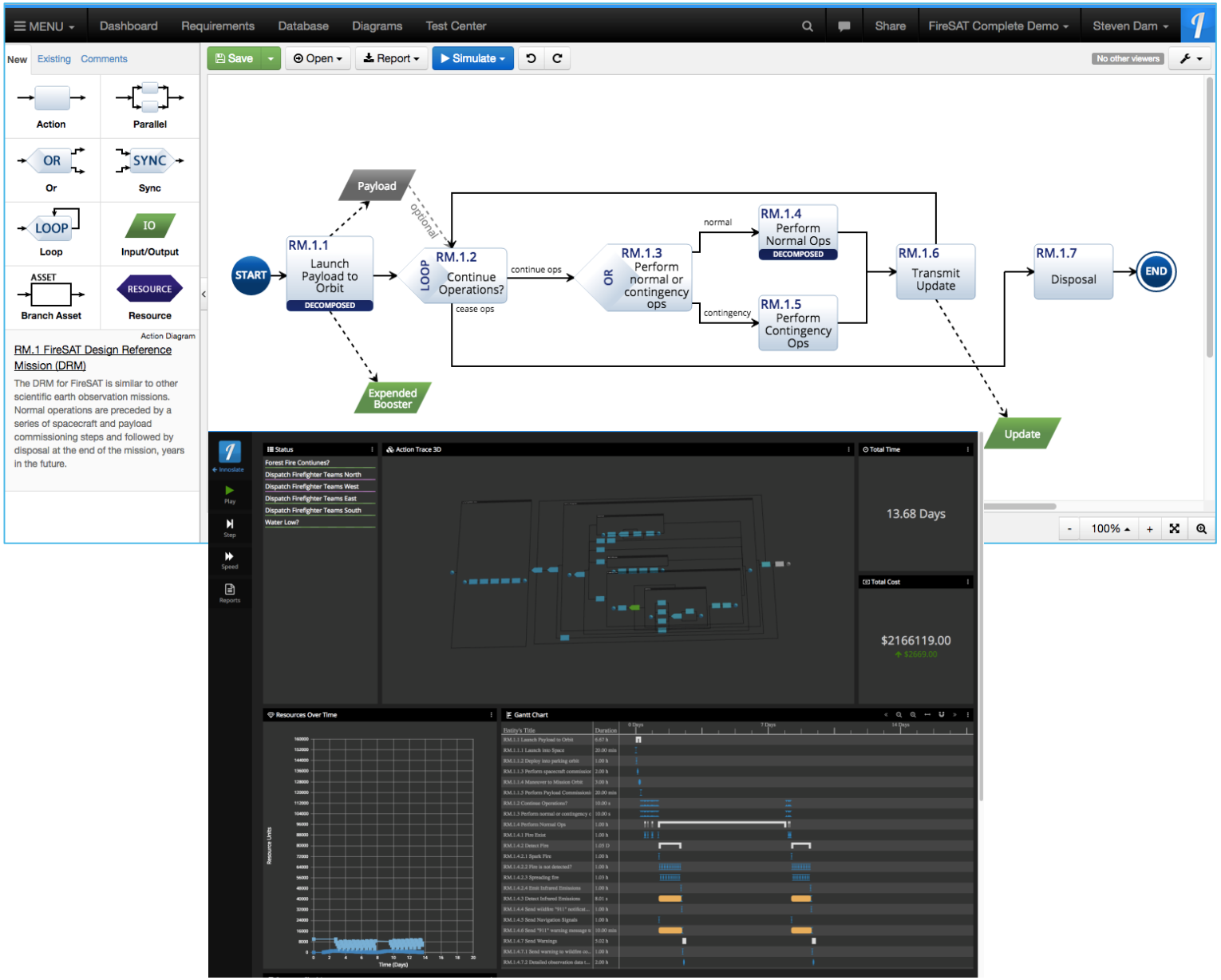
This table shows how each scenario addresses each of the scenario characteristics.

Scenario	*No. of Platforms*	*Orbit*	*Scope*	*Environment*	*Capabilities*
1. Launch and operate single satellite (GEO	1	GEO	Localized to Northern US	Benign (normal space environment)	Single Sensor/Comms platform
2. Launch and operate single satellite (LEO)	1	LEO	Worldwide coverage in tracks	Benign (normal space environment)	Single Sensor/Comms platform
3. Launch and operate small constellation (GEO)	3	GEO	Worldwide in three major locations	Benign (normal space environment)	Single Sensor/Comms platform
4. Launch and operate small constellation (LEO)	3	LEO	Worldwide coverage in multiple tracks	Benign (normal space environment)	Single Sensor/Comms platform
5. GEO Constellation with flares	6	GEO	Worldwide in six major locations	Solar flare environment	Single Sensor/Comms platform
6. LEO constellation with debris	6 + stand-by	LEO	Worldwide coverage in multiple tracks	Debris environment (maneuver)	Separate comms and sensor satellites (at LEO)
7. Complex set of small satellites, interacting, with solar flares	>6 + stand-bys	LEO and GEO	Simultaneous worldwide coverage; 24x7	Extra hostile radiation environment and debris	Separate comms (at GEO) and sensor satellites (at LEO)

5.3Scenario Models and Validation

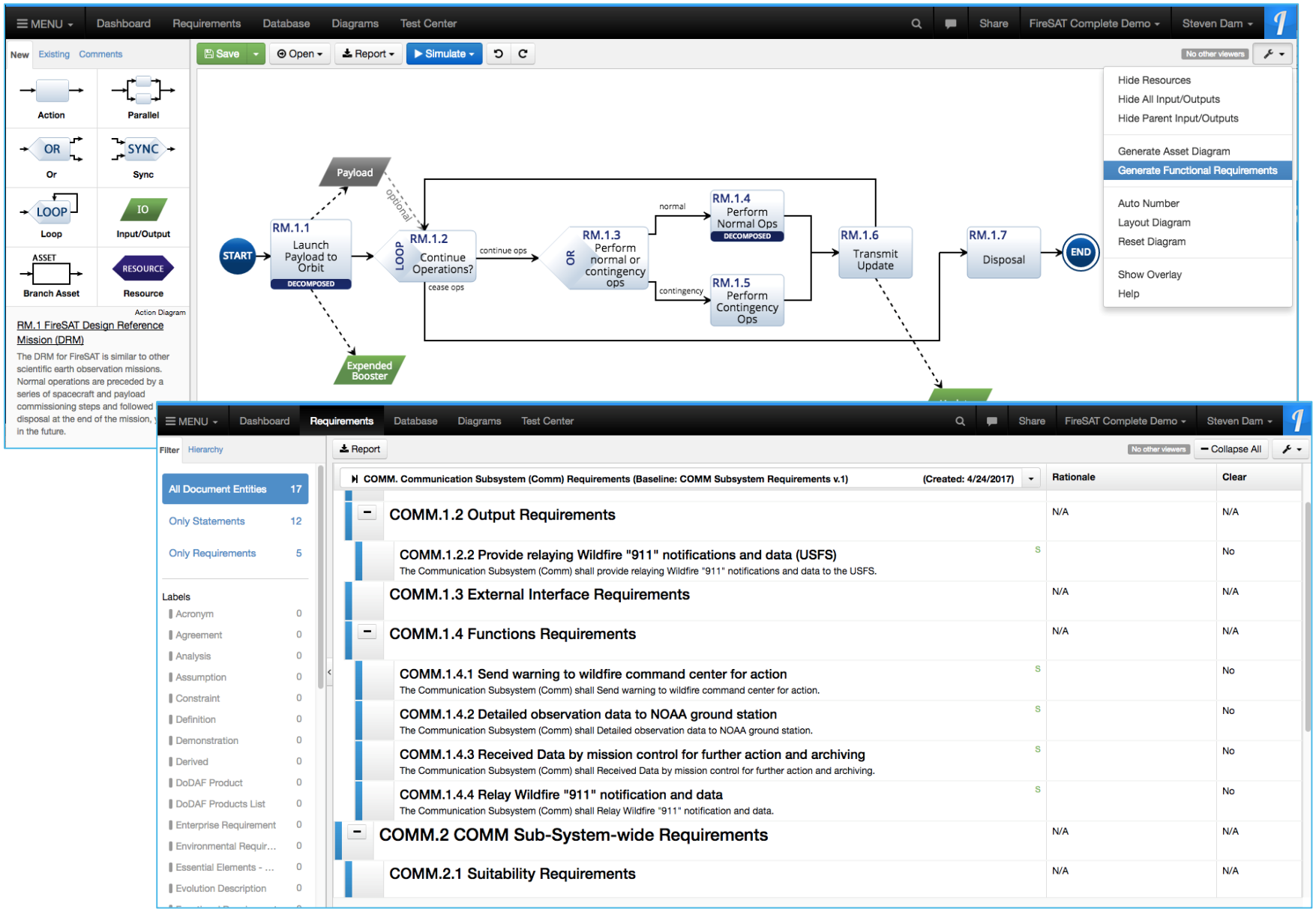
- Scenarios are used to validate user needs and identify functional requirements
- Use CONOPS to create a good set of scenarios

Step 5: Model and Verify Scenarios



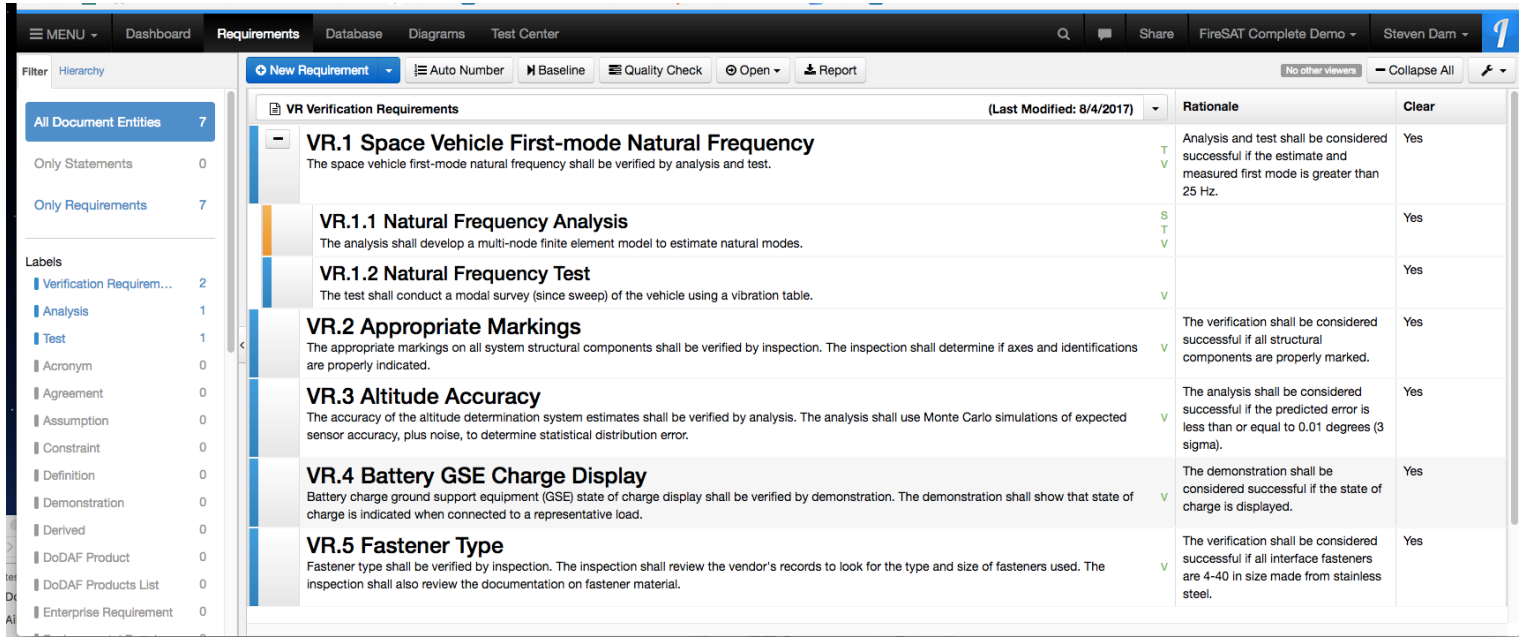
- Decompose to get more detailed functional requirements
- Include physical constraints and resources to obtain non-functional (performance) requirements
- Verify models/ requirements via simulation

Step 6: Generate Lower Level Requirements



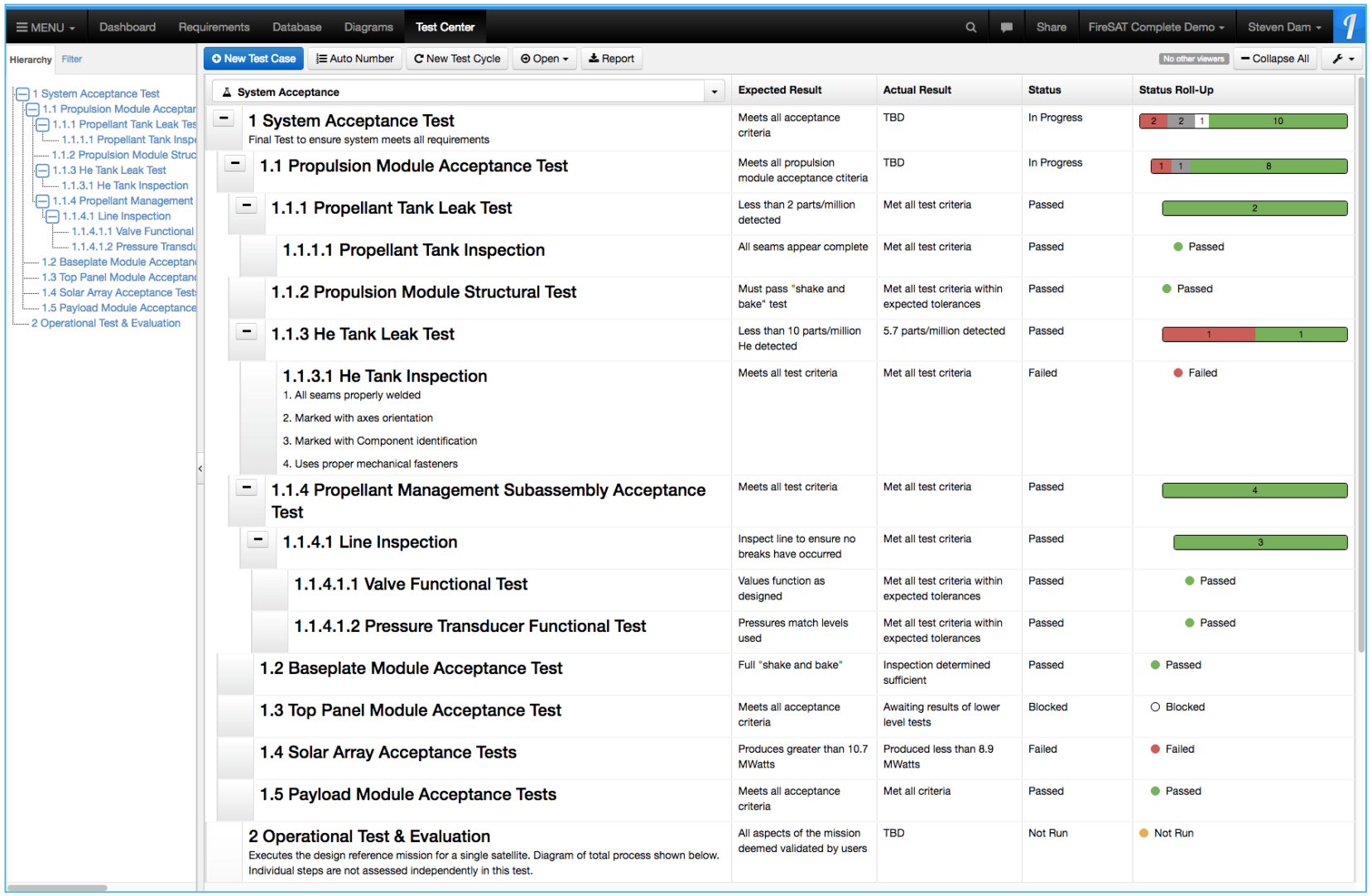
- Generate requirements" from models
- Edit lower level requirements
- Publish (baseline) requirements

Step 7: Develop Verification Requirements



- In parallel with steps 3-6, you can derive the verification requirements
- These requirements specify the verification methods as well

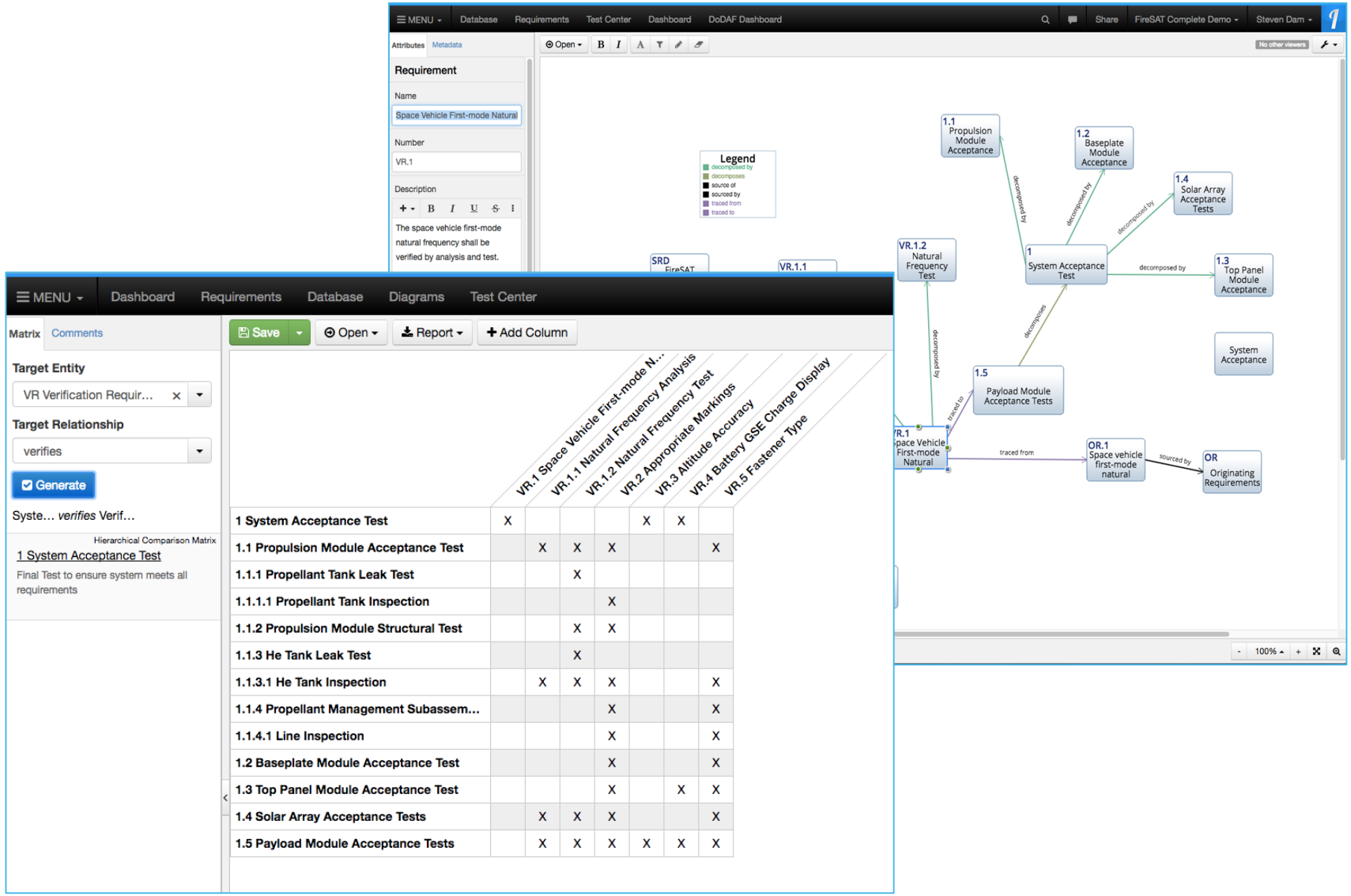
Step 8: Develop Test Cases



- Capture test cases and results (when it's time)
- Roll-up more detailed test cases to higher levels
- Link to test plan and requirements (next slide)

Step 9: Trace Verification Requirements to Test Cases

- Use tools to show all relationships or comparison matrix for a specific relationship
- Modify attributes and relationships as needed
- Produce RVTM and other reports to show requirements are met



Next Steps

- Repeat steps 1-9 as needed for lower levels of decomposition
- Stop when you have the selection criteria to decide what to buy or build
- Then go through the integration and verification process (right side of “V”) and document results as you go
- Make sure that the overall model meets good modeling practices
- Perform risk analysis and other analyses as needed

Summary

- Requirements analysis is a critical part of requirements management
- Modeling and simulation are critical to ensuring you have the requirements you need and are developing systems that work
- To be successful in moving from spreadsheets to model-based systems engineering you need help from your process and tool
- You will know you are successful when you system gets fielded ahead of schedule and under budget