INTUITIVE®

# Bridging the Gap Between MBSE and RAM Using Parametric Diagrams

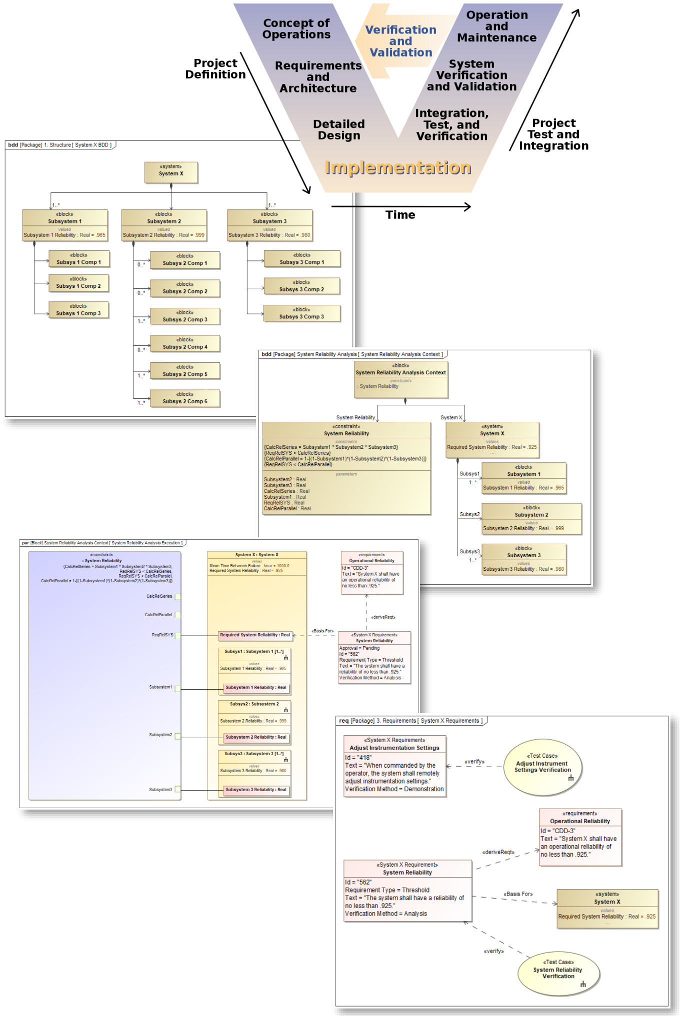
Keith Zook

# Purpose and Topics

- **Purpose**

***Describe how RAM analyses can be performed within a system's Model Based Systems Engineering (MBSE) environment***

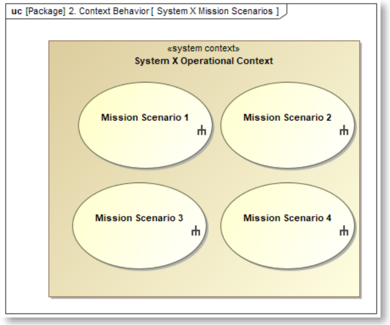
- **Topics**
  - Introduction to MBSE
  - Model and tool description
  - MBSE language and diagram types
  - Modeled behavior, requirements, and structure
  - A simple Reliability analysis
  - Correlation to requirement verification
  - Other potential application areas
  - Summary



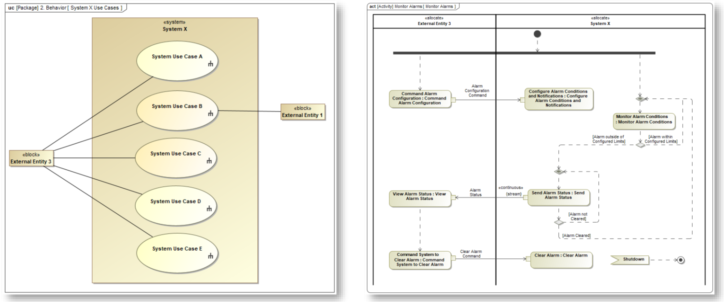
# MBSE – Model Based SYSTEMS ENGINEERING

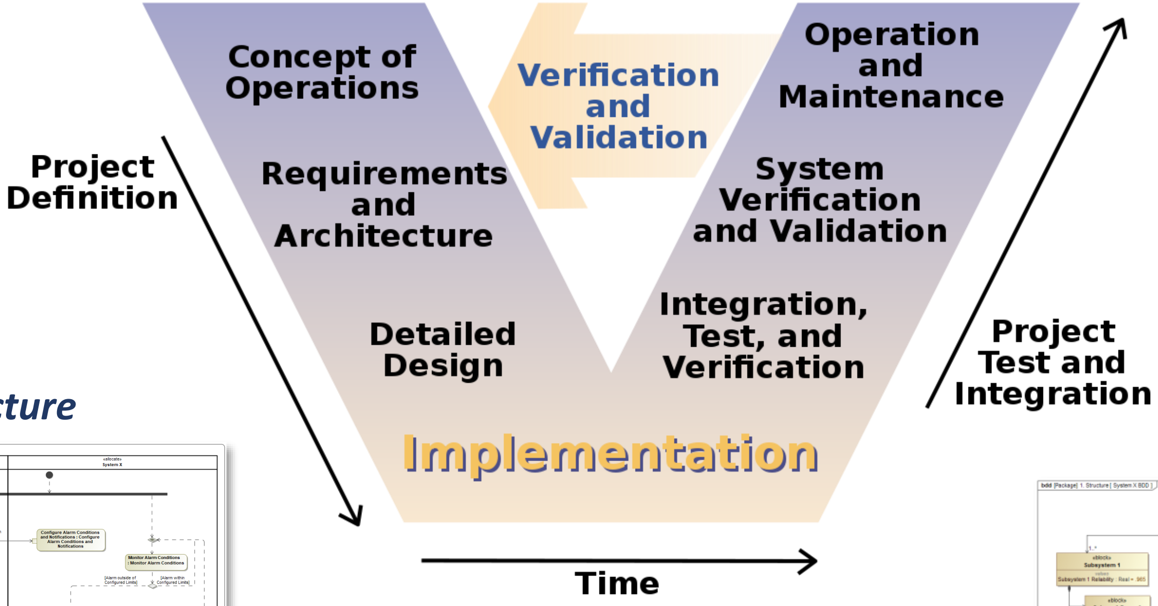
INTUITIVE®

*Operational Needs Analysis*

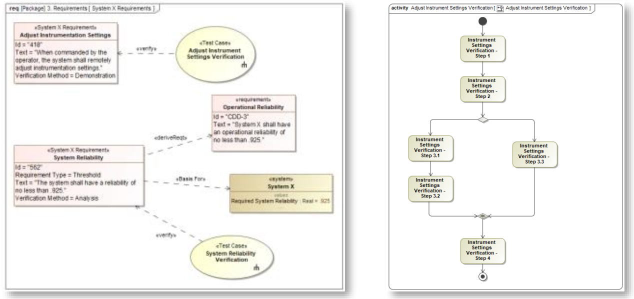


*Functional Architecture*

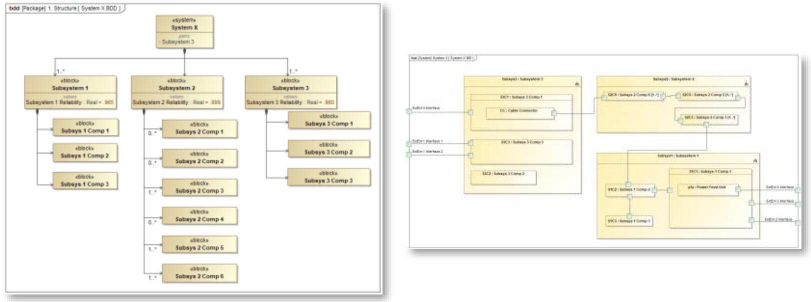





*Test Plans and Procedures*



*Physical Architecture, Subsystems, and Interfaces*

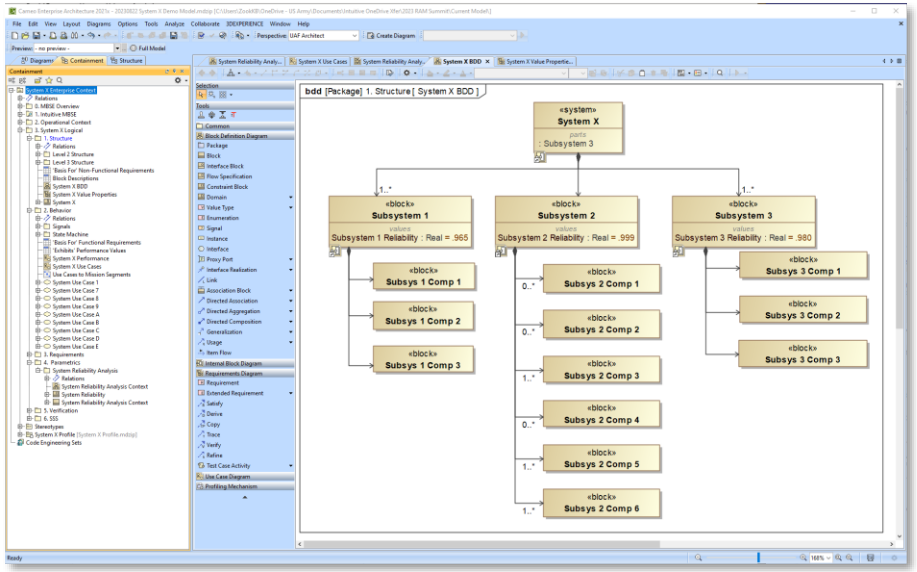


***We use models that aid us in performing the Systems Engineering 'V' process***

# Today’s Model and MBSE Tool

INTUITIVE®

- MBSE Tool
  - Cameo Enterprise Architecture
    - aka MagicDraw
  - Cameo Simulation Toolkit
- Notation
  - SysML
- System of Interest
  - System X
  - Completely notional
  - It could be anything …

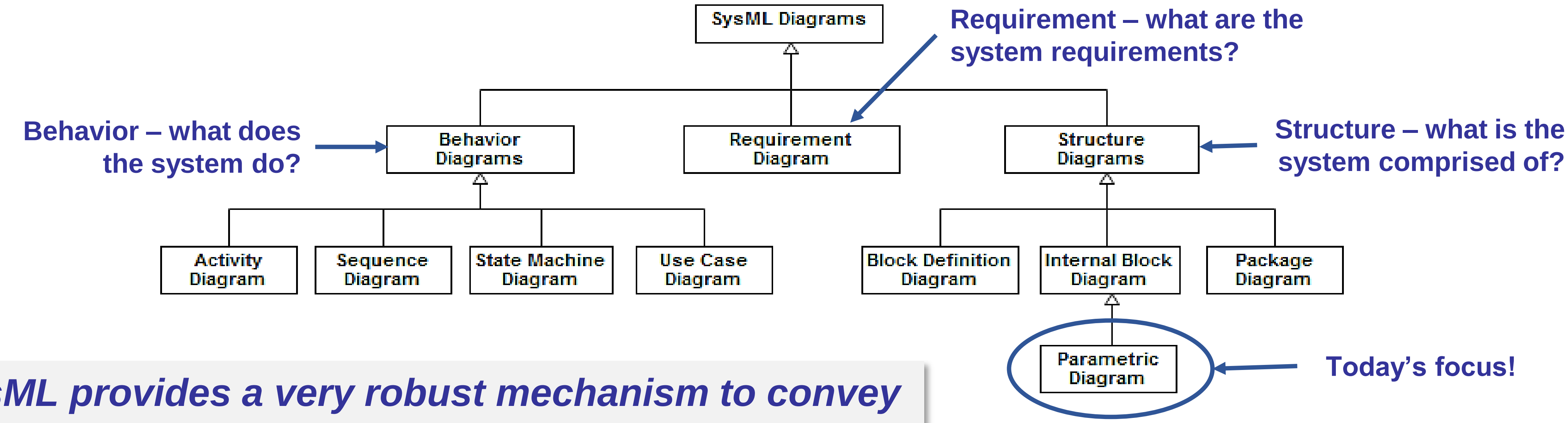



# The MBSE Language

We use the Systems Modeling Language (SysML)

- 3 diagram categories; 9 diagram types
- Defined element types and relationships between elements (the metamodel)
- Everything on the diagram has semantic meaning; not just a ppt shape or line
- Customize as necessary
- Other notations and frameworks exist; SysML is foundational

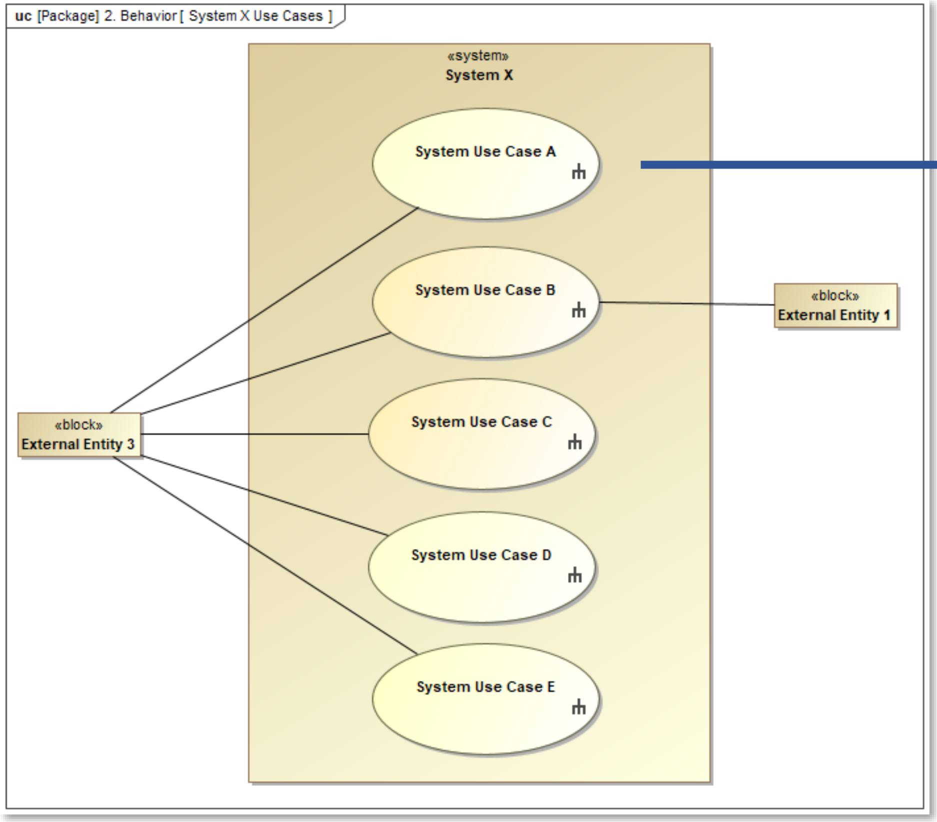
**OMG – Object Management Group**



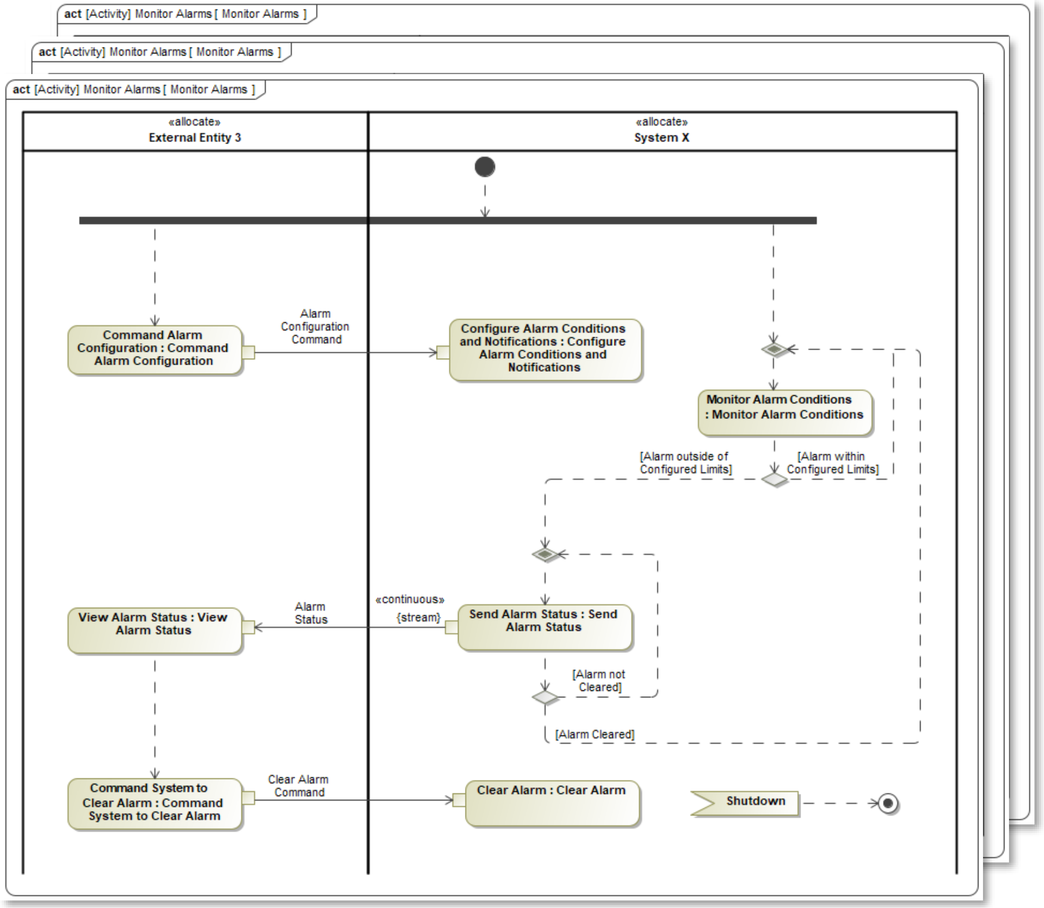

***SysML provides a very robust mechanism to convey system behavior, structure, and requirements***

# System X Behavior

### System Use Cases – identify top-level functions and external interactions




### Activity Diagrams elaborate each Use Case with detailed functional flow

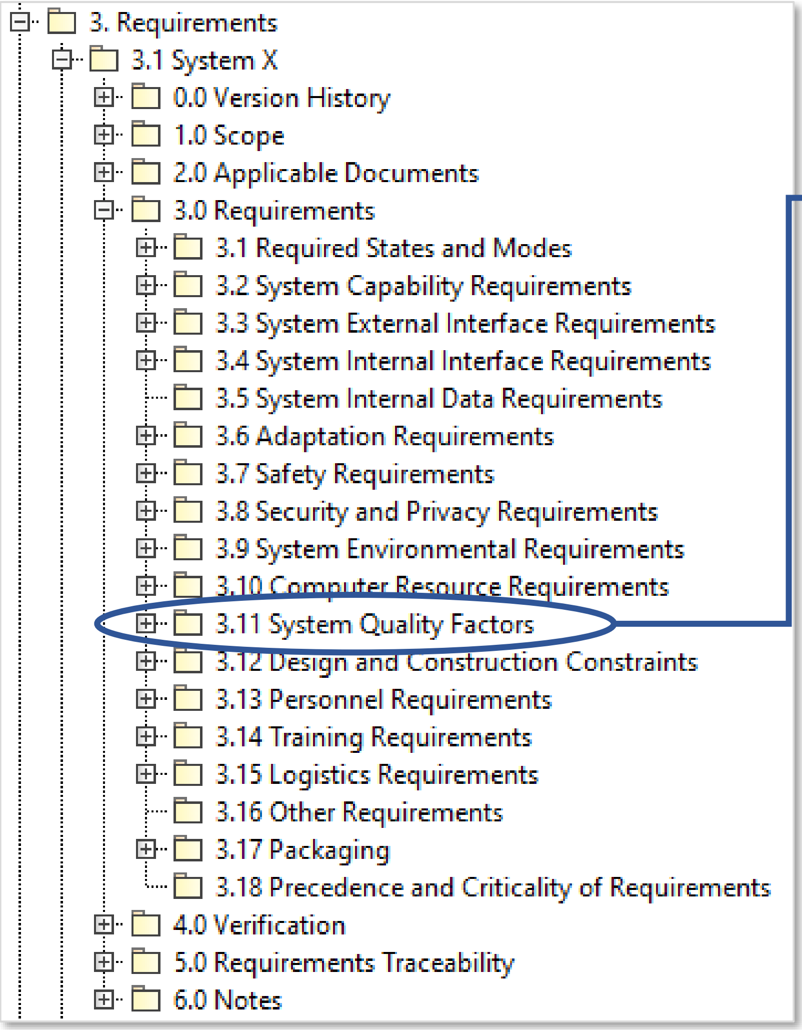



***Basis for the system's functional requirements; supports Functional Hazard Analyses (FHA)***

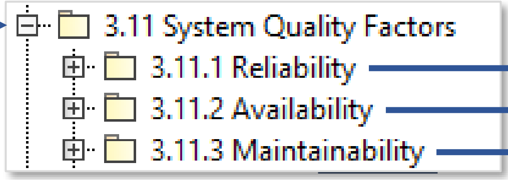
# System X Requirements

INTUITIVE®

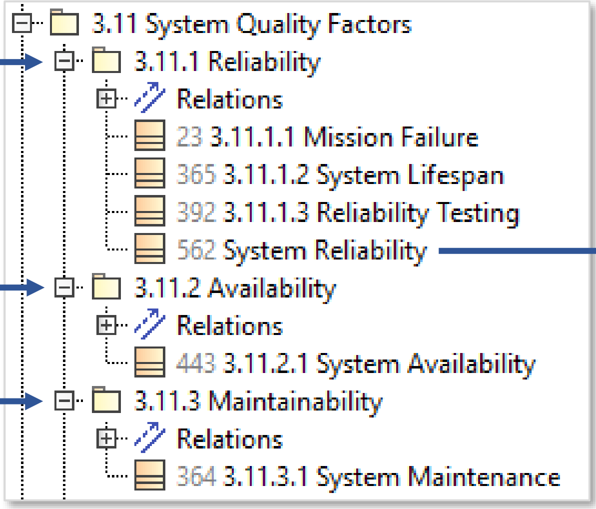
**Model organization – reflects standard specification content (MIL-STD-961, DI-IPSC-81431A)**



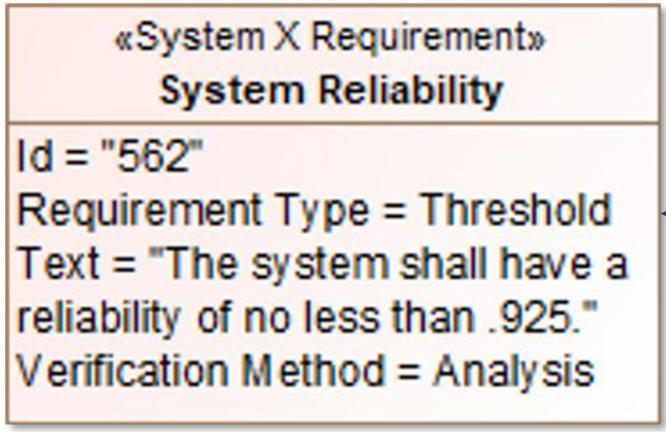

**Typically where RAM requirements live**




**Expanded to show actual Requirement elements**



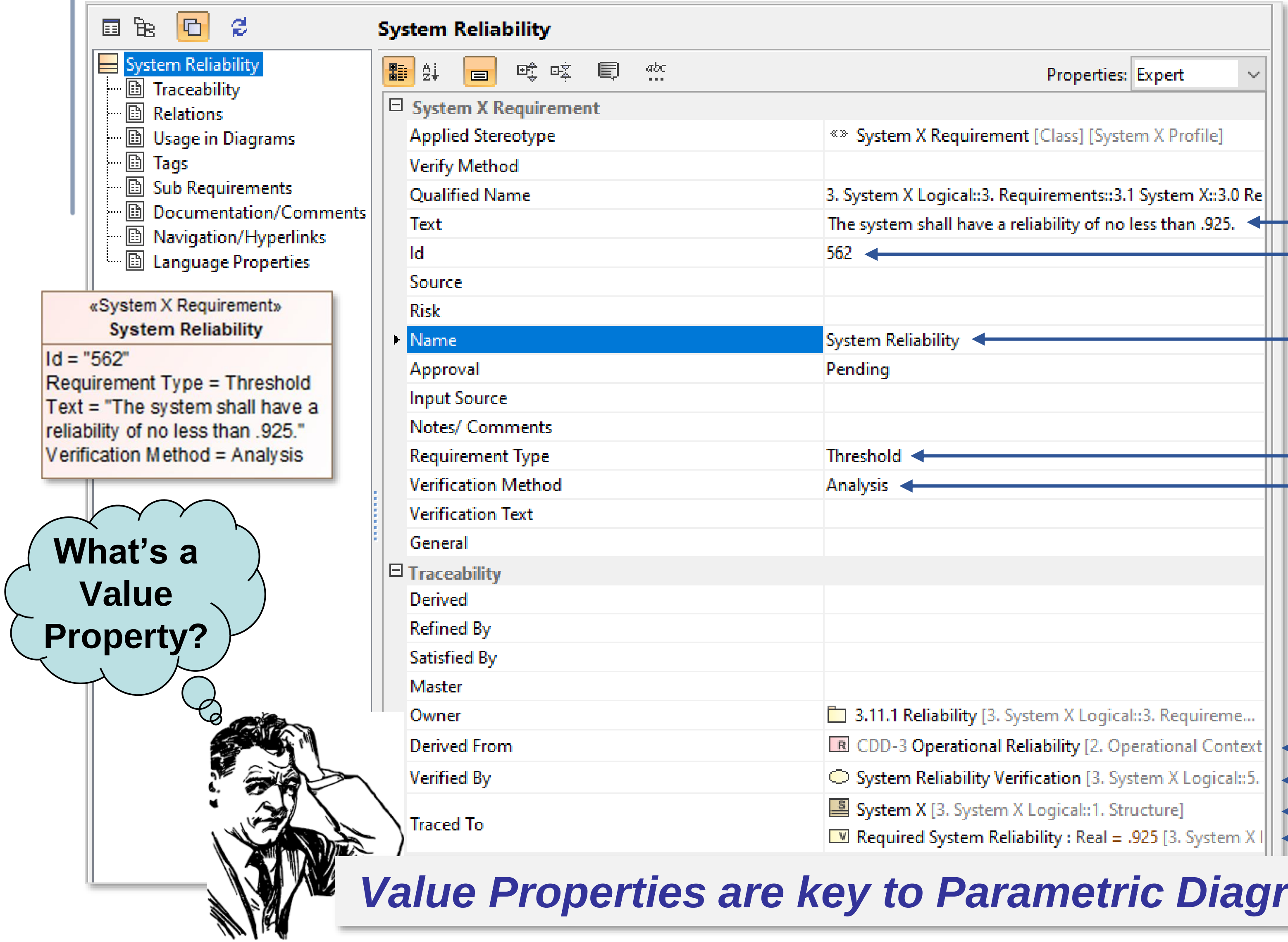

**Reliability Requirement Element**




***Requirement elements can directly populate specifications or simply be used in tables and diagrams***

# Reliability Requirement Element Specification

INTUITIVE®




## Attributes

**Requirement Text – The system shall have a reliability of no less than .925.**

**Requirement ID – 562**

**Requirement Name – System Reliability**

**Requirement Type - Threshold**

**Requirement Verification Method - Analysis**

## Relations

**Parent Requirement – CDD-3**

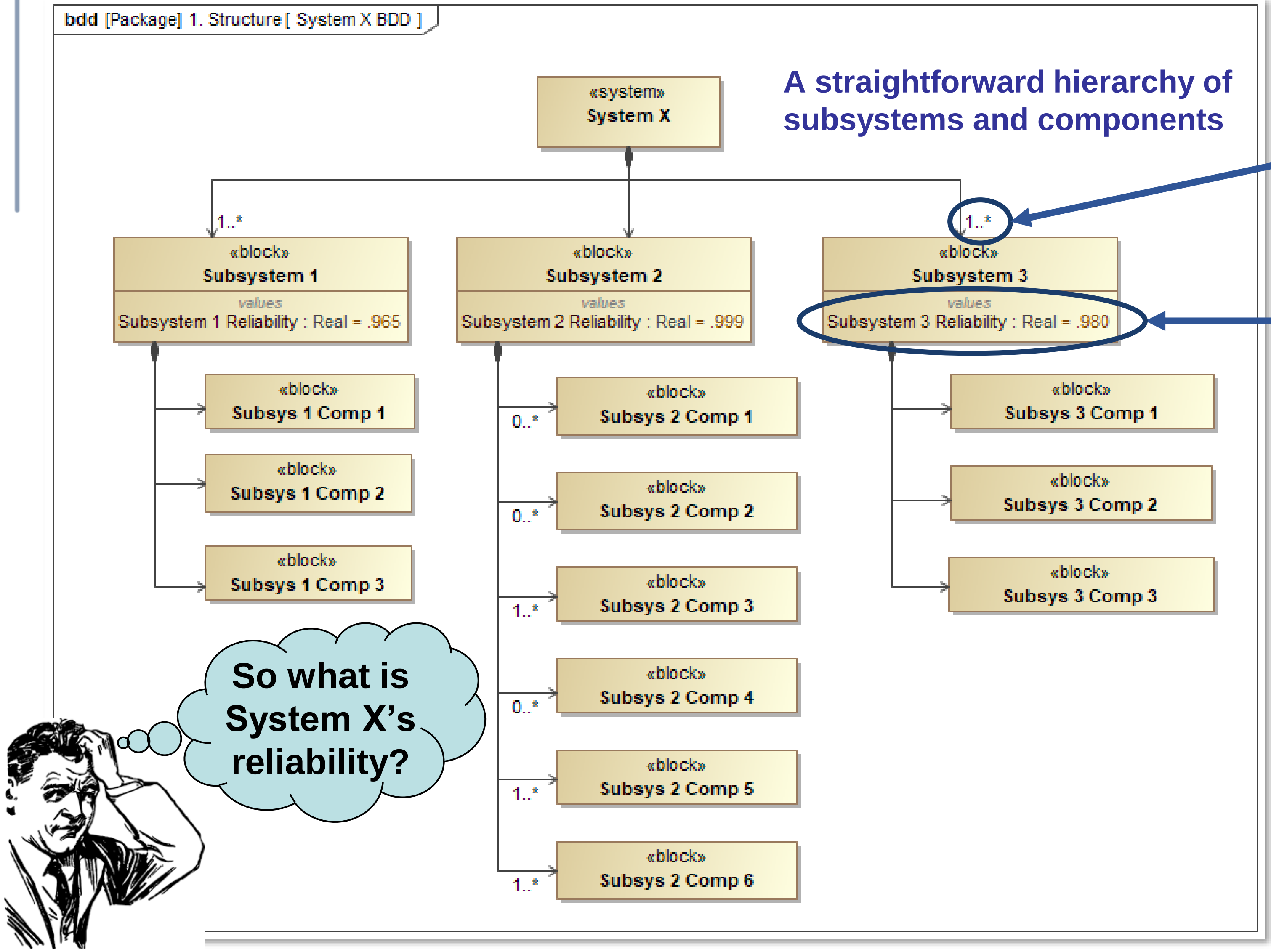
**Requirement Test Case – how will verification be performed?**

**Requirement for - System X**

**Required System X Reliability Value Property – required reliability value owned by System X**

***Value Properties are key to Parametric Diagrams!***

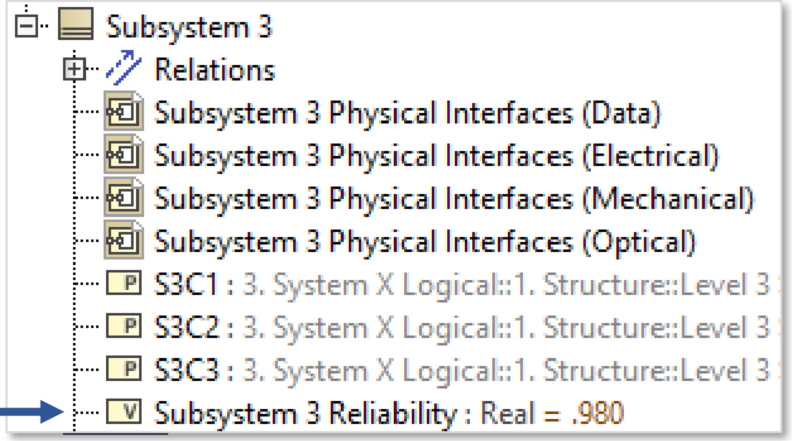
# System X Structure




**A straightforward hierarchy of subsystems and components**

**1..* Multiplicity – there is at least 1 Subsystem 3 but there could be several**

**Values that represent non-functional constraints are represented using Value Properties**




**Anatomy of a Value Property:**

- **Name – Subsystem 3 Reliability**
- **Type – Real**
- **Value - .980**

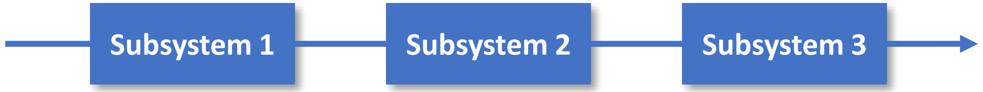
# Reliability Calculations

**We're going to calculate the reliability of System X**

- Based on the reliability of its subsystems
- Simple series and parallel system configurations
- Compare calculated reliability against the Required System Reliability

***We can mechanize all these calculations, and tie them directly to System X elements, within the model***

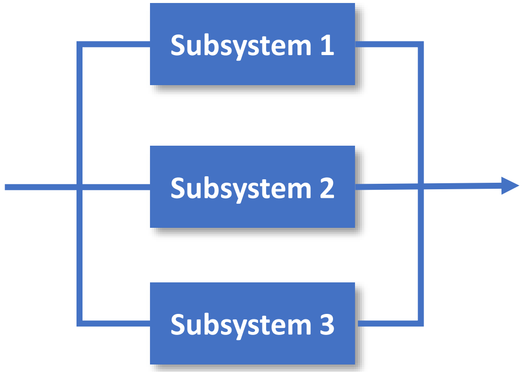
Series: System 'success' depends on the success of every individual component




System X Series Reliability
= Subsystem 1 * Subsystem 2 * Subsystem 3

System X Reliability > Required System Reliability ?

Parallel: If not all components fail, the entire system works




System X Parallel Reliability
= 1 – (1 - Subsystem 1) * (1 - Subsystem 2) * (1 - Subsystem 3)

System X Parallel Reliability > Required System Reliability ?

# Mechanizing MBSE Parametric Analyses

INTUITIVE®

## 4 primary steps:

**1. Define the analysis context**

- Identify the model elements that comprise the analysis
- Define the equations and parameters
- Identify the related System X subsystems and their Value Properties

**2. Create the Parametric Diagram**

- Implement the analysis
- Establish associations between equation parameters and System X Value Properties

**3. Execute the simulation**

- Cameo Simulation Toolkit
- Execute the analysis equations
- Provide results

**4. Analyze and refine the results**

- View the results
- 'Change' values to refine results
- Perform 'what-if' scenarios

# Step 1: Establish the Analysis Context

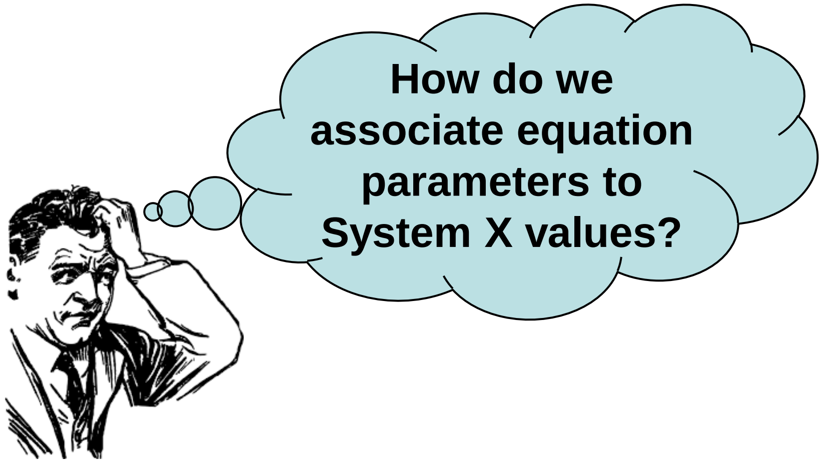
INTUITIVE®

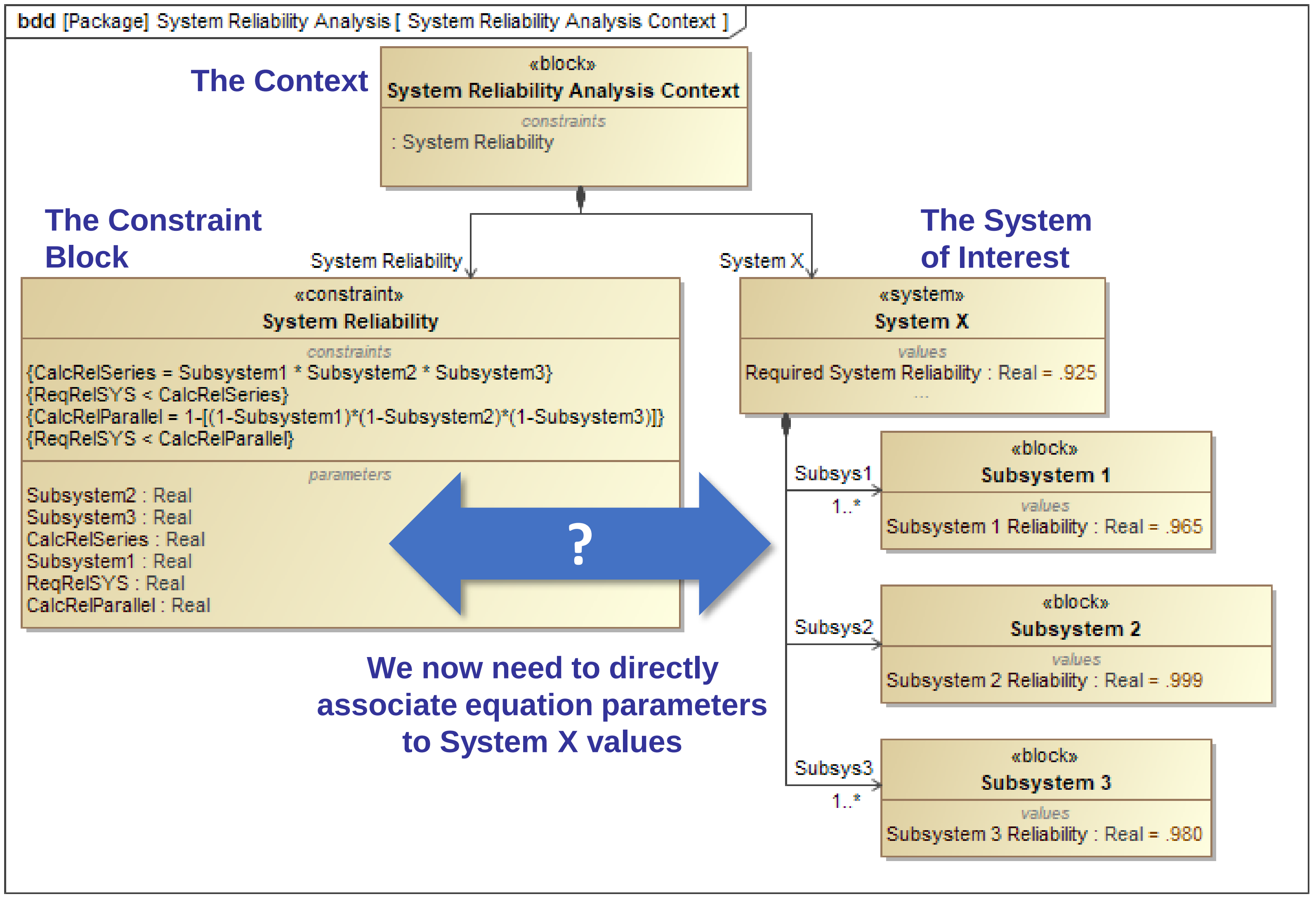
**Context diagram is represented as a Block Definition Diagram (BDD)**

- **A SysML Structure Diagram**

**The Context is comprised of:**

- **The System of Interest (System X)**
- **System X Subsystems**
  - **and their Reliability Value Properties**
- **Constraint Block**
  - **Constrains System X**
  - **Defines the Reliability equations and parameters**
  - **Defines Reliability equation parameters as Real**



# Step 2: Create the Parametric Diagram

An instance of the Constraint block

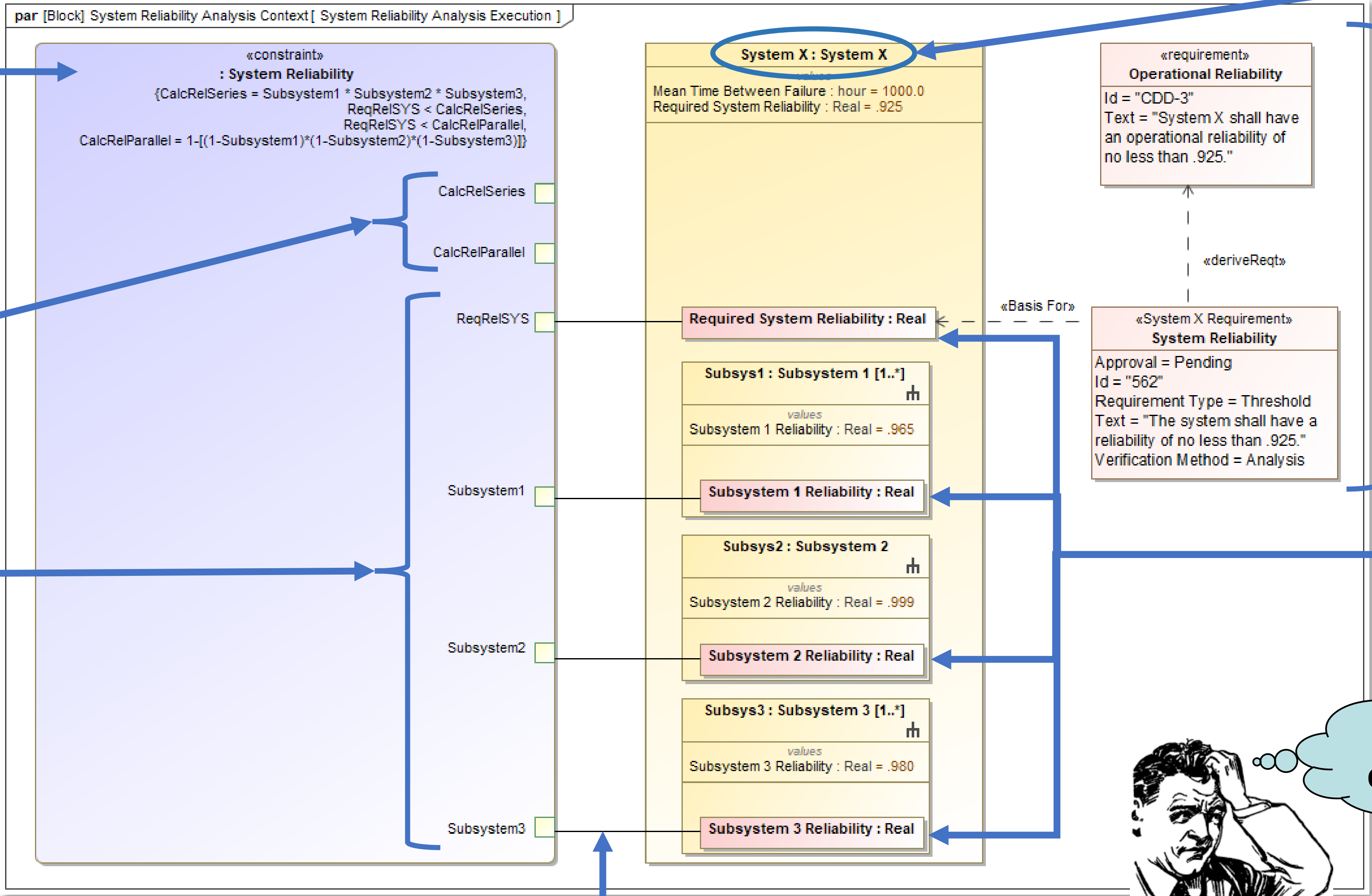
Calculated equation parameters

- Not bound to Value Properties

'Known' Equation parameters

- From bound Value Properties

Binding Equation Parameters to Value Properties




An instance of System X

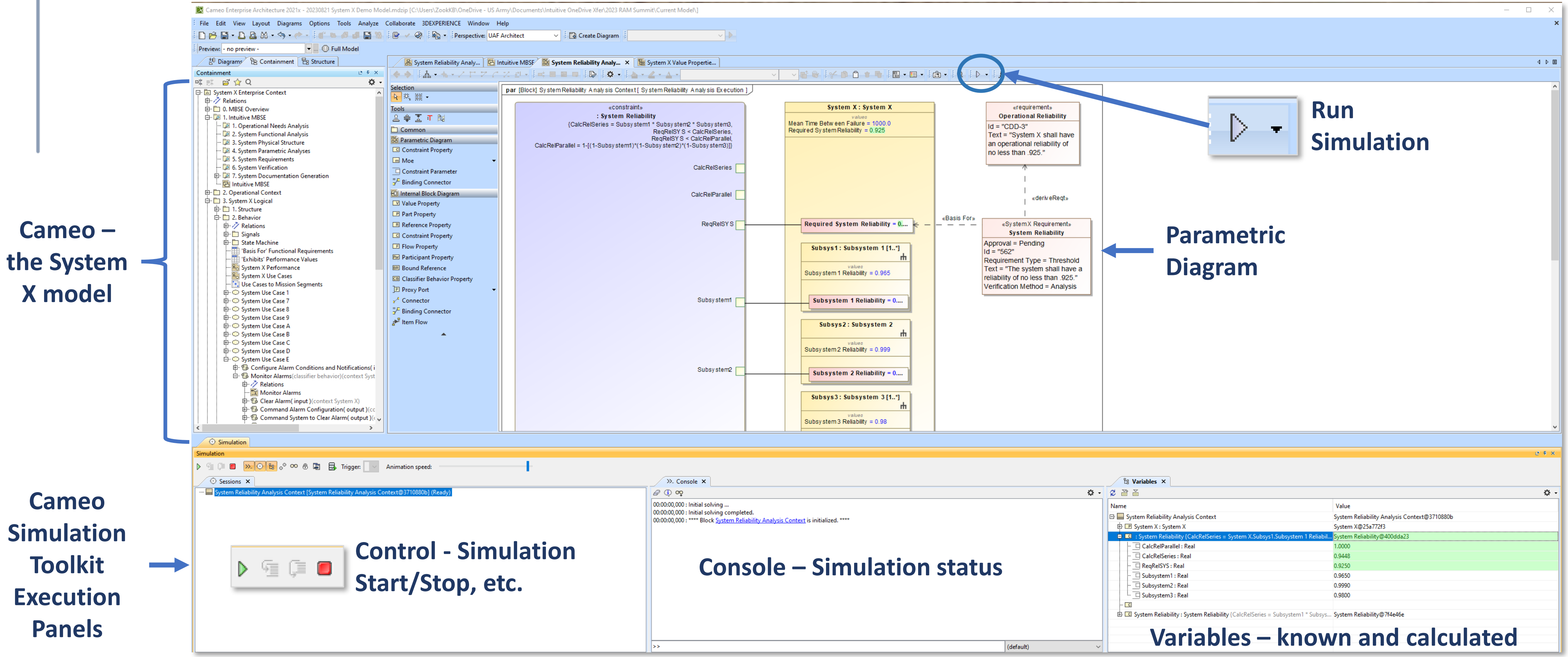
Requirement elements included for information

Value Properties

- System Required Reliability
- Subsystem Reliability

# Step 3: Execute the Simulation

INTUITIVE®



INTUITIVE®

# Step 4: Analyze and Refine the Results

Variables ✕

| Name | Value |
|---|---|
| System Reliability Analysis Context | System Reliability Analysis Context@3710880b |
| System X : System X | System X@25a772f3 |
| : System Reliability {CalcRelSeries = System X.Subsys1.Subsystem 1 Reliabil... | System Reliability@400dda23 |
| CalcRelParallel : Real | 1.0000 |
| CalcRelSeries : Real | 0.9448 |
| ReqRelSYS : Real | 0.9250 |
| Subsystem1 : Real | 0.9650 |
| Subsystem2 : Real | 0.9990 |
| Subsystem3 : Real | 0.9800 |
| | |
| System Reliability : System Reliability {CalcRelSeries = Subsystem1 * Subsys... | System Reliability@7f4e46e |

The simulation has completed execution and…

**Green:** Both parallel and series calculated values exceed System X required reliability!

Variables ✕

| Name | Value |
|---|---|
| System Reliability Analysis Context | System Reliability Analysis Context@3710880b |
| System X : System X | System X@25a772f3 |
| : System Reliability {CalcRelSeries = System X.Subsys1.Subsystem 1 Reliabil... | System Reliability@400dda23 |
| CalcRelParallel : Real | 0.9111 |
| CalcRelSeries : Real | 0.1637 |
| ReqRelSYS : Real | 0.9250 |
| Subsystem1 : Real | 0.4800 |
| Subsystem2 : Real | 0.5500 |
| Subsystem3 : Real | 0.6200 |
| | |
| System Reliability : System Reliability {CalcRelSeries = Subsystem1 * Subsys... | System Reliability@7f4e46e |

We can actually change values directly within the Variables pane

- further 'what-if' analyses, so…

Tweaking the Subsystem reliability values…

**Red:** Neither calculated value exceeds System X required reliability

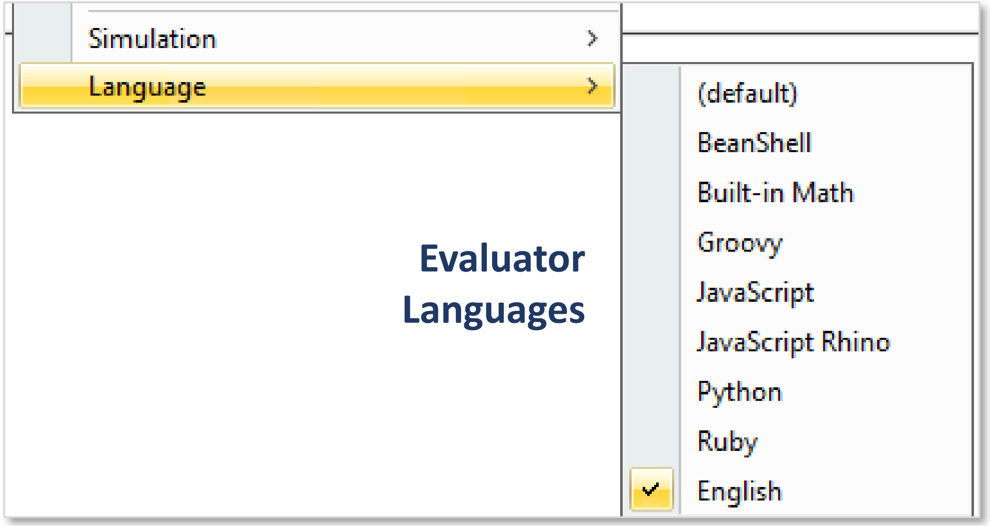
# External Evaluator Capability

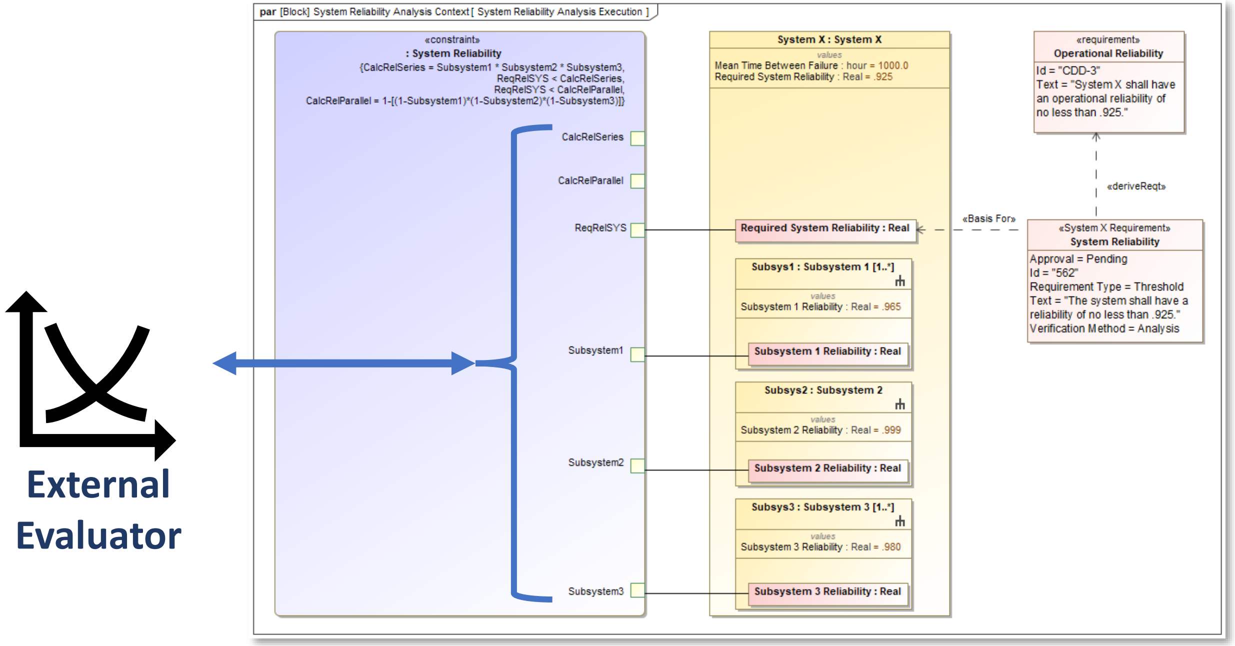
Cameo math functions are fairly robust; however, ‘external evaluator’ capability exists

- More custom/intense analyses
- Several external evaluator platforms supported
    - Matlab
    - MapleTM
    - Mathematica
    - more
- Custom coding
    - Python
    - Ruby
    - more

Fundamental premise is ability to ‘expose’ constraint parameters to external evaluator platforms

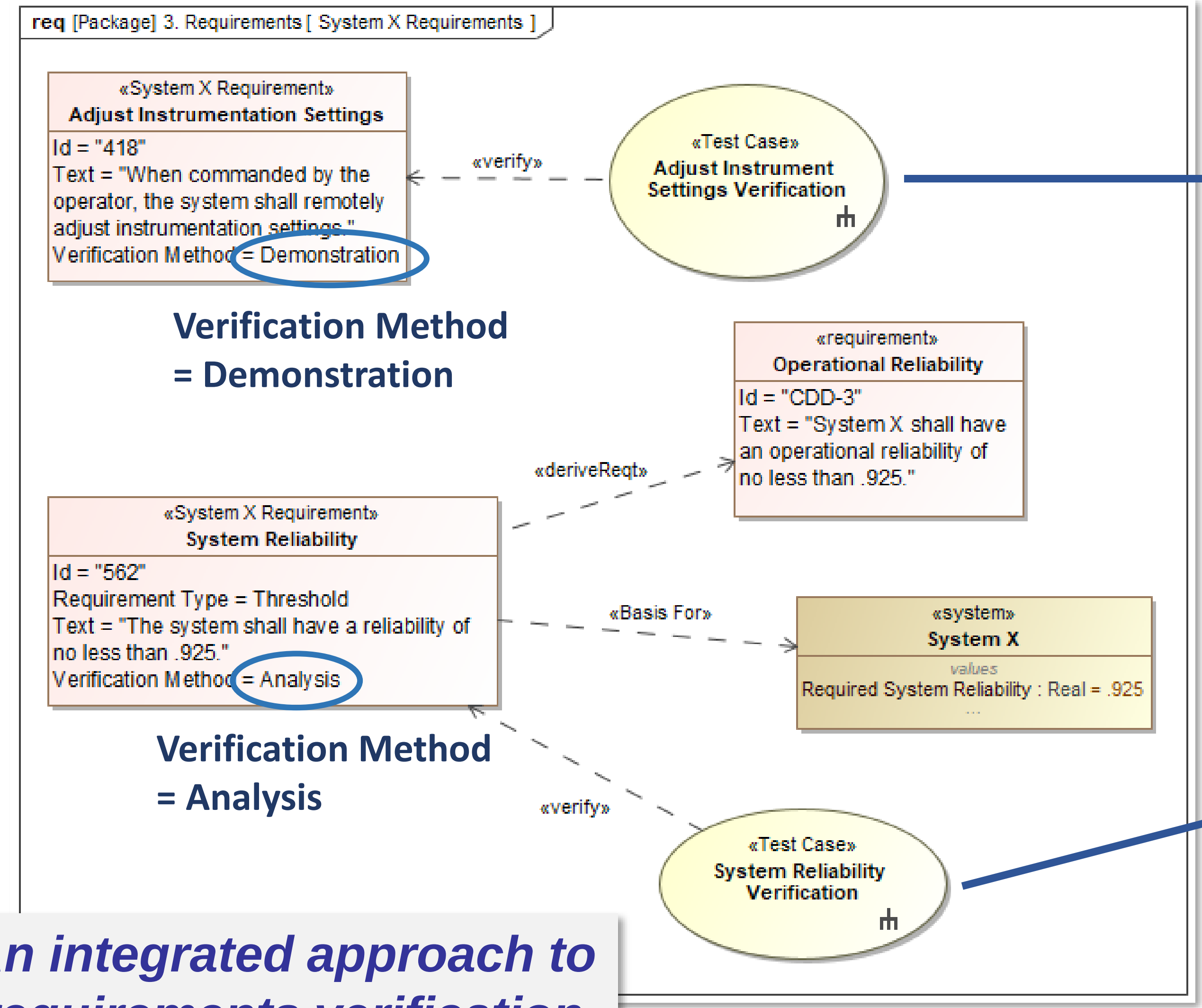
- Results come back to the model
- Key is still associating analysis parameters to System of Interest Value Properties



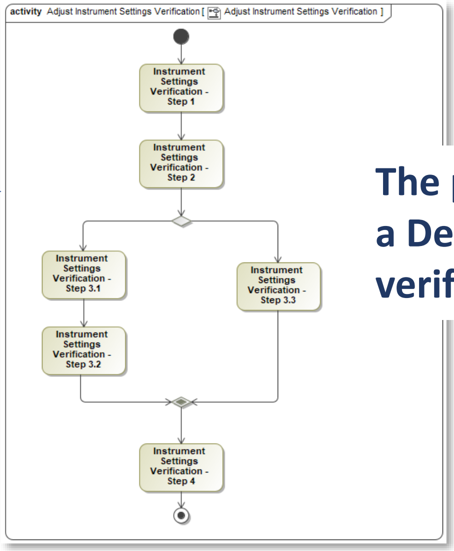
# Reliability Requirement Verification

INTUITIVE®

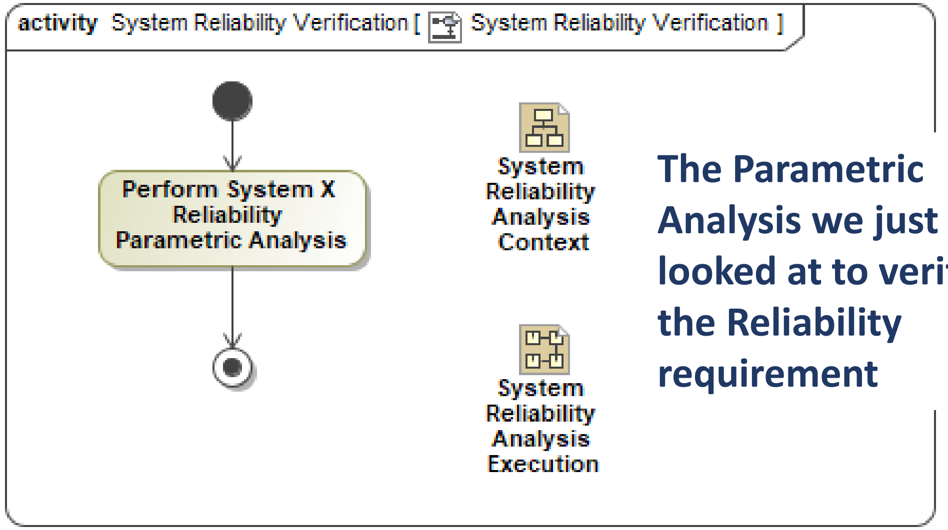



**Verification Method = Demonstration**

**Verification Method = Analysis**




**The process flow for a Demonstration verification method**

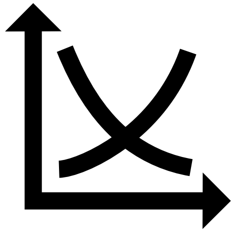


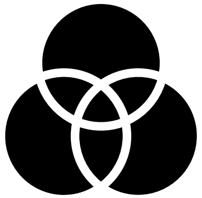

**The Parametric Analysis we just looked at to verify the Reliability requirement**

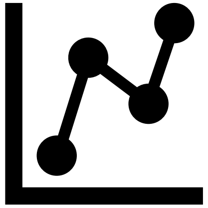
***An integrated approach to requirements verification***

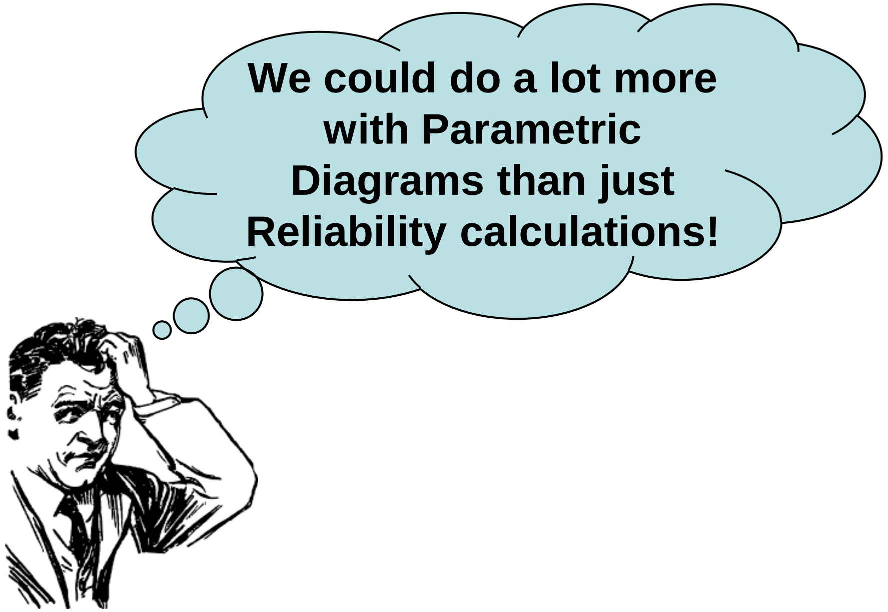
# Other Potential Applications

- Other RAM calculations
  - Availability
  - MTBF
  - etc.
- System physical aspects
  - Aircraft weight rollup
  - Aircraft center of gravity
  - etc.
- Risk calculations
- Anything that can be represented as a value, owned by the system, can be used in a Parametric Diagram analysis!



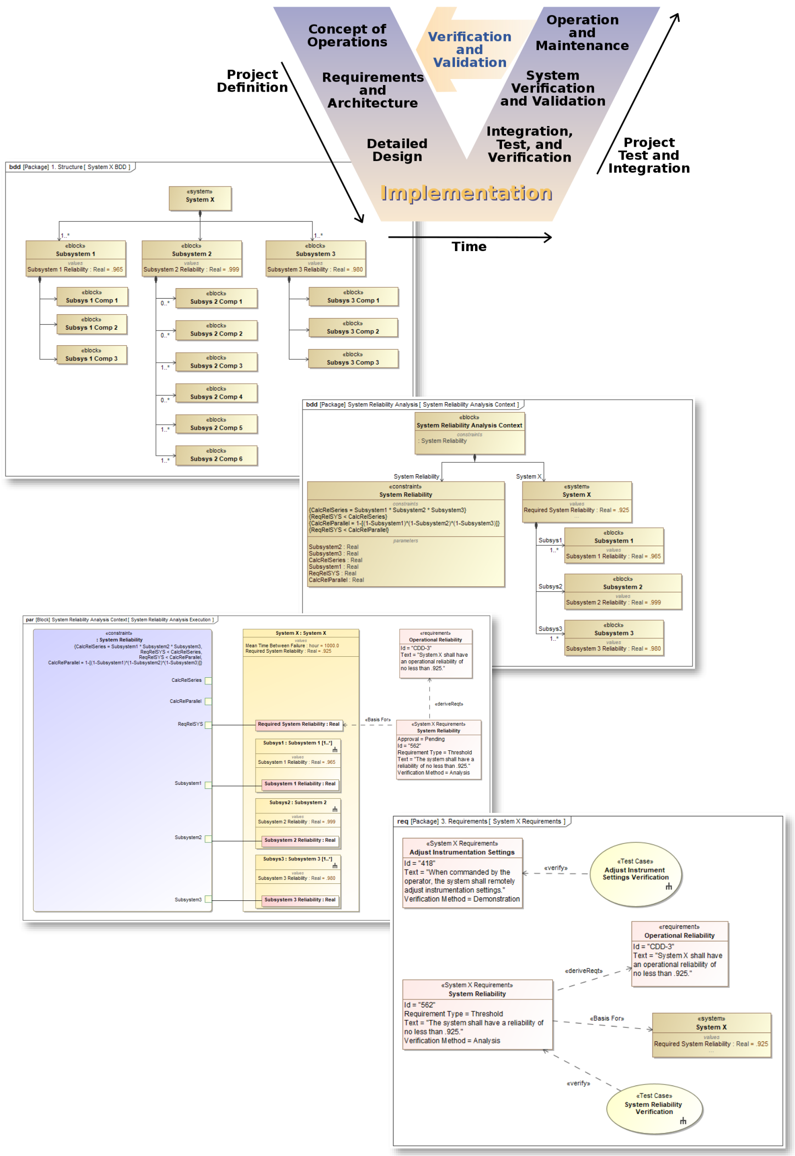






# Summary

- MBSE directly supports the Systems Engineering process
- Parametric Diagrams provide a robust analysis capability within the MBSE model environment
- RAM activities can leverage a model’s Parametric Diagram capability for an integrated model-centric approach



# Wrap Up and Discussion

INTUITIVE®

- Questions?
- Comments?

- Contact: Keith Zook
  Keith.Zook@IRTC-HQ.com