The Benefits of a Requirements Verification Architecture Model

… or How to Tie the Left and Right Sides of the Life Cycle V-Model Together

Charley Patton, CSEP
Systems Engineer

September 21, 2022

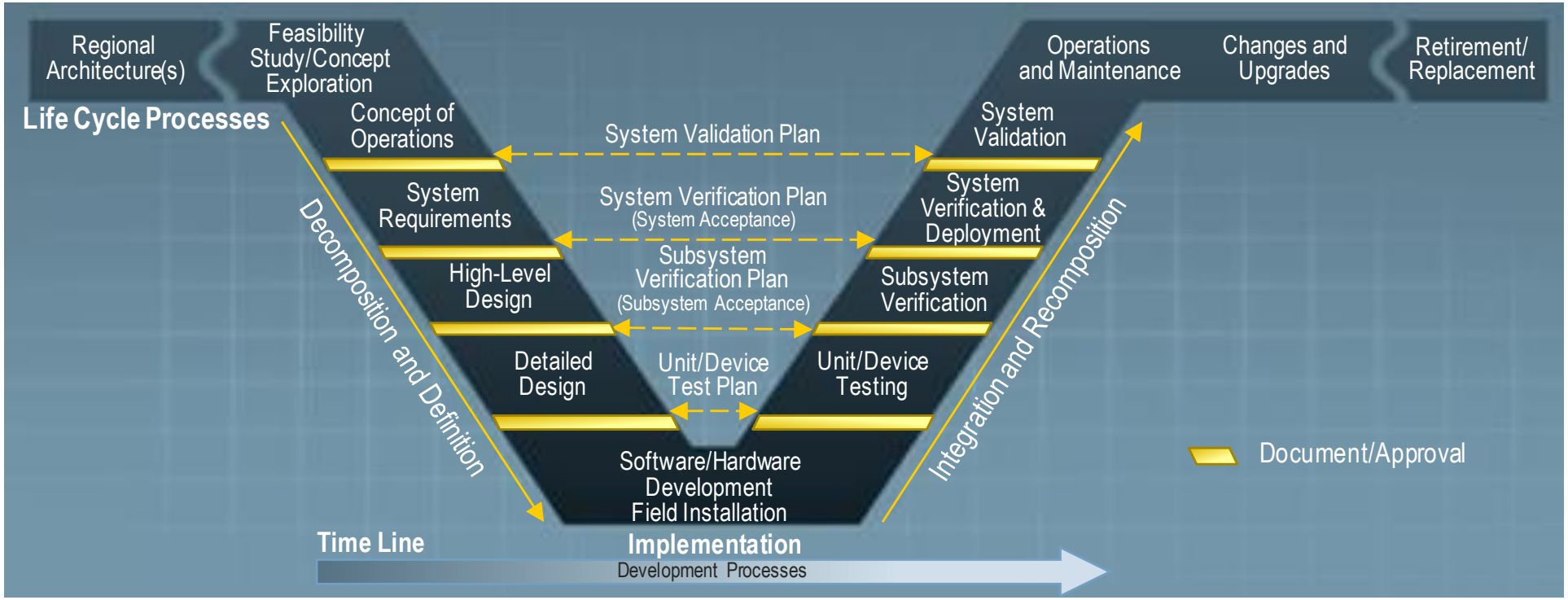
What You're About to See

- Why you want a Requirements Verification Architecture Model (RVAM)
- What makes up the RVAM
- How to create the RVAM
- Q&A

WHY You Want the RVAM

WHY | WHAT | HOW

- Horizontal Integration: Increases coordination across development life cycle stages earlier
- Vertical Integration: Reinforces need for top-down requirements and architecture development
  - Typical Req ←→ Test traces may be insufficient
  - Increases collaboration among SE, I&T, and Development teams
- Model-based visual artifacts



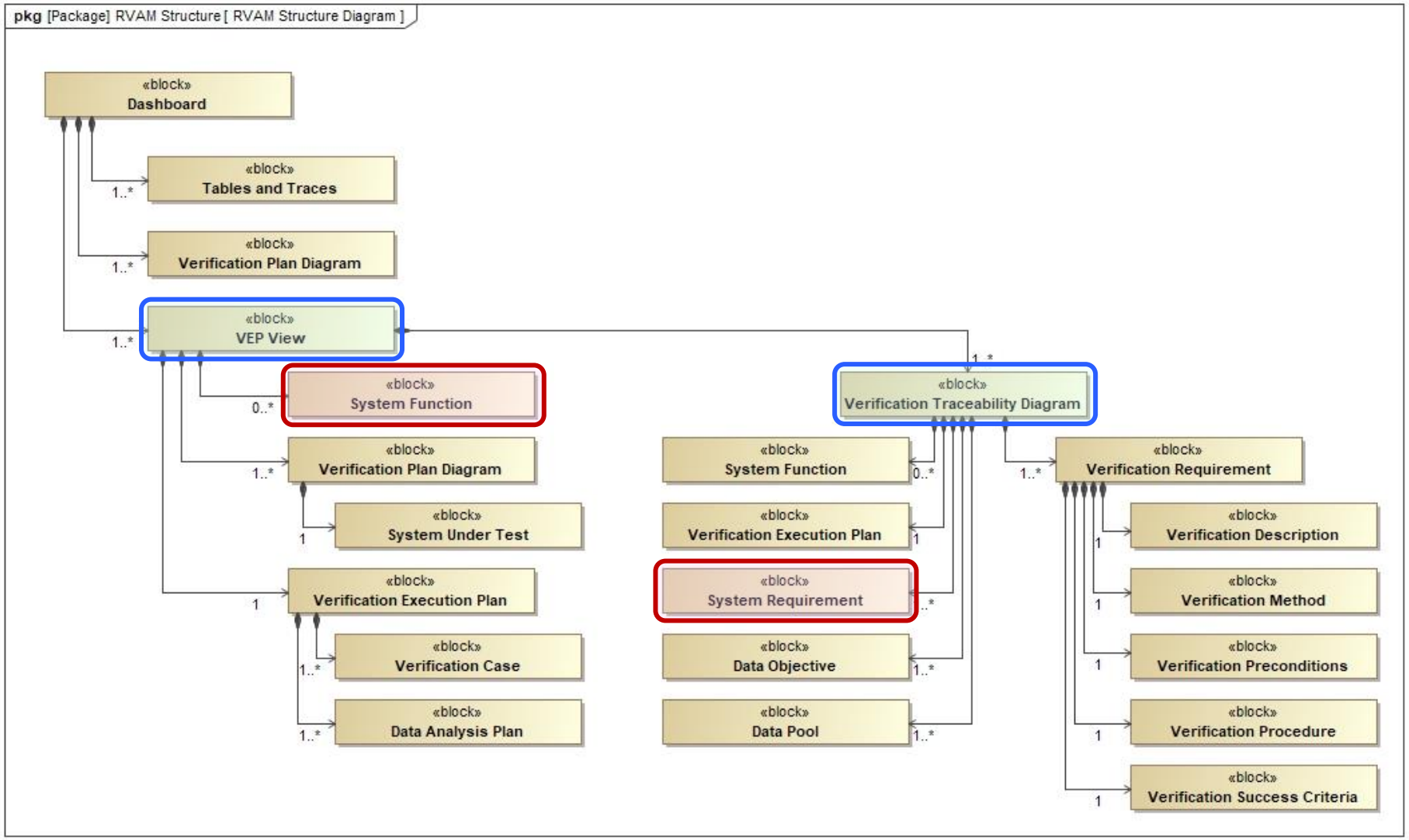
Communicate, Coordinate, Collaborate Over the Development Life Cycle

WHAT the RVAM Parts Are

Structure

WHY | WHAT | HOW

- Model elements are:
  - Either inputs from system model
  - or generated specifically for RVAM

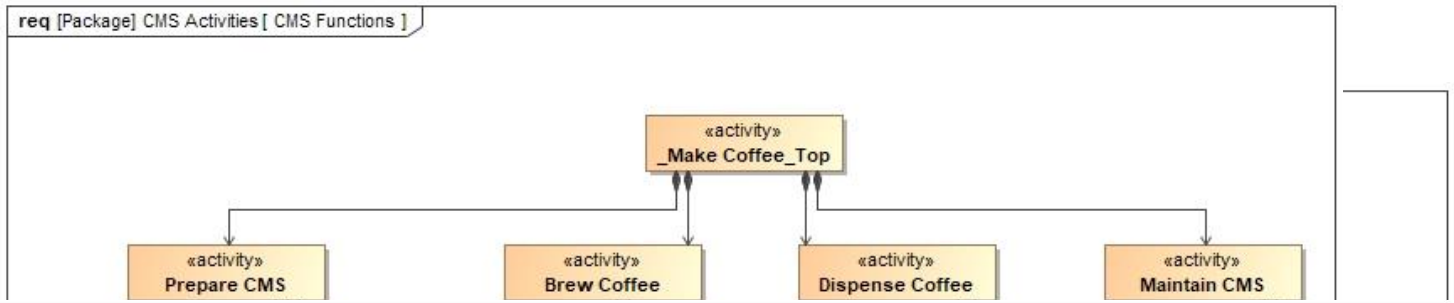


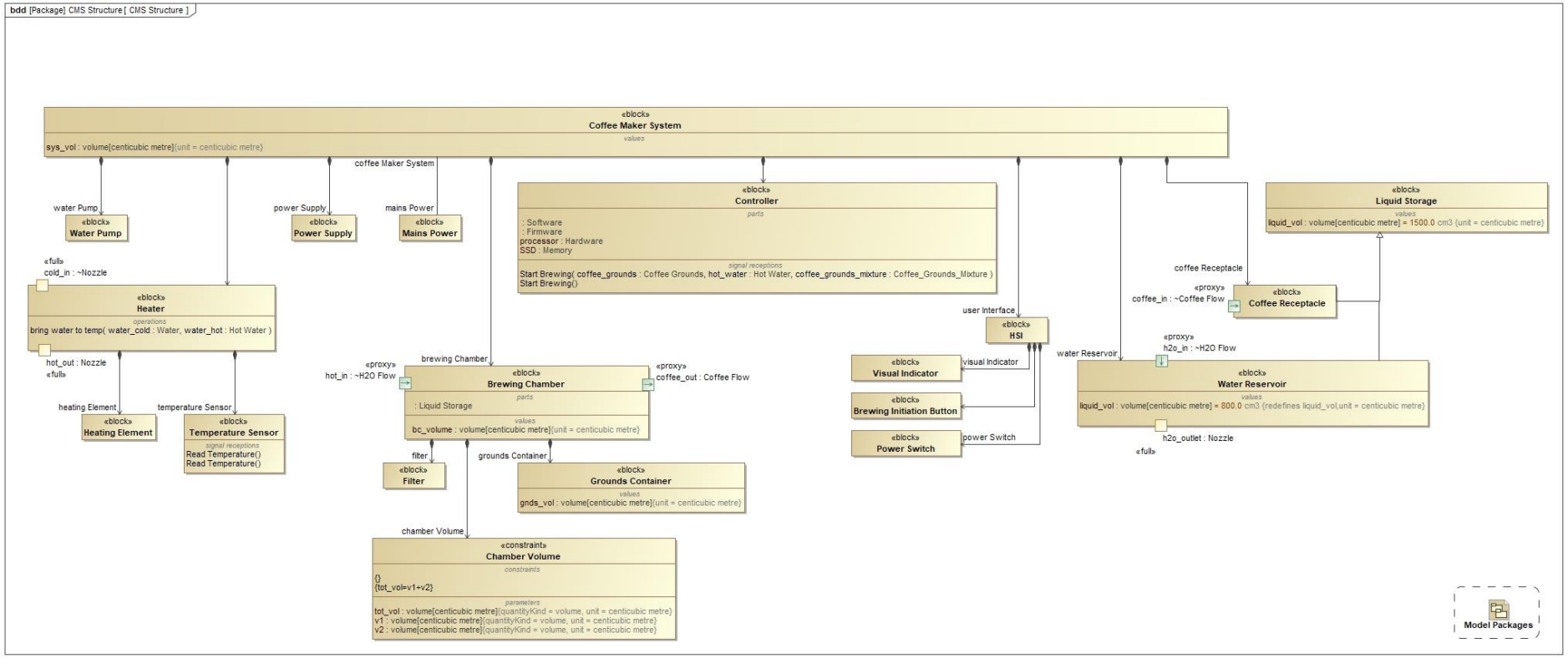
Comprehensive top-to-bottom description of verification efforts

Input Artifacts: from Req and Arch development

WHY | WHAT | HOW

- System Functions
- System Requirements
- System Architecture
  - Physical Entity Decomposition

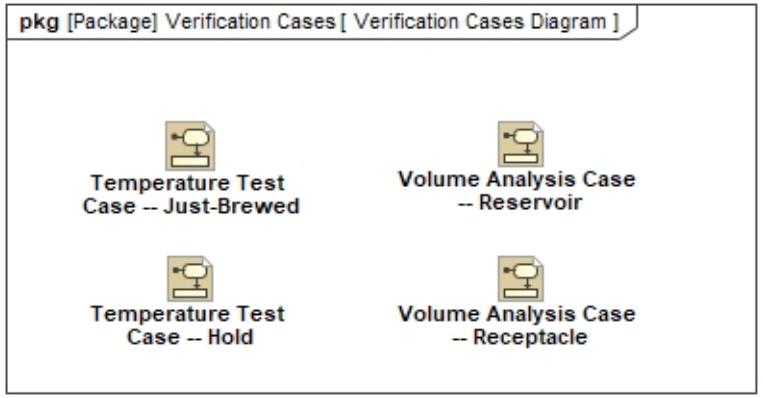


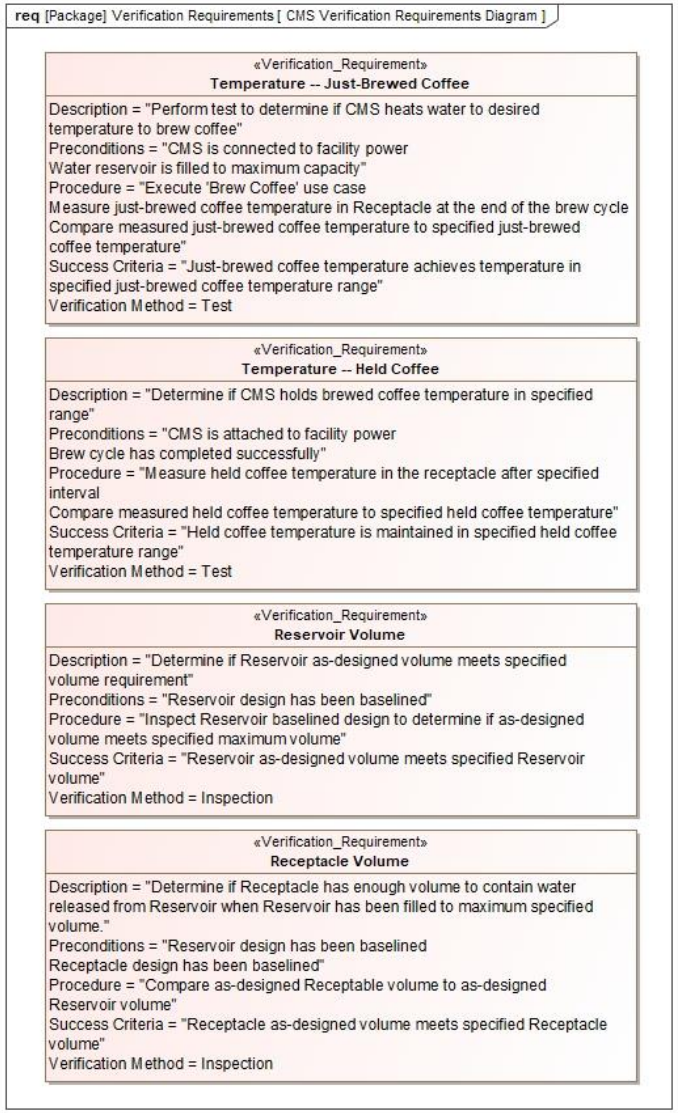


Artifacts typically generated by Systems Engineering team

Input Artifacts: from Verification development

- Verification Requirements (VRs)
  - Description/Objective
  - Method (e.g., Test, Demonstration, Analysis, Inspection/Examination)
  - Preconditions/Environment
  - Procedure
  - Success Criteria
- Verification Cases (VCs)
  - Verification environment
  - Verification inputs and outputs
  - Data collection method
- Verification Plans (VPs)
  - Verification environment
  - Verification configuration
  - Resources
  - Exercised Verification Cases



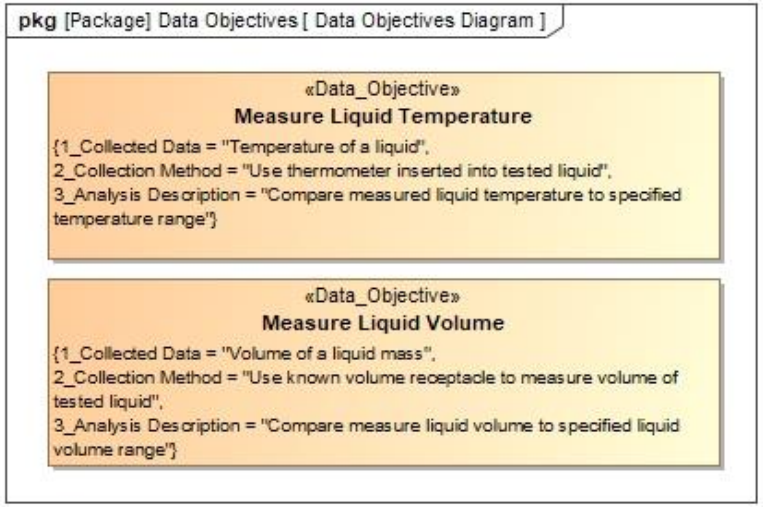


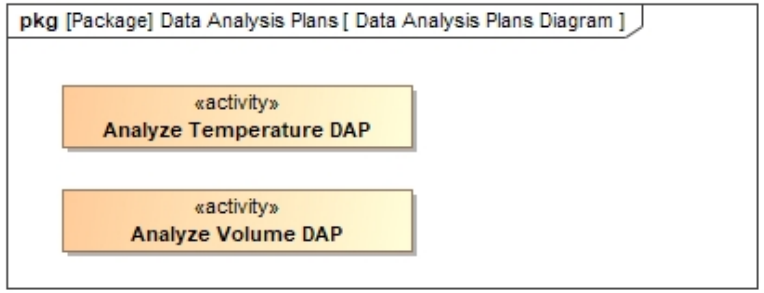
Artifacts typically generated by Integration and Test team

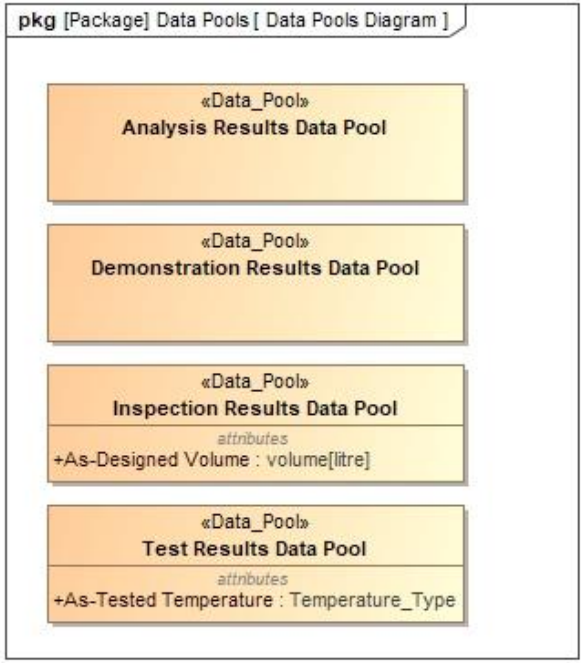
Input Artifacts: from Verification development

WHY | WHAT | HOW

- Data Objectives (DOs)
  - What data must be collected, the collection method, and data analysis description
- Data Analysis Plans (DAPs)
  - How data will be analyzed to determine verification success of the system requirements
- Data Pools (DPs)
  - Where the data inputs and outputs will be managed (e.g., DOORS for requirements, CM tool for plans)







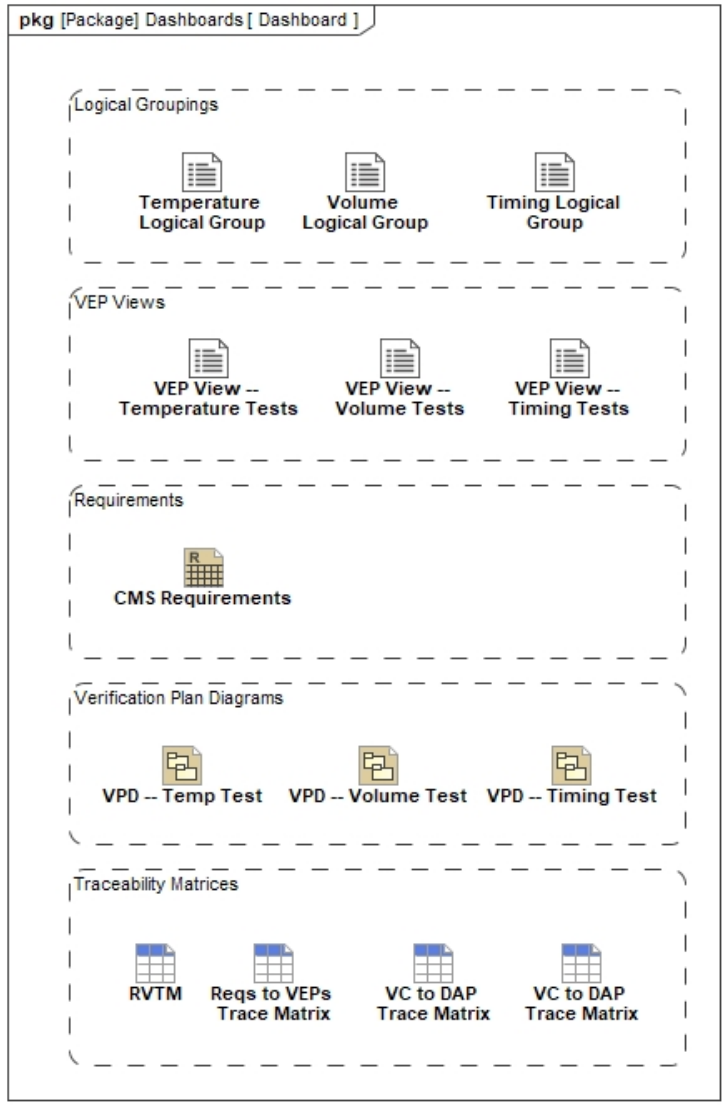
Artifacts typically generated by Integration and Test team

Generated Artifacts

WHY | WHAT | HOW

- Dashboards
  - Content diagrams
- Tables and Views
  - Requirements Traceability Matrices (e.g., RVTM)
  - Reqs ←→ VEP
  - VEP ←→ VCs
  - VCs ←→ DAPs
  - etc.

#	Id	Name	Text	Verify Method	Refined By	Satisfied By	Derived From	Verified By
1	1	1 Heat Water	The CMS shall heat water from room temperature to 185F within 1 minute.	Test	Heat Water	Heater		Measure Temperature Rise VEP -- Temp Test
2	2	2 Reservoir Volume	The CMS shall contain 6 cups of water.				1 Heat Water	
3	3	3 Remove Grounds	The CMS shall remove grounds from brewed coffee.	Demonstration		Filter		

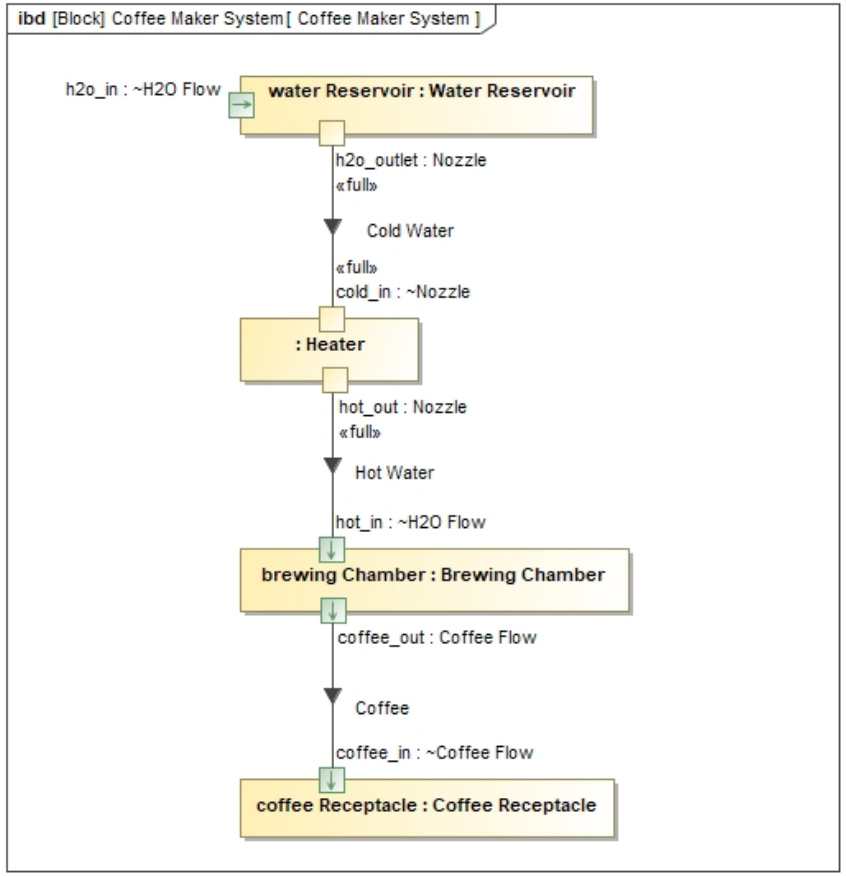


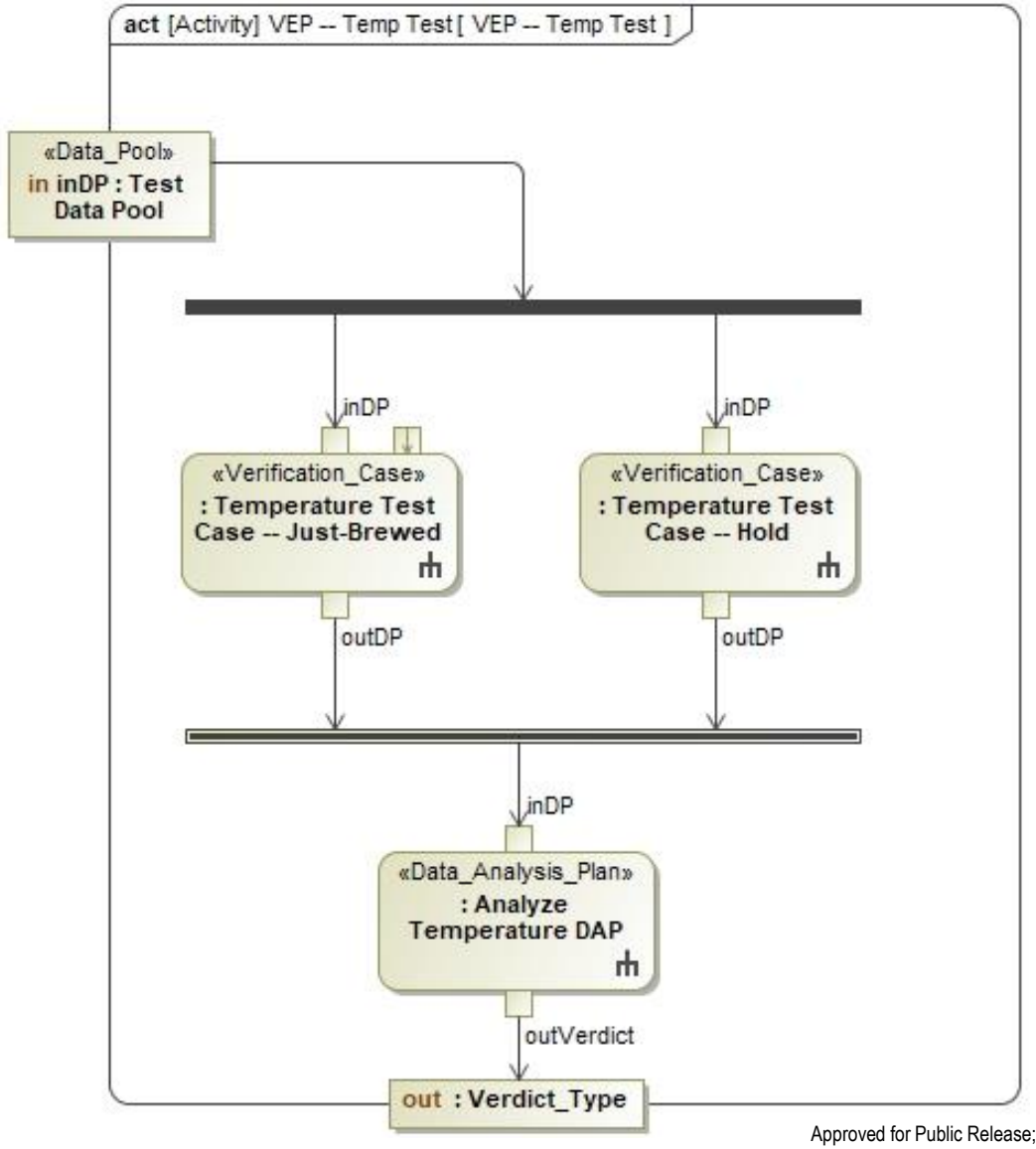
Dashboards help navigate the model

Generated Artifacts

WHY | WHAT | HOW

- Verification Execution Plan (VEP)
  - Verification and Analysis activity flow
- System Under Test (SUT)
  - Tested system components shown in test configuration

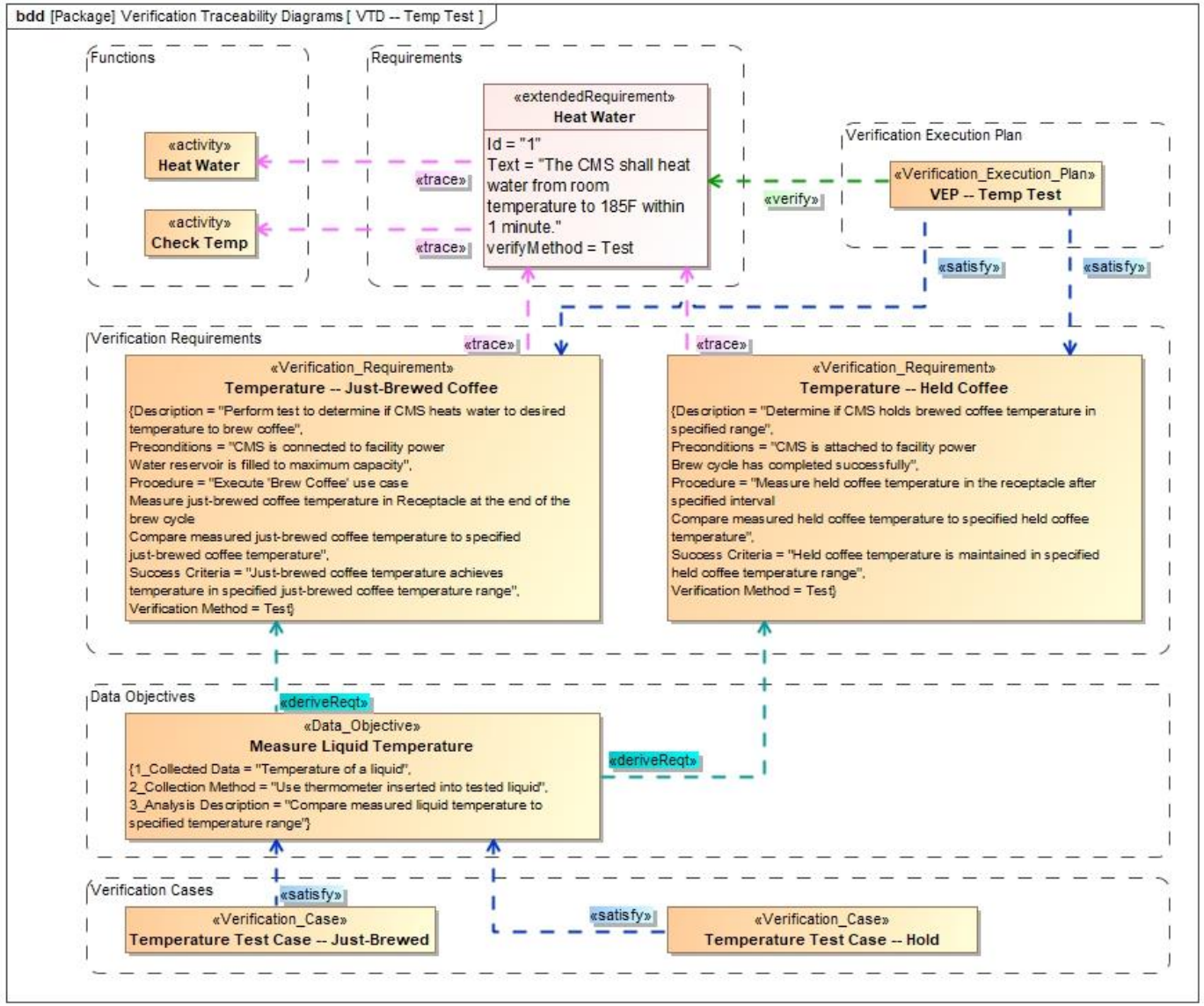




Generated Artifacts

WHY | WHAT | HOW

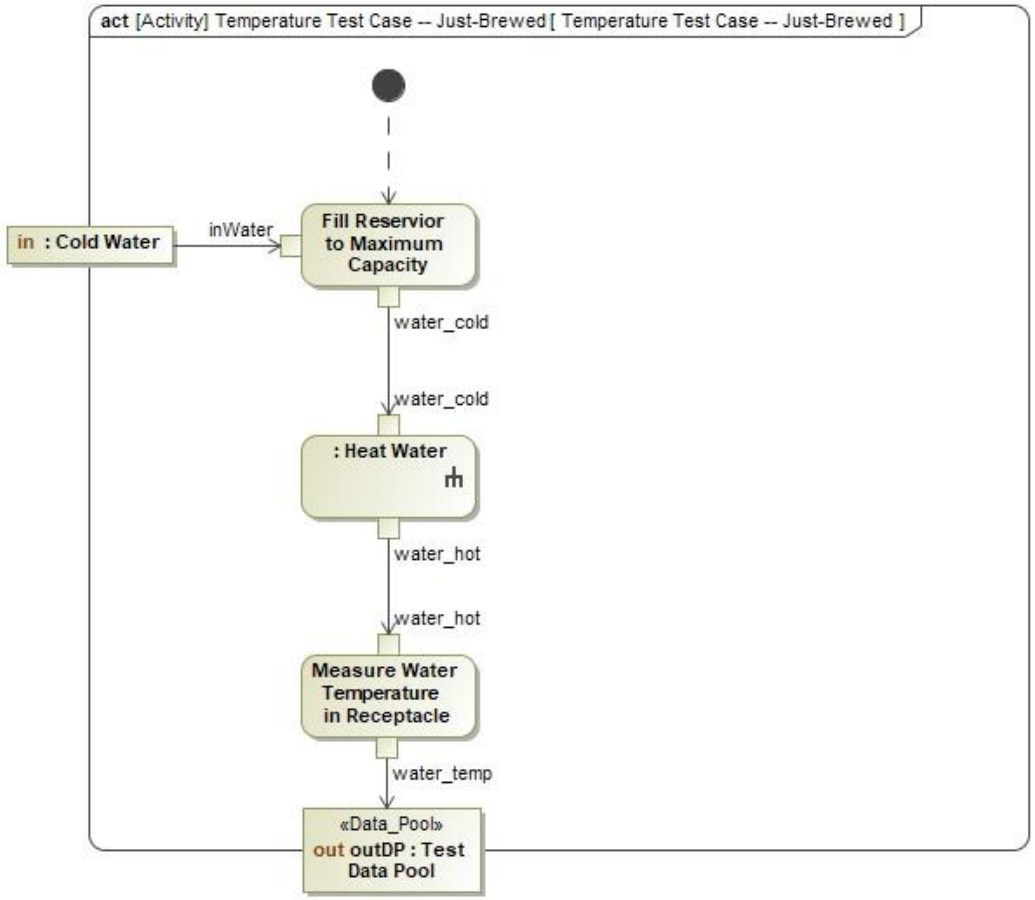
- Verification Traceability Diagram (VTD)
  - Traces
    - System Reqs <<trace>> Functions
    - VEP <<verify>> System Reqs
    - VEP <<satisfy>> Verification Reqs
    - Verification Reqs <<trace>> System Reqs
    - Data Objectives <<derive>> Verification Reqs
    - Verification Cases <<satisfy>> Data Objectives

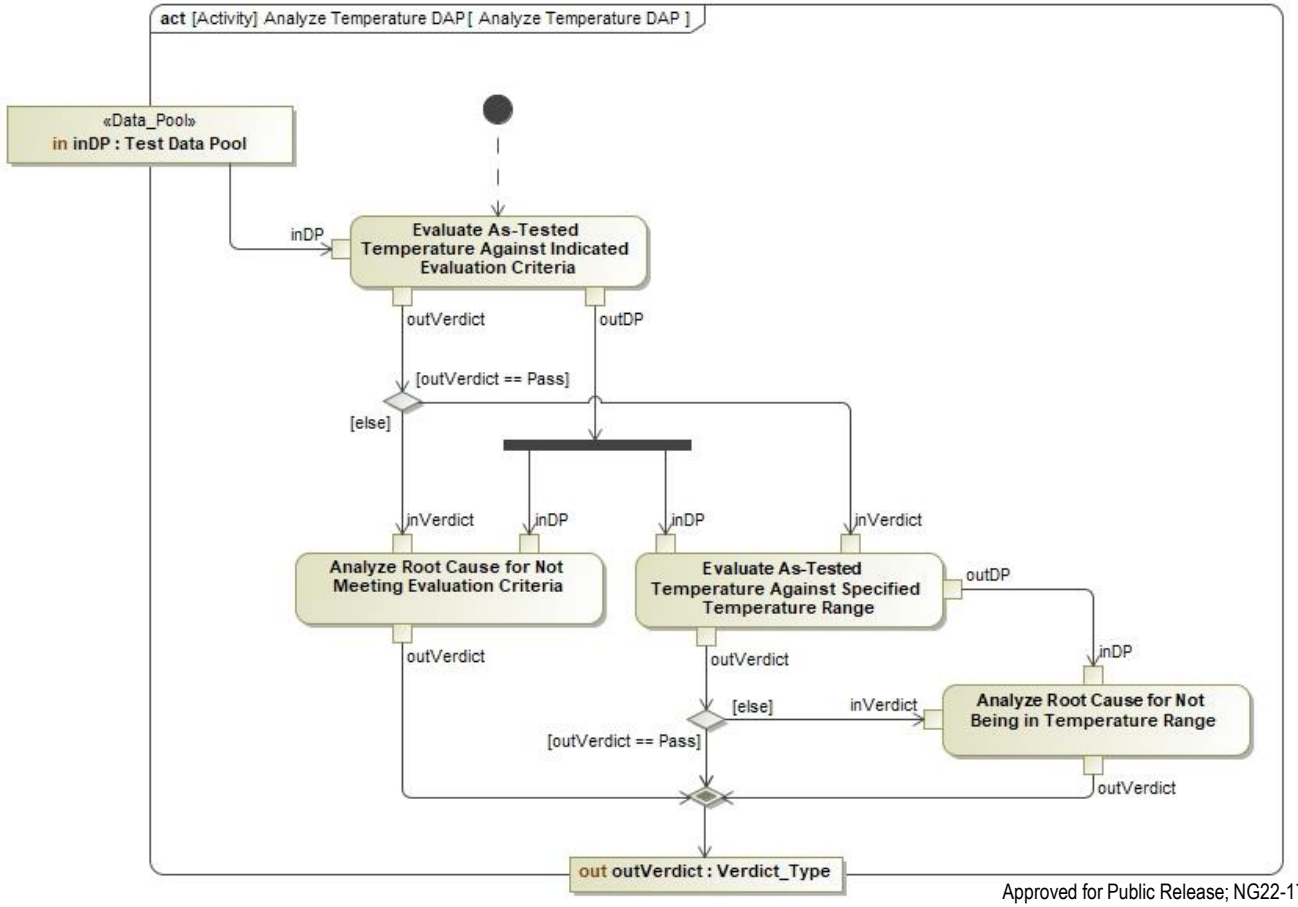


Generated Artifacts

WHY | WHAT | HOW

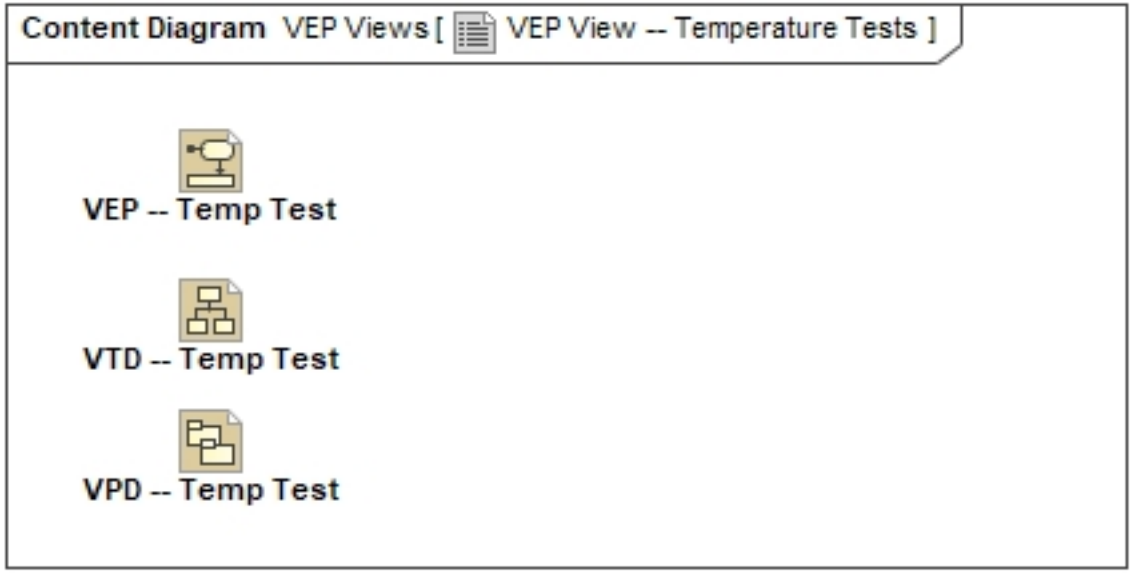
- Verification Case activity diagrams
- Data Analysis Plan activity diagrams
  - Activity diagram showing how data will be analyzed to determine verification success of the system requirements





Generated Artifacts

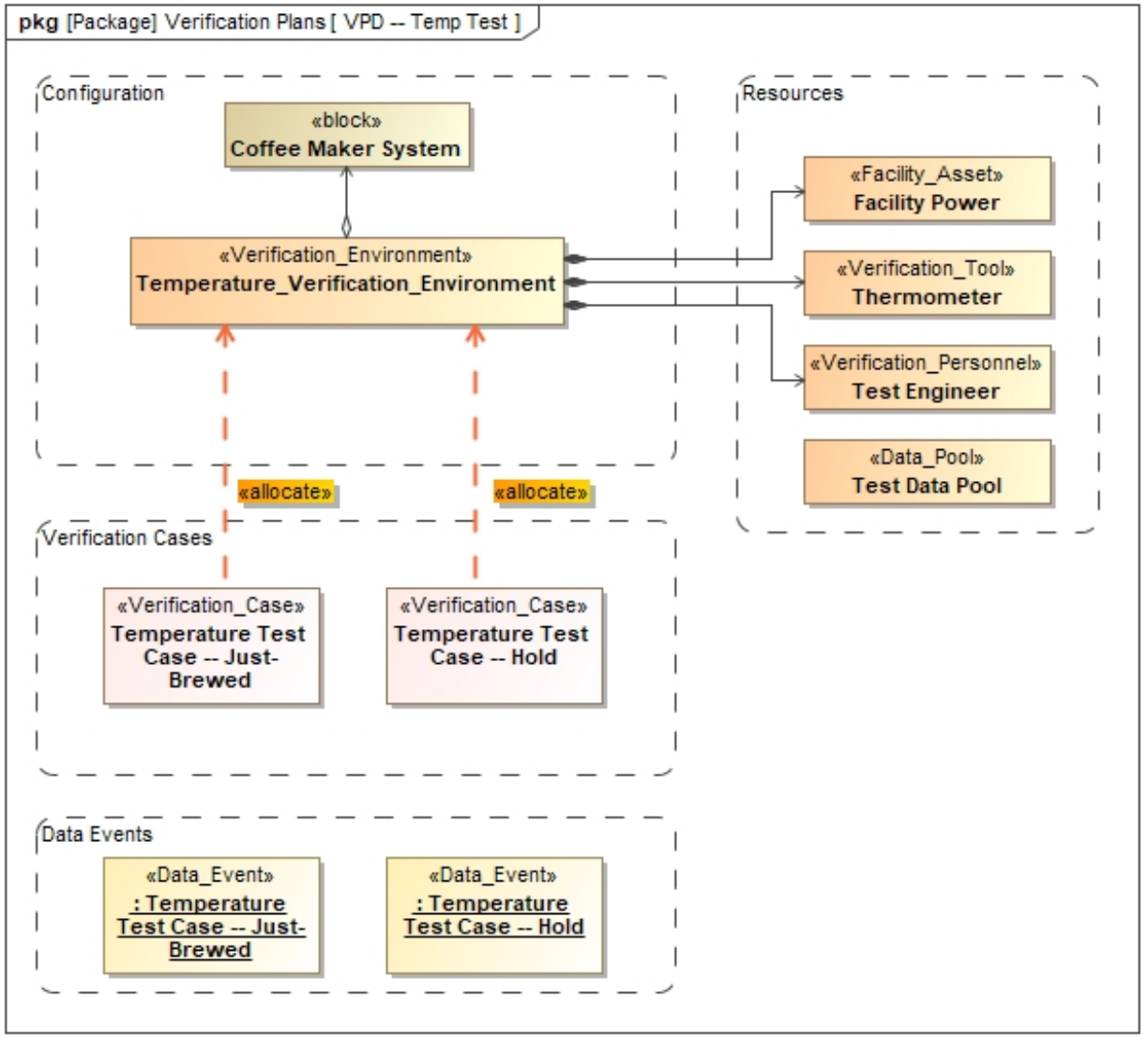
- VEP View
  - Dashboard (content diagram) identifying VEP, VTD, VPDs
  - 1 per logical group of requirements



Generated Artifacts

WHY | WHAT | HOW

- Verification Plan Diagram (VPD)
  - Dashboard (content diagram) identifying
    - Verification Plan
    - Verification Environment
      - Including the System Under Test (SUT)
    - Verification Cases
    - Data Events
    - Verification Resources

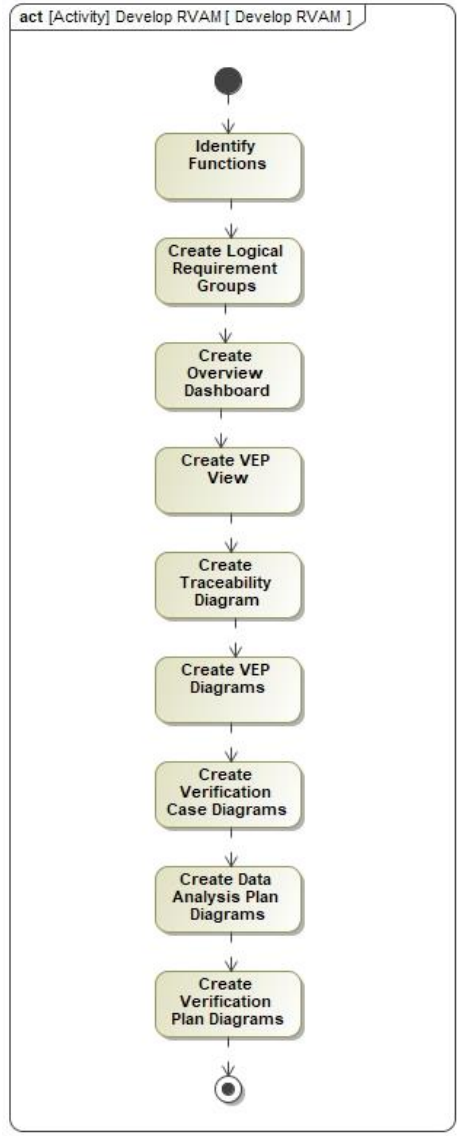


HOW to Create the RVAM

HOW to Create the RVAM

1. Identify relevant system functions
2. Determine logical grouping of system requirements
   - For example: (1) Temperature reqs, (2) Volume reqs, (3) Timing reqs, (4) etc.
3. Create summary overview dashboard
4. Create VEP View (1 per logical req group)
   - Identifies VEP, Traceability Diagrams, VPDs
5. Create the Traceability Diagram
   - Consider creating multiple so that each contains no more than 10 requirements, for readability
6. Create the VEP (1 per logical req group)
7. Create Verification Case activity diagrams
8. Create Data Analysis Plan activity diagrams
9. Create Verification Plan Diagram

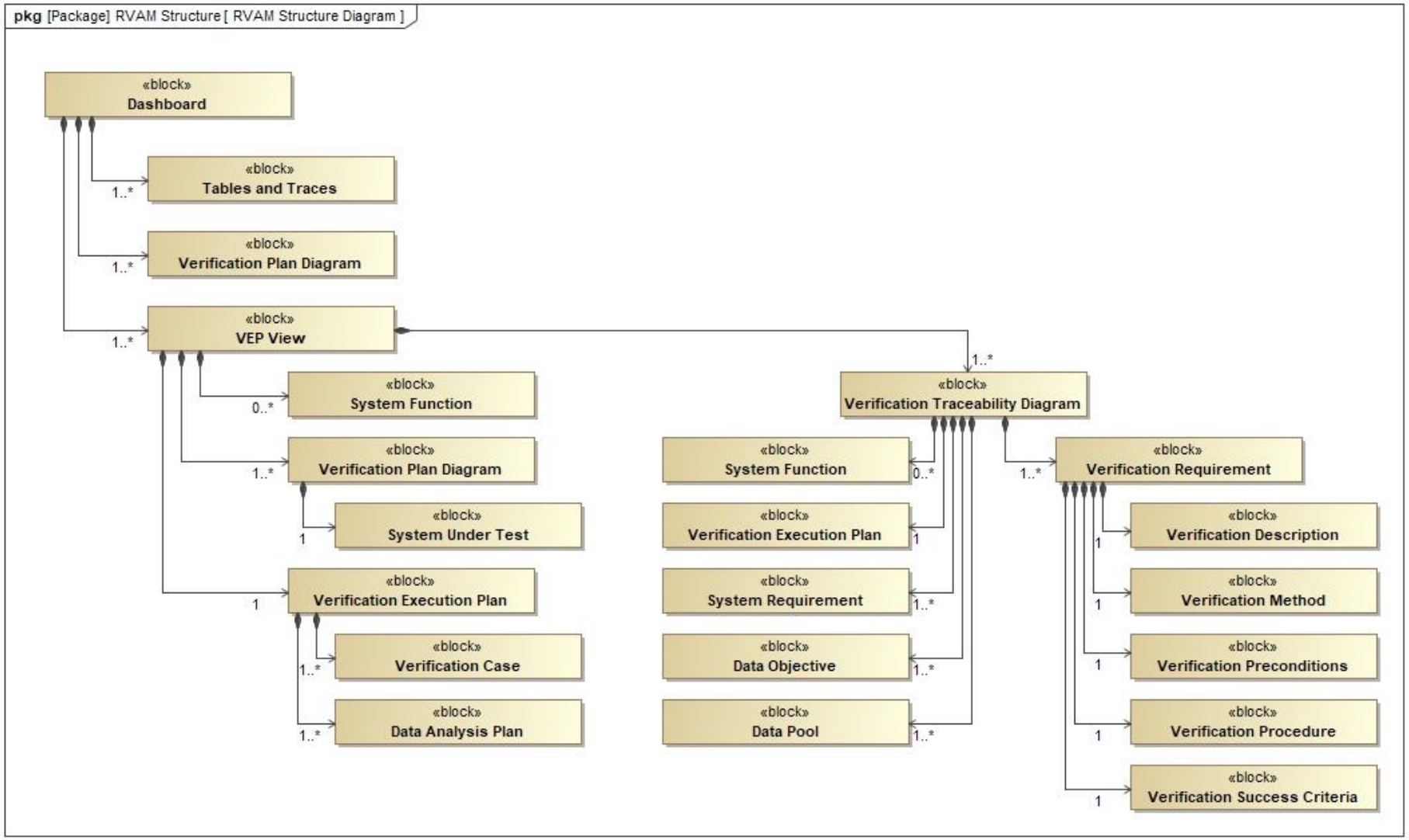
- In General:
  - Create placeholder diagrams
  - Create lower-level objects, then populate higher-level objects
  - Iterate, especially over the dashboards



Division of Labor

WHY | WHAT | HOW

- Systems Engineering
  - System Functions
  - System Requirements
  - System Architecture
  - Top part of VTD
- Integration and Test
  - Verification Requirements
  - Verification Descriptions, Plans, Procedures
  - Data Objectives and Analysis Plans
  - Bottom part of VTD



www.incose.org