Raytheon

Pulling the Digital Thread with Model Based Systems Engineering

10/25/2017

Chris Finlay
finlayc@raytheon.com
401.842.2691

Stacy Gottesman
smgottesman@raytheon.com
520.794.8474

Julie DeMeester
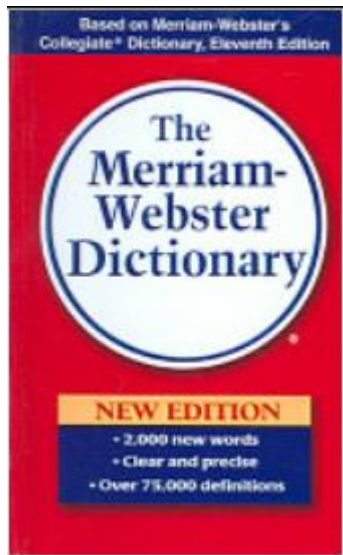*julied@raytheon.com*
978.858.4759

Agenda

- MBE Vision
- Digital Thread Process
- Creating the Systems Digital Thread
- Pulling the Digital Thread through SW Development
- Pulling the Digital Thread through HW Development
- Benefits
- Lessons Learned

First... Some definitions

Digital Thread vs. Digital Twin

The ***digital thread*** refers to a **collaborative engineering framework** that digitally connects data flow and data views of a system throughout its lifecycle across traditionally "siloed" engineering functions.

The ***digital twin*** refers to a physics-based set of digital models representing a physical system, its surrounding environment and real time data feeds. The digital twin represents each unique as-built system instance and operational and environmental data unique to that specific serial number it represents.

Based on Merriam-Webster's
Collegiate® Dictionary, Eleventh Edition

The
Merriam-
Webster
Dictionary

NEW EDITION
• 2,000 new words
• Clear and precise
• Over 75,000 definitions

This Paper focuses on the Digital Thread
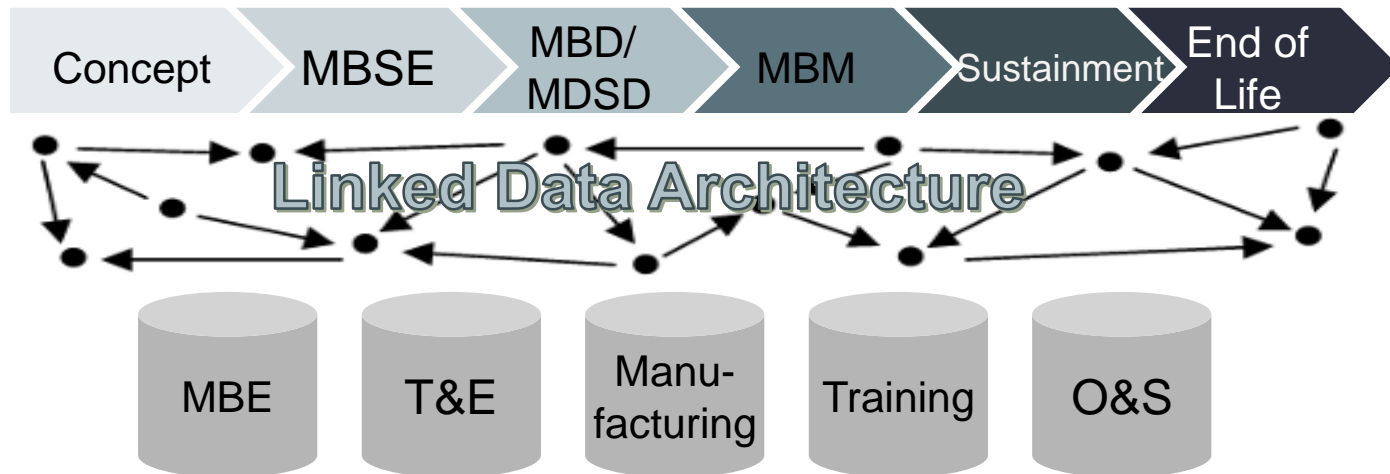
Model Based Engineering

*Engineering solutions composed as a set of models **linked** through an **information** infrastructure forming a **Digital Thread** that provides **authoritative source of truth***

- Our model data is then turned in to **actionable information** as part of the overall design processes
- Our models become the source of information for deliverable documents which are **produced automatically**
- **Design decisions** are then **linked** and **consistent** across the solution space
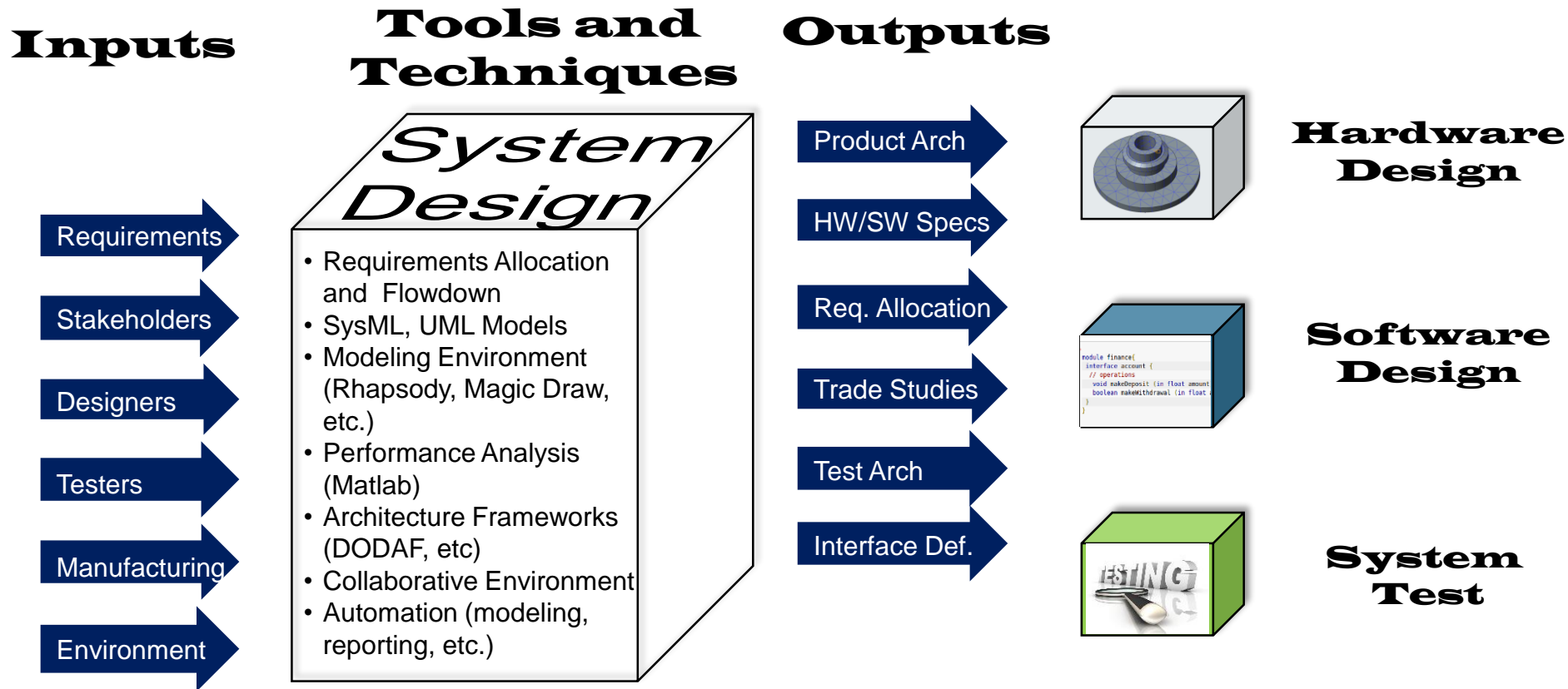
The Models are the Master

Digital Thread Process

- Provides end-to-end information flow across the product lifecycle
- Enables a digitally linked data architecture (OSLC-enabled)
- Determines "what" information is important
- Enhances value-stream mapping and eliminates "air gaps"

Concept	MBSE	MBD/ MDSD	MBM	Sustainment	End of Life

Linked Data Architecture

MBE | T&E | Manu-facturing | Training | O&S

MBSE = Model Based Systems Engineering
MBD = Model Based Definition
MDSD = Model Driven SW Development
MBM = Model Based Manufacturing

**Provides actionable information through upstream and downstream impact analysis**

System Digital Thread

Inputs

- Requirements
- Stakeholders
- Designers
- Testers
- Manufacturing
- Environment

Tools and Techniques

System Design

- Requirements Allocation and Flowdown
- SysML, UML Models
- Modeling Environment (Rhapsody, Magic Draw, etc.)
- Performance Analysis (Matlab)
- Architecture Frameworks (DODAF, etc)
- Collaborative Environment
- Automation (modeling, reporting, etc.)

Outputs

- Product Arch
- HW/SW Specs
- Req. Allocation
- Trade Studies
- Test Arch
- Interface Def.

Hardware Design

Software Design

System Test

MBSE enables our system design process to yield more accurate and consistent digital thread outputs

Creating the System Digital Thread

Requirements
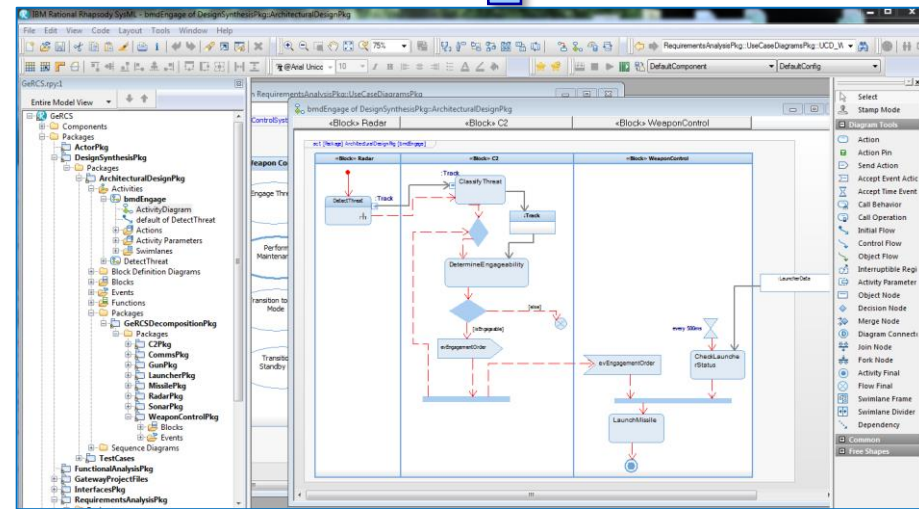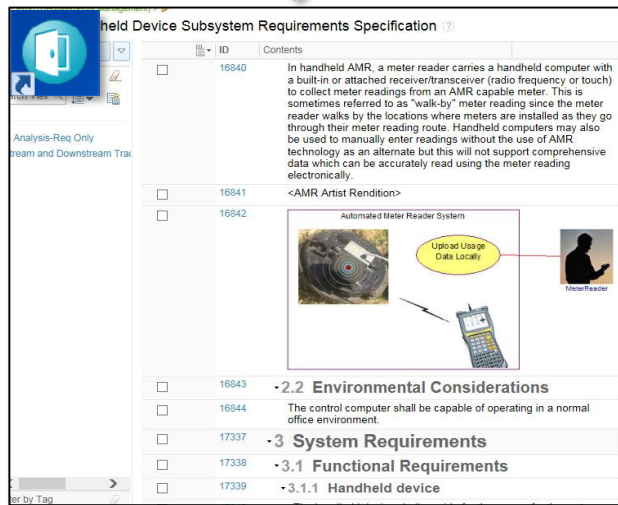
SysML Models

Requirements Allocations/Flowdowns - digital linkages between requirements in a requirements management tool (DNG)

Generate Integrated SysML Model - typically in Rhapsody or MagicDraw. Power Point and Visio SysML diagrams **do not** count
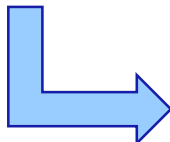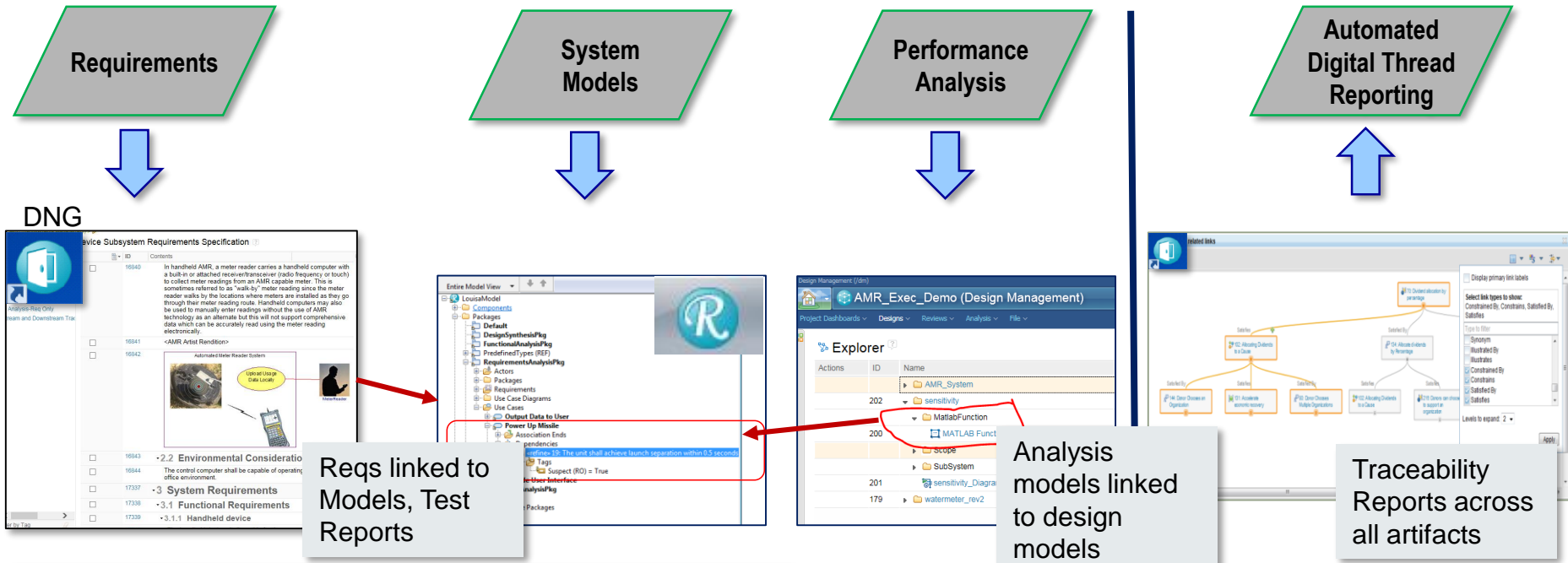
- System Requirements
- Software Requirements
- Hardware Requirements
- Test Requirements

Use Case Modeling

- System Use Cases
- Behaviors
- Interfaces
- Functions

Creating the System Digital Thread

Requirements → **System Models** → **Performance Analysis** → **Automated Digital Thread Reporting**

DNG

Reqs linked to Models, Test Reports

Analysis models linked to design models

Traceability Reports across all artifacts

Requirements Allocations/Flowdowns - digital linkages typically between requirements and the SysML/UML models, HW Design Models, test Artifacts (RQM) and analysis models

System Design Model Traceability – digital linkages between SysML models and other models such as UML models, HW design models, Test Artifacts and analysis models

Automated Report Generation – reports are generated automatically using the tools that contain the digital linkages.

- Software Requirements
- Hardware Requirements
- Test Requirements

Reporting Actionable Information

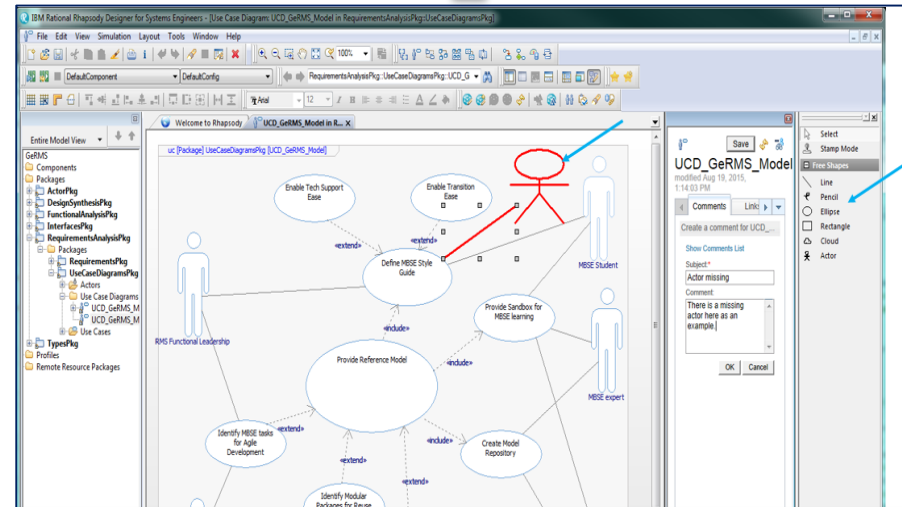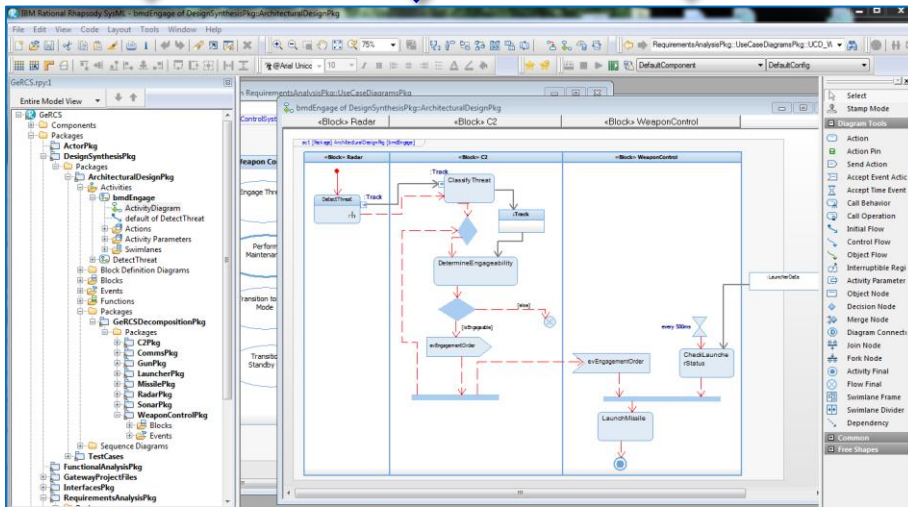- Requirement Traceability
- Verification Matrix
- Impact Analysis

Creating the System Digital Thread

SysML Model

Requirements Links

Performance Analysis Links

Reviewed Models and Digital Thread

Generate Integrated SysML Model - typically in Rhapsody or MagicDraw. Power Point and Visio SysML diagrams **do not** count
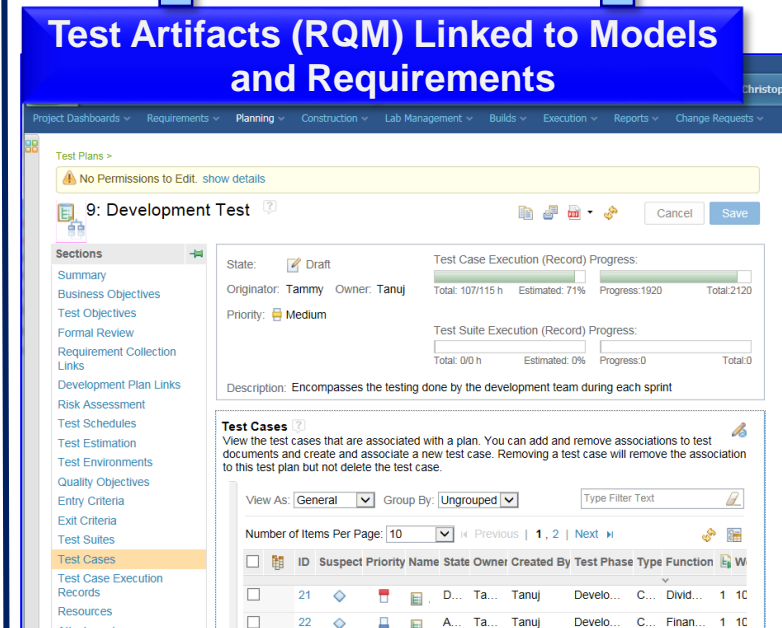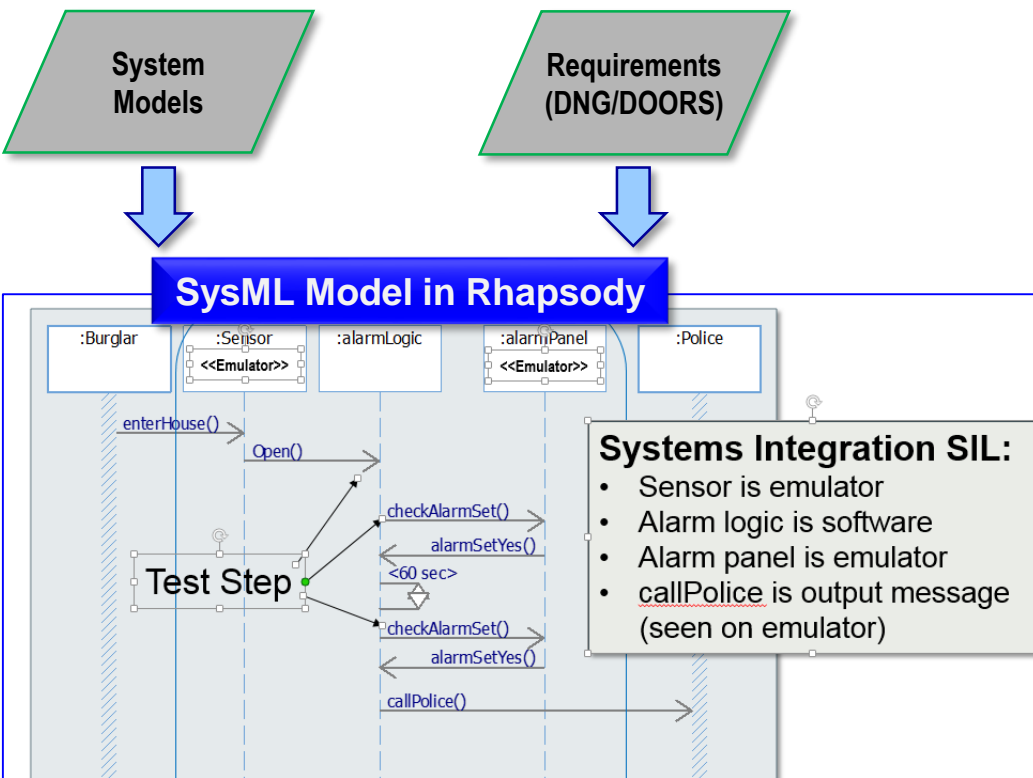
Perform Model Based Peer Reviews - typically in Rhapsody Design Manager (RDM) for Rhapsody or Collaborator for MagicDraw.

- System Use Cases
- Behaviors
- Interfaces
- Functions

Team Reviews

- Web-based (Don't need design tool)
- Comment directly on model (eliminate air-gap)
- Archives with Model View Versions

Creating the System Digital Thread

System Models

Requirements (DNG/DOORS)

Test Plans and Procedures

Test Result Reports and Metrics

SysML Model in Rhapsody

Test Artifacts (RQM) Linked to Models and Requirements

Systems Integration SIL:
- Sensor is emulator
- Alarm logic is software
- Alarm panel is emulator
- callPolice is output message (seen on emulator)

Model Driven Testing - Test Sequences, Vectors and Stimulators defined in models. Test artifacts (e.g., cases, plans, procedures) link to the model(s) to define the scope and interactions required for each test event.

Test Defintion- Test artifacts (e.g., cases, plans, procedures) linked requirements and model. Documents and reports automatically generated

Test Artifact Development

Maintaining the System Digital Thread

| Requirements | System Models | Test Artifacts | | Requirements Baseline | System Models Baseline | Test Artifacts Baseline |

RPE
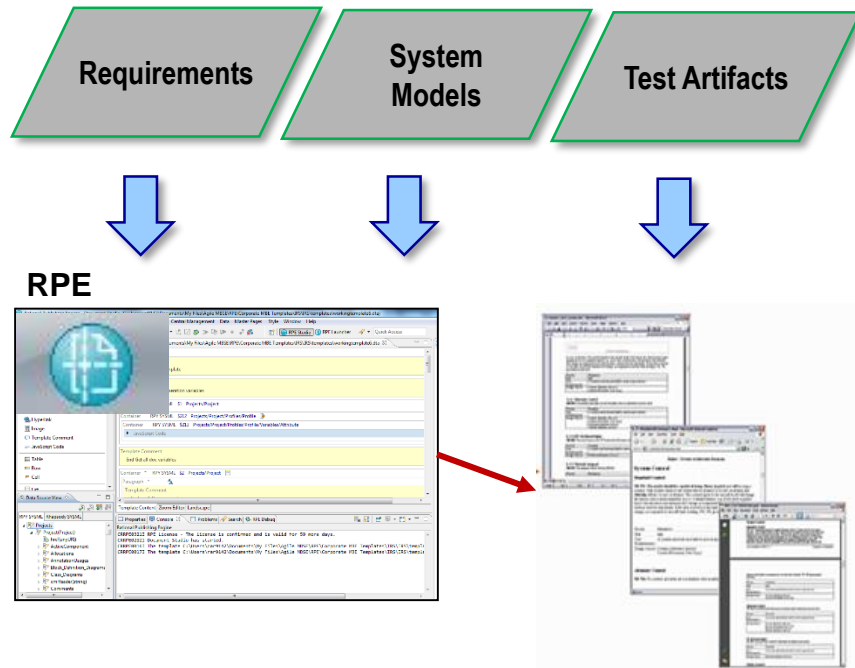
Automatic Creation of Derivative Artifacts - typically with Rational Publishing Engine (RPE) for Rhapsody
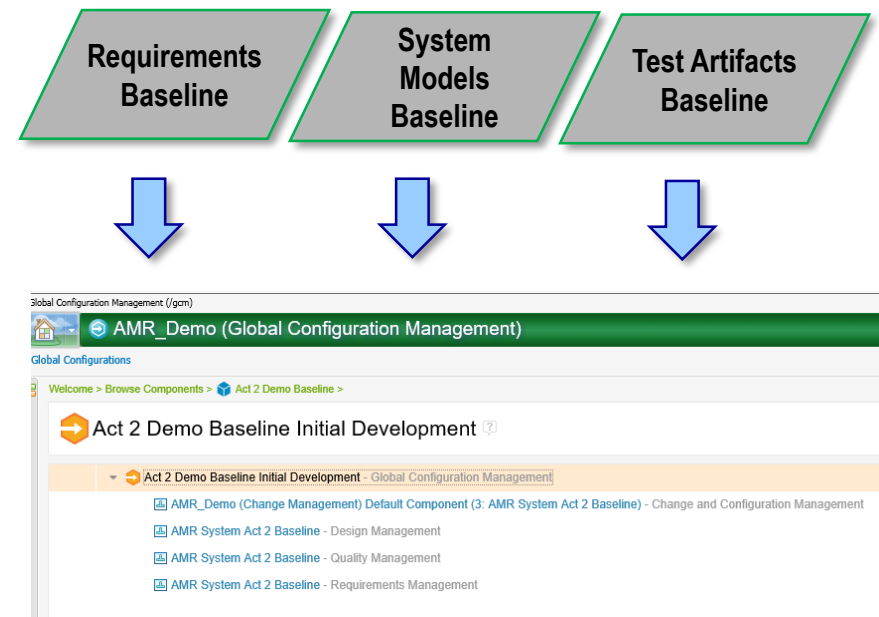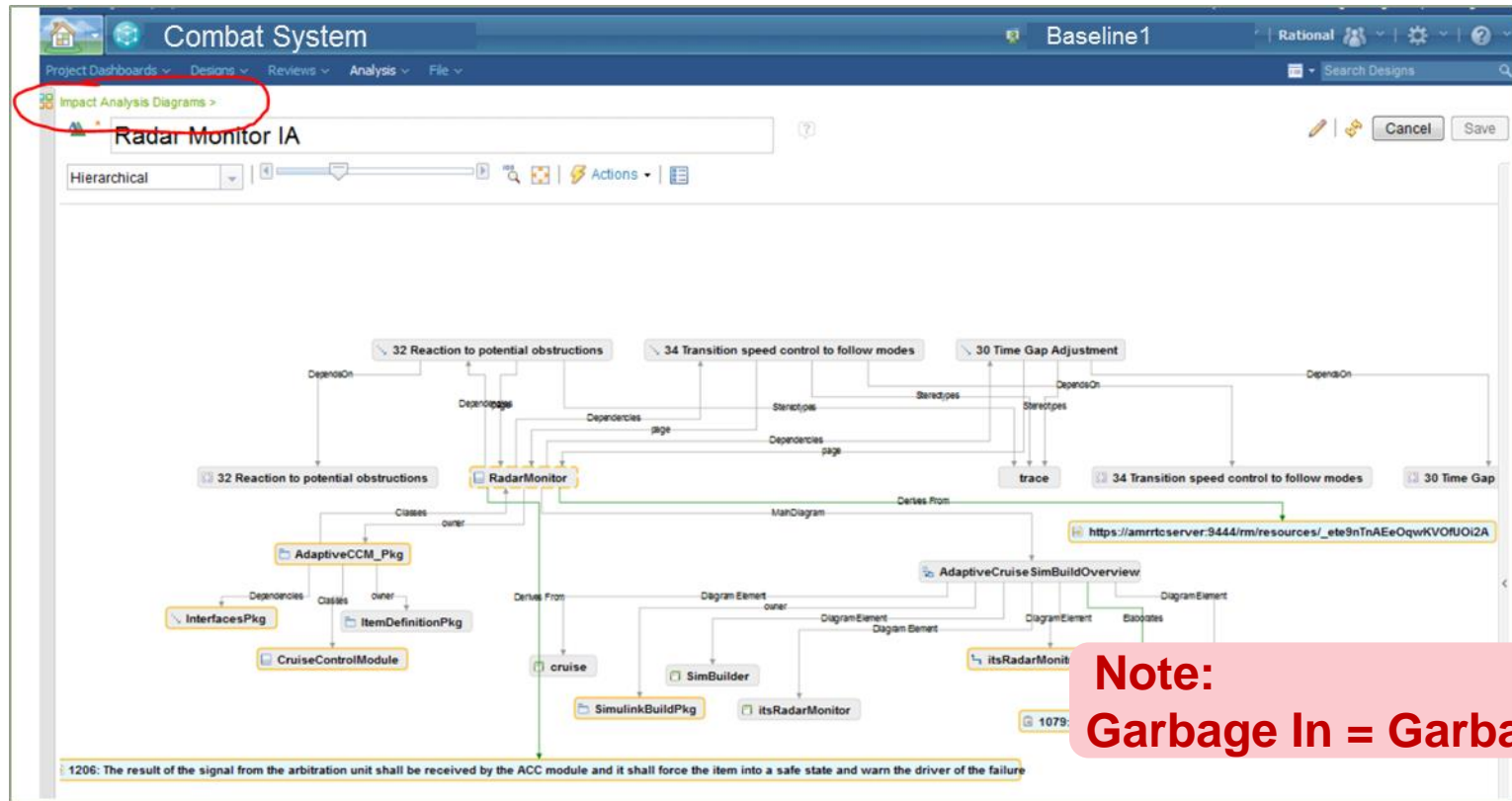
CM of Models – Configures baselines across multiple contributing applications forming a "configuration of configurations"

Keeping the Digital Thread maintained is just as important as creating it in the first place
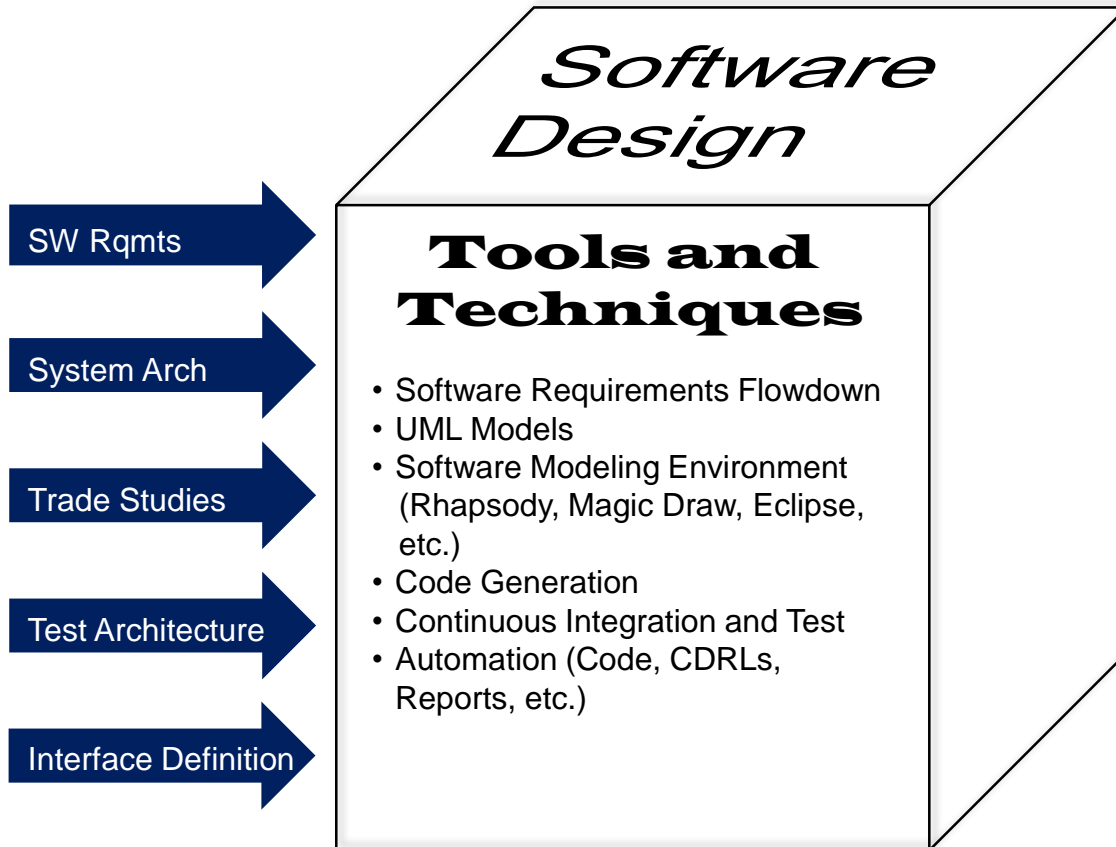
Getting Actionable Information Out

Digital Thread Impact Analysis

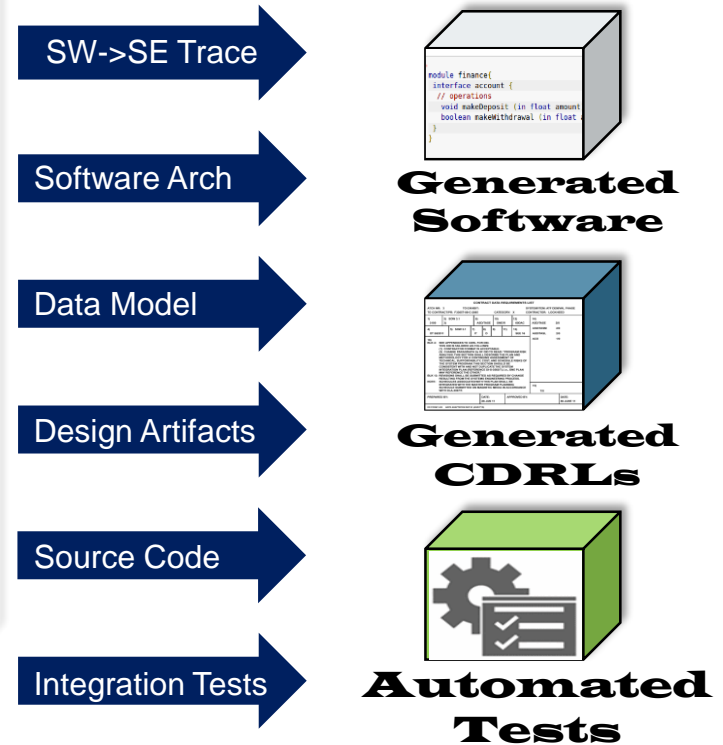

Note:
Garbage In = Garbage Out

Digital Thread rapidly and confidently identifies potential upstream and downstream impacts to design modifications.

Software Digital Thread

Inputs

Outputs

Software Design

Tools and Techniques

- Software Requirements Flowdown
- UML Models
- Software Modeling Environment (Rhapsody, Magic Draw, Eclipse, etc.)
- Code Generation
- Continuous Integration and Test
- Automation (Code, CDRLs, Reports, etc.)

Inputs:
- SW Rqmts
- System Arch
- Trade Studies
- Test Architecture
- Interface Definition

Outputs:
- SW->SE Trace
- Software Arch
- Data Model
- Design Artifacts
- Source Code
- Integration Tests

Generated Software

Generated CDRLs

Automated Tests

Connecting the Digital Thread across engineering functions further enhances design consistency
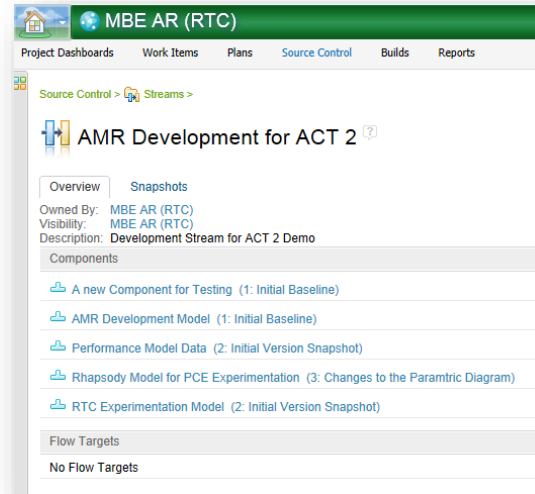
Pulling the Digital Thread through Software

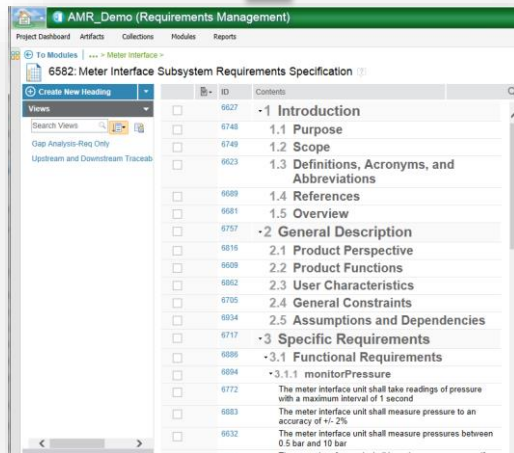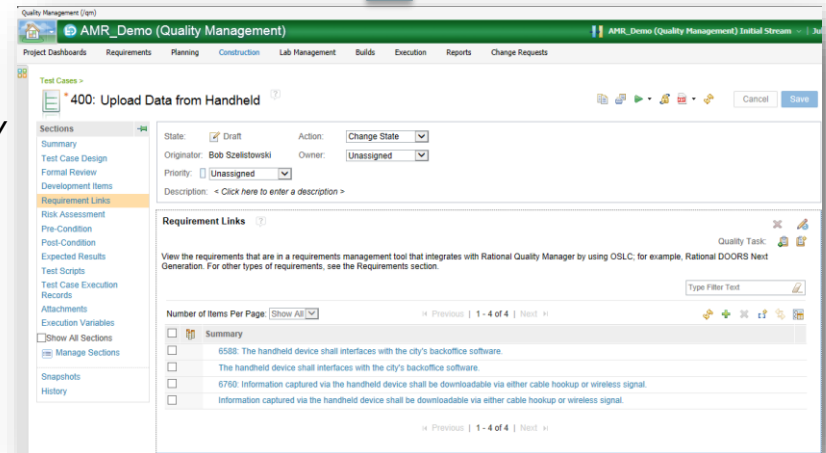*Create Software & Data Model in Rhapsody/RDM*
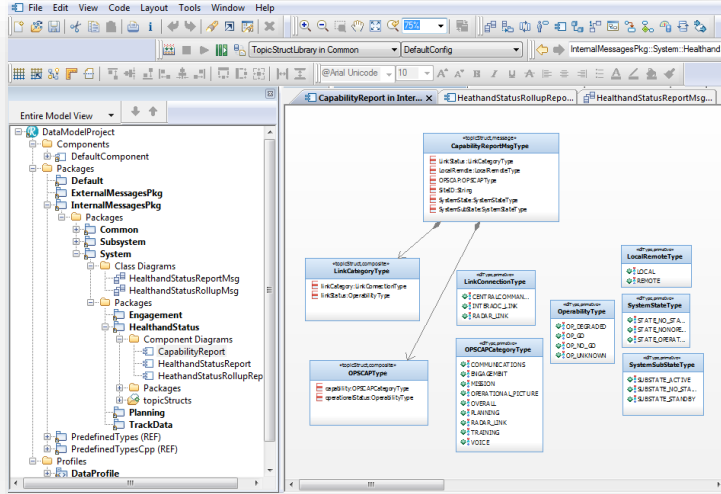
Manage/Track Changes in RTC

Code Generated from Model

Trace to Model
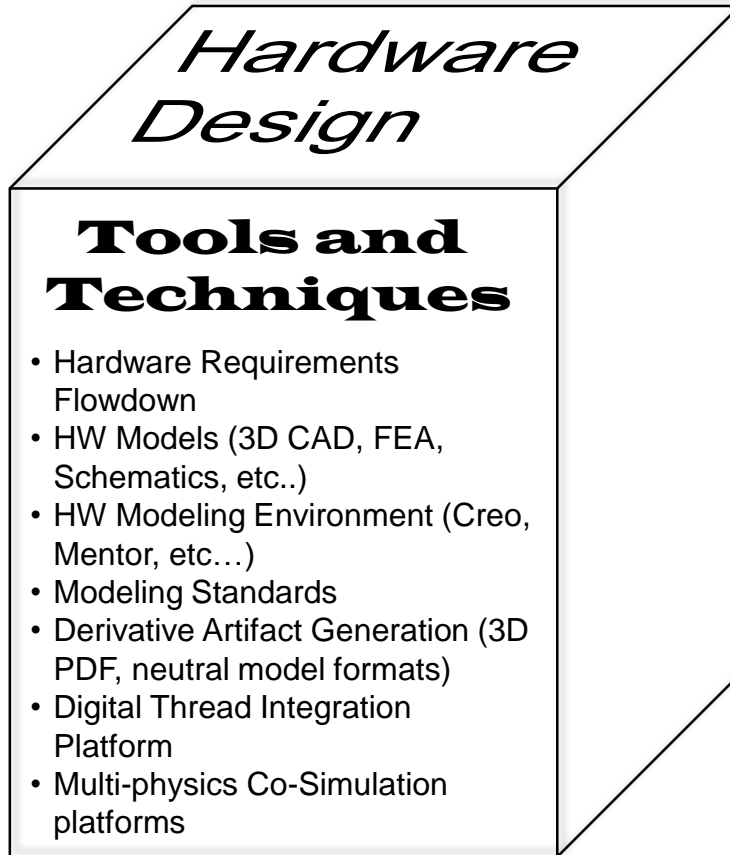
Continuous IV&V

Requirements in DNG

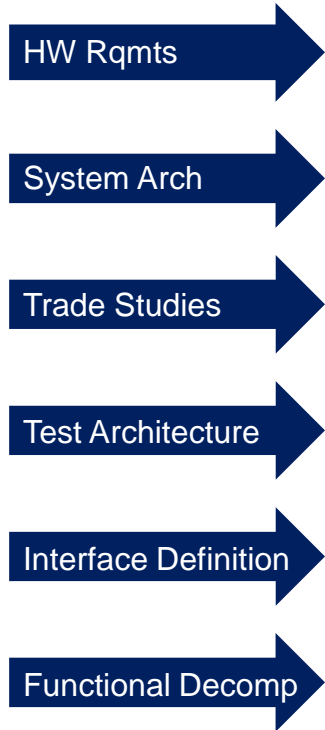*Validated by Test Case*

Test Cases & Execution Results in RQM

HW Digital Thread

Inputs

- HW Rqmts
- System Arch
- Trade Studies
- Test Architecture
- Interface Definition
- Functional Decomp

Hardware Design

Tools and Techniques

- Hardware Requirements Flowdown
- HW Models (3D CAD, FEA, Schematics, etc..)
- HW Modeling Environment (Creo, Mentor, etc…)
- Modeling Standards
- Derivative Artifact Generation (3D PDF, neutral model formats)
- Digital Thread Integration Platform
- Multi-physics Co-Simulation platforms

Outputs

- HW->Sys Trace
- HW Arch
- HW Models
- Design Artifacts
- Integration Tests

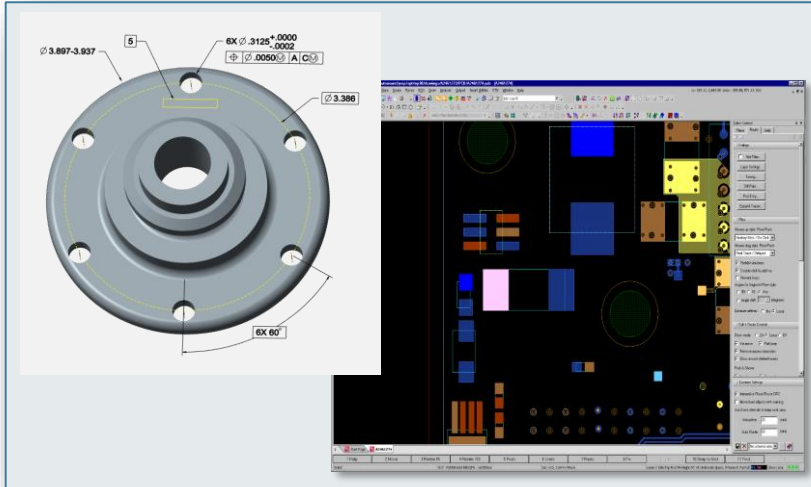**Hardware Design**
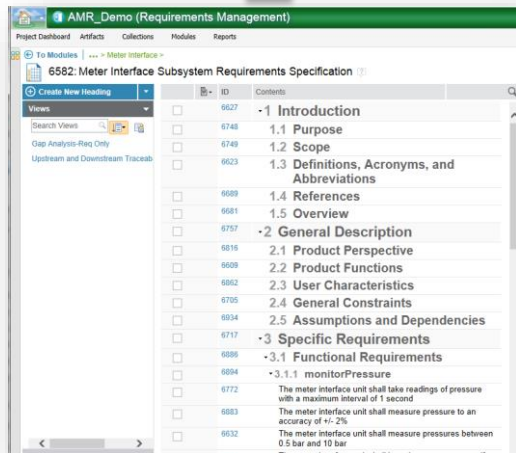
Technical Data Package

Digital Twin Artifacts

The HW Digital Thread provides the basis for Model Based Manufacturing and the Digital Twin

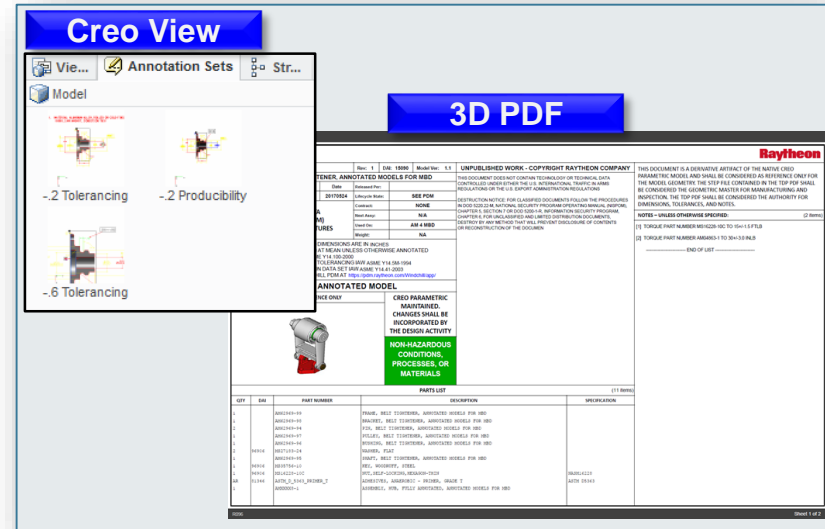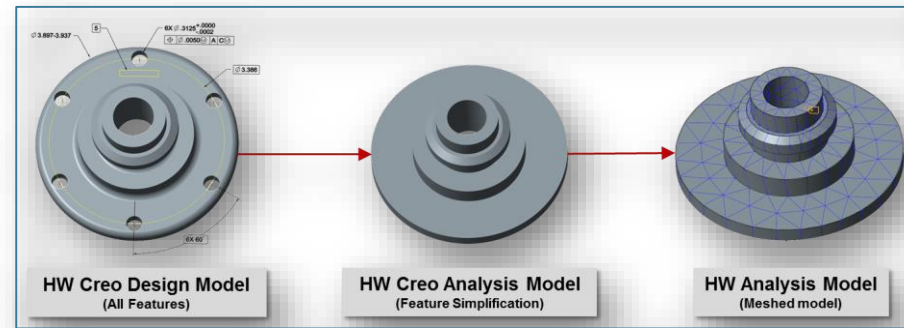# Pulling the Digital Thread through HW

Create ME/EE Design Models

Model Based Peer Reviews

Derivative Artifacts Generated from Model

Trace to Model

Design Analysis and Optimization

Validated by Test Case

HW Requirements in DNG

Analysis Models Linked and Sourced to Design Model

MBE Digital Thread Benefits

- Because information is linked and does not live as stove-piped information in documents or disconnected models…
  - **Eliminate manual transfers**, data redundancy and increase data integrity (*removes "air gaps"*)
  - **Provides automated impact analysis** on proposed changes
  - Facilitates **traceability of design decisions** for life of design
  - Make **changes in one place** and propagate change through linkages (lowers risk of missing key work products or causing disconnects / escapes
  - Can perform **early and continuous design refinement** with easy cross reference to design details
  - Models may be **re-used** across disciplines, across the life cycle of a program and across programs
  - Enforced rigor **reduces risk** associated with system complexity
  - **Communicate more effectively** across stakeholders because of the graphical nature of many types of models. (*shift defect detection curve to the left*)
  - Facilitates **knowledge transfer** of our system design decisions.

Lessons Learned

- Technology is still emerging, we can't do everything we need to yet to eliminate all the "air gaps"

- Some 3rd party OEMs collaborate more openly with others
  - Digital Thread will only survive if tools integrate with each other through common standards… no one tool meets all needs
  - Need more collaboration amongst the tool vendors

- Customers are starting to ask for MBSE/MBE specifically in RFPs ☺… RFP language does not accurately reflect common MBE conventions or specifies the MBE digital thread vision but does not reflect the current state of technology ☹

- There is still a cultural barrier both within industry and with the Customer on MBE adoption. Good news is that we are all making headway