DRAPER

Efficacy of Model Based Systems Engineering (MBSE) for Test and Evaluation

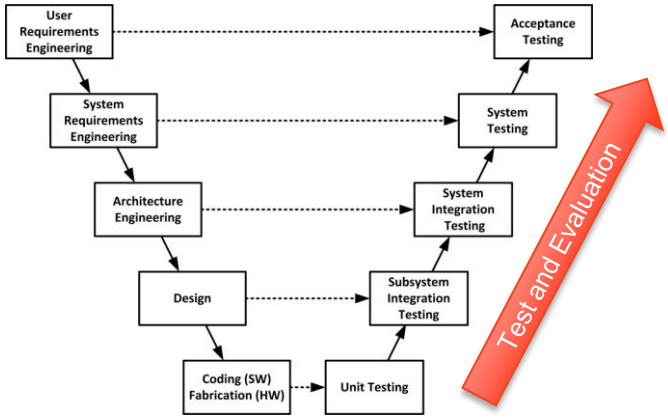
National Defense Industrial Association
19th Annual Systems Engineering Conference
10/24/2016 - 10/27/2016

Steven Monzon, Matthew Pittard, Damien Boudreau, Buphus Nall, Peter Castelli, Brenan McCarragher

Draper

Model-Based Systems Engineering applied to Test and Evaluation

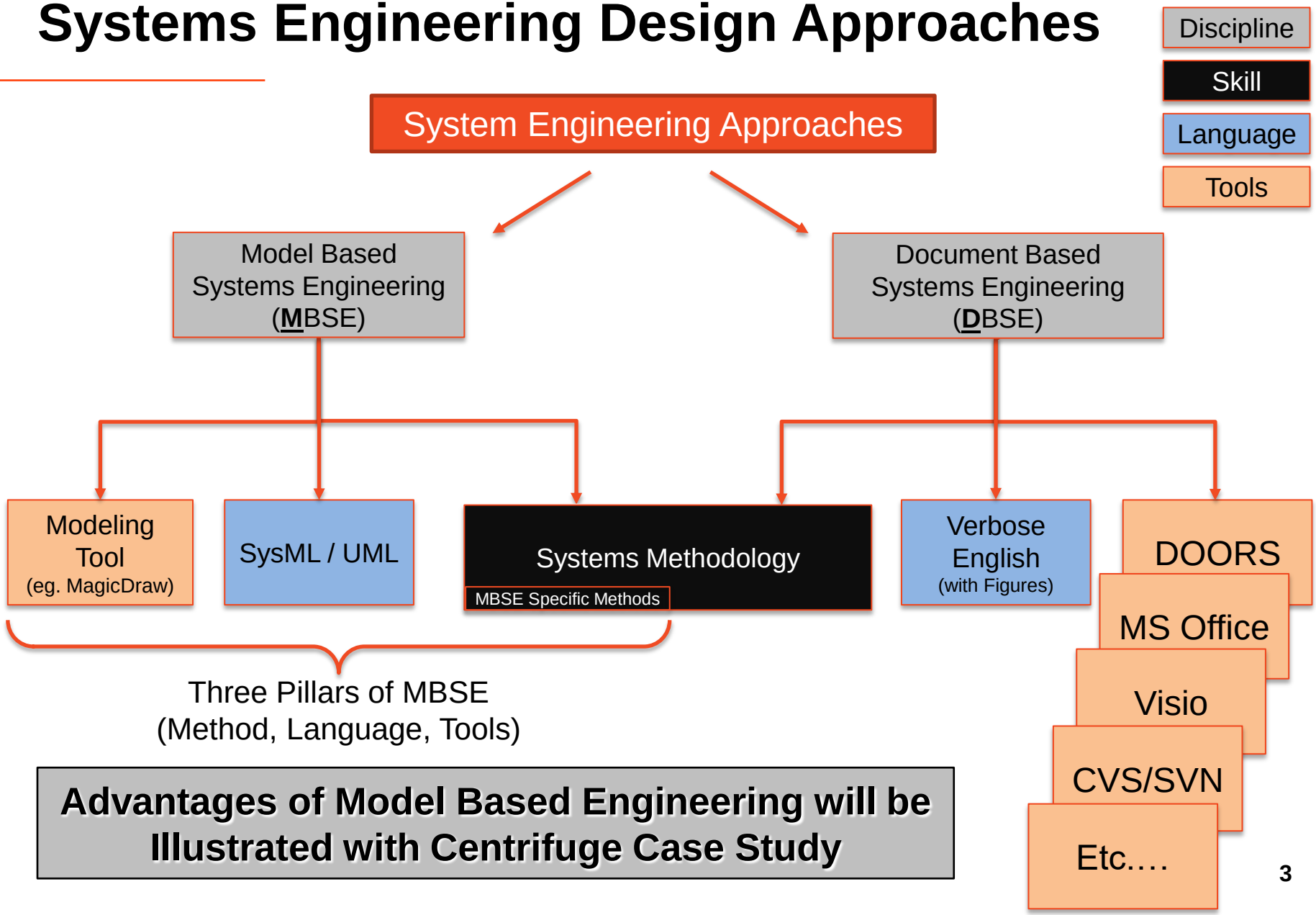
- **Evaluating MBSE for application on test and evaluation program**
  - *Little information on applying MBSE to T&E*
  - *Difficulty in finding information on cost/benefit of MBSE*



- **Centrifuge Case Study**
  - *Modeling of a centrifuge test program was performed to evaluate the cost and benefit of applying MBSE to a T&E program*
  - *Modeling was limited to the initial development of a centrifuge test program*

Centrifuge Case Study Performed to Explore MBSE Benefits

Document Based vs Model Based Systems Engineering Design Approaches



Advantages of Model Based Engineering will be Illustrated with Centrifuge Case Study

MBSE Case Study Observations

1. Up front training is key in reducing inefficiency in model re-architecting
2. Establishing model structure should be performed early in model development
3. Complex programs that share similar components will benefit most from model re-use
4. Efficiency in design change is the most significant benefit of MBSE
5. Significant return on investment can be realized through use of MBSE
6. Commercially available MBSE tools are still maturing and have limitations fully expressing the language
7. Modeling can range from top level abstraction to the software algorithm level. Appropriate level should be selected according to program need.

MBSE Supports T&E The Same Way it Would The Design of a Product

Observation 1:

Training and Assistance

- **Upfront MBSE training and mentorship is necessary**
  - *Team-wide training would have helped set realistic expectations of MBSE*
  - *Embedding an expert on the design team helps bring everyone up to speed through real time feedback and training through apprenticeship*
  - *Learning how to properly express the language as well as a method of modeling is important up front; lessons learned later in development may cause the need for significant design changes*
- **Modeling in SysML requires acclamation**
  - *Getting up to speed with the language and methodology required a large time investment and personal commitment*
  - *MBSE can often look and feel much different than what most stakeholders are used to, it may require culture change for full adoption*

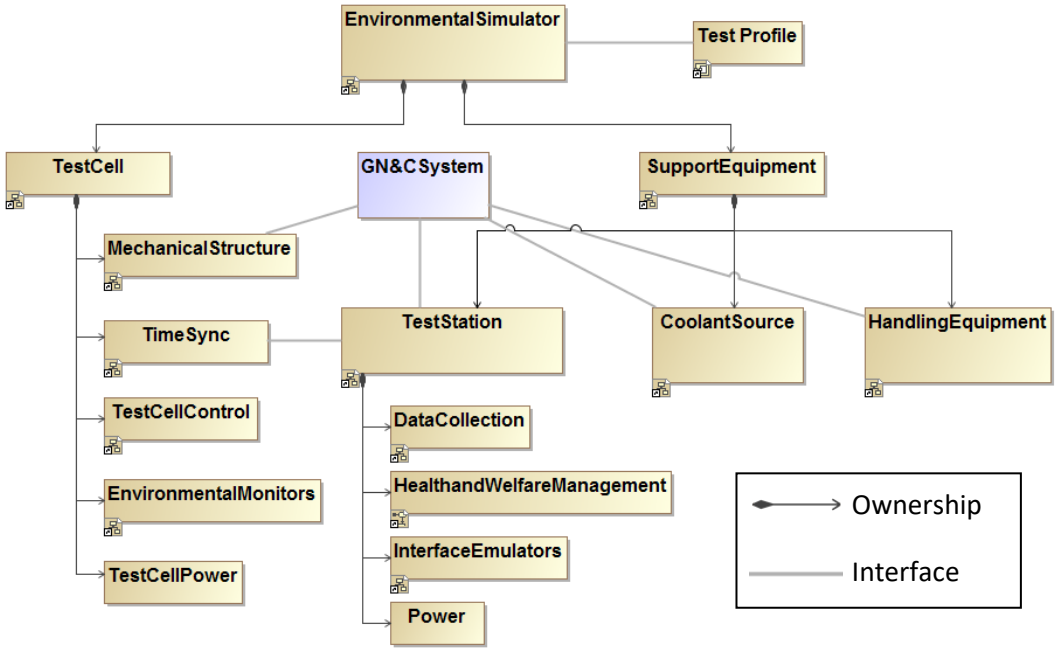
- **Case Study Example:**
  - *Up front training: 3 days*
  - *In depth language training: 4.5 days*
  - *Methodology training: 2 days*

Training and Team Buy-In are Key to Full Adoption of MBSE

Observation 2:

Establish Model Framework Early

- **Establishing a model framework is an important first step and requires team consensus**
  - *Proceeding without a consensus on model structure caused several redesigns*
  - *Consensus took several iterations*
- **Model framework sets the foundation for the MBSE modeling effort**
  - *The model framework was the entry point for the centrifuge definition*
  - *Model provided the foundation a single source of the most up-to-date information, something that DBSE methods lack*



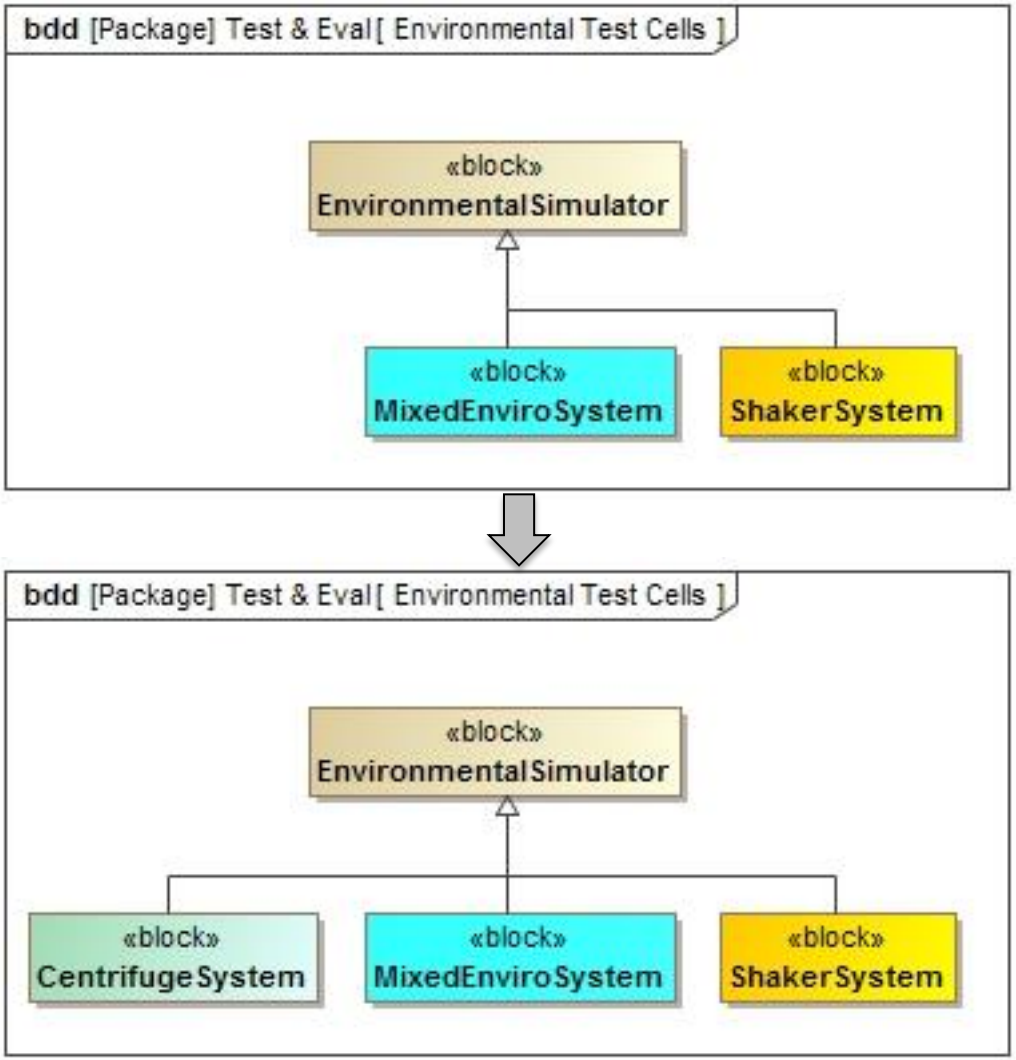
- **Case Study Example:**
  - *Initial model framework took ~30 hours*
  - *~5x 1-hour meetings with leadership to agree on structure*
  - *~6 model re-designs taking ~50 hours combined*

Establishing a Model Framework Requires Team Consensus

Observation 3:

The Benefits of Model Re-Use

- **MBSE helped save time in development by promoting model reuse**
  - *This test and evaluation program required the creation of multiple similar test cells, a process that would have required creating multiple sets of similar documentation using DBSE*
  - *Creating a general test cell structure and having the ability to easily copy and customize things such as requirements, parts, handling procedures, etc. saved modeling time with each added test*



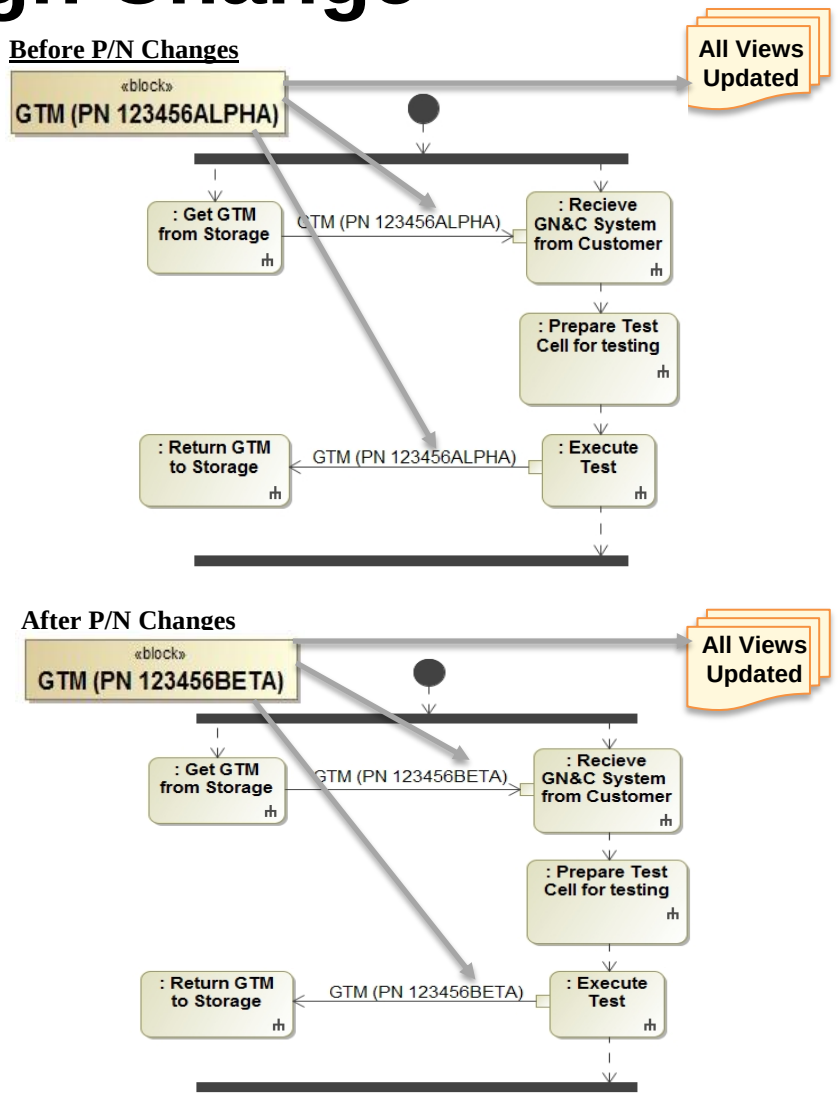
- **Case Study Example:**
  - *Defining centrifuge specific test cell structure ~20 hrs*
  - *Defining common features (health & welfare, handling procedures) ~120 hrs*

Reuse of Common Systems can Save Time in Development

Observation 4:

MBSE Benefit to Design Change

- **Model consistency and a better understanding of system interactions lead to quicker, more accurate design changes**
  - *One of the main observed benefits was the ability to quickly perform a design change*
  - *A design change with traditional documents required an extensive search in order to identify all necessary changes and a series of approvals for each document*
  - *MBSE streamlined the process of design change by often requiring only one change that was propagated through the model*

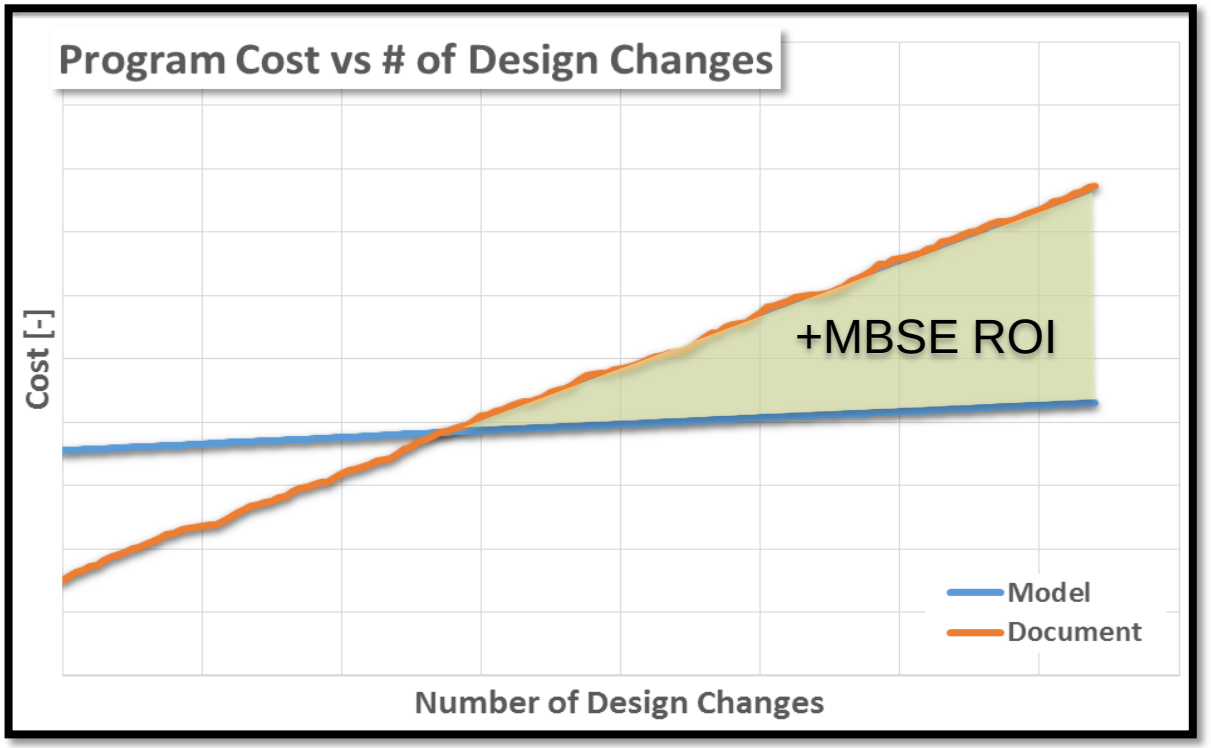


"Automated Consistency Among Work Products When Change Occurs"
– L. Delligatti, *SysML Distilled, 2013*

Observation 5:

Realizing Return on Investment

- **Greater return on investment with more complicated, longer term programs**
  - *Cost to change design was found to be significantly less with MBSE*
  - *Number of design changes generally scale with program size*
  - *As MBSE becomes part of engineering culture, return on investment (ROI) will be realized earlier in the development lifecycle.*

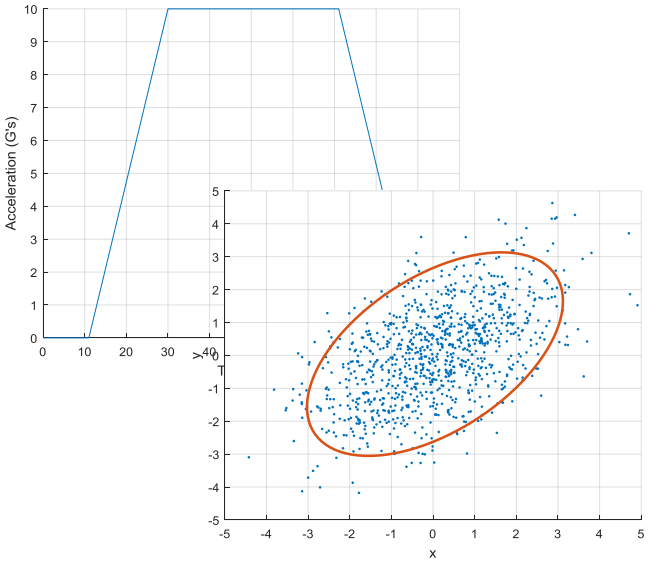


Return on Investment Most Evident with Larger and Longer Term Projects

Observation 6:

MBSE Misconceptions & Limitations

- **Use the right tool for the job**
  - *MBSE is primarily used for modeling system structure and behavior*
  - *Simulation of physics using complicated mathematical relationships are best achieved using separate tools*



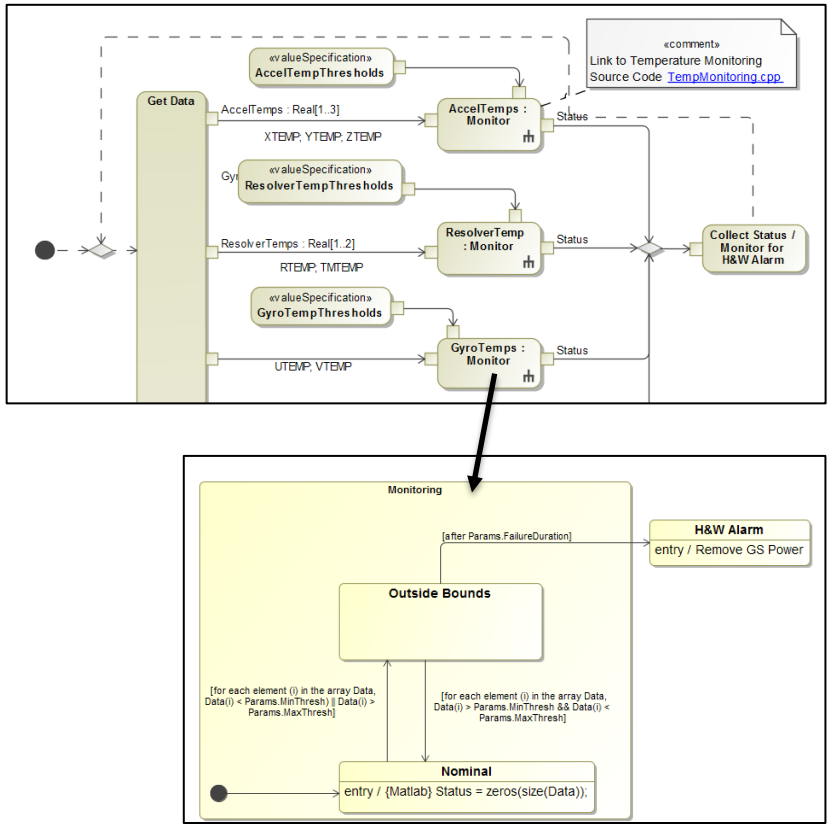
- **MBSE language and tools have limitations**
  - *Geometric design was not performed due to difficulty of implementation in SysML*
  - *While not necessarily forbidden by the language, it was found to be impossible to display particular information such as value properties on behavior diagrams*

MBSE is not a Panacea

Observation 7:

Appropriate Level of Abstraction

- **Level of abstraction depends on stakeholder needs**
  - *The chosen modeling level of detail was directly related to stakeholder desires*
  - *Requirement verification was required to ensure validity of design updates and detailed logic was implemented to ensure this*
- **Not all documents need to be included in the model**
  - *Very detailed documents that are not linked to other aspects of design did not prove beneficial to include*
  - *With existing documents, especially those that were very complex, it often made more sense to reference or make external links*
  - *Even if detailed design descriptions are not currently required, it was found to be beneficial to leave room for expansion in order to avoid potential restructuring*



Appropriate Level of Abstraction Depends on Stakeholder Needs

MBSE Case Study Observations

1. Up front training is key in reducing inefficiency in model re-architecting
2. Establishing model structure should be performed early in model development
3. Complex programs that share similar components will benefit most from model re-use
4. Efficiency in design change is the most significant benefit of MBSE
5. Significant return on investment can be realized through use of MBSE
6. Commercially available MBSE tools are still maturing and have limitations fully expressing the language
7. Modeling can range from top level abstraction to the software algorithm level. Appropriate level should be selected according to program need.

MBSE Supports T&E The Same Way it Would The Design of a Product