Collaborative Modeling

Agile Model-Based Systems Engineering

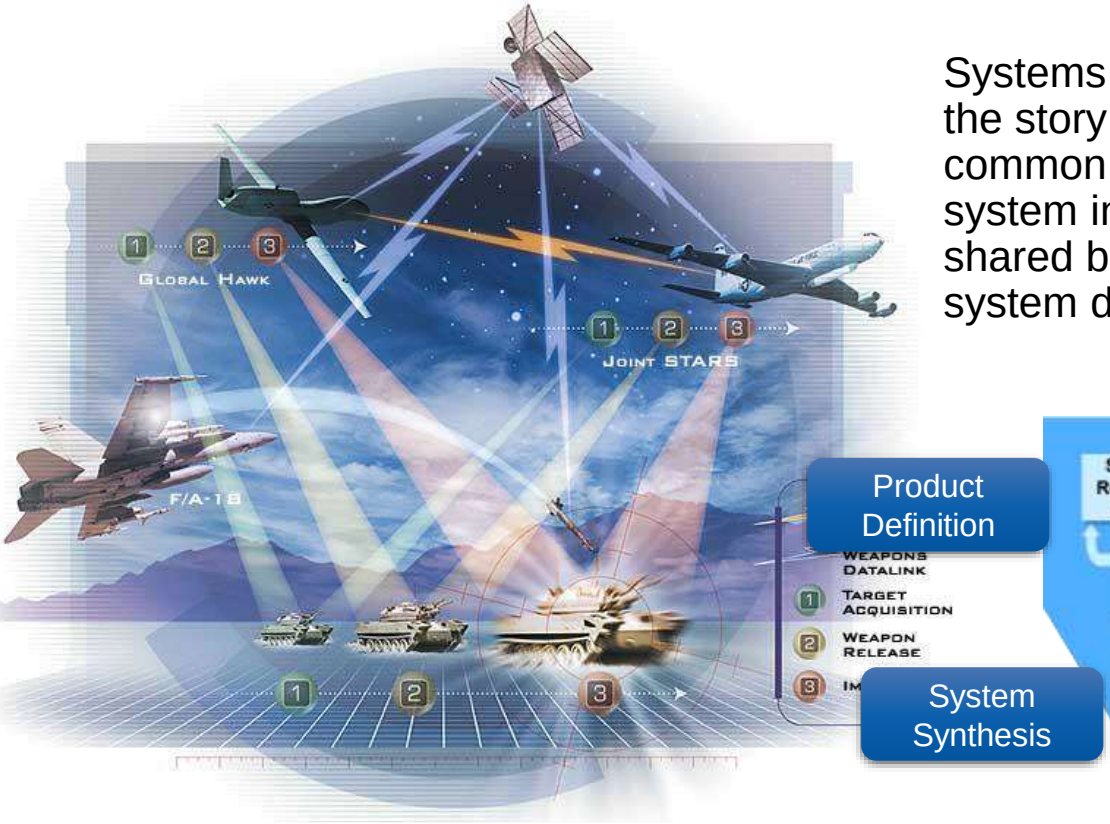
Ken Kubo
Northrop Grumman

November 2022

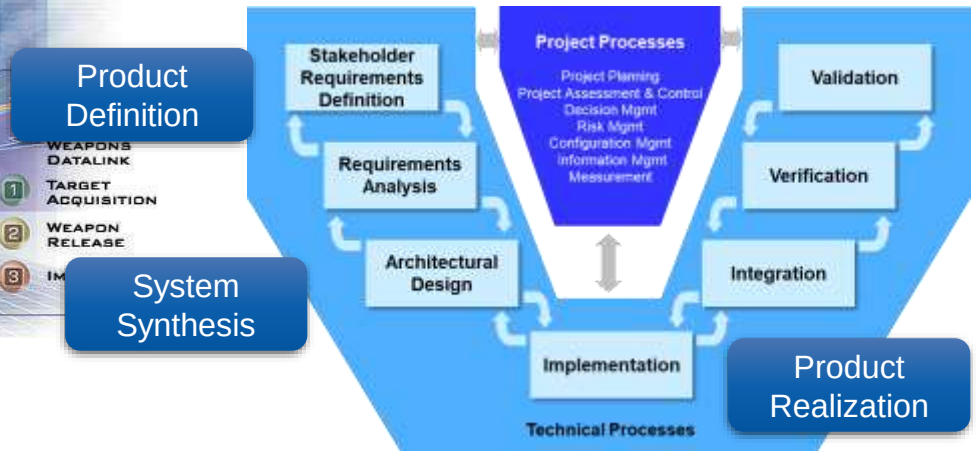
Overview

- Development Processes
- Agile Model-Based Systems Engineering
- Key Enablers and Practices

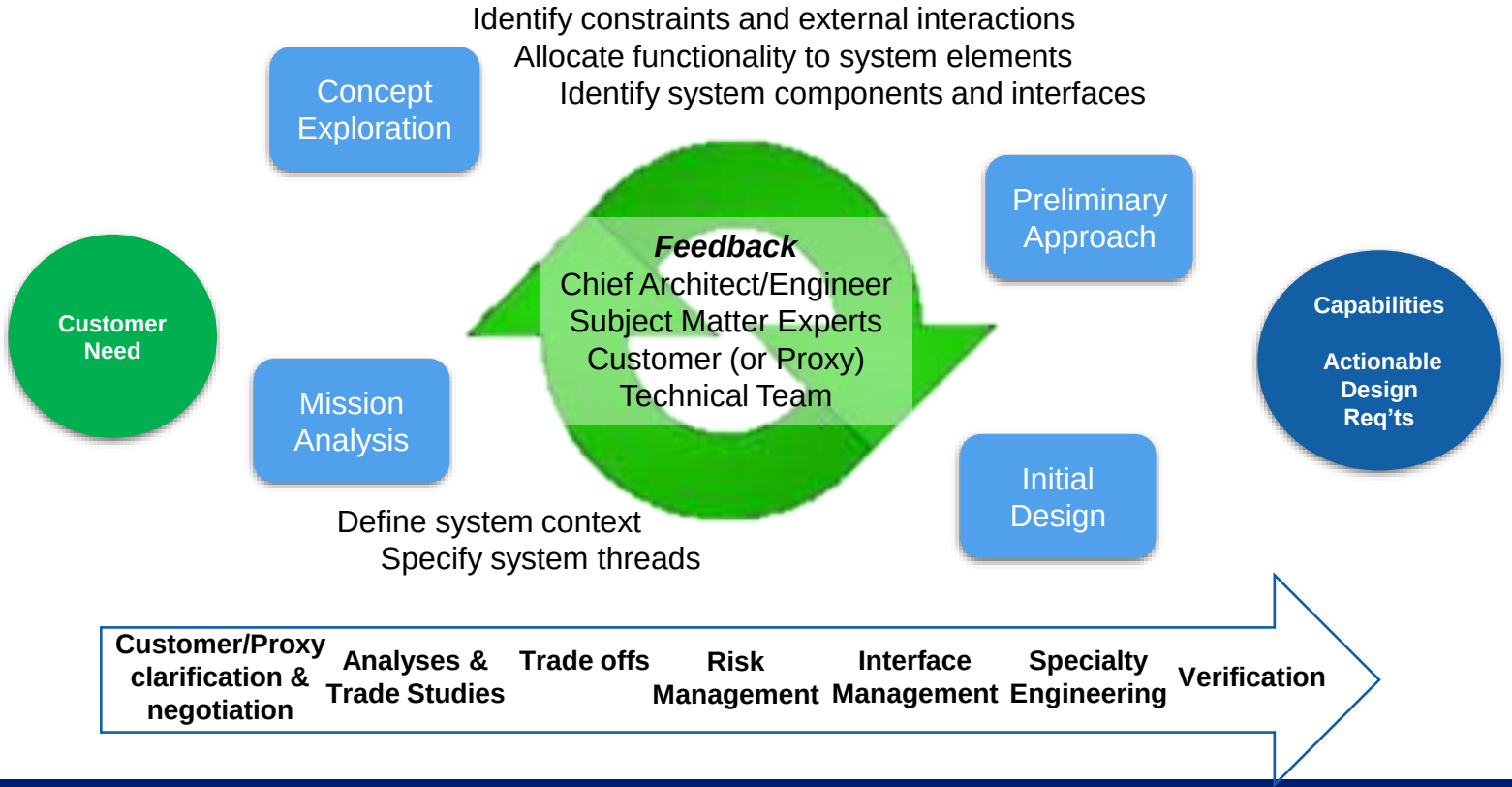
The Big Picture



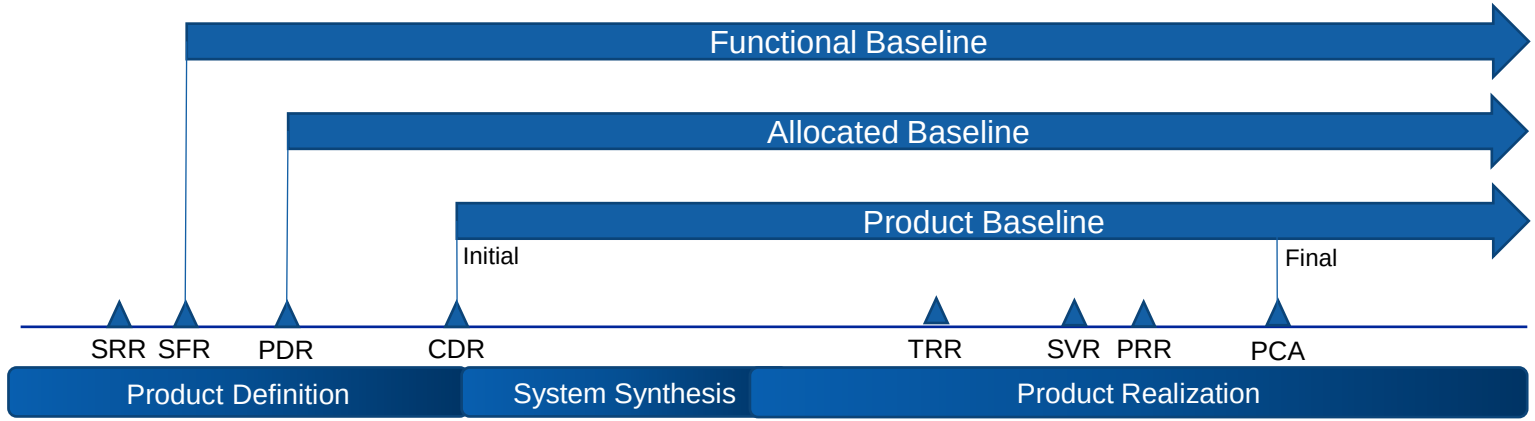
Systems Engineers have the responsibility to tell the story of the system, ensuring that there is a common consistent understanding of the system intent, context, and implementation shared between all stakeholders at all levels of system development and operations.



Requirements Development



Systems Engineering Technical Reviews



CON OPS

Component Specs

Verification Plan

System Spec

Component Analysis
Requirements Allocation

Hardware and Software Integration
System Integration

Produce and Deploy
(and Sustain)

IEEE Std 15288-2-2014
Technical Reviews and Audits

…the traditional requirements development, preliminary design review, and critical design review events may be replaced by incremental design reviews, and, if needed, system-level reviews.
GAO 20-590G

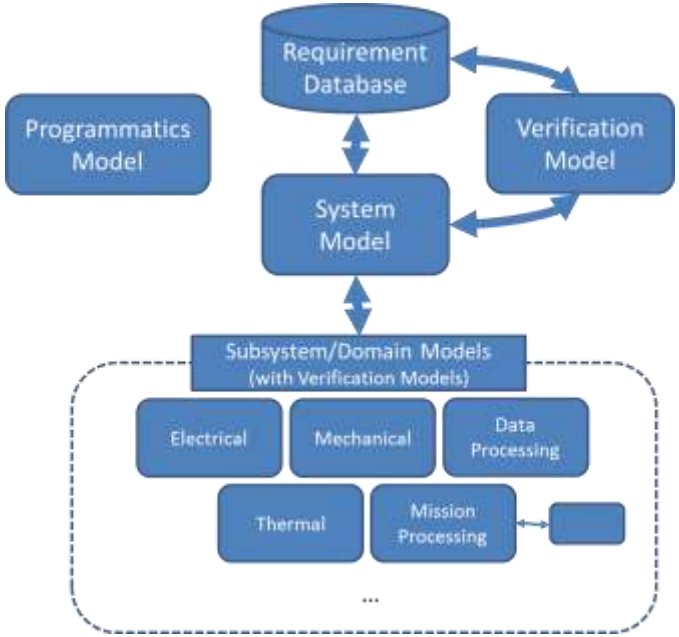
System Specification
Functional Specification

Incremental Design Review

- Confirm user needs: update specs, test approach
- Review change impacts to baselines
- Review updates to architecture and design (and design sets)
- Review Architectural Runway

Need a persistent communication vehicle to maintain consistent understanding of the system, incrementally refined

Model-Based (Systems) Engineering

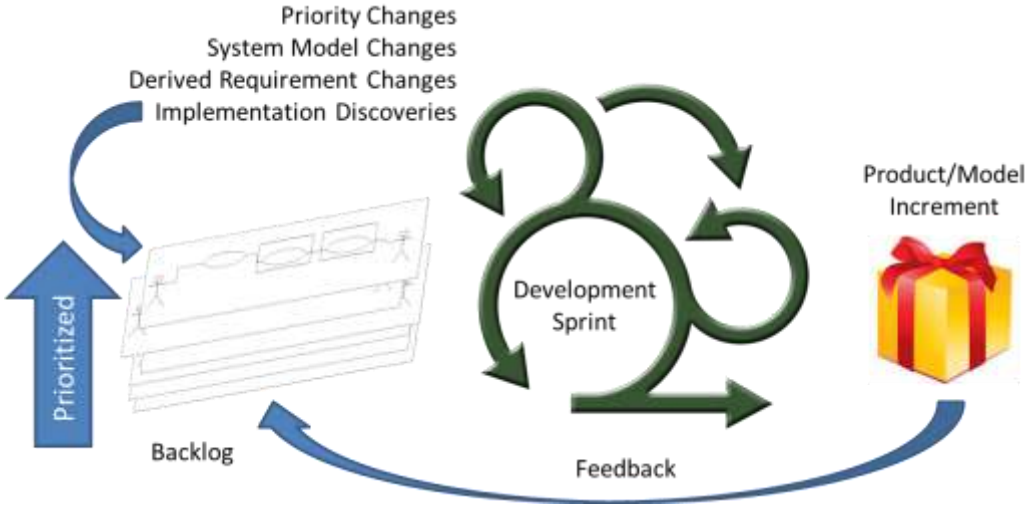


"Model-Based Engineering (MBE): An approach to engineering that uses models (DoD 5000.59-M 1998: a physical, mathematical, or otherwise logical representation of a system, entity, phenomenon, or process) as an integral part of the technical baseline that includes the requirements, analysis, design, implementation, and verification of a capability, system, and/or product throughout the acquisition life cycle."

– Final Report of the Model Based Engineering (MBE) Subcommittee, NDIA Systems Engineering Division, M&S Committee, 10 February 2011

Models provide the consistent context throughout execution

Agile Engineering



"Agile development refers to a group of development methods based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams."

– Alistair Cockburn, et al.

Agile methods provide controlled product evolution

NORTHROP GRUMMAN

The Lean-Agile Environment

- Requirements/desirements are not fully understood (or communicated)
- Priorities shift during the development timeframe
- Unexpected discoveries affect the design (or even the requirements)
- Technical debt impacts development resources and productivity

Requirements must be stable for reliable results. However, the requirements **always** *change.*

- Niels Malotaux, “Evolutionary Project Management Methods”

[Agile methods] embrace change by segmenting … development into manageable change-resistant increments and allowing change to take place at increment boundaries

- Matthew R. Kennedy, David A. Umphress, “An Agile Systems Engineering Process: The Missing Link?”

“Big Modeling Up Front” does not support responsivity to change

NORTHROP GRUMMAN

Agile Principles (for System Development)

1. Our highest priority is to satisfy the customer through *early and continuous delivery* of **a *valuable* system**.
2. *Welcome changing requirements*, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver **a working system** *frequently*, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people *and* developers must *work together* daily throughout the project.
5. Build projects around *motivated individuals*. Give them the environment and support they need, and *trust* them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is *face-to-face conversation*.
7. ***A working system*** is the primary measure of progress.
8. Agile processes promote *sustainable* development. The sponsors, developers, and users should be able to maintain a constant pace *indefinitely*.
9. Continuous attention to *technical excellence and good design* enhances agility.
10. *Simplicity* – the art of maximizing the amount of work not done – is essential.
11. The best architectures, requirements, and designs emerge from *self-organizing teams*.
12. At regular intervals, the team *reflects* on how to become more effective, then *tunes* and *adjusts* its behavior accordingly.

NORTHROP GRUMMAN

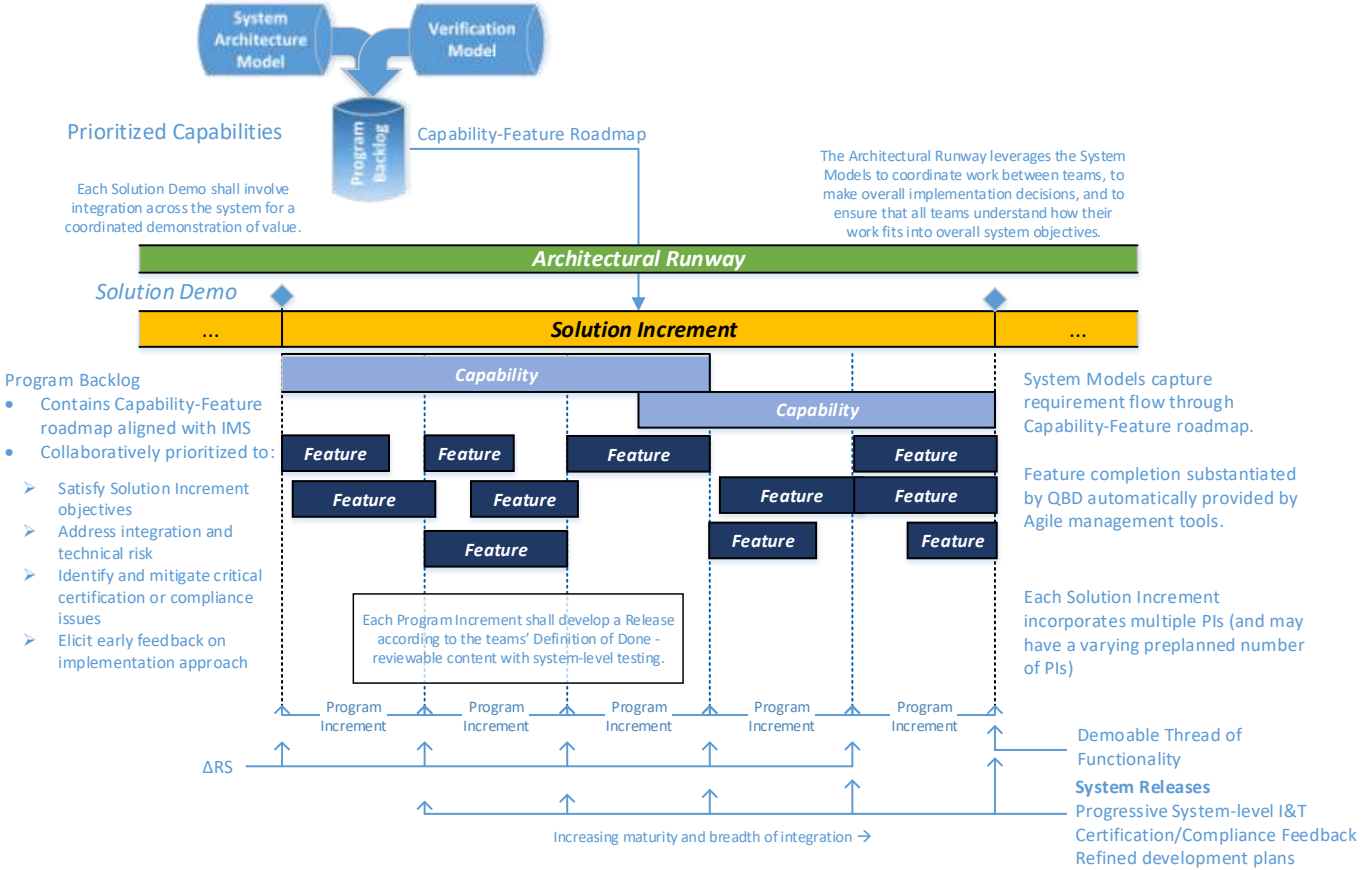
Modeling Agile

- Models are (still) the persistent (and controlled) artifact in the development flow.
- Models are a living artifact that keep the requirements and product architecture (and documentation) consistent in the technical baseline.
- Models enhance communication with the customer and within the team and represent a single source of truth.
- Models support the consistent elaboration of requirements to reduce speculative work.
- Models highlight risk areas (interfaces and ignorance).
- Models help provide the context to drive “system thinking” to the individual developers.

Agile Modeling

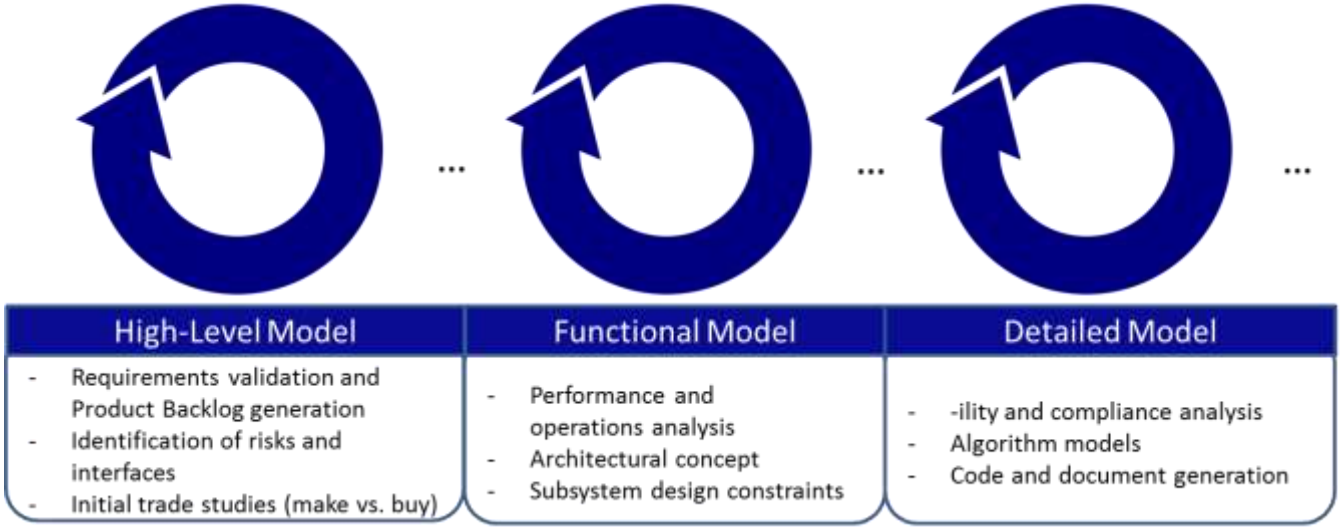
- Agile manages changing requirements and discoveries through prioritized work.
- Agile captures requirements as system interactions, supporting use case development.
- Agile focuses on delivering stable capabilities or threads of execution through a system, allowing deferral of unneeded detailed modeling.

Agile Systems Engineering Flow



- Each Feature includes all elements for a testable (demoable) function, including documentation/plan deltas, model/drawing/code updates, and feature-level testing.
- Systems Engineering supports development of SE products as Features become actionable. SE also supports clarification/refinement of requirements (acceptance criteria, Features, Stories) during implementation.

Agile Model-Based Engineering



- Refine the System Model
- Identify Capabilities and derive Features
- Model the functionality of Features for clarity
- Leverage models to refine implementation
- Maintain models with incremental learning

Key Practices

- “Right Sized” Modeling - avoid Big Modeling Up Front (BMUF)
- Model Sufficiency
- Model as the primary persistent artifact – Configuration Control!
- Model-Based, Acceptance Test-Driven Development (MBATDD)
- Collaborative Model Evolution – Rules of Engagement
- Pair Modeling

The Definition of Done

- Format – is the model expressed in the correct framework and notation and in the correct tool? Are the units, dimensions, data types consistent with the parent system model(s)? Are the key views still valid?
- Linkage – are there appropriate links connecting the model to the parent system model?
- Architecture – does the model fit within the existing system model without affecting any interfaces to other parts of the system? Did the team check their model with the appropriate chief engineer/architect?
- Meta-Data – are all required meta-data fields completed?
- Traceability – are all appropriate requirements properly traced through this level of model?
- Documentation – can appropriate documentation (still) be generated for this system?
- Testing – have appropriate tests been run on the model (compliance, functionality, performance)? Does the model function appropriately if the parent system model is tested?

Know Before You Go…

- Tool constraints
- Tool integration
- Model framework and interfaces
- Model-based documentation constraints
- Version control standards and workflow

Retrospective

- Agile and MBSE practices reinforce each other.
- Agile provides controlled responsiveness to change.
- Modeling maintains consistency, shared understanding of an entity, management of solution intent.
- Standard tools support these practices.

Embrace change in a consistent and traceable manner

Questions?

"Efficiency is important, but it's not a value differentiator. Working harder and faster means nothing if you're on the wrong path—it only means you're getting to the wrong place faster. What is important in today's business age are two things: adapting to change, and adapting to change *as a team*."

– Jeff Boss, Chaos Advantage