



Engineering, Operations & Technology
Boeing Test & Evaluation

Integrating Operational Testing into Contractor Verification and validation with Mission Based Testing

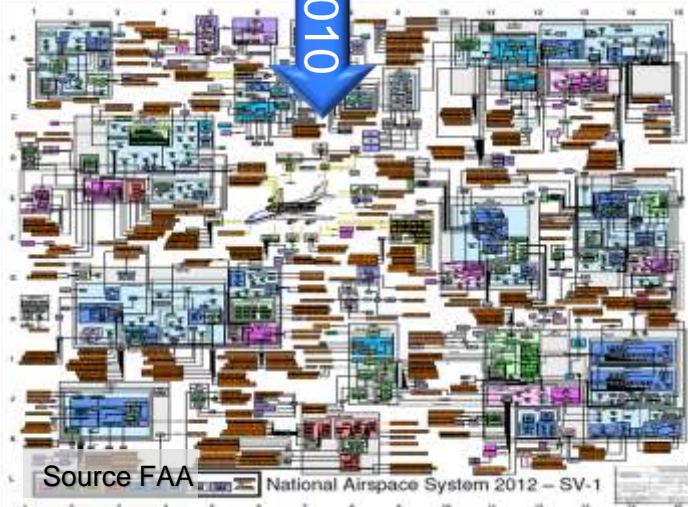
David Allsop – The Boeing Company

2016 System-of-System's Engineering Workshop
January 26-28, 2016
El Paso, Texas

Modern Highly Integrated Systems



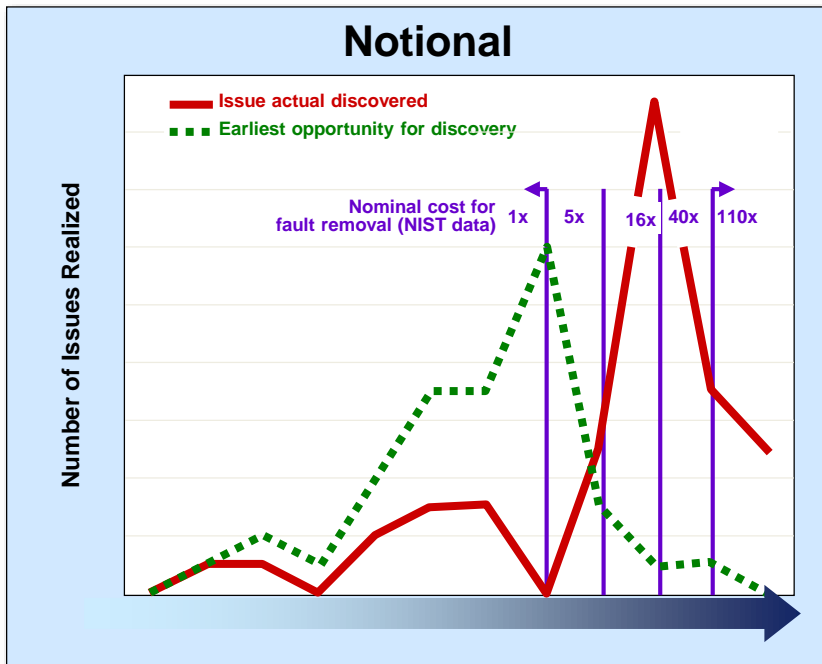
10101010



Today's Products Integrate Into Complex Systems of Systems

Late Discovery is Driving Program Cost and Schedule

The Cost

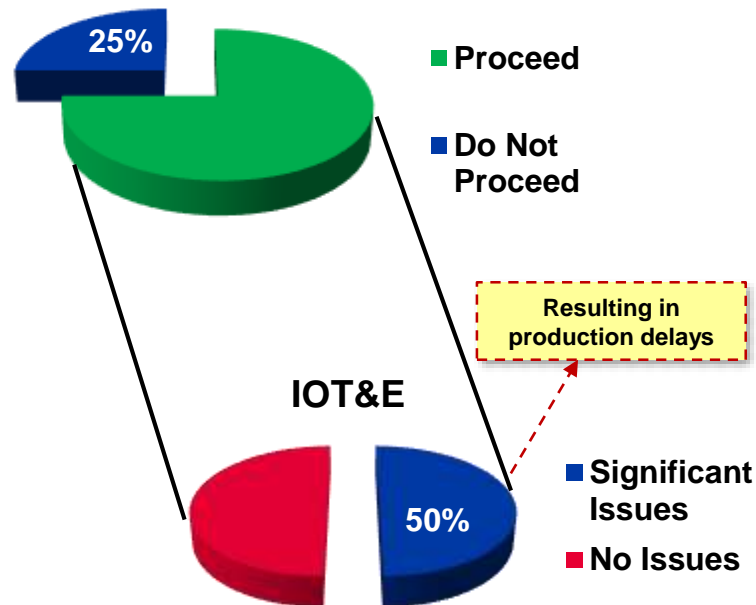


Copyright © 2015 Boeing. All rights reserved.

Late discoveries are considerably more costly!

The Impact

DT&E Assessments recommending to proceed to IOT&E (2012 DT&E Annual Report)



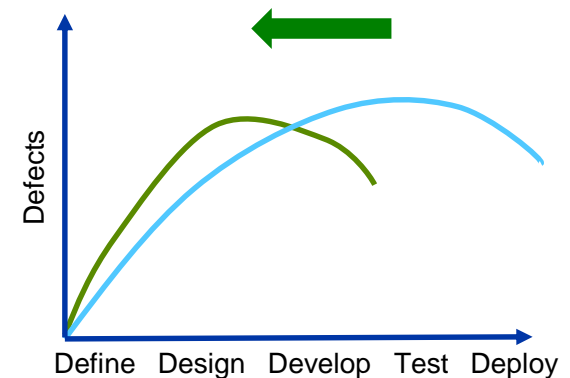
Only 41% (5 of 12) programs made it to & through a successful IOT&E

Sources

- NDIA ICOTE, Chief Developmental Tester & Industry Test Lead – Partnering for success, August 2013
- NIDS Planning report 02-3. The economic Impacts of Inadequate Infrastructure for Software Testing May 2002.
- D.Galin, Software Quality Assurance From Theory to Implementation, Pearson/Addison-Wesley (2004)
- B.W. Boehm, Software Engineering Economics, Prentice Hall (1981)

Shift Left

- Early validation and developmental test is commonly called “shift left”.
- Response to the ever increasing lifecycle cost.
- Declining budgets and poor cost and schedule performance suggest new methods are needed.
- Systems and products are becoming ever more complex with their internal and external design approach evolving into systems of systems.
- Verifying and validating the design early can prevent late problem discovery during test or during customer operation.



Shift Left to Discover Defects Early to Improve Program Performance

Model Based Systems Engineering

“...is the formalized application of modeling to support system requirements, design, analysis, verification and validation activities. It begins at conceptual design & continues throughout the development life cycle.”

INCOSE Vision 2020 (Reference: “International Council on Systems Engineering, Systems Engineering Vision 2020,” INCOSE-TP-2004-004-02, Version/Revision: 2.03, Dated September 2007)

Functional Analysis

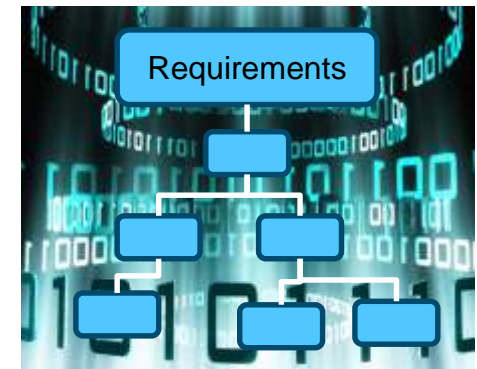
- Systematic process of identifying, describing and relating the functions a system must perform, but not how they will be performed

Functional Thread

- System model based on functions

Mission Thread

- System model based on Mission



ptcdigitalworld.wikispaces.com/immersed+in+digital+technology

Formal application of modeling to support system requirements, design, analysis, verification and validation

Mind the Gap

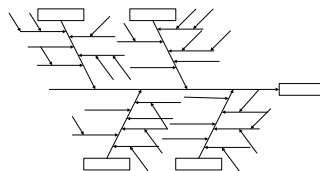


Gap between MBSE and V&V

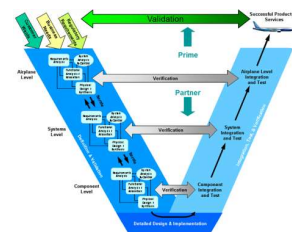
- Validation gap between phases
- Complex systems of systems
- Complex organizations
- Individual systems maturing at different rates
- Unintended system interaction



MBSE



The Gap



MBSE

V&V



Concurrent Integration and Validation

Functional testing with mission threads

- Closes the Gap between upfront MBSE and integration and test
- Enables complex system validation
- Validates the system model based on missions
- V&V Plan

Mission Thread Validation

- Mission validation leverages functional validation
- Validation metric

Benefits

- Discover Integration Issues
- Discover and retire risk
- Discover assumed risk
- Validate requirements and integration
- Capture schedule and prevent EAC growth



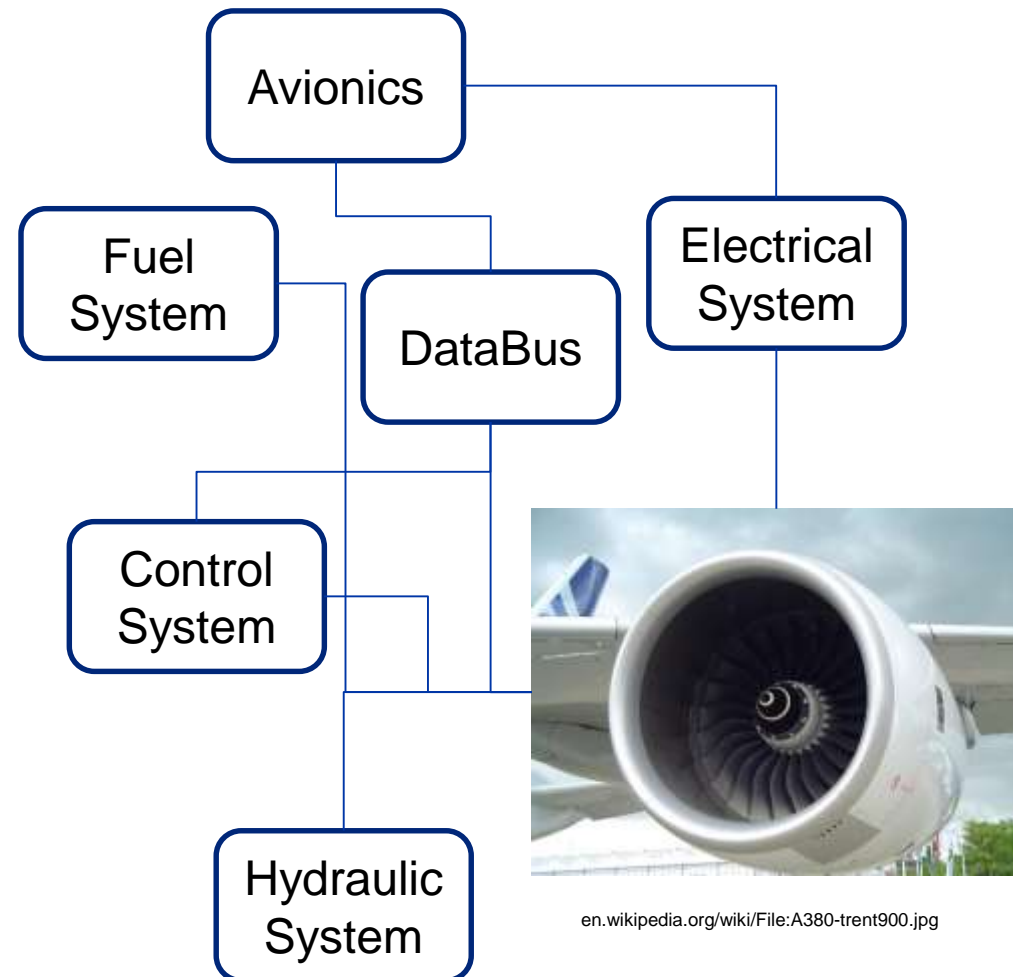
www.Wikipedia.org/wiki/Burr_puzzle

Functional Thread Example

Testing Thread Example

- Validation - Start the Engine
- Complex interaction between many systems
- What model or function system fidelity is needed?
- Functional models maintained in MBSE

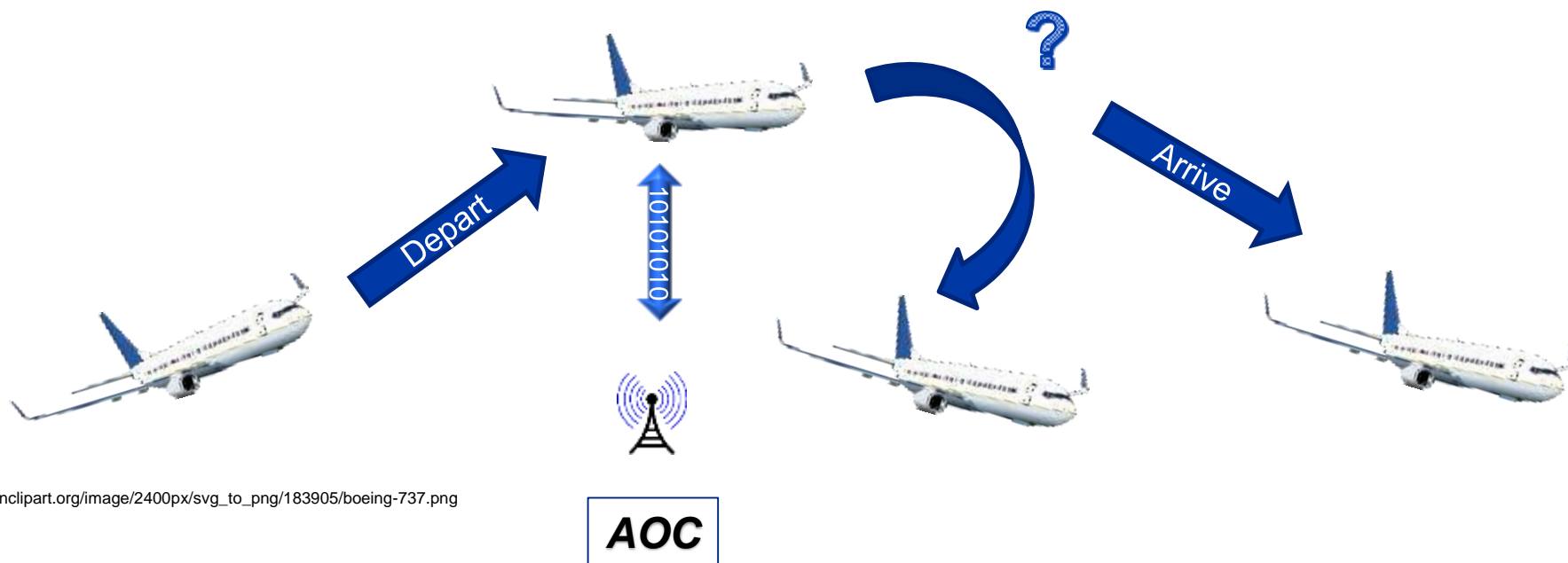
Notional Functional Diagram



Mission Thread Example

Mission Thread Example

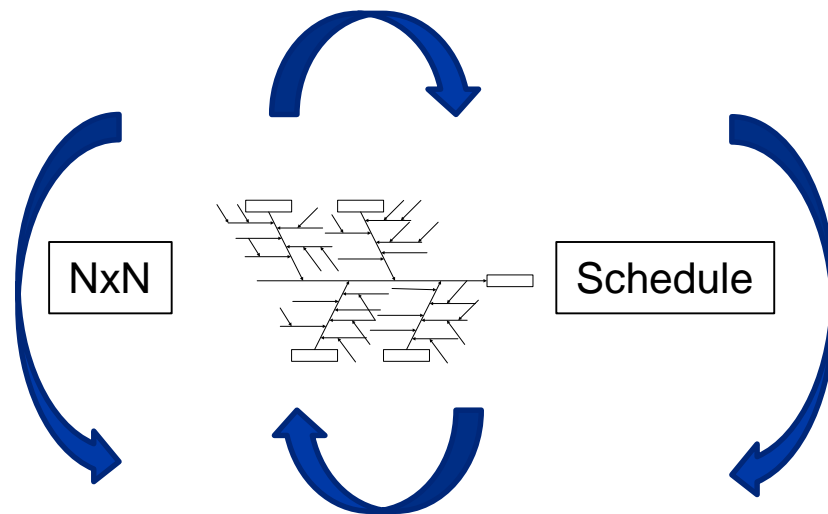
- Validation – On/Off board fault communication
- Complex interaction between many systems
- Modeling, Simulation and physical test environment
- What model or function system fidelity is needed?
- Modeling of external environment



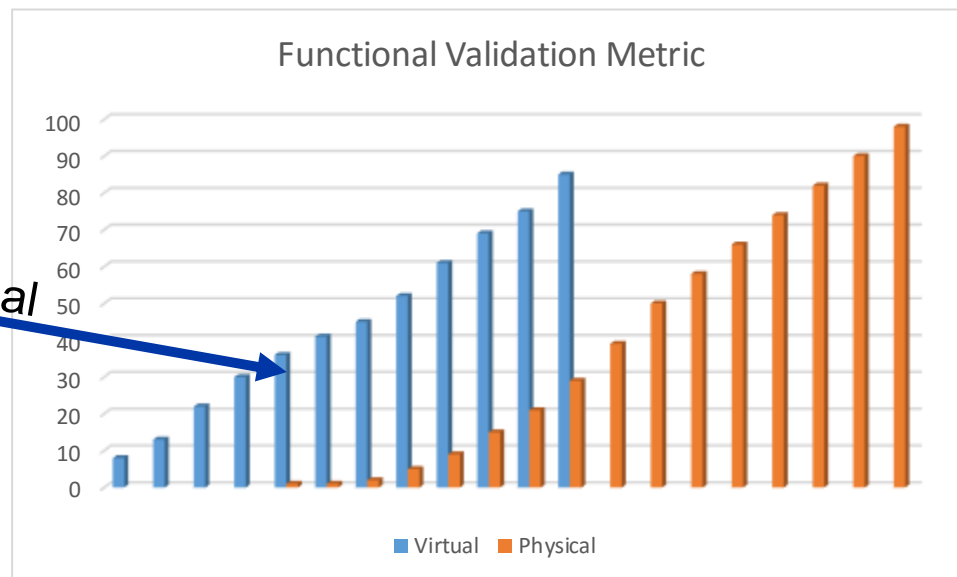
openclipart.org/image/2400px/svg_to_png/183905/boeing-737.png

Tracking Validation

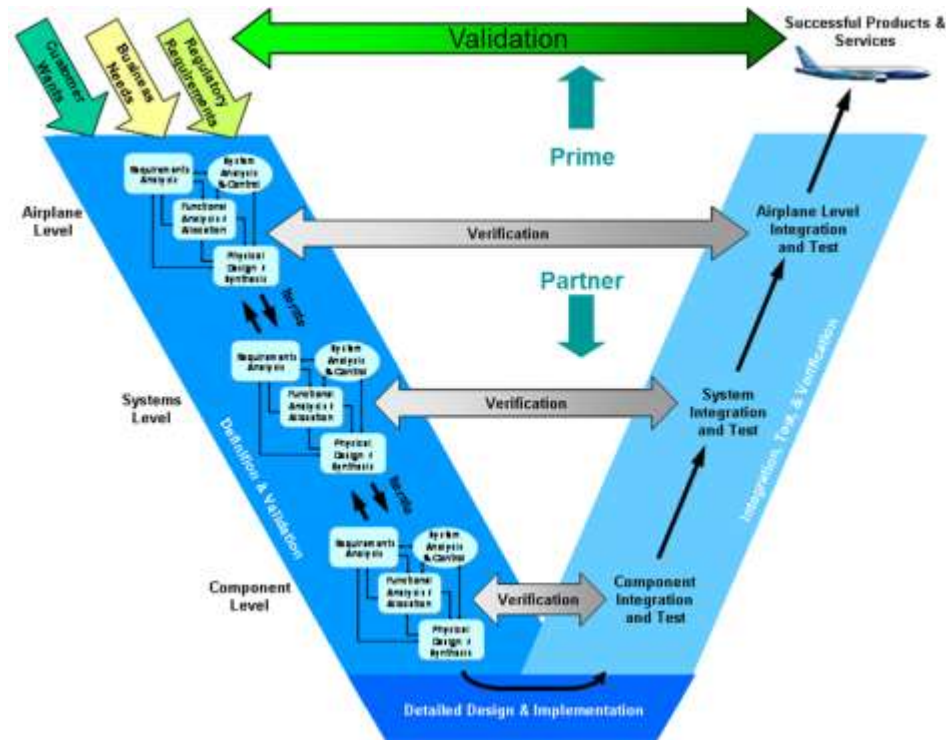
- Visibility at Program and Team Level
- Leading indicator
- Aligns with program & system integration plan
- Data flows directly from MBSE model
- Function Threads align to engineering integration plan
- Called out in V&V Plan



Notional



Challenge - Early Validation and Shifting Left



Challenge – Testers need ‘real’ requirements

Engineers need to prototype and validate requirements to define ‘real’ requirements



System Validation Metric

Validation Phases

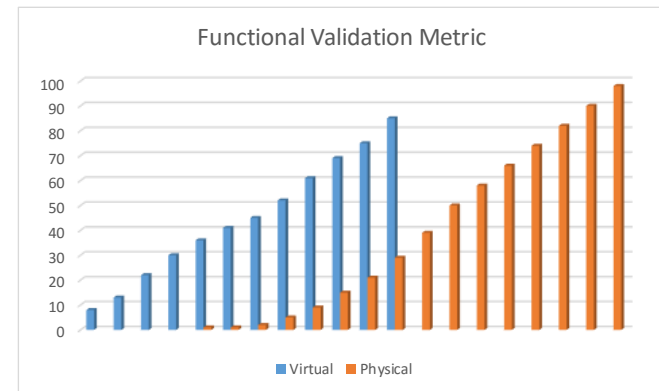
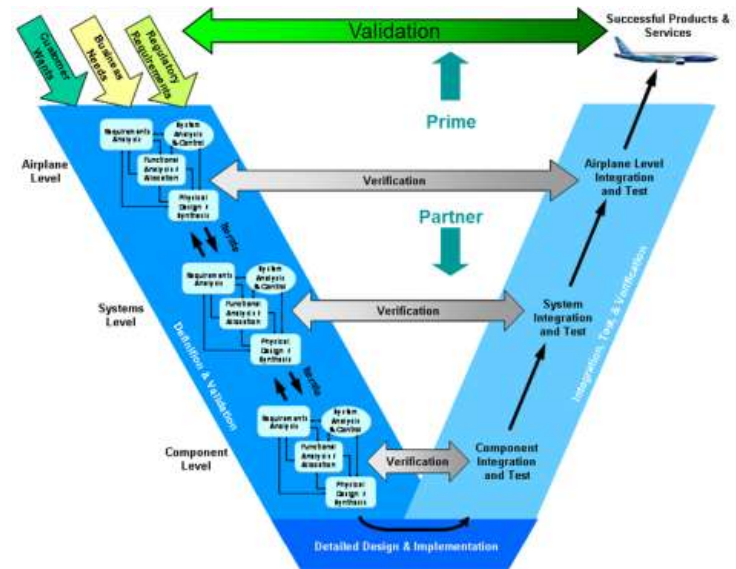
- Short cycles (< 2 months)
- Each phase builds on the previous
- Tracked at SEIT level

Virtual Validation

- Limit is what can realistically be performed virtually
- Aligns to integration plan
- Validates using functional threads

Physical Validation

- Overlaps with virtual validation
- Validates using functional threads



Early Risk Discovery and Retirement is Key to Project Execution

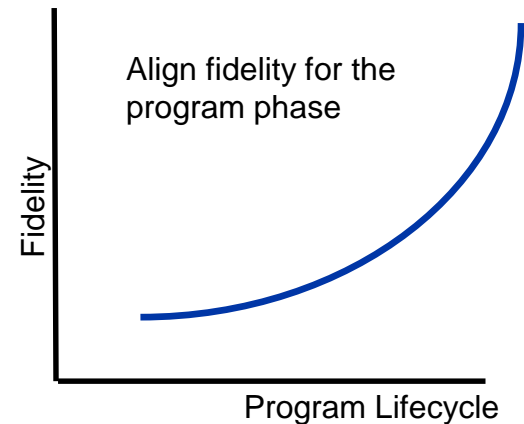
Creating the Validating Environment

Modeling & Simulation environment supports requirements development, validation and system sell off

- MBSE – Requirements source
- Desktop – Quick feedback to engineers
- SIL/HIL – Simulation to drive system integration and V&V
- Platform – Simulation to validate testing leading to formal V&V
- Environment – Simulation of external factors

Right sized Modeling

- Integrating test environment
- Mission relevant environment
- Reusable through lifecycle
- Modeling aligns with testing need
- Fidelity grows with program



Integrating Test Environment is Fundamental to Verification and Validation

Conclusions

- Mission testing ensures end to end validation
- Continuous validation and metrics helps close the integration gap
- Discover and Retire Risk Early Through Mission Thread Testing
- Schedule can be maintained
- Ensure the program will pass IOT&E
- You'll have a happy customer

Discover and Retire Risk Early

