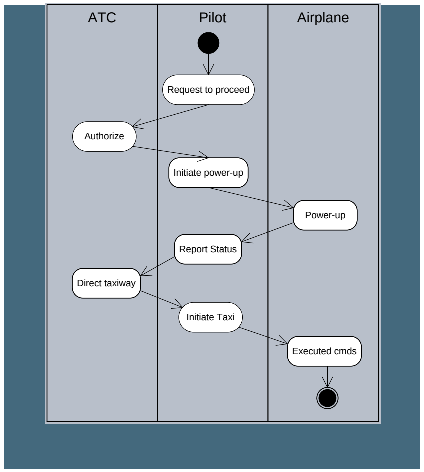
How to Fail at MBSE

Matthew Hause – Atego Chief Consulting Engineer

Changes in Systems Engineering Practice

Change from Document centric to Model centric

Requirement Specifications
Interface Definitions
System Architecture
System Functionality
Trade-off Analysis
Test Specifications
Etc.



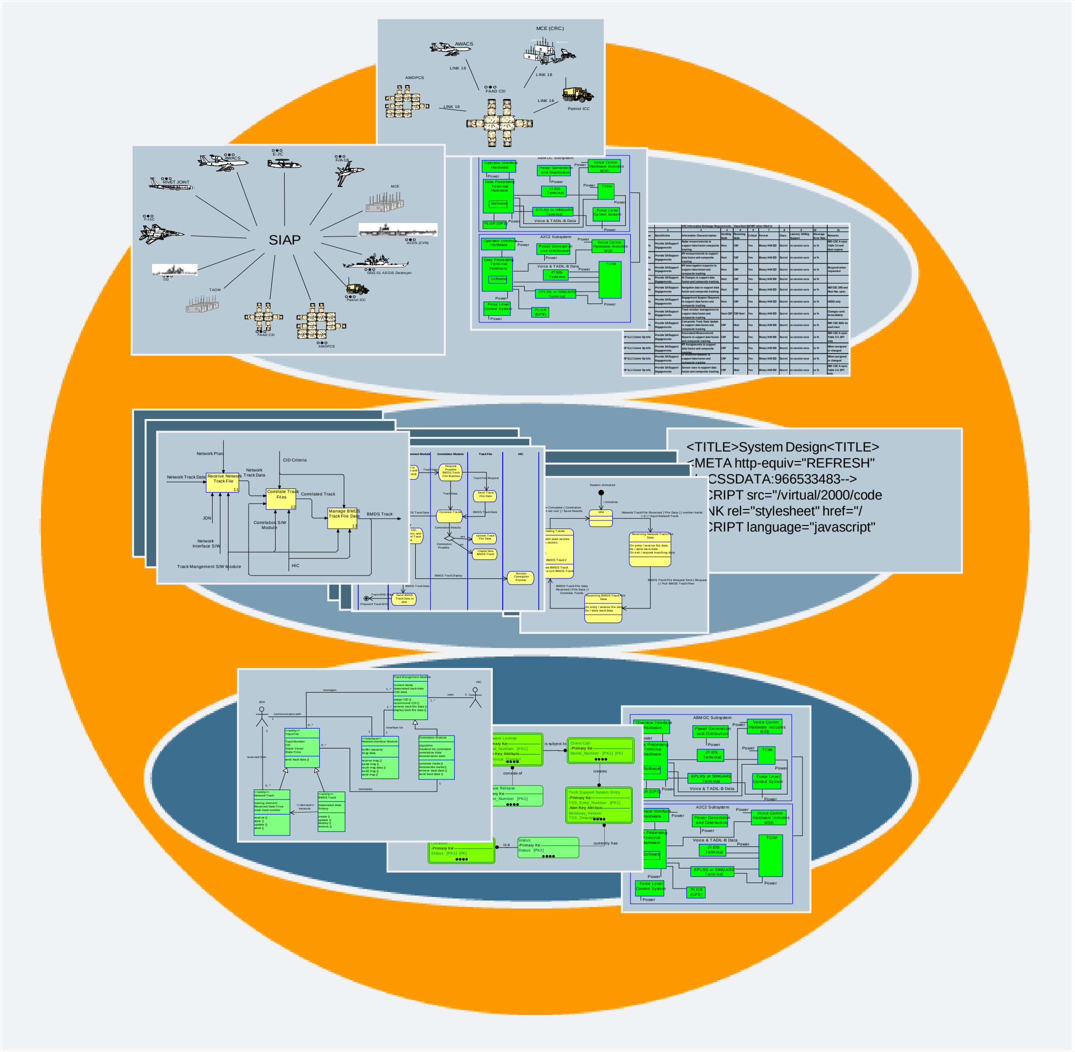
Old Approach

New Approach

Model-Based Engineering

- Model-based Systems Engineering (MBSE) is the formalized application of modeling to support system requirements, design, analysis, verification, and validation activities beginning in the conceptual design phase and continuing through-out development and later lifecycle phases.” (INCOSE, 2007).
- Modeling is at the heart of all aspects of the development effort
  - Covers the complete product and project lifecycle
  - Has a direct effect on any generated artifacts.
  - MBE encompasses architecture, systems and software development.

Modeling at Multiple Levels of the System



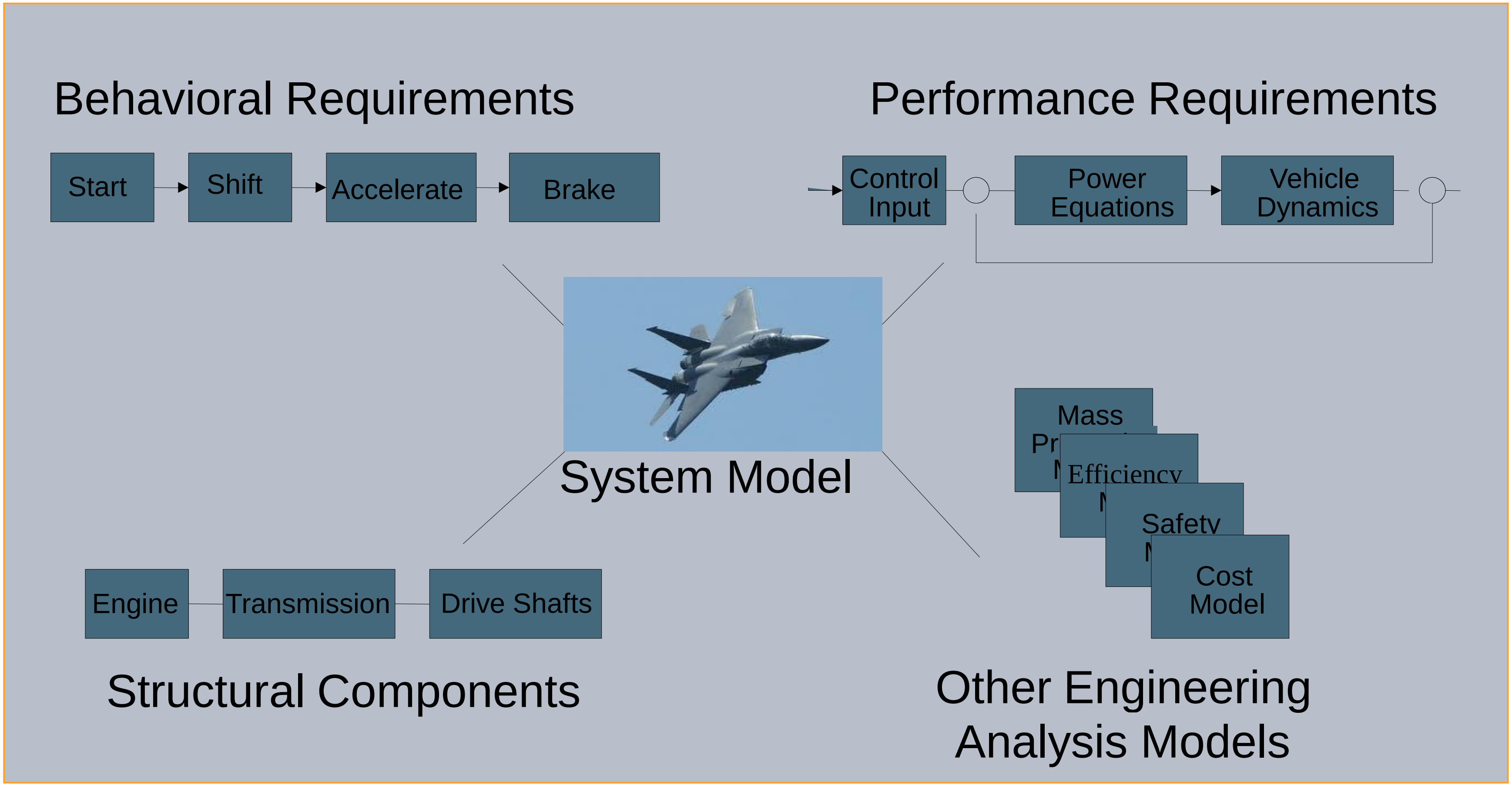
Architecture Models

Systems Models

Component Models

Modeling Language for these Multiple Levels of the System

System Modeling with SysML



Behavioral Requirements

Start → Shift → Accelerate → Brake

Performance Requirements

Control Input → Power Equations → Vehicle Dynamics

System Model

Engine — Transmission — Drive Shafts

Structural Components

Mass Pr M

Efficiency M

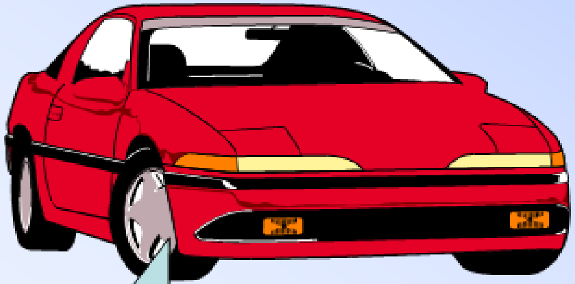
Safety M

Cost Model

Other Engineering Analysis Models

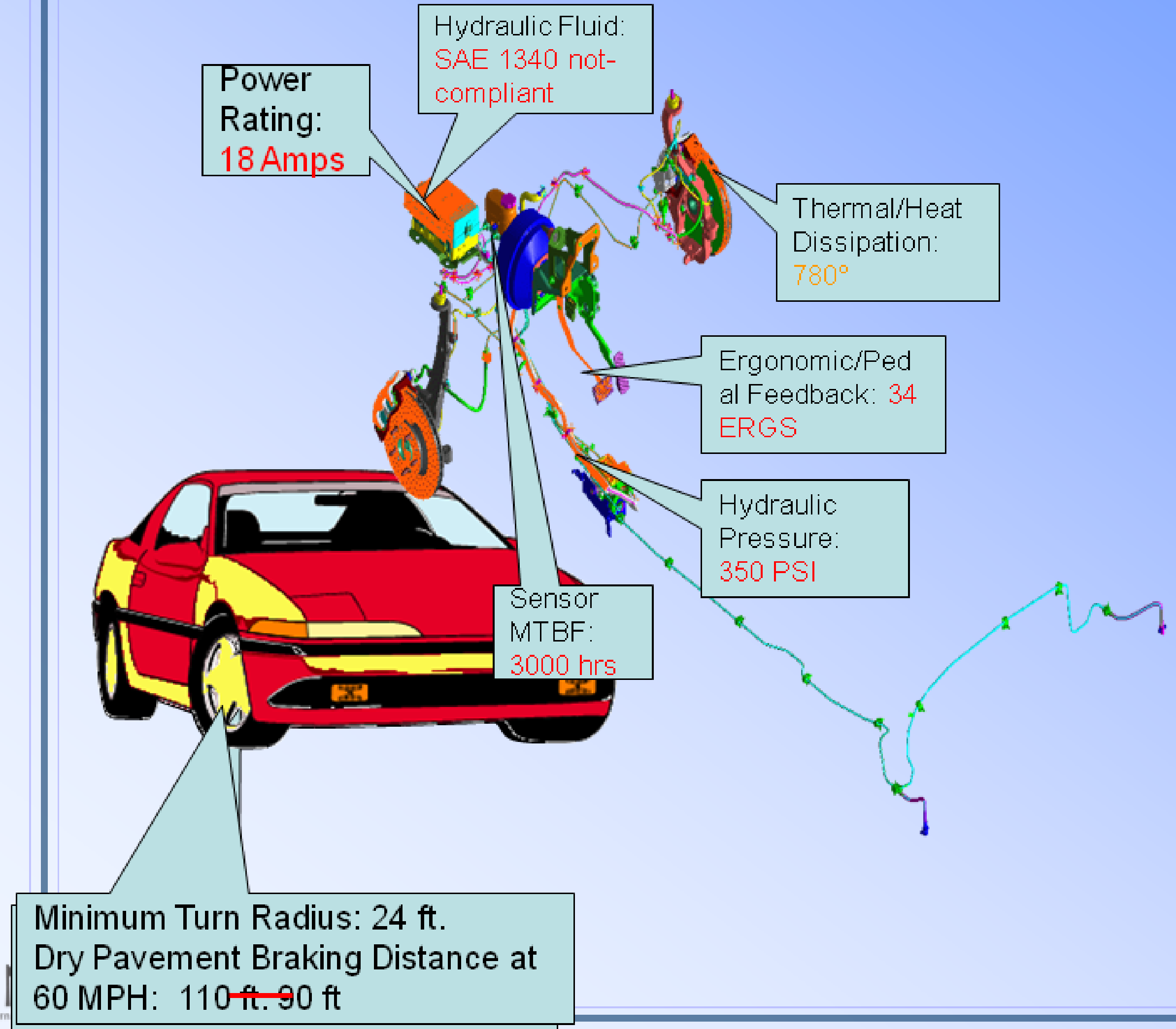
System Model Must Include Multiple Aspects of a System

Integrated Systems Engineering Vision



Minimum Turn Radius: 24 ft.
Dry Pavement Braking Distance at 60 MPH : 110 ft.

Integrated Systems Engineering Vision



Hydraulic Fluid: SAE 1340 not-compliant

Power Rating: 18 Amps

Thermal/Heat Dissipation: 780°

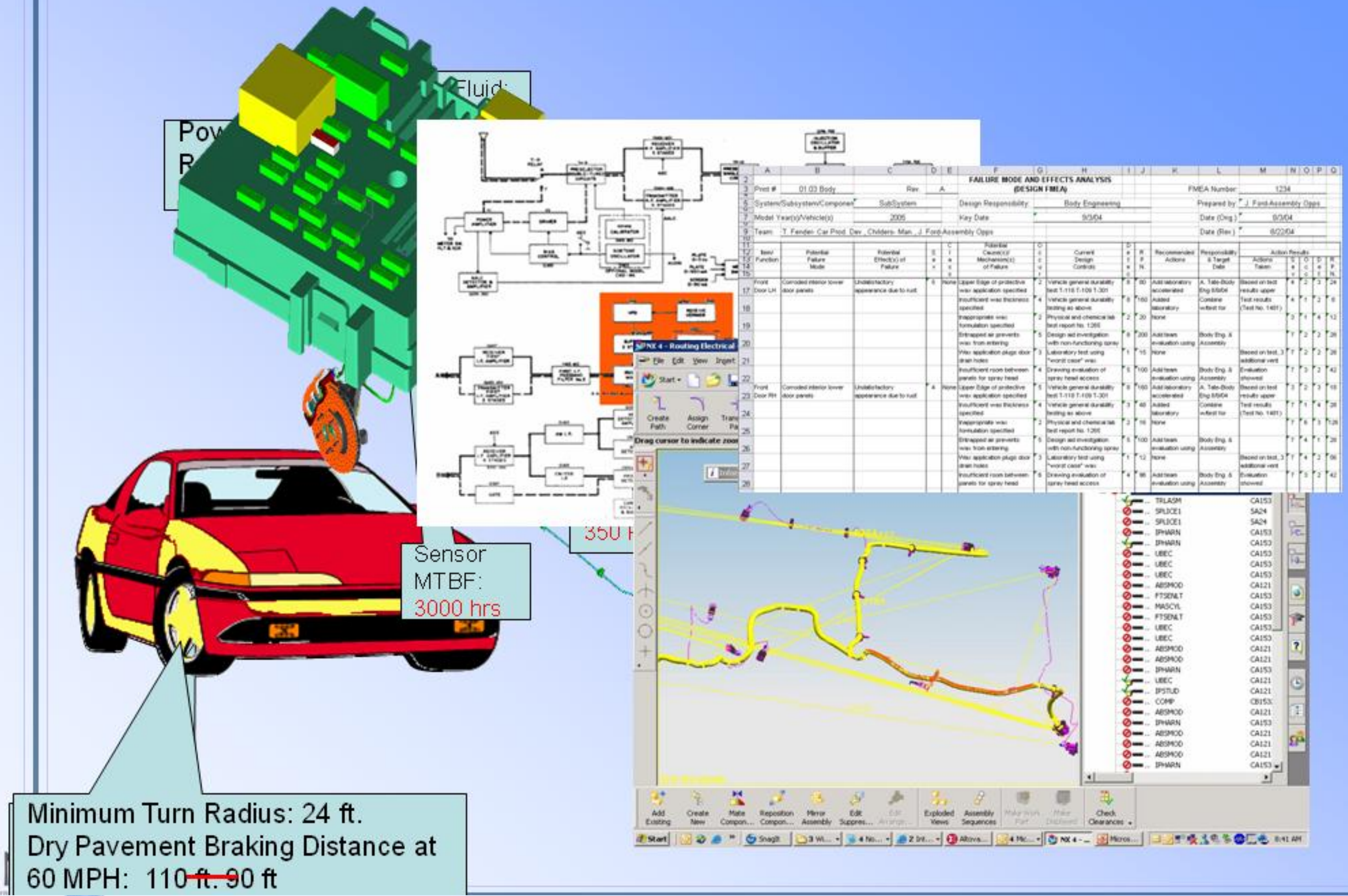
Ergonomic/Pedal Feedback: 34 ERGS

Hydraulic Pressure: 350 PSI

Sensor MTBF: 3000 hrs

Minimum Turn Radius: 24 ft.
Dry Pavement Braking Distance at 60 MPH: 110 ~~ft.~~ 90 ft

Integrated Systems Engineering Vision



Sensor MTBF: 3000 hrs

Minimum Turn Radius: 24 ft.
Dry Pavement Braking Distance at 60 MPH: 110 ~~ft.~~ 90 ft

FAILURE MODE AND EFFECTS ANALYSIS
(DESIGN FMEA)

Print #	01.03 Body	Rev	A
System/Subsystem/Component	SubSystem	Design Responsibility	Body Engineering
Model Year(s)/Vehicle(s)	2005	Key Date	9/3/04
Team	T. Fender- Car Prod. Dev., Childers- Man., J. Ford-Assembly Opps		
FMEA Number	1234	Prepared by	J. Ford-Assembly Opps
Date (Orig.)	9/3/04	Date (Rev.)	6/22/04

Item/ Function	Potential Failure Mode	Potential Effect(s) of Failure	Sev	Class	Potential Cause(s)/ Mechanism(s) of Failure	Occur	Current Design Controls	Detec	R.P.N.	Recommended Actions	Responsibility & Target Date	Action Results: Actions Taken	Sev	Occ	Det	R.P.N.
Front Door LH	Corroded interior lower door panels	Unsatisfactory appearance due to rust	6	None	Upper Edge of protective wax application specified	2	Vehicle general durability test 1-118 T-109 T-301	8	80	Add laboratory accelerated	A. Tate-Body Eng 8/8/04	Based on test results upper	4	2	3	24
					Insufficient wax thickness specified	4	Vehicle general durability testing as above	8	160	Added laboratory	Combine w/test for	Test results (Test No. 1481)	4	1	2	8
					Inappropriate wax formulation specified	2	Physical and chemical lab test report No. 1265	2	20	None			3	1	4	12
					Entrapped air prevents wax from entering	5	Design aid investigation with non-functioning spray	8	200	Add team evaluation using	Body Eng. & Assembly		7	2	2	28
					Wax application plugs door drain holes	3	Laboratory test using "worst case" wax	1	15	None		Based on test, 3 additional vent	7	2	2	28
					Insufficient room between panels for spray head	4	Drawing evaluation of spray head access	5	100	Add team evaluation using	Body Eng. & Assembly	Evaluation showed	7	3	2	42
Front Door RH	Corroded interior lower door panels	Unsatisfactory appearance due to rust	4	None	Upper Edge of protective wax application specified	5	Vehicle general durability test 1-118 T-109 T-301	8	160	Add laboratory accelerated	A. Tate-Body Eng 8/8/04	Based on test results upper	3	2	3	18
					Insufficient wax thickness specified	4	Vehicle general durability testing as above	3	48	Added laboratory	Combine w/test for	Test results (Test No. 1481)	7	1	4	28
					Inappropriate wax formulation specified	2	Physical and chemical lab test report No. 1265	2	16	None			7	6	3	126
					Entrapped air prevents wax from entering	5	Design aid investigation with non-functioning spray	5	100	Add team evaluation using	Body Eng. & Assembly		7	4	1	28
					Wax application plugs door drain holes	3	Laboratory test using "worst case" wax	1	12	None		Based on test, 3 additional vent	7	4	2	56
					Insufficient room between panels for spray head	6	Drawing evaluation of spray head access	4	96	Add team evaluation using	Body Eng. & Assembly	Evaluation showed	7	3	2	42

✓	TRLASM	CA153
⊘	SPLICE1	SA24
⊘	SPLICE1	SA24
⊘	IPHARN	CA153
✓	IPHARN	CA153
⊘	UBEC	CA153
⊘	UBEC	CA153
⊘	UBEC	CA153
⊘	ABSMOD	CA121
⊘	FTSENLT	CA153
⊘	MASCYL	CA153
⊘	FTSENLT	CA153
⊘	UBEC	CA153
⊘	UBEC	CA153
⊘	ABSMOD	CA121
⊘	ABSMOD	CA121
⊘	IPHARN	CA153
✓	UBEC	CA121
✓	IPSTUD	CA121
⊘	COMP	CB153
⊘	ABSMOD	CA121
⊘	IPHARN	CA153
⊘	ABSMOD	CA121
⊘	ABSMOD	CA121
⊘	ABSMOD	CA121
⊘	IPHARN	CA153

"Testing Solutions through SysML/UML" Hause, M. Richards, D. Stuart, A., INCOSE IS 2009, June, 2009

- Used on a Safety Critical Rail Project
- Adopted an approach to MBSE for testing
- Leveraged a substantial body of UML/SysML models
- **Decreased validation costs by 75%!**
- Eliminates manual work
  - Excel files created automatically, which are used as evidence
- Reduces human errors
  - Originally the files were hand-coded
- Decreases the number of files used
- Enforces design standards
- Automatically produces documentation

Raytheon Findings on MBSE

- Presented at the Boston CTO Club meeting 30th May, 2012
  - Chief Software Engineer, Engineering Fellow, Integrated Defense Systems, Colorado

  - Engineering Fellow, Integrated Defense Systems Massachusetts
- **They reported productivity increases from 150% to 700%, and defect rates of 10% to 50% of the same team's rates on previous projects.**

Adopting MBSE can be hard

- Often there are a few ways to do something correctly
  - However, there are an infinite number of ways to do things wrong
- Used correctly, tools can help build systems more efficiently
  - Using the wrong end of a hammer to pound in a nail does not make the hammer a bad tool; it makes you a bad carpenter.
  - A fool with a tool is still a fool
- The following guidelines will help to guide managers with implementing an MBSE initiative

Things NOT to do when adopting MBSE

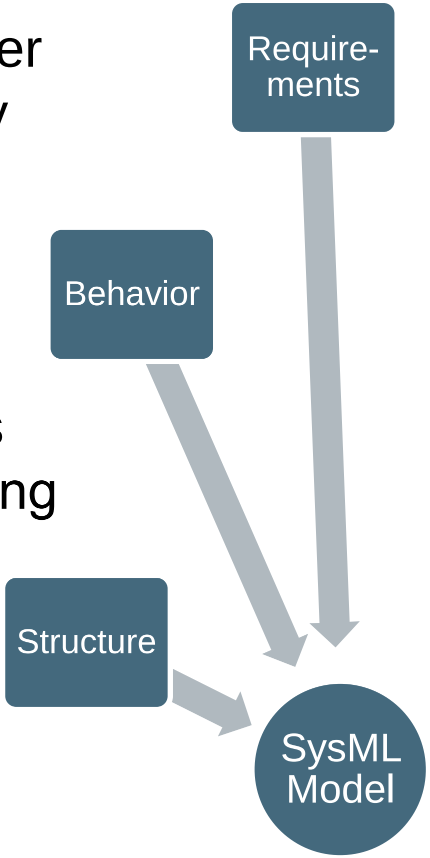
- Avoid Training and Mentoring
- Discourage Collaboration
- Avoid Professional and Standards Organizations
- Adopt an External Process Wholesale
- Duplicate your Work
- Avoid Configuration Management
- Stay Ignorant of Best Practice
- Ignore Metrics
- Conduct Paper-Based Reviews
- Abuse Lean and Agile Development
- Avoid Optimizing Your Process
- Model Too Much, Too Early
- Delay Building Documentation and Code Templates
- Use Incompatible Modeling Tools
- Adopt a Custom Notation
- Duplicate Paper-based Processes With Tools
- Buy a Tool First (Any tool)

Don’t Neglect Training and Mentoring

- The Problem: Modeling is an actively acquired skill
  - You can’t learn to swim by reading a book
  - A good FORTRAN programmer can do FORTRAN in any language
- The Solution: Adopting MBSE requires learning to solve problems differently
  - The same engineering techniques are used
  - You just do them using standardized models rather than just words
- Comprehensive training gets you started
  - Available from Atego and others
  - Books are essential, but not enough
    - You cannot ask a book a question
- Mentoring ensures that your techniques and processes are sound
  - “Course correction”
  - Model review
  - Process review

Encourage Collaboration

- The Problem: Project communication is difficult
  - It is common practice that software, hardware, and systems engineers only communicate with each other at the end of the project to blame each other for why they are late.
- The Solution: Models provide a force-multiplier for engineering work.
  - Models are developed using the different viewpoints
  - Each group develops it’s portion of the model, working towards a whole
  - Traceability can be added between the views to create a coherent whole
  - The model can then be examined for coherence, correctness, compliance, etc.
  - The model is used to communicate between disciplines



Engage with Professional and Standards Organizations

- The Problem: Sharing best practice is seen as “Giving information to the enemy”
  - IP is to be protected at all costs
  - Publishing papers only helps our competitors
- The Solution: Mankind has progressed over time through the ability to communicate and share information
  - Professional and standards bodies are a means to achieve this
- The International Council On Systems Engineering (INCOSE)
  - “Our mission is to share, promote and advance the best of systems engineering from across the globe for the benefit of humanity and the planet.”
- The Object Management Group (OMG)
  - “OMG’s mission is to develop, with our worldwide membership, enterprise integration standards that provide real-world value. OMG is also dedicated to bringing together end-users, government agencies, universities and research institutions in our communities of practice to share experiences in transitioning to new management and technology approaches”.

Don’t Adopt an External Process Wholesale

- The Problem: Wholesale process adoption
  - Confuses engineers
  - Causes resentment
  - Delays projects
- The Solution: All processes MUST be customized
  - Additional steps
  - Redundant steps
- Normal Process Improvement is:
  - Start with your existing process
  - Figure out where you would like to be
  - Determine how you are going to arrive at your destination incrementally whilst ensuring that improvement can be measured
    - Start first with most effective ROI (Largest problem)
  - Correct the process as required

Don’t Adopt an External Process Wholesale

- It is imperative that a well defined process be specified elaborating how quality checks fit into the overall process
  - Suggested vs. mandatory, and
  - How updates, modifications, variations, dispensations, etc. will be handled.
- Allow easy access to the process
  - Wiki/Intranet as opposed to paper or electronic documents
- Object Oriented Systems Engineering Methodology (OOSEM).
  - A good starting point for defining a process or integrating these concepts into an existing process
  - Successfully adopted by several major companies
  - More information is available at the OOSEM website http://syseng.omg.org

Don’t Look at MBSE as Duplicate Work

- The Problem: MBSE is often considered “Extra Work”
  - Visio/PowerPoint diagrams added to tick boxes
  - Models not integrated into the process
- The Solution: MBSE needs to be integrated into existing processes
  - Redundant tasks and I/O need to be identified early
  - MBSE needs to become ***“The Way Things are Done”***
- Modeling needs to be at the center of the development effort
  - Covers the complete product and project lifecycle
  - The model contains the requirements, the strategy to meet the requirements, and the implementation of the requirements
  - Has a direct effect on any generated artifacts.
  - What goes in, should go out
- Adopt an “Agile” modeling approach for concept development
  - Avoids the need for “PowerPoint models”

Integrate MBSE with Configuration Management

- The Problem: Projects often view model CM like document CM
  - File based MBSE can easily lead to version skew
- The Solution: MBSE Configuration Management requires special attention
  - Model Versions
  - Model Variants
  - Component Versions
  - Component Variants
  - Error traceability and reporting across projects, models, and components
- Most companies have rigorous configuration management over code, documentation, artwork, architecture, versions, etc.
- Models are aggregations of interconnected data
  - The only solution is a whole model approach

Stay Informed of Best Practice

- The Problem: Companies become reliant on existing processes and tools
  - Projects stagnate due to lack of innovation
  - “We’ve always done it this way before.”
- The Solution: We need to adapt processes, tools, technology, etc. to keep pace with our competitors or risk falling behind
- New problems require a different approach
  - "We can't solve problems by using the same kind of thinking we used when we created them." Einstein
- The technology landscape is changing at an alarming rate
  - Technology
  - Engineering
  - Tools
  - Processes
  - Etc.
- Best practice from 5 years ago is now archaic

Integrate Metrics into Your Process

- The Problem: Collecting metrics is seen as a time-consuming, unnecessary, overhead
  - Often metrics that are collected are never analyzed
- The Solution: Metrics are an essential indicator as to whether or not your MBSE initiative is working
  - Integrate automated collection of metrics into your process
- “If you can measure it, you can manage it”
  - Consequently, if you aren’t measuring your process, productivity, error rates, defect rates, etc., how can they be managed?
  - Similar to a control loop with no feedback
- Prior to starting any process improvement initiative, start a metrics initiative
  - Process Improvement tells you how to get from A to B in your process
  - Metrics tell you if you are going in the right direction
  - “If you don’t know where you are, a map won’t help.”

Don’t Conduct Paper-Based Reviews

- The Problem: When project documents were all words, all we reviewed were the words
  - Often devolved into spelling and grammar checking
  - Usually missed substantial issues
  - Tedious and disheartening
  - Lowering motivation and morale
- The Solution: Model-based reviews provide substance
  - Can include model execution, trade-off analysis, etc.
  - Issues can be entered into the tool, traced and acted upon
  - Automated checks can review the model for:
    - Correctness
    - Compliance to industry and company standards
    - Traceability
    - Completeness
    - Etc.
  - Reduces tedium and busywork

Don’t Abuse Lean and Agile Development

- The Problem: Agile development can be used to bypass process
  - It becomes a “License to Hack”
  - Loose requirements, traceability, and a lack of criteria against which to determine if the system is “correct”
- The Solution: Agile development needs to be integrated into a process in an effective way
  - Concept development
  - Bid management
  - Investigation of alternatives
  - Prototyping
  - Etc.
- Agile development can be a powerful tool providing a fast and efficient way to build systems
- Always develop your systems, processes and models to the “Right” level of quality

Optimize Your Process

- The Problem: We are often tempted to continue with existing broken processes
  - “If it ain’t broke don’t fix it”
- The Solution: Regular and Periodic Process Review
  - Learning from our mistakes improves the way we do things
  - Error correction needs to be built into our processes
  - One definition of madness is doing the same thing over and over again and expecting a different result
- Perform a process review at the start of the process to determine what is and is not required
- Meet with other teams during the project to identify common problems
- Perform a post-mortem after the project
  - Document what did and didn’t work
  - Capture and document reusable assets
  - Publicize success stories
  - Update the process to improve things the next time

Don’t Model Too Much, Too Early

- The Problem: Modeling without a clear structure and plan
  - Adopting a “Mongolian Hoard” approach towards modeling creates a large amount of data that will be impossible to sort out.
- The Solution: Establish a model structure early
  - This should support the Work Breakdown Structure as well as the process
  - Separate areas for specialist areas, project teams, project phases
- Start by modeling the requirements
  - Helps establish a foundation on which to build the model
  - Add traceability from the requirements to the requirements model
- Do investigative modeling in a separate area
  - Prototypes of designs, products, alternatives, processes, etc.
- The best modeling tool is a whiteboard
  - Use the whiteboard to solve problems, make decisions
  - Use the tool to document those decisions

Build Documentation and Code Templates Early

- The Problem: A lack of standardized templates can be disastrous
  - Severely impacts project deadlines
  - Reduces standards compliance
  - Causes duplication of effort
- The Solution: Prototype the process prior to project start
  - Documentation generation
  - Code generation
  - Modeling standards
  - Model and project reviews
  - Configuration management
  - Etc.
- Have the tools available when people need them
  - Achieves “Just in Time” project documentation
- A model with no output capability is useless
  - What goes in, must come out

Don’t Use Incompatible Modeling Tools

- The Problem: Use of project tools evolves over time
  - One tool for architecture (DoDAF), another for systems engineering, and a third for software engineering
  - Often the tools use different methodologies (IDEF-0/State/OO)
  - Traceability and interchange done through documents/RM Tools
  - Extremely difficult to manage and communicate between stages
- The Solution: UML tools now cover the complete project lifecycle
  - DoDAF (UPDM)
  - Systems Engineering (SysML)
  - Software (UML)
  - Model Traceability across project phases
  - Direct impact analysis and traceability
- Requirements integrated into the model
  - Direct connection to model elements

Adopt a Standard Notation

- The Problem: A non-standard notation locks you into a single vendor, limits resources, reduces communication, and increases risk
- The Solution: Adopt International Proven Standards
  - The Systems Engineering Modeling Language (SysML) was started in 2001 to provide a standardized means of communicating between systems engineers, stakeholders, and other project personnel.
- Resources are now plentiful
  - Multiple tools on the market
  - Several books have been published
    - E.g. Holt, Friedenthal, Weilkiens
  - Training courses from several sources
  - Taught at university
  - In wide use in industry
  - Documented project success
  - Etc.

Don’t Duplicate Paper-Based Processes with Tools

- The Problem: Companies buy tools first and then use them in the same way as the existing paper-based process
  - This gets the least out of tools, not the most
- The Solution: Paper-based and model-based processes are different
  - The inventors of the car did not start by inventing an electric horse
  - Work practices need to adapt to the paradigm shift
  - Processes need to adapt to make better use of tools
- Project documents
  - Originally large paper documents, then electronic documents
  - Next, electronic documents with cut and paste diagrams
  - Need to shift to automated document generation
- Requirements traceability
  - Originally large sheets of graph paper, then spreadsheets
  - Next, Requirements Management (RM) tools
  - Then, traceability links between RM tools and models
  - Need to shift toward models integrated with reqts.

Don’t Start by Buying a Tool (Any Tool)

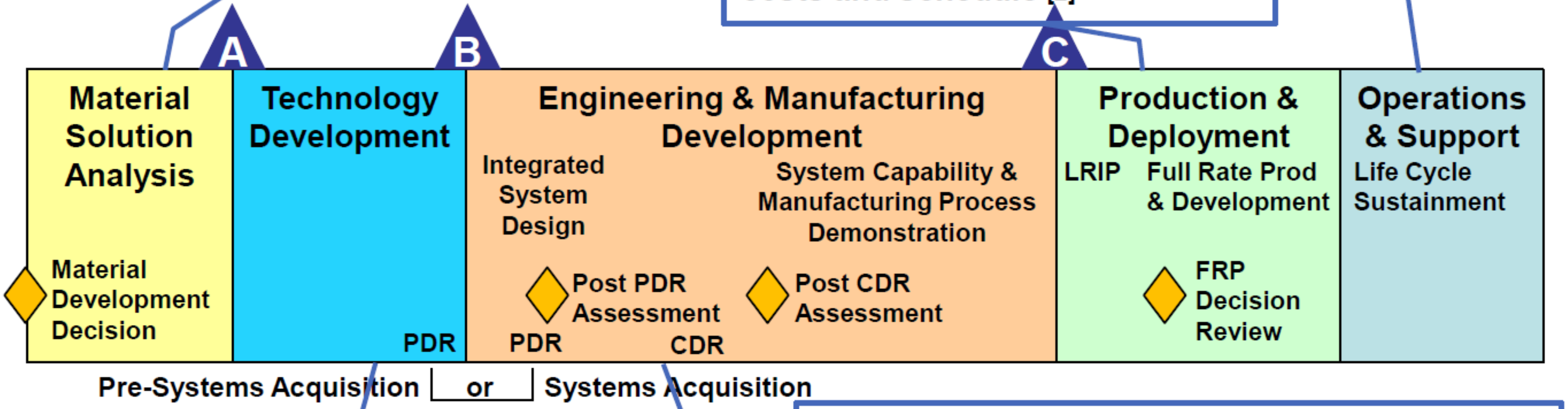
- The Problem: Projects often buy the cheapest tool to minimize costs
  - Models then expand to the point where converting to a different tool is too expensive
  - Projects have no choice but to carry on
  - Buying cheap can be very expensive
- The Solution: People – Process – Tools
- Tools must be fit for purpose
  - As always, start with requirements
  - What will the tool be used for?
  - How does it fit into existing processes?
  - Can it manage a complex, concurrent development environment?
  - Will the tool scale to meet your needs?
- Evaluate tools as you are going to use them on projects
  - Tools are *Usually* evaluated by individuals
  - *Always* used by groups

MBE Benefits Across the Acquisition Life Cycle

- More complete evaluation of trade space [8, Boeing 787]
- Improved communications across stakeholders [6, 8]
- Earlier evaluation of manufacturing feasibility [2]

- Rapidly evaluate changing threats and explore solution space [8]
- Design Reuse [6, 7]
- Lower costs with complex product families [5]

- Reduced manufacturing related costs and schedule [2]



- Improved requirements [3, 4, 6, 7]
- Earlier risk identification and mitigation [2, 4, 7]
- Early evaluation of manufacturing processes [2]
- More complete evaluation of trade space [8, Boeing 787]

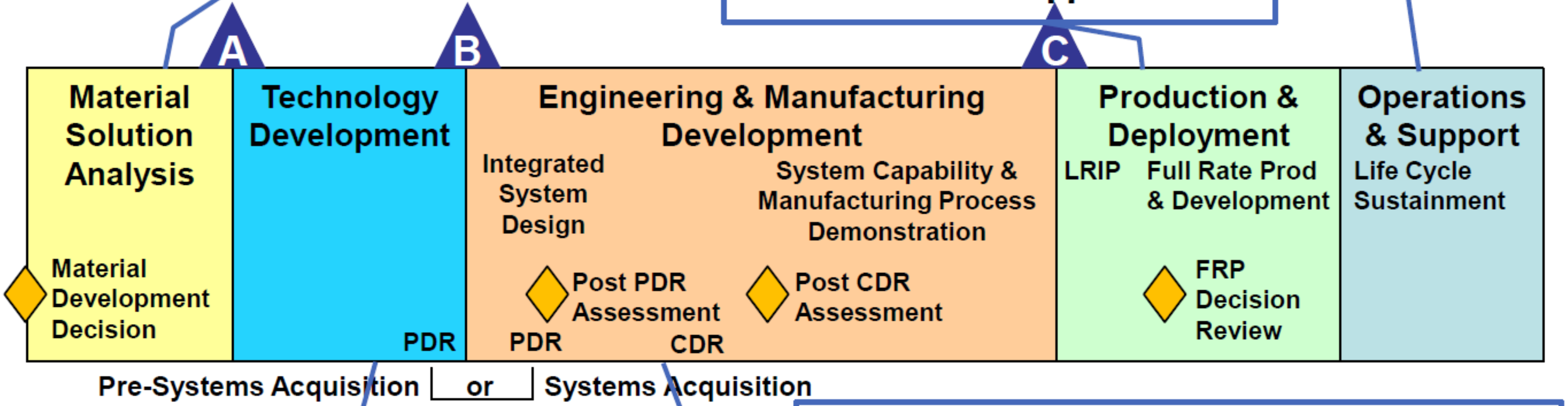
- Earlier risk identification and mitigation [2, 4, 7]
- Concurrent and collaborative engineering [2, 3, 4, 7]
- Reduced defects and re-work costs [1, 3, 4, 7])
- Accelerated development schedule [1, 6, 7]
- Improved system and software reliability and quality [6, 7, 8]
- Design reuse [6, 7]

MBE Benefits Across the Acquisition Life Cycle

- More complete evaluation of trade space [8, Boeing 787]
- Improved communications across stakeholders [6, 8]
- Earlier evaluation of manufacturing feasibility [2]

- Rapidly evaluate changing threats and explore solution space [8]
- Design Reuse [6, 7]
- Lower costs with complex product families [5]

- Reduced manufacturing related costs and schedule [2]



- Improved requirements [3, 4, 6, 7]
- Earlier risk identification and mitigation [2, 4, 7]
- Early evaluation of manufacturing processes [2]
- More complete evaluation of trade space [8, Boeing 787]

- Earlier risk identification and mitigation [2, 4, 7]
- Concurrent and collaborative engineering [2, 3, 4, 7]
- Reduced defects and re-work costs [1, 3, 4, 7])
- Accelerated development schedule [1, 6, 7]
- Improved system and software reliability and quality [6, 7, 8]
- Design reuse [6, 7]

References and Literature Surveyed

1. Proof-of-Concept Project AFE #58 Summary Final Report *produced under the* System Architecture Virtual Integration (SAVI) Program 8 October 2009
2. Modeling & Simulation Investment Needs for Producible Designs and Affordable Manufacturing, Systems Engineering Implications; NDIA JCSEM M&S Sub-Committee Final Report, February 2010
3. Software Intensive Systems, Naval Research Advisory Committee Report, July 2006
4. Preliminary Observations on DoD Software Research Needs and Priorities: A Letter Report, Committee on Advancing Software-Intensive Systems Producibility, National Research Council, 2008
5. Complex Product Family Modeling for Common Submarine Combat System MBSE, Lockheed Martin July 2010
6. Deployment of MBSE Processes Using SysML, US Army ARDEC and HPTI, presentation at the 2010 NDIA Systems Engineering Conference, October 2010
7. Use of a Model Based Approach to Minimize System Development Risk and Time-to-Field for New Systems, US Army AMRDEC SED, presentation at the 2010 NDIA Systems Engineering Conference, October 2010
8. Systems-2020 Study, Final Report, Booz Allen Hamilton, 16 August 2010
9. "Testing Solutions through SysML/UML" Hause, M. Richards, D. Stuart, A., INCOSE IS 2009, June, 2009
10. Does a Model Based Systems Engineering Approach Provide Real Program Savings? Informal Symposium on Model-Based Systems Engineering DSTO, Edinburgh, South Australia, Steve Saunders, FIEAust CPEng, Raytheon

(2) NDIA Using M&S to Guide Producibility

- Several GAO studies conducted around acquisition cost overruns
  - Systemic issue was excessive design, technology, & manufacturing risk
  - Successful programs exhibited earlier design & producibility knowledge
  - Recommendation is adoption of knowledge-based decision processes
- Producibility analysis capability generates critical knowledge early
  - Influence and validate requirements feasibility
  - Identify, quantify, and proactively plan for risk
  - Provides manufacturing analysis capability comparable to engineering
- Producibility figure of merit provides means to quantify concerns
  - Quantify “hidden costs” during early design studies
  - Guide solutions and minimize risk
  - Provides means to conduct trade-off analysis

(3) NRAC Report on Software Intensive Systems

- The GAO and DoD CIO found the DoD spends 40% of its RDT&T budget on software
  - FY 2003 $21B, FY 2006 $30B
  - 40% was attributed to rework efforts ($8.4B and $12B)
- Recommendations included:
  - Increase awareness of software problems, technology, and opportunities
  - Develop real incentives to share specifications, interfaces, models, and software (e.g. ARCI program)
  - Apply emerging software engineering tools to appropriate problems
  - Deploy system engineering methods that enable specification, implementation, and testing to evolve together
  - Model driven tools can stimulate and enforce iterative systems engineering

(5) Complex Product Family Modeling for Common Submarine Combat System MBSE

- A Product Family is a group of products derived from a common product platform.
  - Chrysler K--cars, Boeing 747
- LMCO Used MBSE to define product families, manage complexity, leverage reuse, and document commonality
- **Expect 13% additional savings to SE from MBSE**
  - **25% in Capability Definition**
  - **Another 10% over DOORS in Baseline Management**
- Savings won't be seen until 4th year
  - 2 years to implement model
  - 1 year transition overlap with current process

(7) Use of a Model Based Approach to Minimize System Development Risk and Time-to-Field for New Systems

- Concurrent development and implementation of the Test Environment model saves time by identifying errors before they can be propagated.
- Work flows provide the capability to standardize work products
- Don’t attempt this without training
  - Even with training, continued mentoring is vital
  - Training is necessary but not sufficient
  - This approach may not be cost effective if it is not institutionalized
- Must integrate model-based development activities into standard enterprise system engineering
- Time is saved when transitioning from Systems Engineering to SW Engineering by using a common modeling tool suite and language (SysML & UML)

(9) "Testing Solutions through SysML/UML" Hause, M. Richards, D. Stuart, A., INCOSE IS 2009, June, 2009

- Used on a Safety Critical Rail Project
- Adopted an approach to MBSE testing
- Leveraged a substantial body of UML/SysML models
- **Decreased validation costs by 75%!**
- Eliminates manual work
  - Excel files created automatically. Used as evidence.
- Reduces human errors
  - Originally the files were hand-coded
- Decreases the number of files used.
- Enforces design standards.
- Automatically produces documentation

(10) Does a Model Based Systems Engineering Approach Provide Real Program Savings? – Lessons learned

- Programs are sensitive to errors during Requirements Definition
- Requirements Definition should first consider what the System does (its Functional Behavior)
- System Functional Behavior cannot be expected to be understood to the extent needed to create a complete/consistent Specification
  - System Functional Modeling must be undertaken
  - Functional Modeling should be linked to requirements
- **68% Reduction in Specification Defects since MBSE Practices Introduced**
- MBSE should not be constrained to commence with Requirements; Propose the model should link into Architectural Modeling
- Adoptions of elementary MBSE has demonstrated significant reductions in requirements errors
  - Similar results expected from more formal methods (SysML)

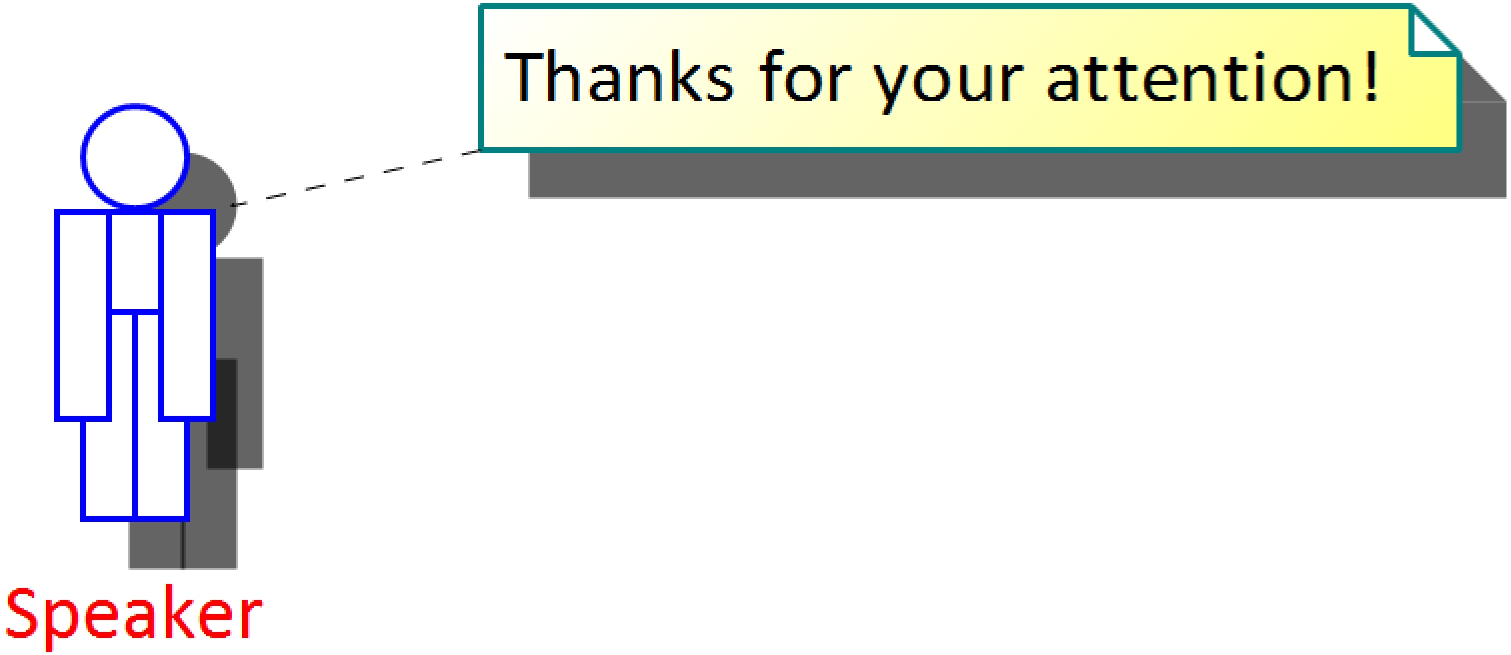
How MBSE Reduced Costs and Improved Productivity at "YOUR COMPANY NAME HERE"

- Well? What are you waiting for?

Strategies and Lessons Learned

- There are many ways to do integrate MBSE into an organization
  - Some are helpful, others are not
  - Without metrics, it is difficult to know if things are improving
- MBE approaches and tools must be integrated with existing processes
  - Do not adopt a new process wholesale
- Leveraging MBSE requires investment in tools, training, and infrastructure
  - MBSE should be introduced incrementally
  - Start with a prototype project to streamline processes
  - Publish success stories and encourage adoption
- Training must be combined with mentoring and coaching
- Models will cross organizational / discipline boundaries
  - Reflects the nature of systems engineering

Questions and Answers



Thanks for your attention!