Automotive Software Systems Complexity: Challenges and Opportunities

ONE FORD
ONE TEAM • ONE PLAN • ONE GOAL

INCOSE International Workshop

MBSE Workshop
January 26th-28th, 2013

Christopher Davey
Senior Technical Leader
Software & Control Systems Engineering
Global EE Systems Engineering Group
Ford Motor Company

Go Further

Presentation Outline:

1. Ford Motor Company & EE Systems

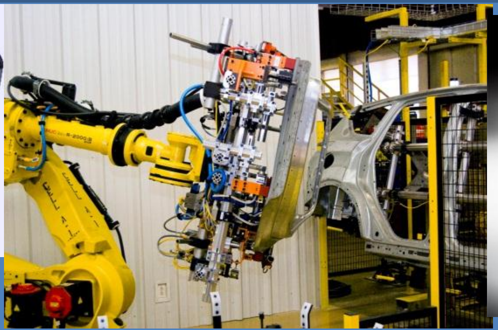
2. Automotive EE & Software Complexity Challenge and Opportunity

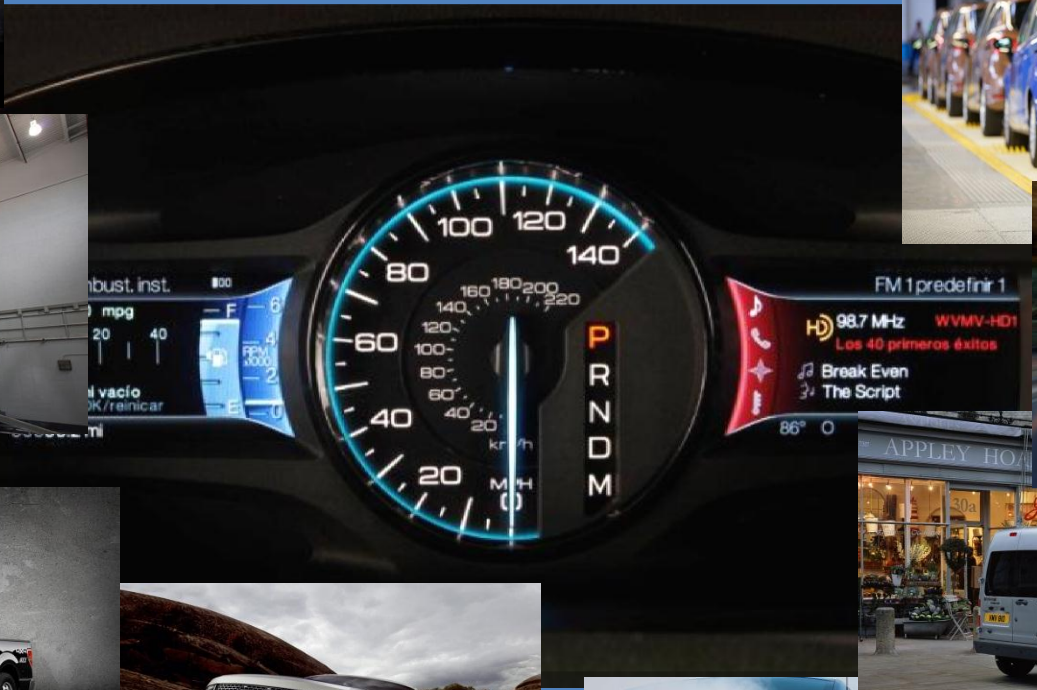
3. A Paradigm Shift: Fully Integrated Systems Engineering
    1. MBSE
    2. Model Based Architectures
    3. Why, When, Where & How

4. Model, Information, Intellectual property re-use: The need for enterprise wide - PLM/ALM Solution

5. Lessons Learned so far…..

Ford Motor Company

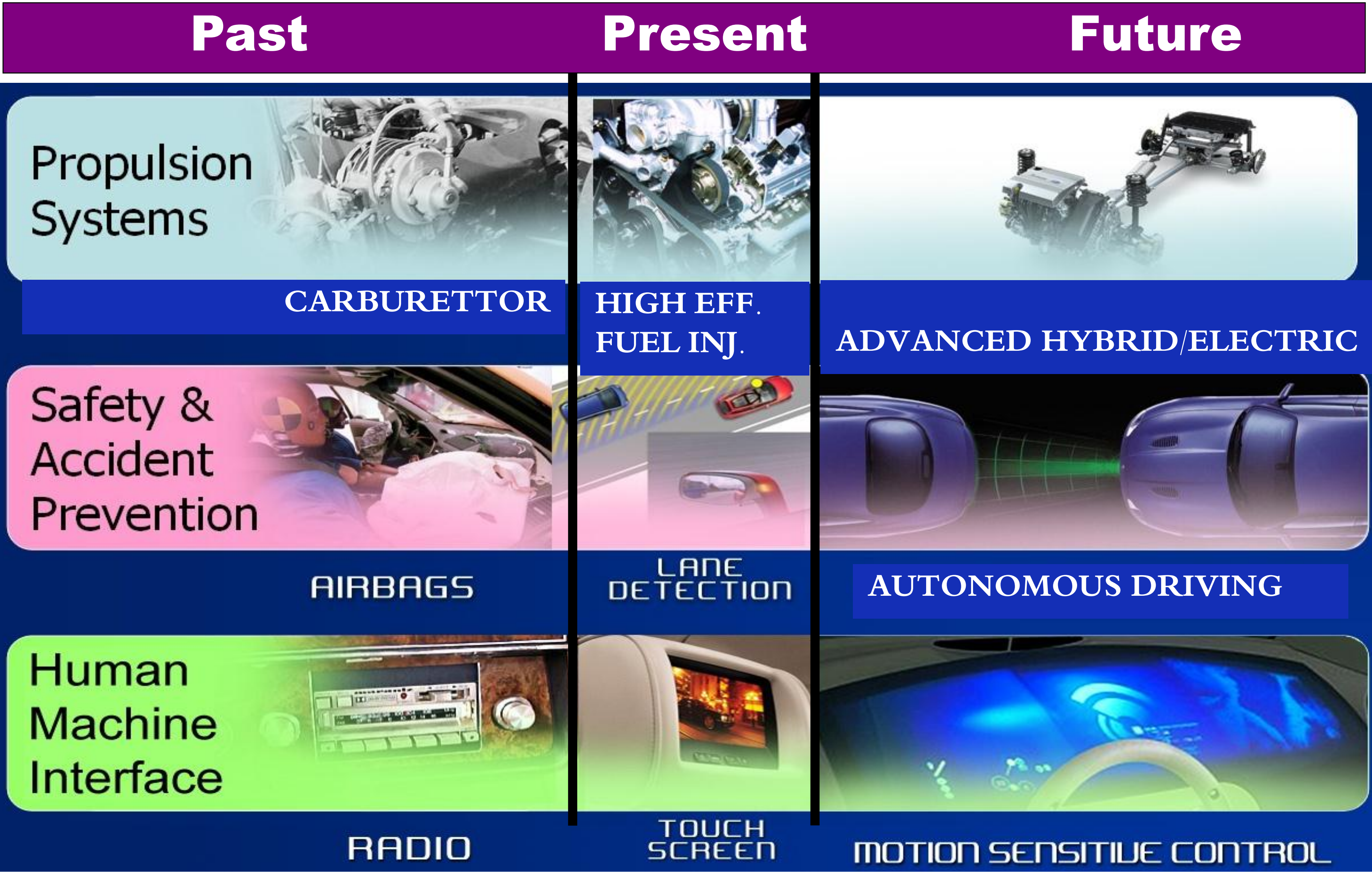








- All Photographs are courtesy of Ford Motor Company

Why do we need all these software Systems?



- **Technology Progression leverages Vehicle Software & Control Systems -> Proliferation**

Properties of EE Software Systems

ONE FORD
ONE TEAM • ONE PLAN • ONE GOAL

*Courtesy of NASA

Space Shuttle (Ref. 1)
- 5 Computers on board
- 700 kByte Software
- 500.000 LOC/Instructions

*Courtesy of Boeing

Boeing 777 (Ref. 2)
- Approx. 3 Million LOC

*Ref: Les Hatton

Enhanced
Vehicle Software and
Electrical Management

Capabilities are required
to maintain the

Robustness
and
Quality

of
Electrical Hardware
Software and
Control Systems

as

Feature Quantity &
Complexity Grows

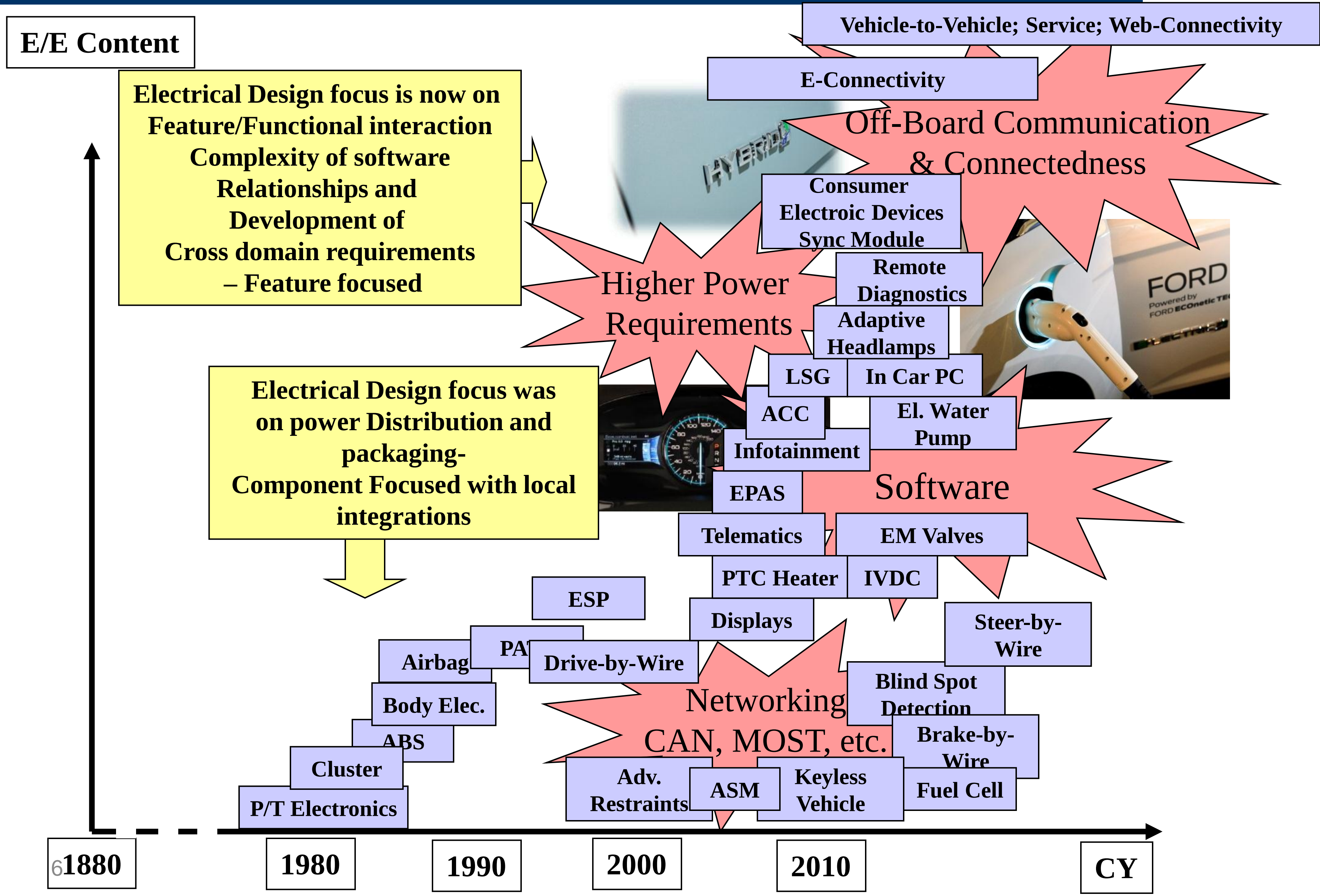
- > 80% of new features are Software enabled EE systems.
- > 70% of new systems are distributed in nature

CD Class Vehicle has approx.
- 50-70 Computers on board with multi-core SW process management
- > 10 MByte of Control Software
- >15 Million LOC/Instructions
- > 5,000 Software Parameters
- > 50,000 functional requirements
- > 1,000,000 pages of specifications
- > 10,000 buildable Vehicle-Series-Variants (based on ECU component permutations per Vehicle line)

⇒ In some aspects a CD class vehicle has higher system, software and build complexity than a commercial aircraft....

General Trends in Vehicle Electronics

ONE FORD
ONE TEAM • ONE PLAN • ONE GOAL



HMI and Off-board Distributed Functionality

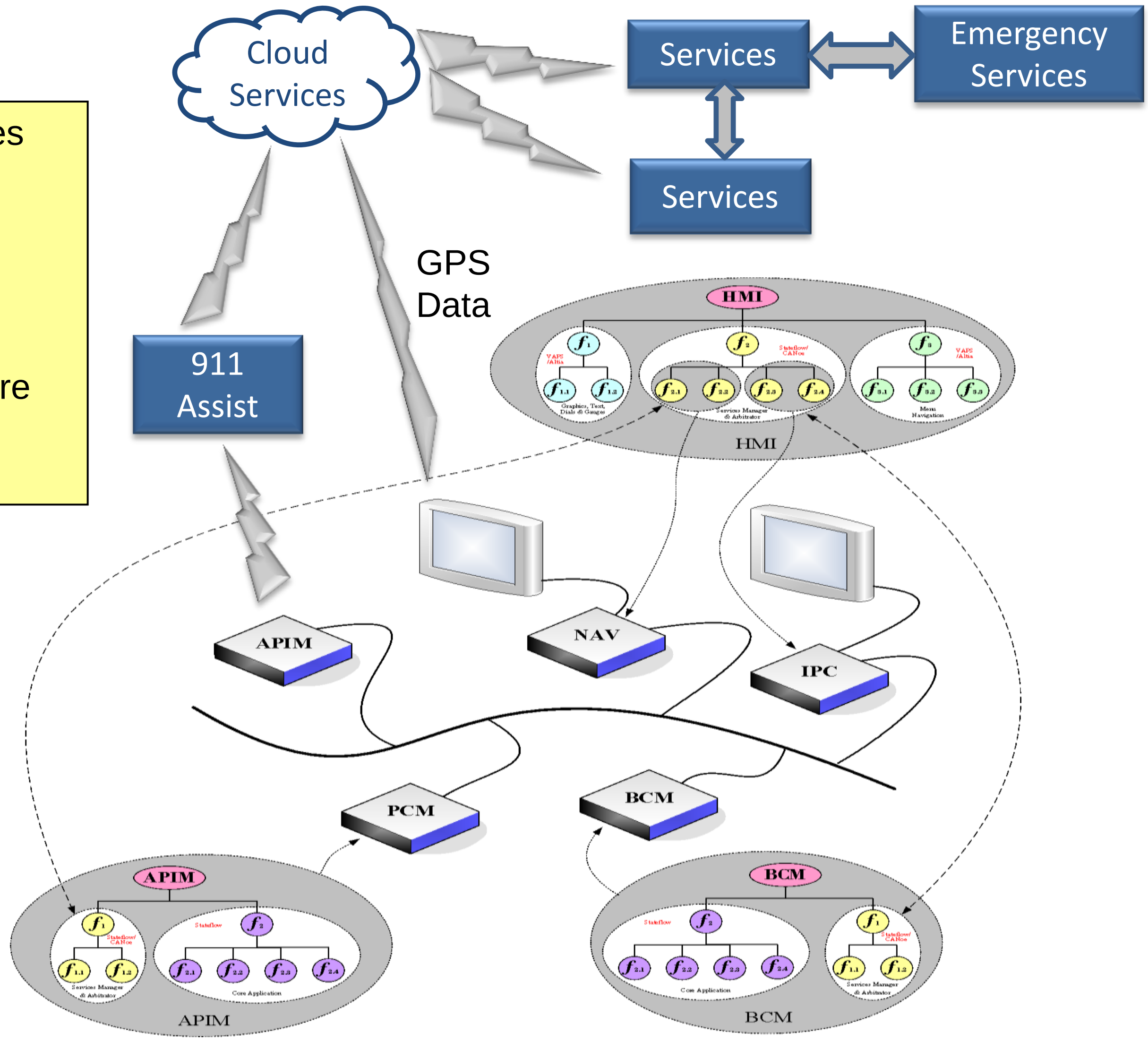
ONE FORD
ONE TEAM • ONE PLAN • ONE GOAL

- Delivery of customer features requires the coordinated execution of multiple modules exchanging signals both wirelessly and over a number of multiplexed networks.

- >70% of all new customer features are software enabled and distributed in nature.

- Service Oriented Architectures (SOA) and well defined functional/software architectures are required to deliver efficient coordination between modules.

Human Machine Interface & Off-board services example

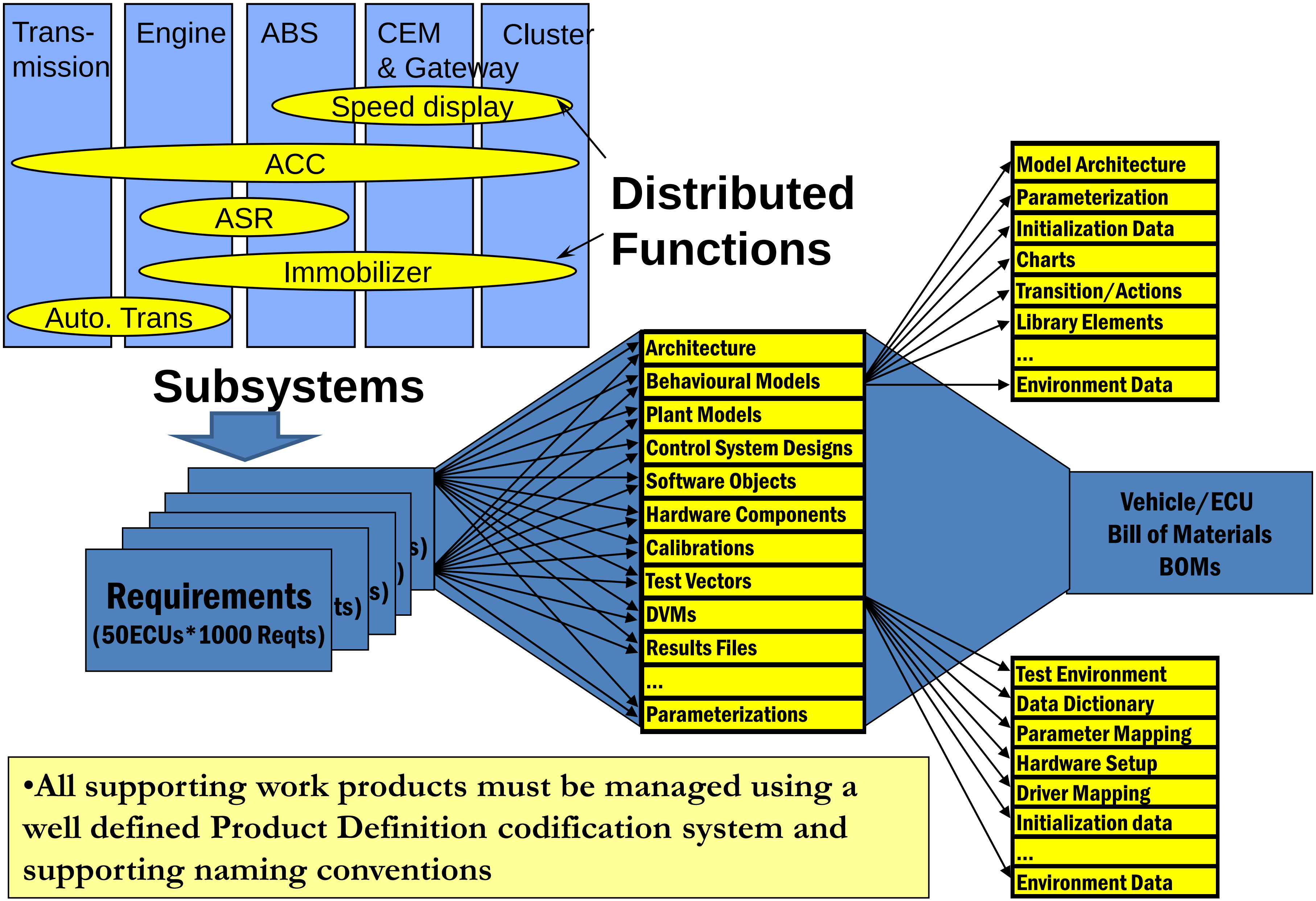


APIM – Sync Module
IPC – Instrument Cluster

NAV – Navigation Unit
BCM - Body Control Module
PCM – Power Train Control Module

Complex Mapping of CTQ Work products

ONE FORD
ONE TEAM • ONE PLAN • ONE GOAL



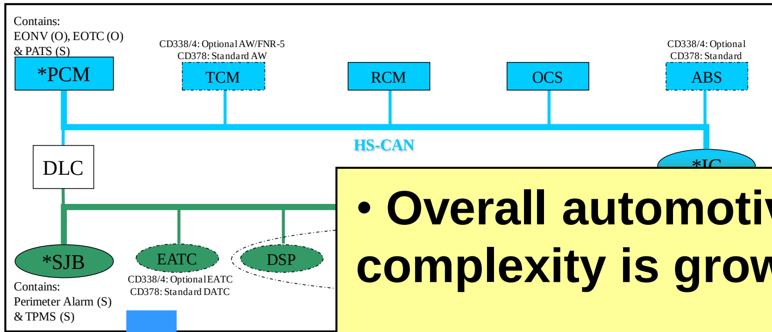
•All supporting work products must be managed using a well defined Product Definition codification system and supporting naming conventions

Example Vehicle Electrical System Growth

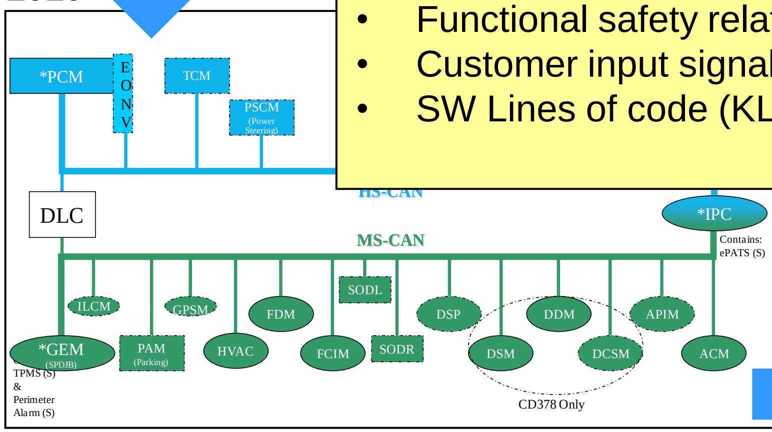
ONE FORD
ONE TEAM • ONE PLAN • ONE GOAL

	Networked ECU [#]	Signals [#]	Software Lines of Code [#]
MY2006	10 - 15	200 - 300	~ 3 Million
MY2010	20 - 30	1000 - 1500	~ 10 Million
MY2012+	50 – 70	3000 - 4000	~15 Million

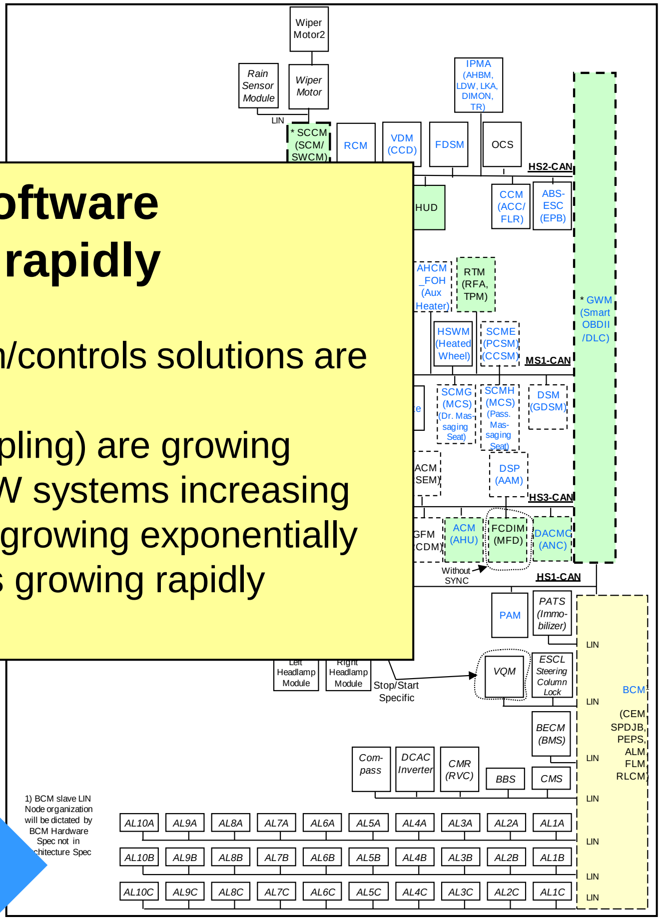
2006



2010

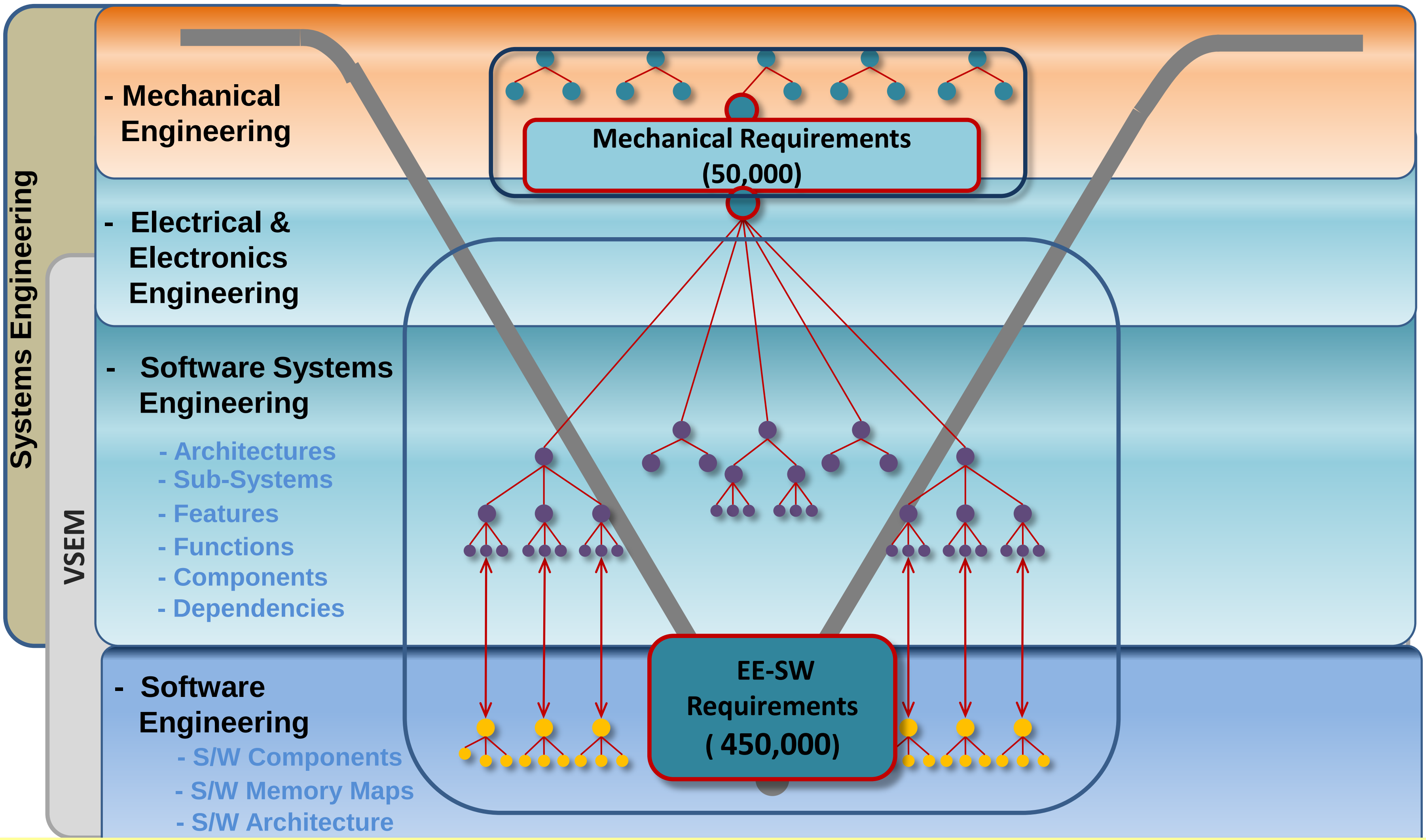


2012+



- **Overall automotive software complexity is growing rapidly**
  - Distributed software-system/controls solutions are growing rapidly
  - System dependencies (coupling) are growing
  - Functional safety related SW systems increasing
  - Customer input signals are growing exponentially
  - SW Lines of code (KLoC) is growing rapidly

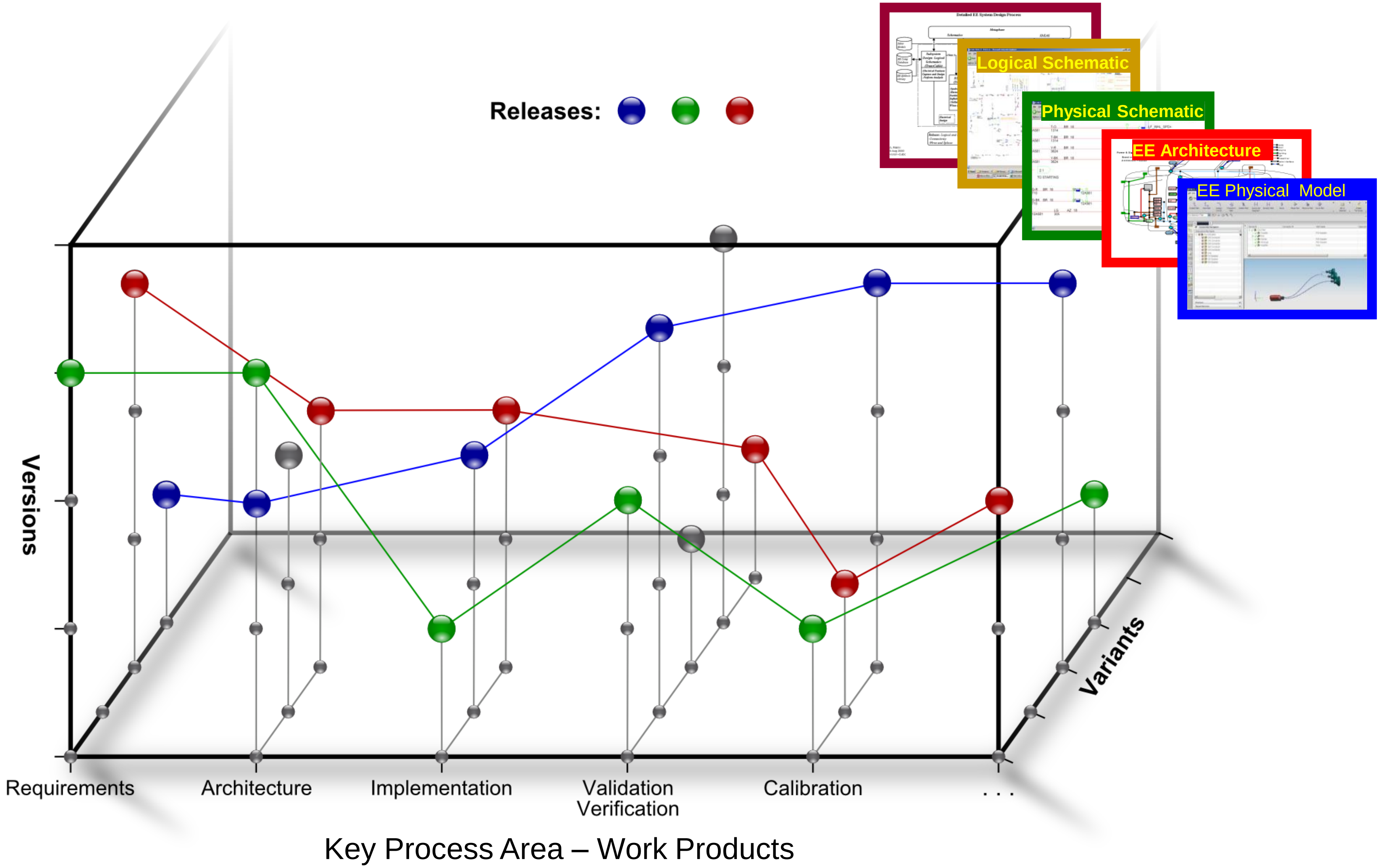
Requirements Engineering



- Mechanical Engineering requirements are typically less complex than EE-SW requirements and more stable
- EE-SW System requirement generate many decomposed levels of requirement-specifications that are more dynamic in nature.
- Software Systems Engineering bridges the gap between Traditional Systems Engineering and Software Engineering, it is an elaboration of the SE principles/practices.

Version – Variant – Release Management

- **Release Configurations / Configurable Baselines**

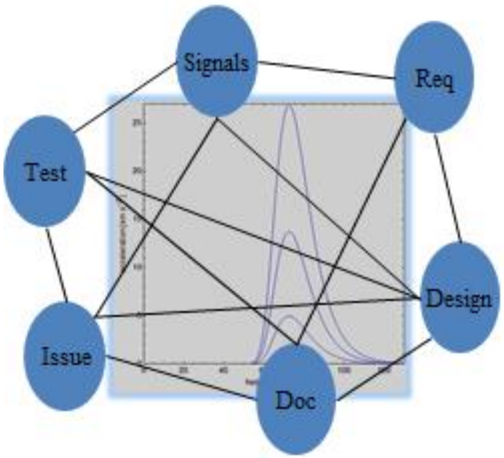


Courtesy of Reiner Busch, Ford Aachen Research Laboratories

PLM/ALM HW-SW Relationship Management

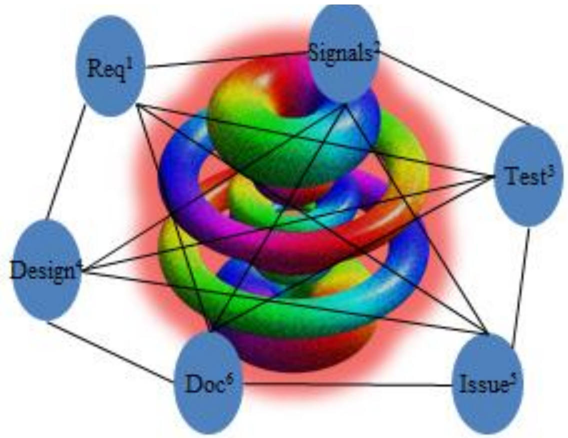
ONE FORD
ONE TEAM • ONE PLAN • ONE GOAL

Mechanical Domain Complexity
(10^3 Dependencies)



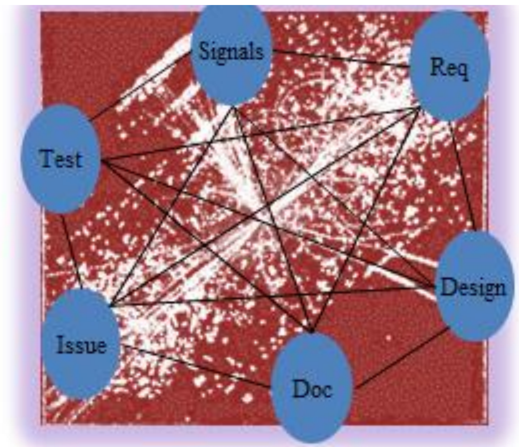
Newtonian Physics

Software Domain Complexity
(10^5 Dependencies)



Quantum Physics

On-Board/Off-Board Software Domain Complexity
(10^6 Dependencies)



Higgs-Boson event

- All product design work products/artifacts must be managed effectively; preserving their dependencies between content, versions & builds
- This presents a Multi-dimensional, multi-variant, multi-domain challenge that extends off-vehicle.

In conventional engineering, we work in the linear zone where stress is linearly related to strain and the behaviour of the engineering system at ‘run-time’ is much more predictable. Software is in general, fully chaotic.
*** Les Hatton, IEEE Software July/1999.**

Why do Model Based Systems Engineering ?

ONE FORD
ONE TEAM • ONE PLAN • ONE GOAL

- Generic Software E/E Failure Mode Mapping:

Work Product Objects

- Architectures
- Features
- Requirements
- Interfaces
- Parameters
- Executable Models
- Test Vectors
- Software Binaries

Failure Modes

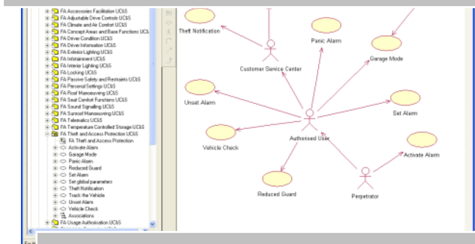
- Missing
- Incomplete
- Inconsistent
- Incorrect (content)
- Incorrect Version / no versioning
- Inappropriate design
- Unaligned (incorrect configuratio
- Un-accessible (Global Engineering Centers)
- Non-Validated
- Duplicate and/or Conflicting

- **Number of Critical-to-Quality work products and associated failure modes typically increase with: growth in SW-EE Feature content, growth of organizational interfaces, growth in global markets and re-use strategies**

Model Based Systems Engineering/Validation

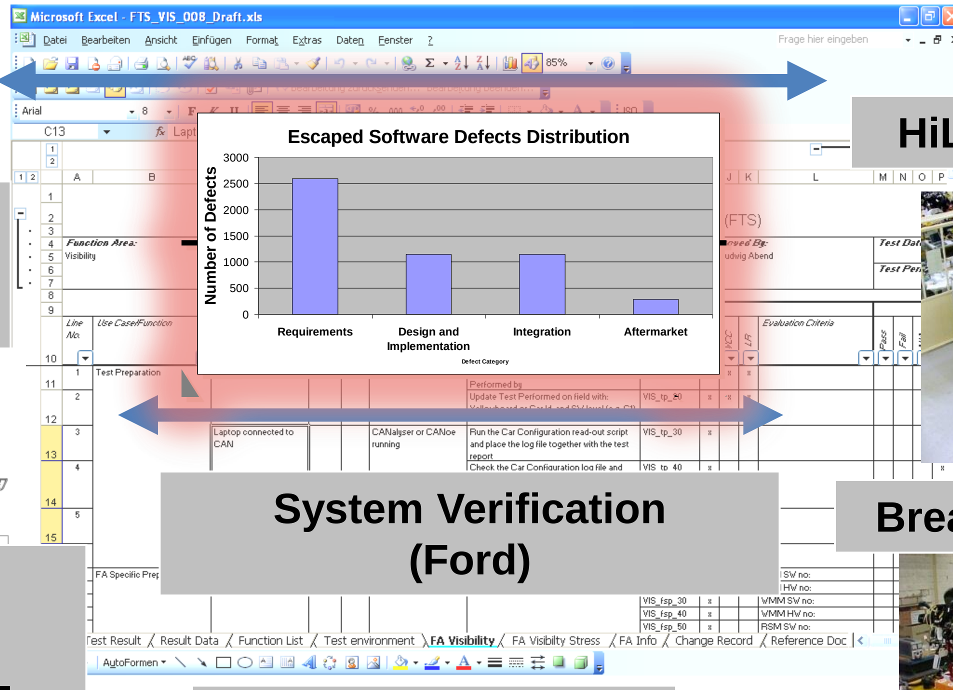
ONE FORD
ONE TEAM • ONE PLAN • ONE GOAL

Customer Requirement - Operational View – (MBA)



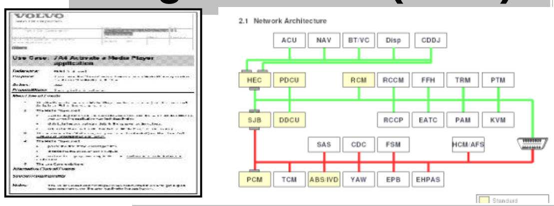
Vehicle Level Validation (Ford)

Vehicle Level Testing



HiL System Testing

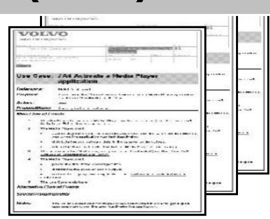
System Requirements - Logical View- (MBR)



System Verification (Ford)

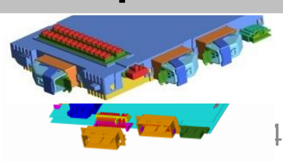
BreadBoard Testing

Component Specifications - Physical View – (MBD)



Component Creation/Testing (MBAutoCode/Test)

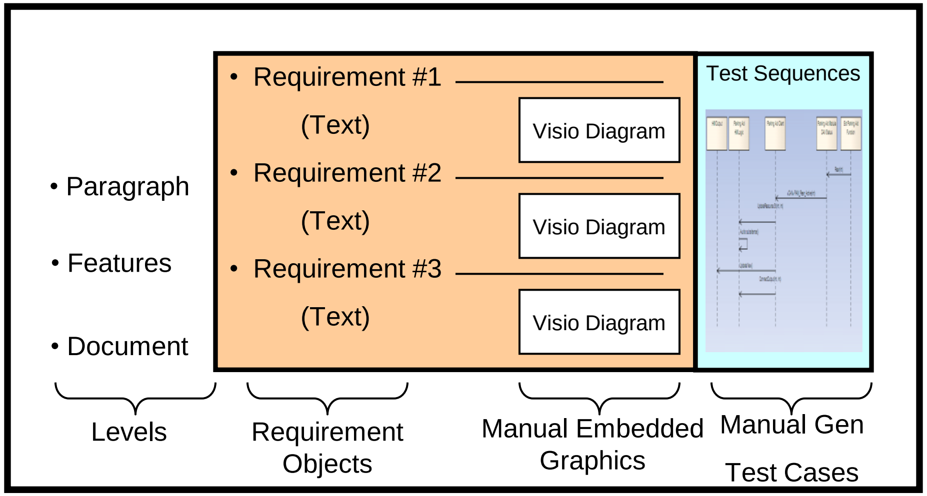
Model Based Component Testing



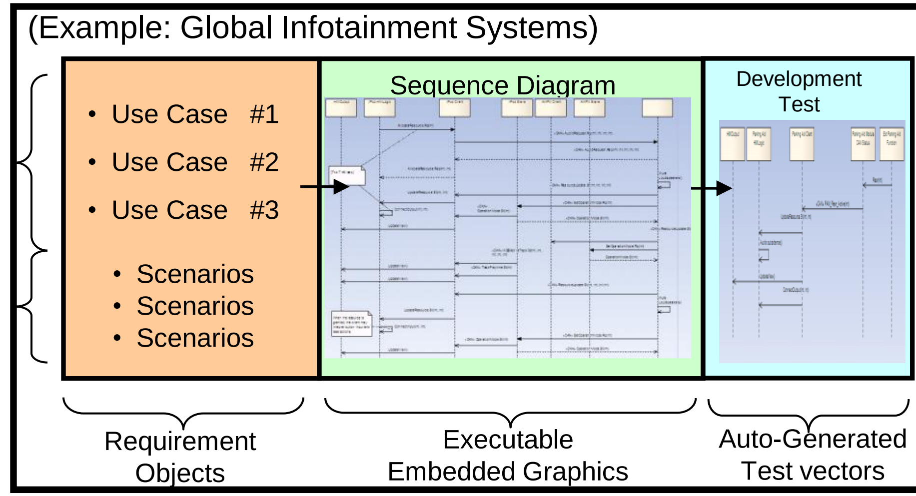
- Software Development/CMMi/SPICE
  - Spiral
  - Rapid development
  - Agile
  - Extreme

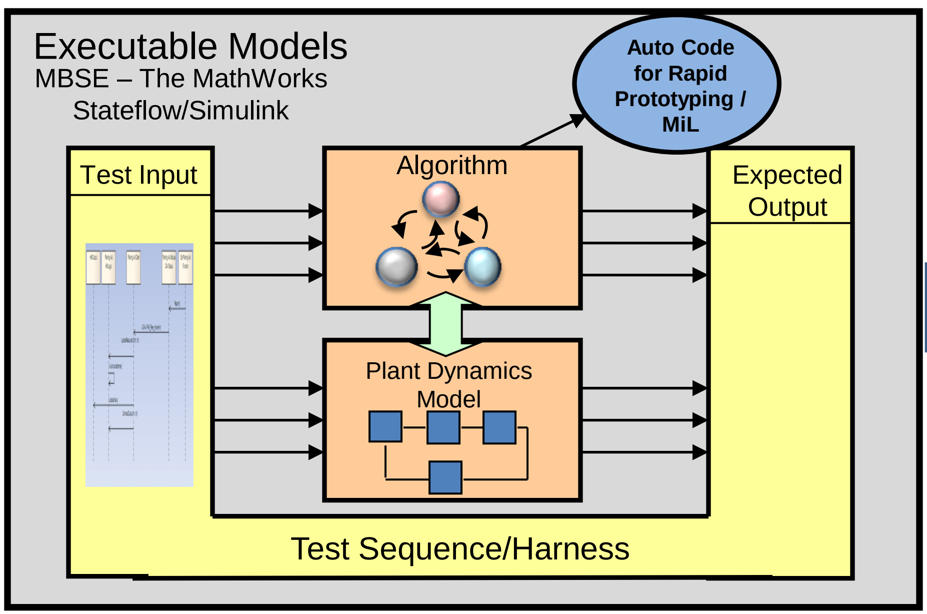
Model Based Re-Use

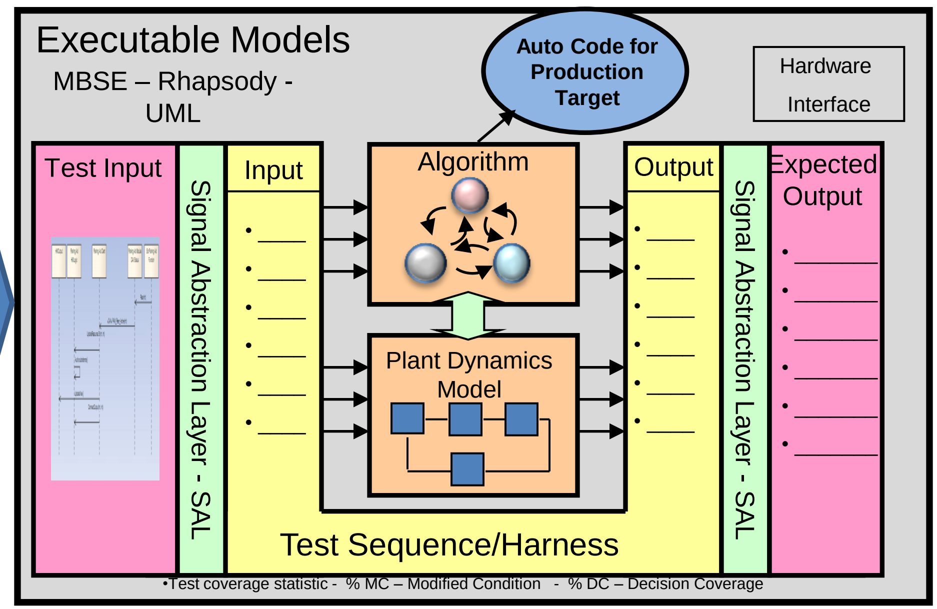
Traditional Textual Document Approach



Use Case - Object Oriented Approach (UML)



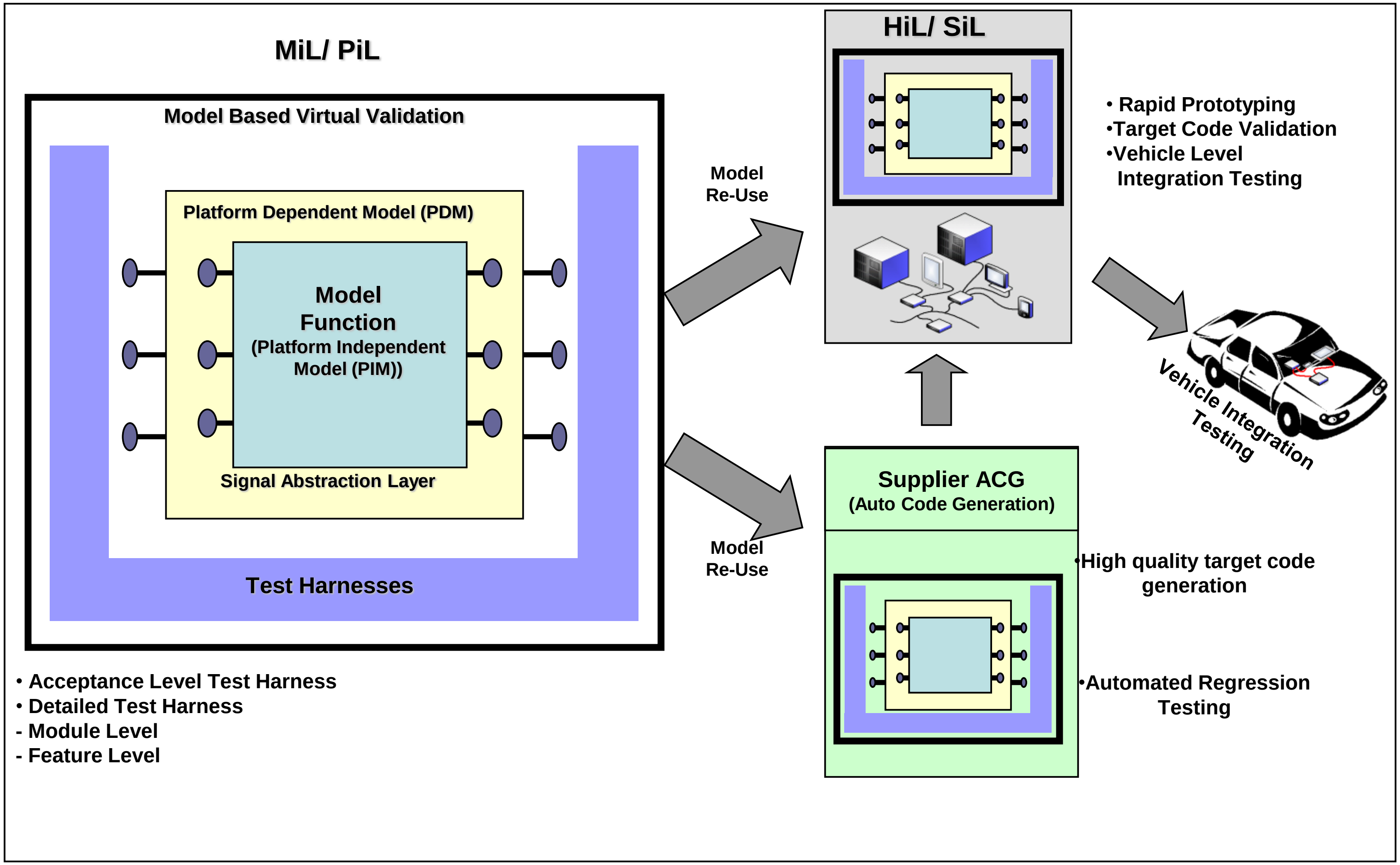




Platform Independent Model – PIM (Fully Re-usable)

Platform Dependent Model – PDM(Vehicle Specific)

Model Based Testing/Hardware-in-the-Loop



- Acceptance Level Test Harness
- Detailed Test Harness
- Module Level
- Feature Level

<u>Efficient Utilization of Model-in-the-Loop, PiL & Hardware-in-the-Loop Testing requires</u>:

- Adoption of Platform Independent Models
- Platform Dependant Model
- Signal Abstraction Layer Mapping to Maximize Artifact Re-use
- Management of Test Artifacts: Test Environments, Test Vectors, Model Parameterization Data, Plant Models

When, What and How to Model - Considerations

- **Business Process Modelling**
  - How dependent are the Development, Production and Service Process-Objects?
  - What degree of Business/Engineering visibility do you want to provide?

- **Business Operational Models**
  - In-house Software System Development (SwSystDev)
  - Outsourced-Supplier SwSystDev
  - Hybrid SwSystDev (Internal/Supplier) (Feature/Function Co-linking)

- **Suitability of Feature/Subsystem for Executable modelling**
  - Degree of re-use
  - Feature Complexity
  - Feature Stability
  - Feature Distributed Nature
  - Degree of newness…
  - Domain Competency/skillset

- **Type of Model Framework**
  - Standards (Industry, Corporate, Domain, Organisation.).
  - Degree of Compliance/Leveraging of Standards

Business Models, Process Models, Product Models…

- Big picture processes and models…

CORE Product Development

1. Business model
2. Application/Architecture model
3. Detailed Design
4. Code * test
5. System & Test
6. Issues Management

Business Co-ordination

1. Project management
2. Quality Management
3. Change Management
4. Cost Management

APPLICATION Product Implementation

1. Detailed Implementation Design
2. Implementation Code and test
3. Systems Test
4. Acceptance Test
5. Issues Mnagement

Business Facilitation

1. Configuration and Variant Management
2. Interface Management
3. Training
4. Quality Assurance
5. Metrics and Reporting

Product Customer Usage/Dealerships

1. Release Mngmnt
2. Performance Mngmt
3. Customer Support
4. Warranty Mngmt
5. Change Mngmt

Architectural Views

	Module Views					C&C Views	Allocation Views				Other documents					
	Decomposition	Uses	Generalization	@@@	Data Model	@@@	Deployment	Implementation	@@@	@@@	@@@	@@@	@@@	@@@	@@@	
Project Manager	S	S		S												
Development team	D	D	D	D												
Test Engineers	D	D	D	D												
Integration Engineers	D	D	D	D												
Design Engineers					S											
Service Engineers	D	D	D	D	D											
Product-Line Engineer	D	D	S	O	S											
Internal Customers																
End Users/Customers																
Analysts	D	D	S	D	D											
Infrastructure/EcoSystem IT support	S	S			S											
New Stakeholders	X	X	X	X	X											
Current/Future Architects	D	D	D	D	D											

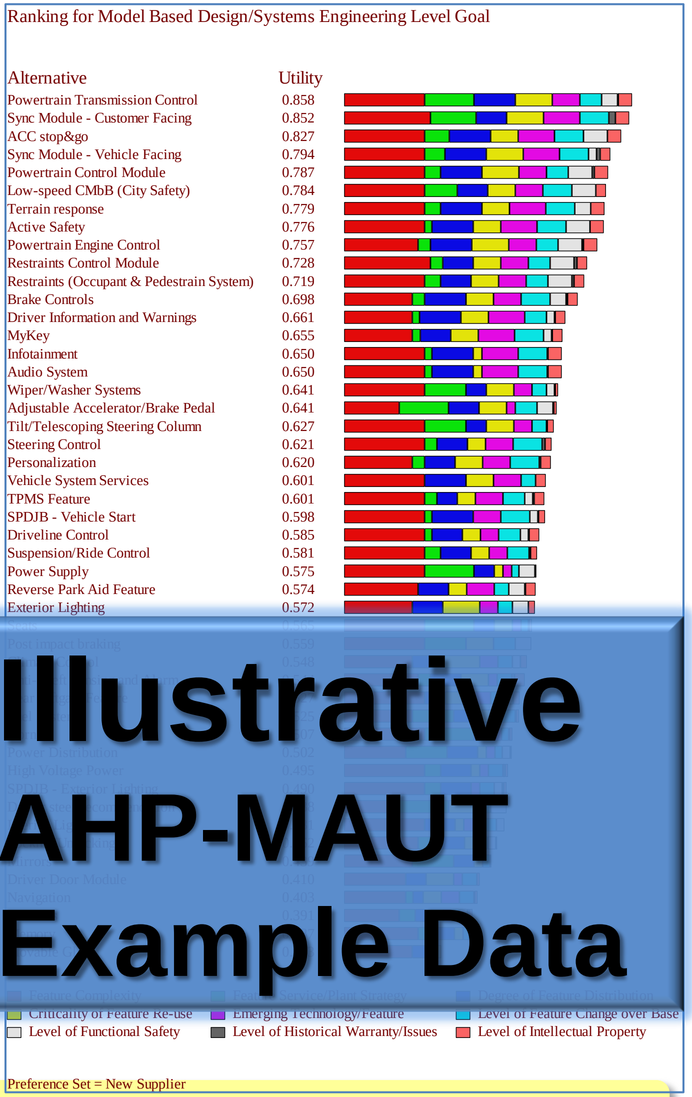
- Which Architectural Perspectives are Critical to Business Success?
- What degree of interoperability do you need between the Architectural Views?
- What Degree of Executable Architecture Model do you need to implement ?

Key: D - Detailed Information; S - Some Details; O - Overview Information; x - anything

Derived from Data within "Documenting Software Architectures, 2nd edition, Paul Clemens et al"

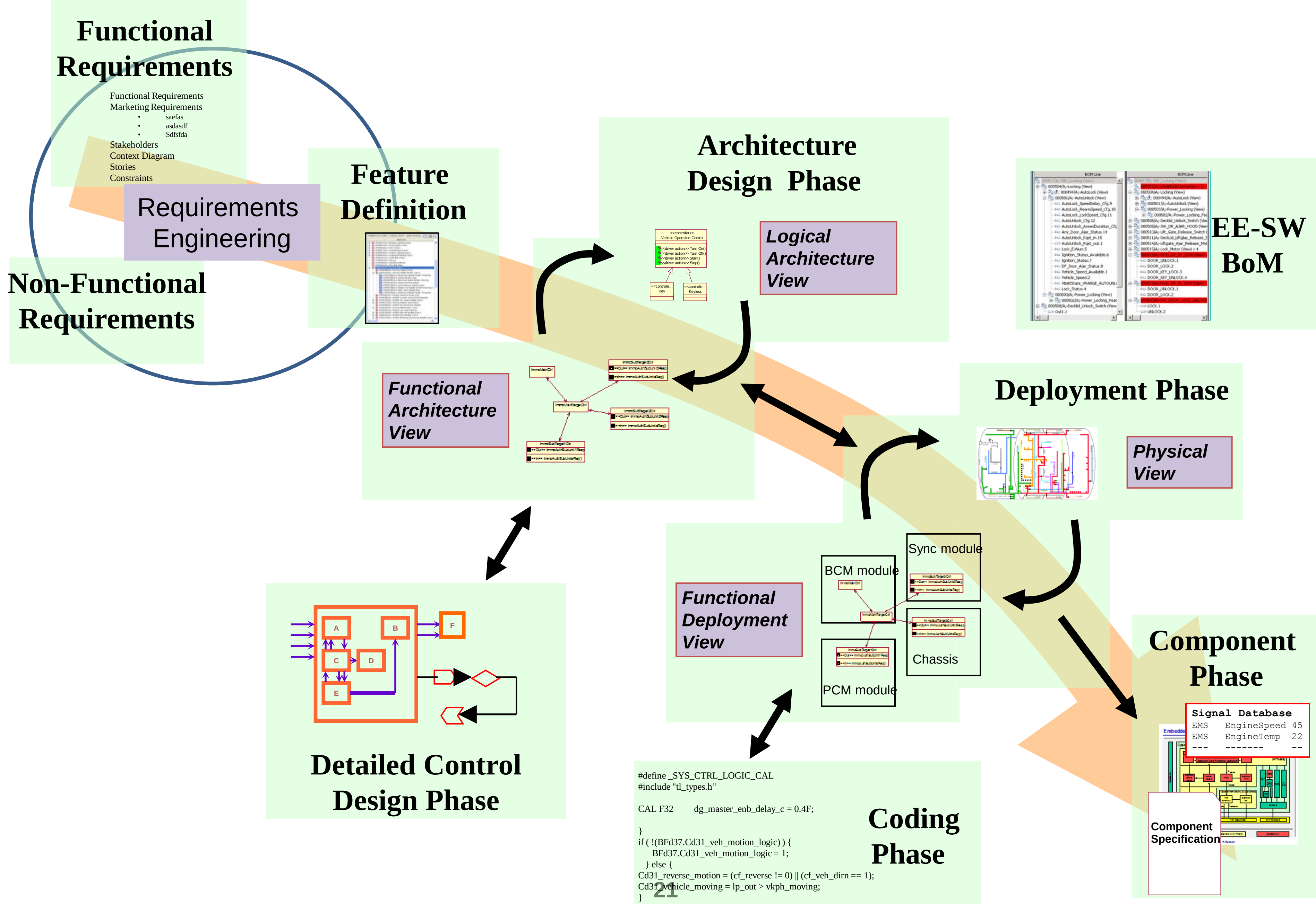
Effectiveness? Failures Models, Causal Factors & PCA Study

- Six Sigma Studies (DFSS & DMAIC)
  - Process Maps
  - Hidden Factories
  - Capability Assessments
  - Causal factor - Defect Correlation statistical studies
  - The Critical Few – Critical to Business (CTB) Process, Methods, Tools and Information Management
  - CTB Metrics
- Hierarchical Decision Process to Prioritise the Critical Few



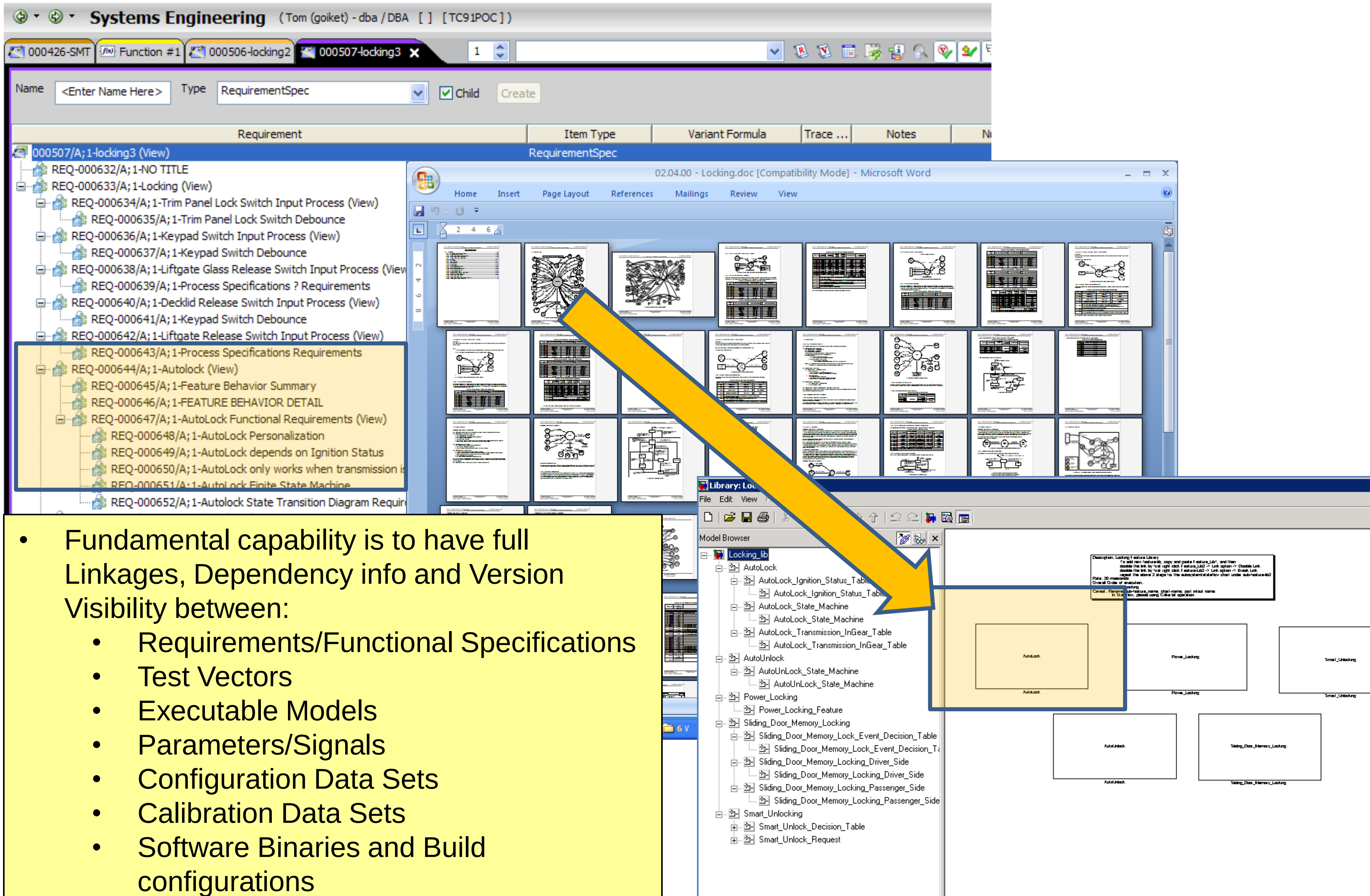
- **DMAIC and DFSS based analysis identifies and prioritises the key areas that need to be addressed**

Model Driven Architecture - MBSE Process?



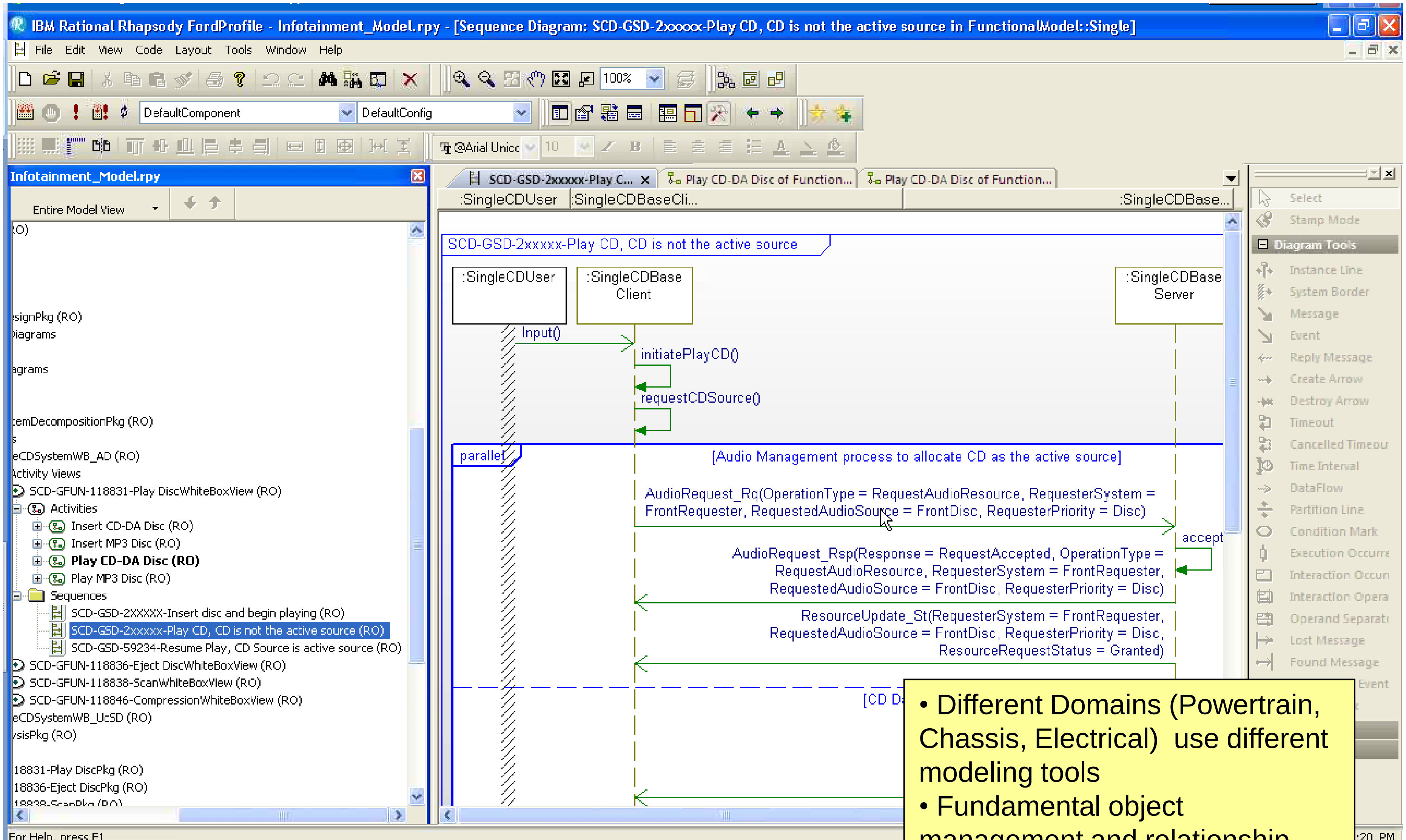
Textual Requirements & Executable Model Mapping

ONE FORD
ONE TEAM • ONE PLAN • ONE GOAL



- Fundamental capability is to have full Linkages, Dependency info and Version Visibility between:
  - Requirements/Functional Specifications
  - Test Vectors
  - Executable Models
  - Parameters/Signals
  - Configuration Data Sets
  - Calibration Data Sets
  - Software Binaries and Build configurations

UML – Infotainment Modeling

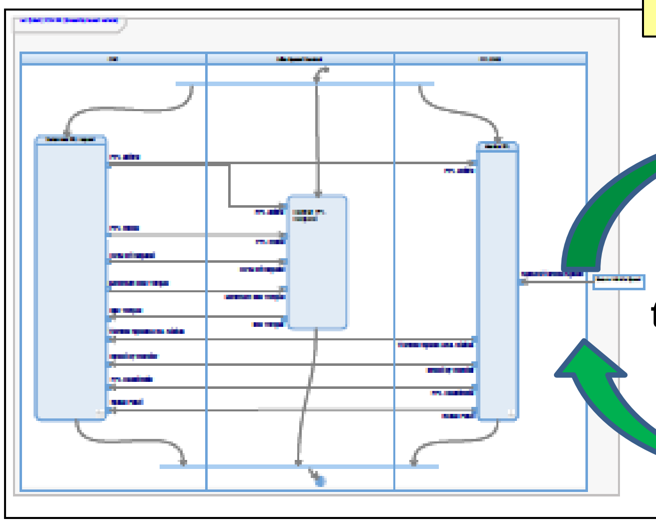


- Different Domains (Powertrain, Chassis, Electrical) use different modeling tools
- Fundamental object management and relationship tasks are similar

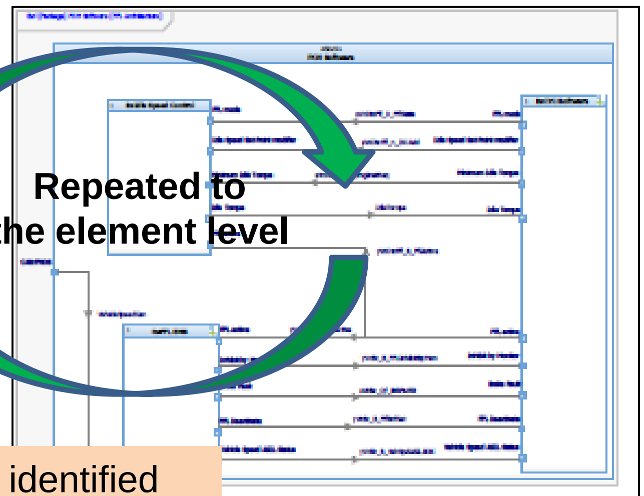
Courtesy of Nick Collela, Ford Motor Company

SysML – Powertrain Feature Modeling

System high level functional requirements are modeled in SysML which are then decomposed into lower level behavior models

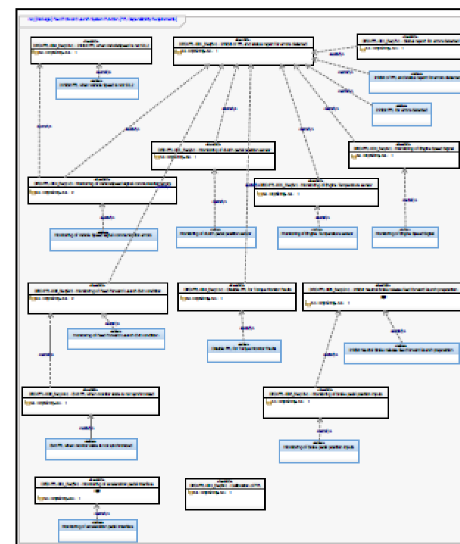


Behaviors are partitioned into a logical architecture and physical architecture



Repeated to the element level

The functional system requirements and derived requirements are linked to the SysML behaviors using a Requirements Diagram



Courtesy of Kyle Post, Ford Motor Company

Requirements gaps were identified when modeling the behaviors which were not apparent from the textual requirements alone

Managing Vehicle System Complexity

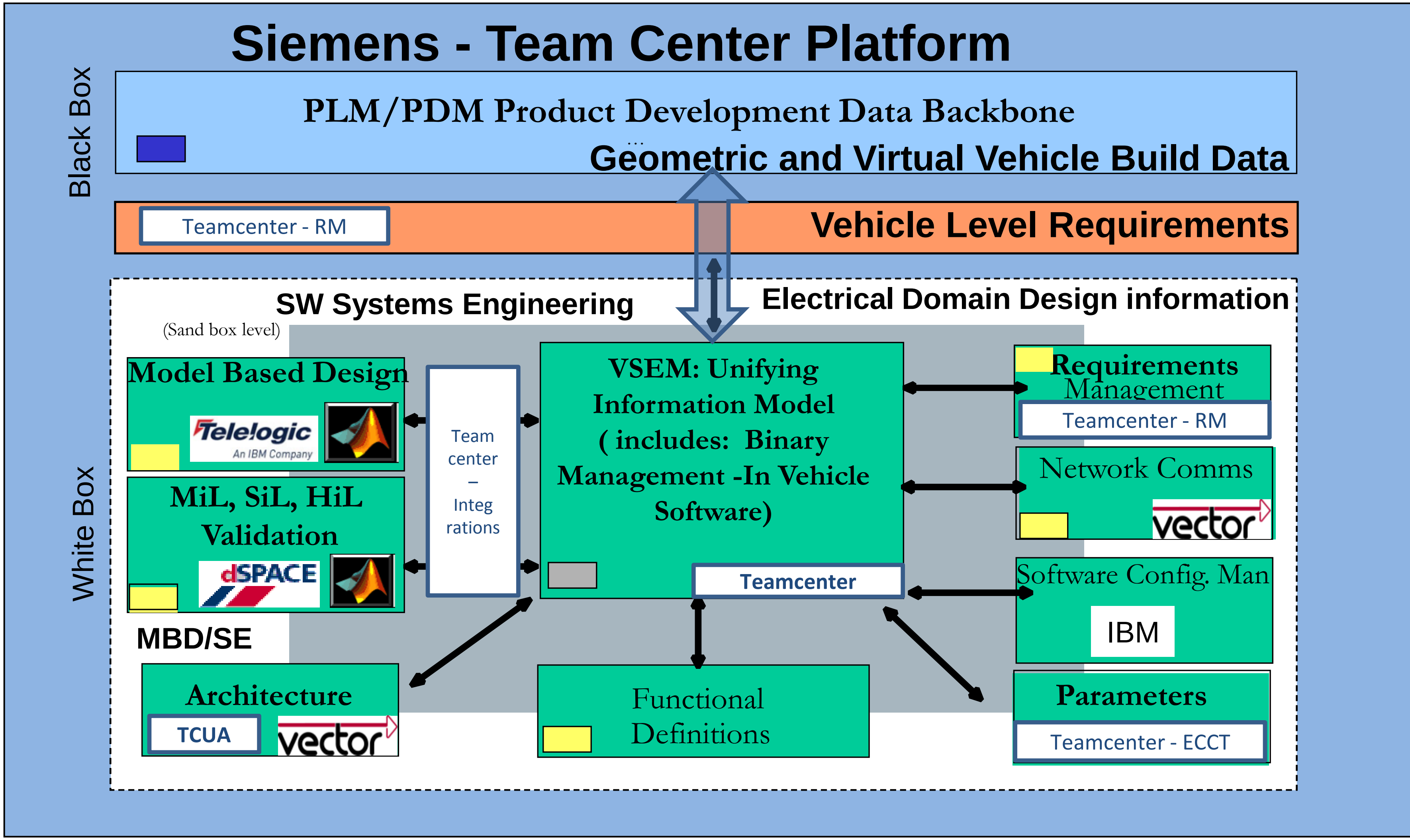
ONE FORD
ONE TEAM • ONE PLAN • ONE GOAL

Vehicle Software & Electrical Management – VSEM

Create a global information platform with seamless information management of engineering data from function to ECU S/W & H/W inclusive to create a complete PLM/ALM solution for electronic/software area.

- **Provide a fully traceable, object level work product framework**
- **Support a Model Based Systems Engineering EE development process**
- **Single source for all information E/E development data**
- **Support change, version, and configuration management** with full traceability
- **Based on standardised open tool framework** to enable utilisation of “best-in-class” commercial state-of-the-art “plug-in” tools.
- **Designed on industry engineering standards** and aligned to the AUTOSAR architectural framework

VSEM Solution Architecture

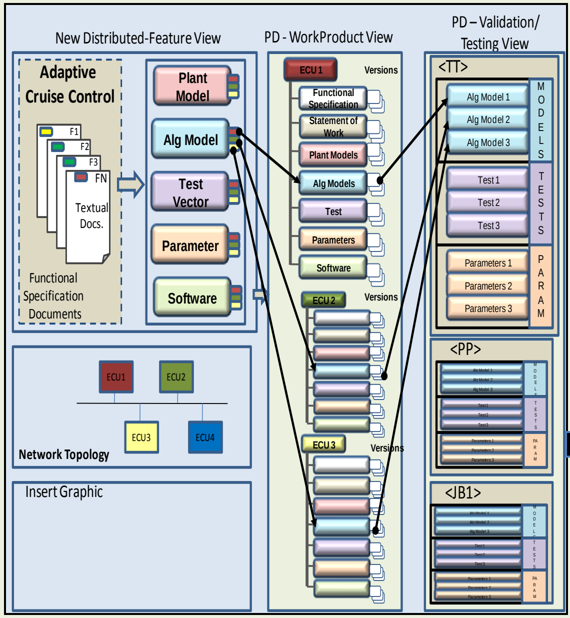


Global Enterprise information | Electrical domain information | Local information in point tools | ↔ Defined Exchange formats

- Adoption of Single Corporate data backbone and authoring tools enables WP and relationship Meta data re-use across Corporate PD systems.
- Must support data objects that require rapid iteration (ALM) Vs structured management (PLM).

VSEM Project Scope and Implementation

Product Development



- Systems Engineering,
- Software Systems Engineering
- Software Engineering

> Model Based Systems Engineering<

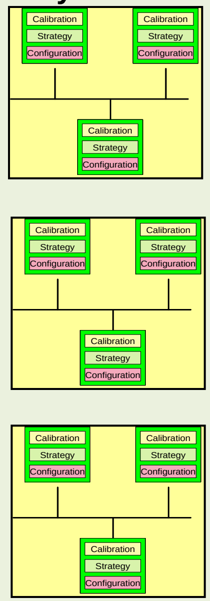
EE System Artifacts that derive a Released BOM

Requirements, Tests, Models, Parameters, Devices, Calibrations, Configurations, Architectures…

(As-Designed)

In-Vehicle Software (IVS)

E/E System BOMs



EE System Released BOMs

(As-Released)

Assembly Plants

Manufactured Vehicle Series Variants

BOM 1
BOM 2
BOM 3
BOM 4
BOM 10
BOM 50

In Plant Flash

- Valencia
- Saarlouis
- Cologne
- Genk
- Auto Alliance
- Cuautitian

Global Manufacturing Plant Reprogramming

(As-Built)

Dealerships and Web Services

FoE - Dealer Updates

FAP- Dealer Updates

As-Modified Vehicle Series Variants

BOM 1	BOM 2	BOM X
BOM ..	BOM ..	BOM ..
BOM ..	BOM ..	BOM ..
BOM ..	BOM ..	BOM ..
BOM ..	BOM ..	BOM ..
BOM ..	BOM ..	BOM ..
BOM ..	BOM ..	BOM ..
BOM ..	BOM ..	BOM Y

Customer Web Services Sync My Ride

FSA- Dealer Updates

FNA- Dealer Updates

Global Dealerships & Customer Updates

(As-Modified)

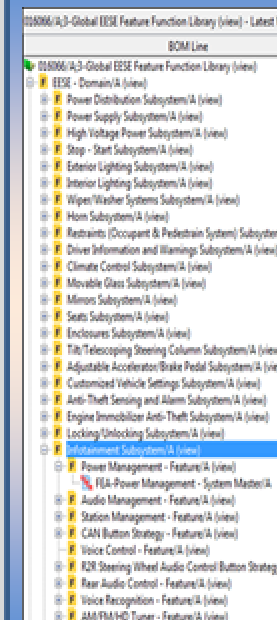
Systems Engineering MB Framework

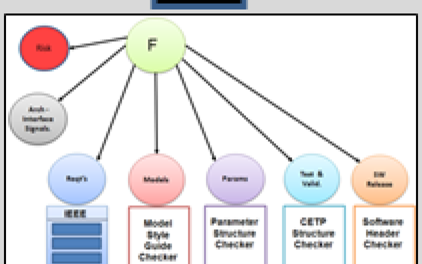
ONE FORD
ONE TEAM • ONE PLAN • ONE GOAL

Systems Engineering Work Products

- FMEA
- P-Diagrams
- Boundary Diagrams
- Interface Matrix
- Noise Factor Mangmt Strategy
- DVP
- Safety Plan
- Prelim Hazard Analysis
- Functional Safety Concept

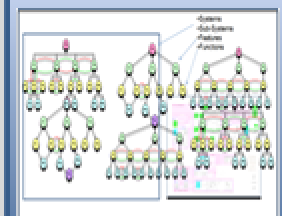
Feature Dictionary





Functional Architecture
(Implementation Independent)

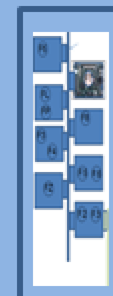
Virtual, Model Based Analysis; Dependencies; Coupling; Cohesion & Arch Quality Attribute Studies



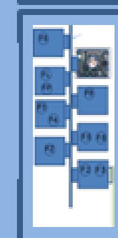
Logical Architecture
(Re-usable Reference Architectures)

Commodity Specific Reference Architectures

Sync Gen 2 Ref Arch V1.0

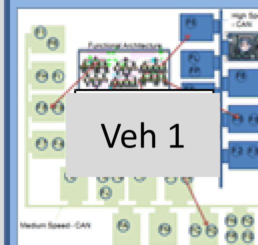


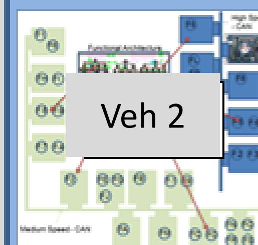
BCM Ref Arch V1.x



Physical Architectures
(Vehicle Program Specific)

Vehicle Program Specific Architectures

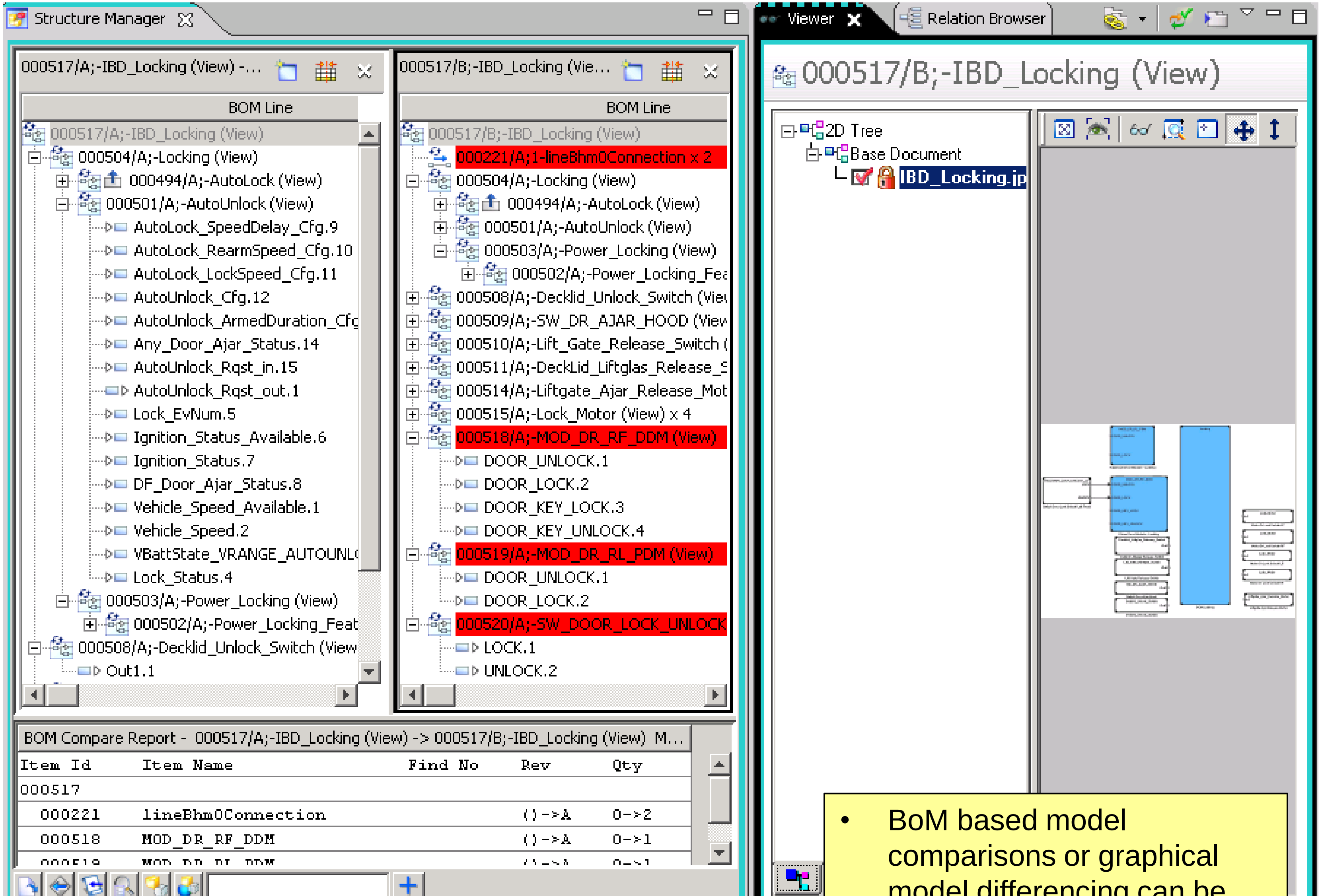




Global Signal Database (GSDB)

- Signal Definitions - Interface Specs
  - -CAN Message relationship
- Global Device Transmittal Database (GDT)
  - SW-HW interfaces
  - EE Devices – Sensors - Actuators

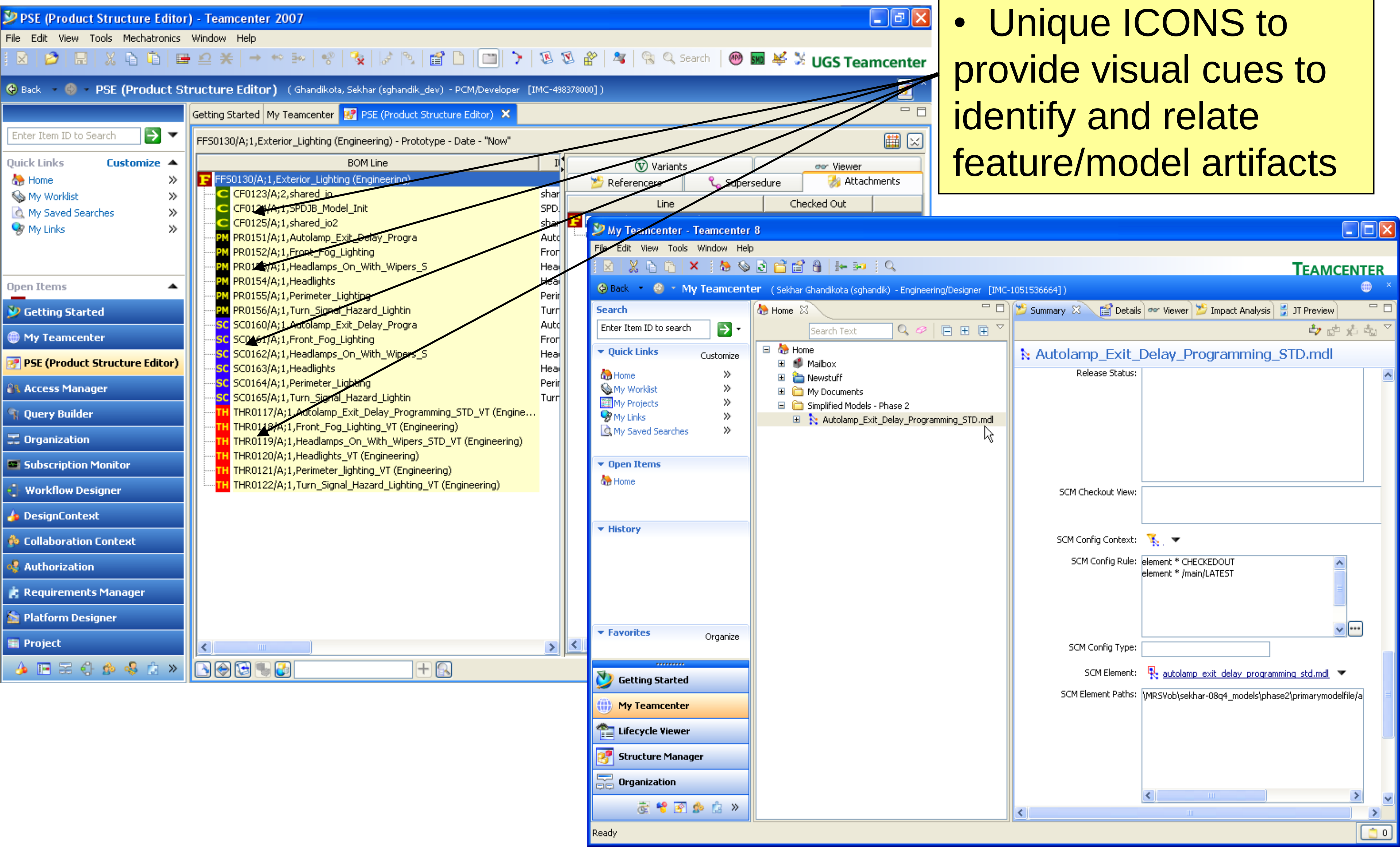
Model Comparisons in BoM View



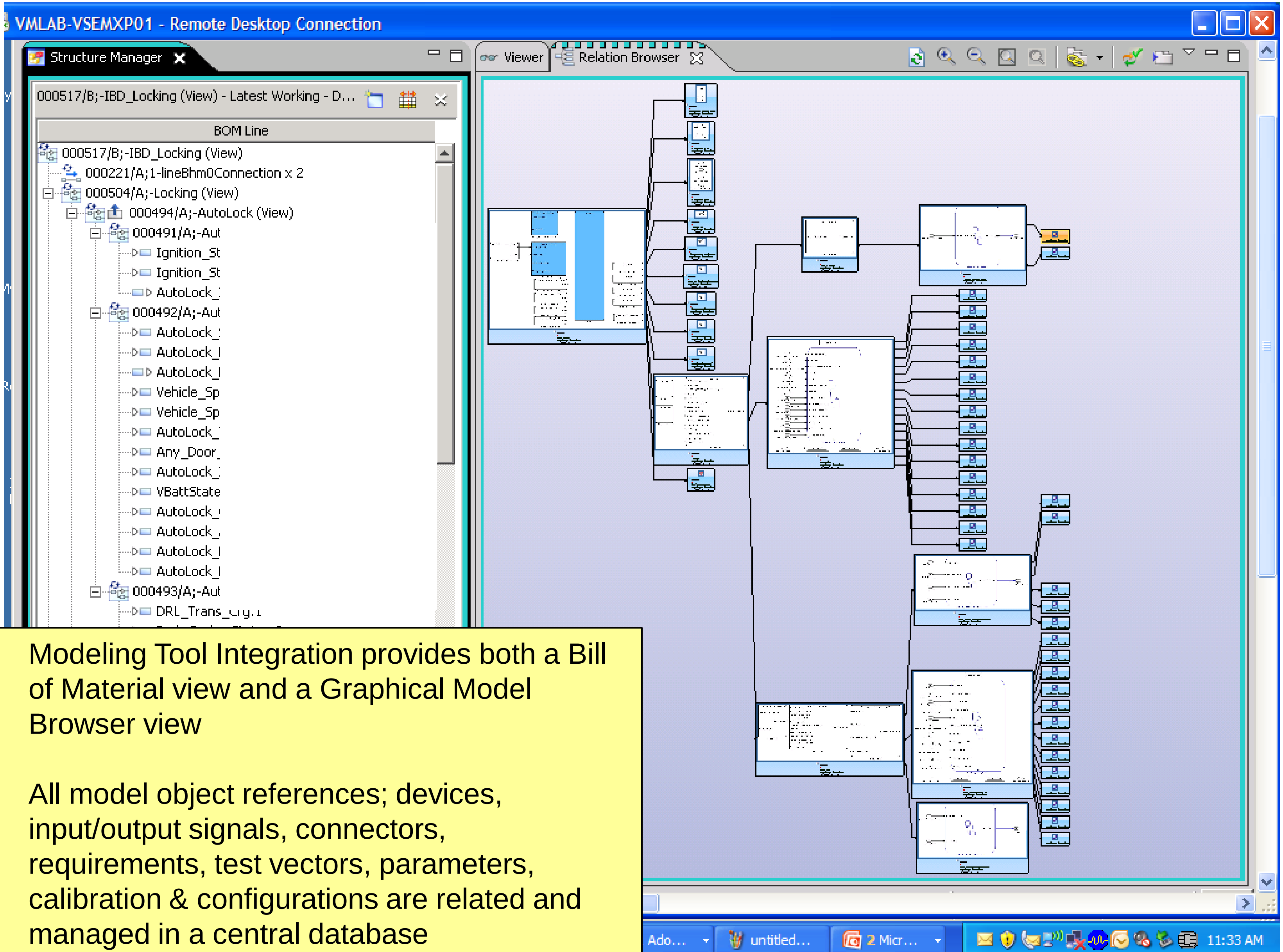
- BoM based model comparisons or graphical model differencing can be performed.

PLM/ALM Based Model Management

- Unique ICONS to provide visual cues to identify and relate feature/model artifacts



Models in a BoM & Object Relationship Viewer

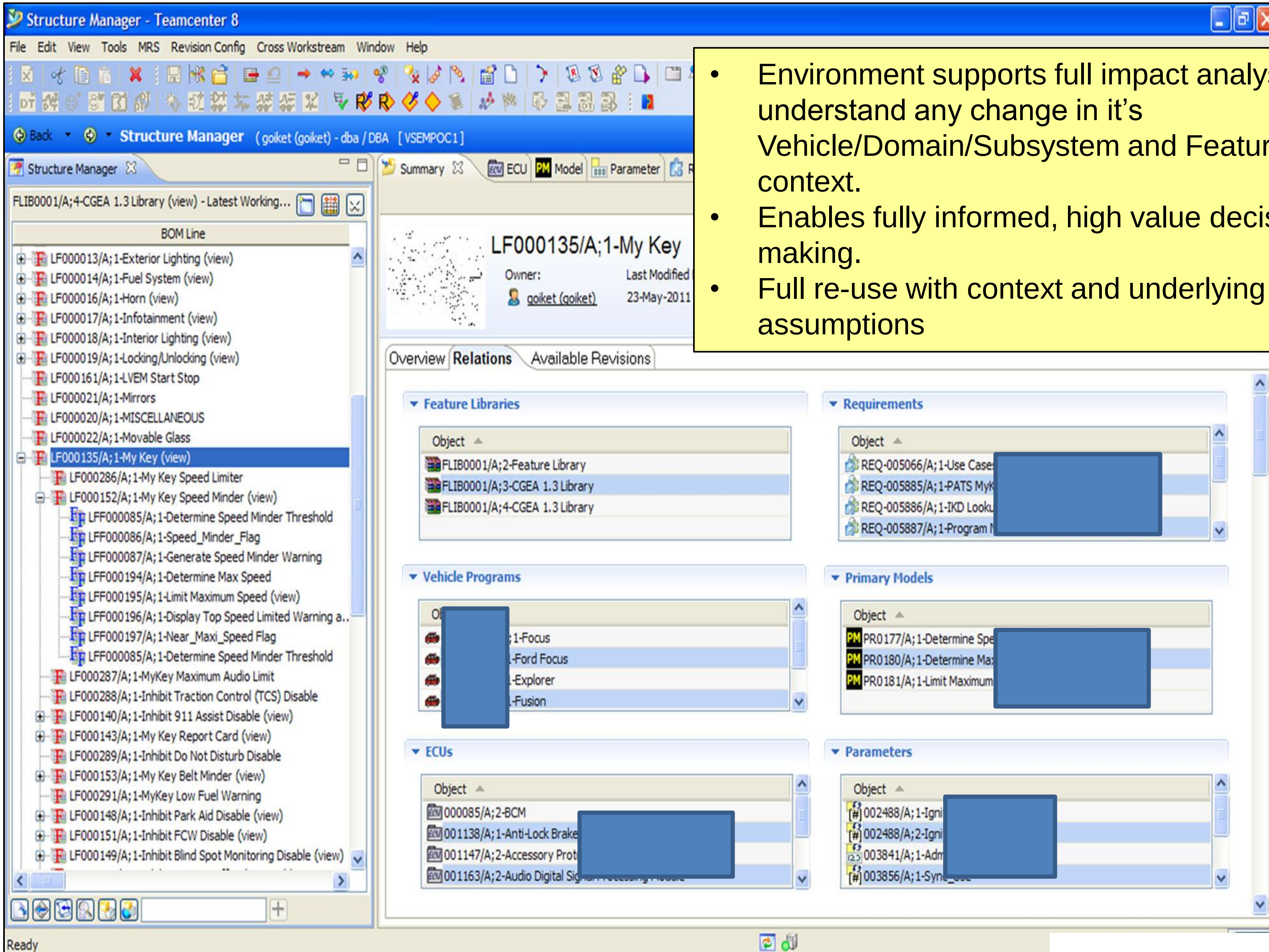


- Modeling Tool Integration provides both a Bill of Material view and a Graphical Model Browser view

- All model object references; devices, input/output signals, connectors, requirements, test vectors, parameters, calibration & configurations are related and managed in a central database

Full Traceability into Impacted Work Products

ONE FORD
ONE TEAM • ONE PLAN • ONE GOAL



- Environment supports full impact analysis to understand any change in it’s Vehicle/Domain/Subsystem and Feature context.
- Enables fully informed, high value decision making.
- Full re-use with context and underlying assumptions

PLM/ALM/MBSE Lessons Learned

"On Ko Chi Shin" – "Studying the old promotes a better understanding of the new". Ancient Japanese proverb.

1. Different domains select different modeling tools due to multiple factors
   1. Ability of their domain specific artifacts to be represented completely within a modeling schema/construct set
   2. Number of engineers with a specific modeling competency within the domain
   3. Amount of legacy models of a given modeling tool type within a domain
   4. Level of abstraction required within a domain and across domains
   5. Availability/cost of existing/new Licenses
   6. Lack of engineering staff with modeling competency or allocated resource/time.

2. Some features/commodities do not benefit from detailed modeling (non-changing features that are truly purchased commodities and do not benefit from executable modeling efforts (still may benefit from static architecture models)

3. These forcing functions lead to the need to support a HYBRID MBSE environment that consists of a set of modelling tools with associated style guides, maturity levels, completeness levels

4. For HYBRID environments it is important that all of the critical to quality/function artifacts/Meta data and associated objects/relationships are managed with supporting lifecycle management tools

5. Enterprise wide change is hard, painful, challenging and rewarding AND far better than the alternative……

VSEM Team Members:

Salwan Ishac
Piyush Karkare
Roy McKinstry
Meisam Yousif
Nana Allen
Erica Denton
Koren Pletcher
Jeff Porter
Jerry Solak
John Haraszkiewicz
Venkat Jakka
Pat Christensen

Anju Hora
David Bethel
Steve Kardel
Mark Anders
Chris Chevela
Chuck Nagi
Tobias Schmidt-Samoa
Bernd Frielingsdorf
Miguel Mancilla
Jorge Filigrana
Oscar Anaya
Hiram Guil Fuentes Rivera

VSEM Global Contributors:

Craig Stephens	- gPCSE
Steve Cibulas	- gPCSE
Bob Griffiths	- gPCSE
Todd Brown	- Chassis
John Joyce	- Chassis
Christopher Davey	- EESE
Frank Kirschke-Biller	- EESE
Susan Young	- EESE
Manu Vedapudi	- IT
Paul Renko	- IT
Claudio Moles	- FSA
Stuart Taylor	- APA
George Walley	- R&AE
Tony Lockwood	- R&AE
Tony Phillips	- R&AE
Kyle Post	- gPCSE

Thank-you for Listening