

- Importance of Systems Engineering
- PTC's Systems Engineering Solution
 - Process Governance
 - Requirements Engineering
 - System Design
 - Product Line Design
 - Test Management
- Systems Engineering Integration Roadmap

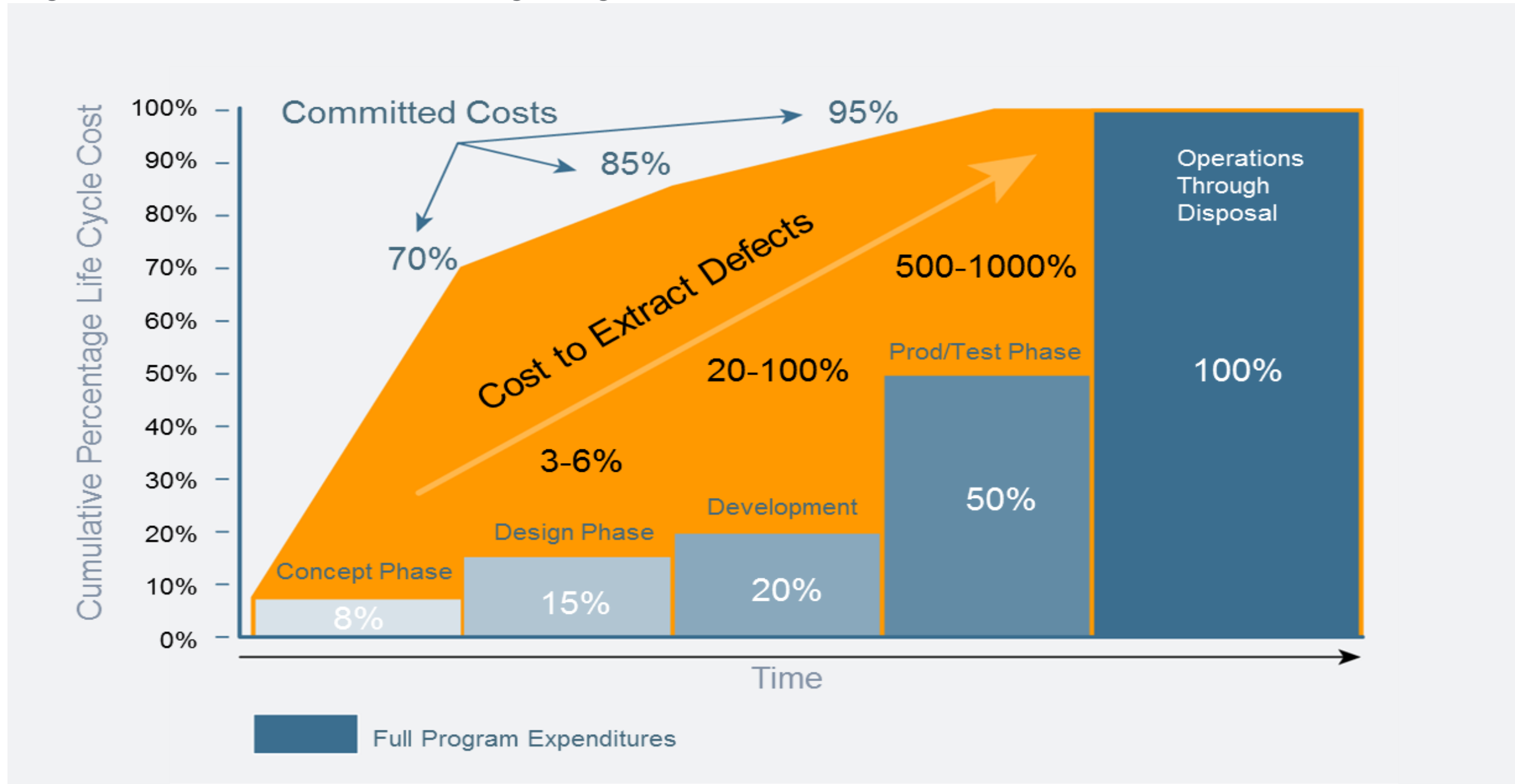


- Importance of Systems Engineering
- PTC's Systems Engineering Solution
 - Process Governance
 - Requirements Engineering
 - System Design
 - Product Line Design
 - Test Management
- Systems Engineering Integration Roadmap

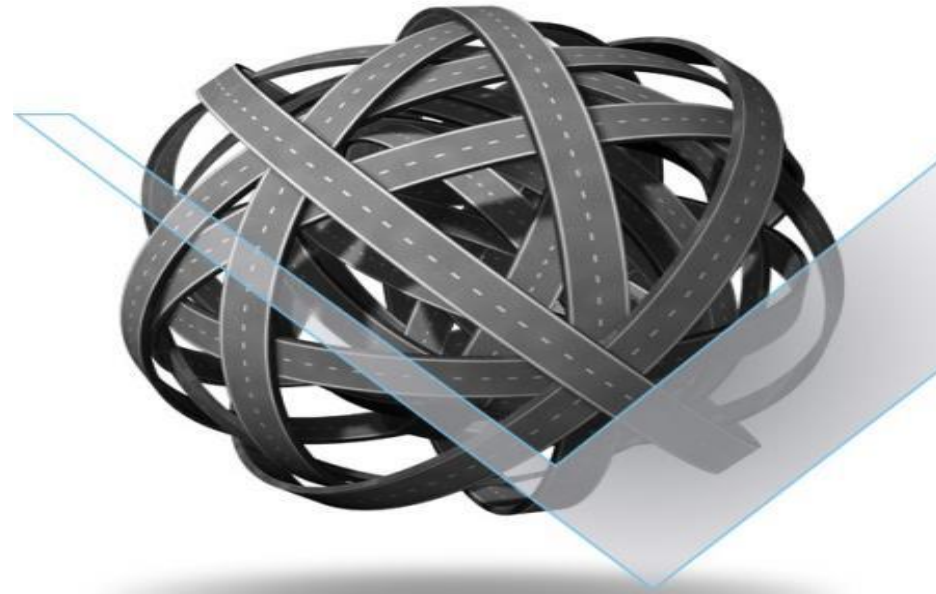


Why Systems Engineering is Important?

Getting it Right First Time...or the cost of getting it WRONG...



70% of the project costs are committed during the **first 8%** of the project and, **85%** after the first **15%**



- Spiraling product development costs
- Inability to innovate
- Increased risk

Product line complexity

- Explosive growth of product variants
- Change volume and velocity overwhelms teams
- Lack of data to make tradeoff decisions

Disconnected teams

- Lack of visibility inhibits collaboration, design and stakeholder buy-in
- Inability to accurately report progress
- Leaves everyone guessing at product readiness

Lack of reuse

- No common design approach or process
- Non-standard work products drive up costs
- No systematic reuse of methodology or tooling

Late validation and verification

- Errors found late in the lifecycle, when most costly to correct
- Skyrocketing costs of compliance
- Manual and error-prone regulatory documentation

Key Systems Engineering Capabilities

Common language for collaborative design

Design system specifications using standards-based SysML modeling notation

Dramatically improve internal and external communications, common standards and process, easier training/recruitment

Model-based product line design & reuse

Design system families and variants that maximize value and profitability

Greatly improve project delivery times and quality

Cohesive data mgnt with traceability across engineering disciplines

Trace requirements to design, code, products structures, and test

Rapidly evaluate requirements changes, efficient impact analysis

Systems Engineering Best Practices

Out-of-the-box expertise that can be tailored to your needs

Improve team skills and capabilities

Validating and verifying requirements and designs

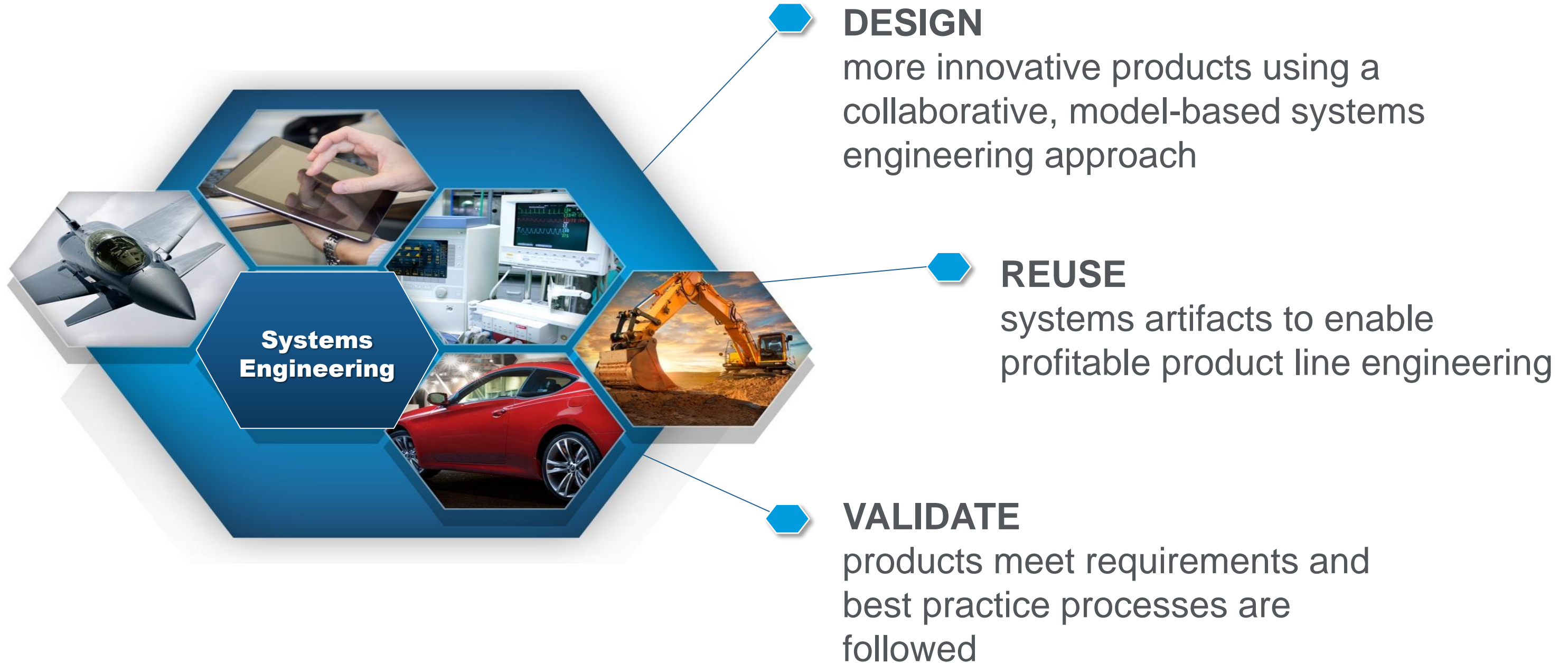
Validating and verifying requirements as well as designs through simulation and testing.

Address product quality early and often

- Importance of Systems Engineering
- PTC's Systems Engineering Solution
 - Process Governance
 - Requirements Engineering
 - System Design
 - Product Line Design
 - Test Management
- Systems Engineering Integration Roadmap

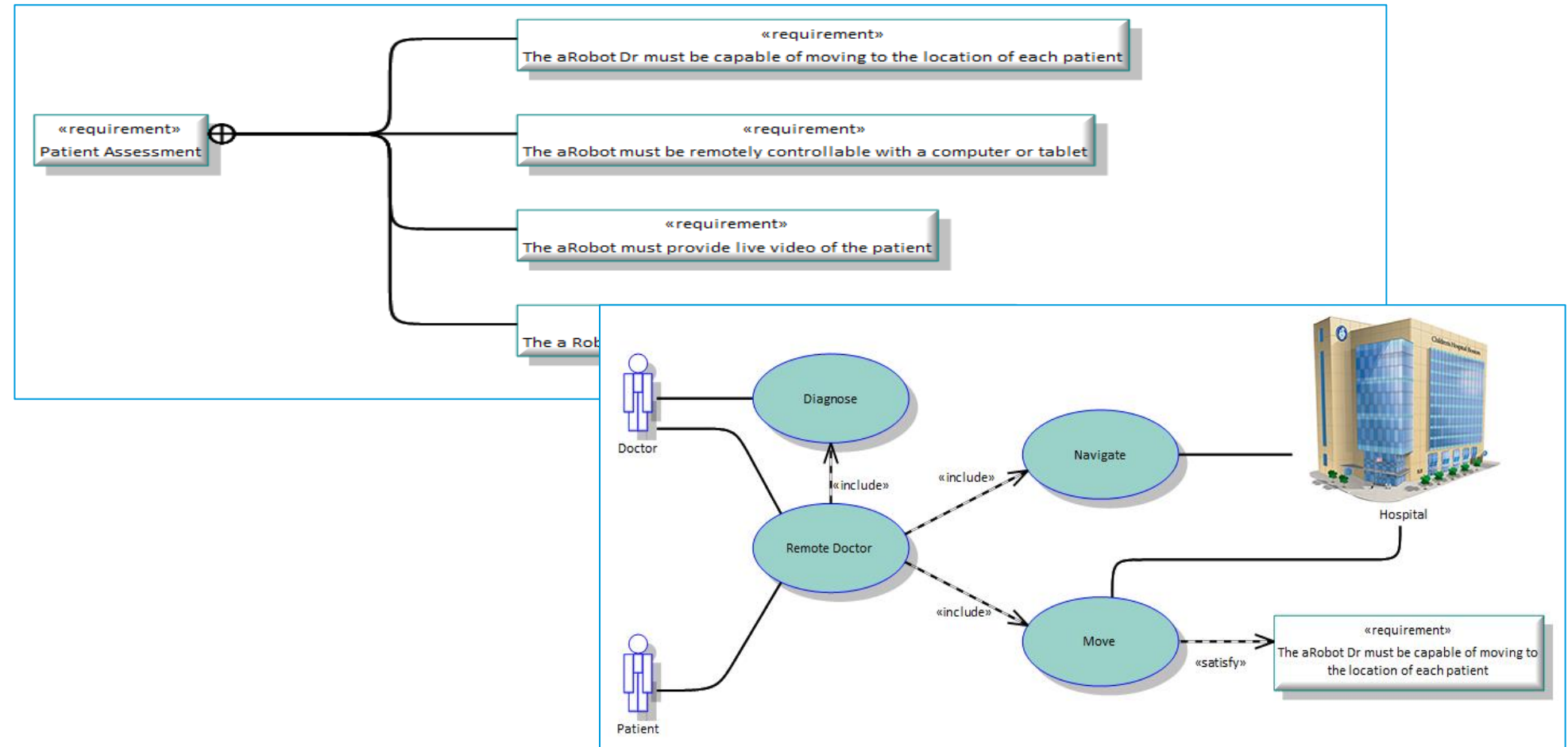


How we do it...



Capabilities

- **Requirements Engineering**
Author and manage complete requirements
- **System Design**
Collaboratively design system specifications using standard (SysML and UML) notations
- **Analyze Design Options**
Define product functional areas (hardware, software, electrical, etc.)



BENEFITS

Improve
product design

Reduce the risk of missed requirements

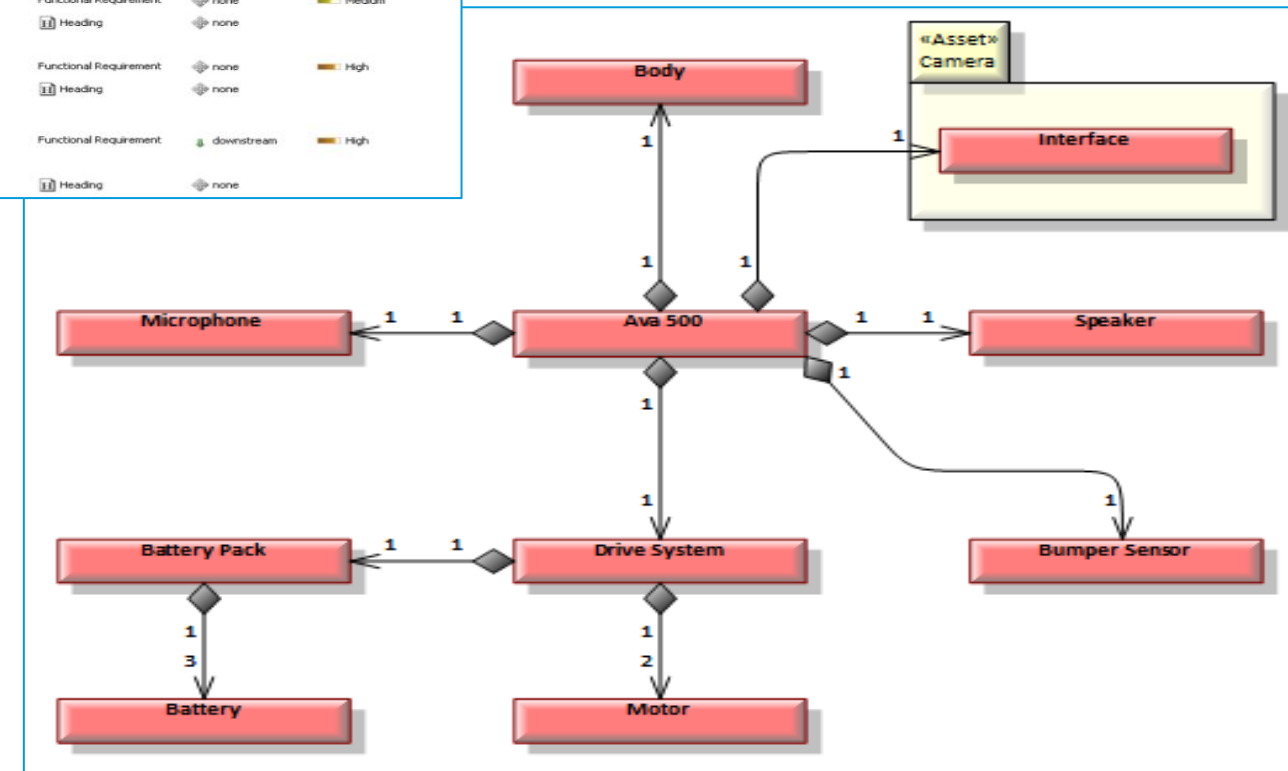
Build consensus

Make more informed
product decisions

Capabilities

- **Requirement, Model and Test Reuse**
Reuse all artifacts across design project
- **Modular Design**
System of Systems design plus architected reuse
- **Product Line Modeling**
Design families of subsystems including commonality and variation

Section	Text	Category	Trace Status	Priority
2	Market	Heading	none	
2.1	Europe	Heading	none	
2.1.1	The vehicle shall be sold into the European Market	Functional Requirement	none	Medium
2.1.2	The vehicle shall be available in left and right hand driver versions	Functional Requirement	none	Medium
2.2	USA	Heading	none	
2.2.1	The vehicle shall be sold into the US market	Functional Requirement	none	Medium
2.3	China	Heading	none	
2.3.1	The vehicle shall be sold into the Chinese Market	Functional Requirement	none	High
3	Functional	Heading	none	
3.1	Load	Heading	none	
3.1.1	The vehicle shall be capable of carrying *medium duty loads (4,5,6)	Functional Requirement	downstream	High



BENEFITS

Reduce product development costs

Shorten time-to-market

Maximize commonality

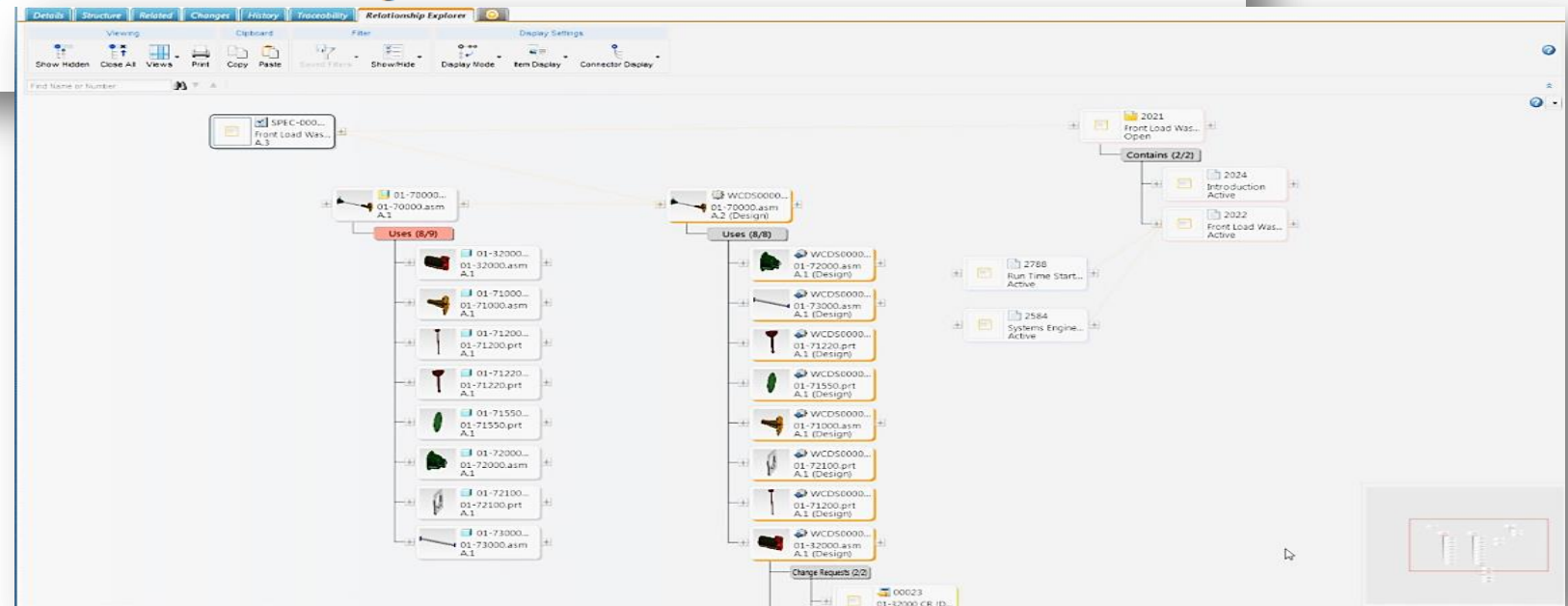
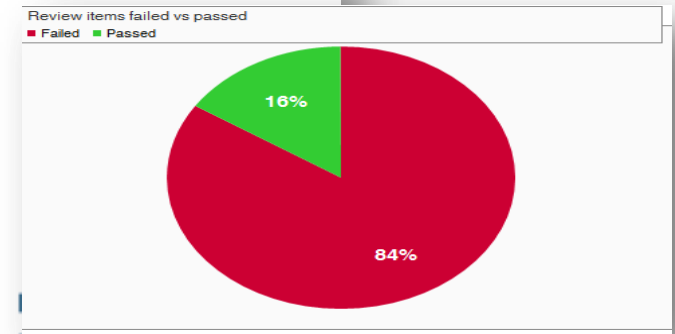
Improve product line profitability

Capabilities

- **Model Verification**
Automate design review to find problems earlier
- **Test Management**
Author and manage test cases, test sessions, and results
- **Traceability**
Capture rich traceability across design artifacts
- **Governance**
Utilize out-of-the-box systems engineering best practices

Overview by review type

Review	Found	Out of	Percentage
Activities not allocated to blocks	0	-	-
Activity diagram has empty swimlanes	0	-	-
Activity pin without connected flow	0	-	-
Constraint block without parametric diagram	0	-	-
Port on an IBD without any connections	0	-	-
Undefined blocks	18	23	78%
Undefined requirements	5	5	100%
Unsatisfied requirements	4	5	80%
Untested requirements	5	5	100%
Untyped constraint parameters	0	-	-
Untyped ports			
Untyped value types			



BENEFITS

Improve product quality

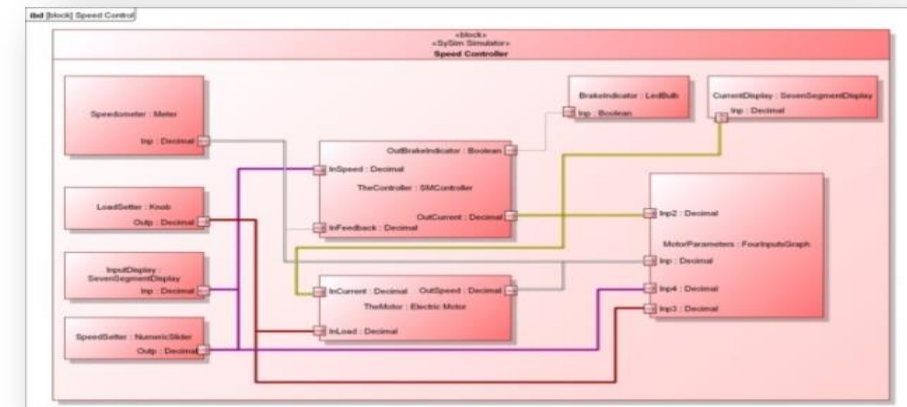
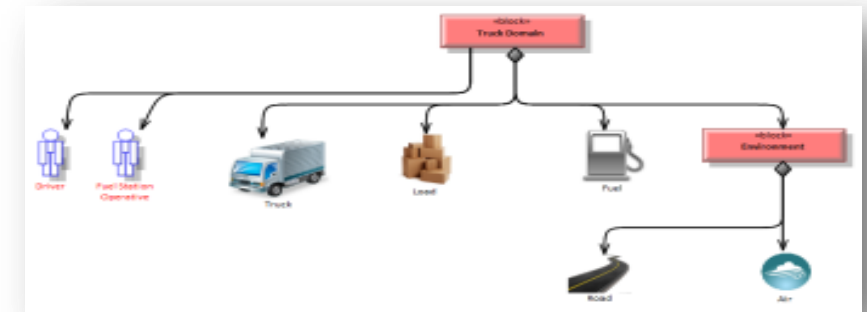
Improve visibility, communication and collaboration

Share knowledge and expertise

Enhance organizational agility

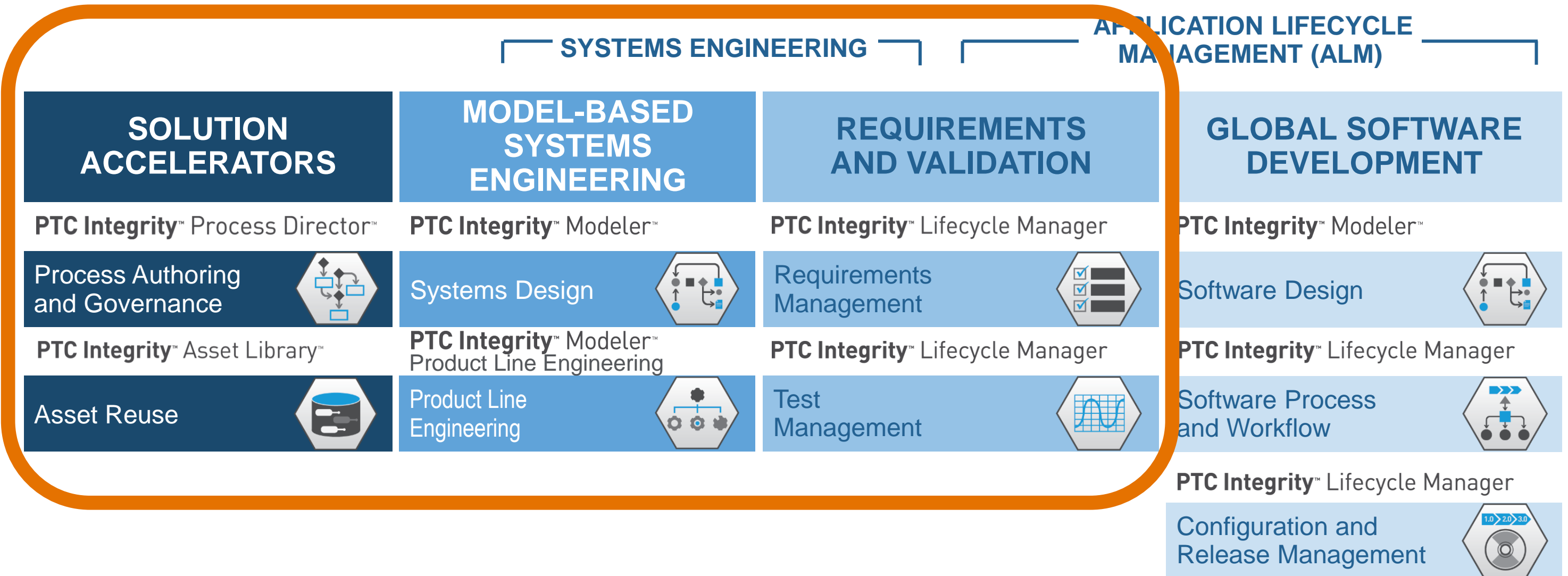
PTC Systems Engineering Solution delivers...

- **Most advanced model-based systems engineering solution in the industry**
 - Our MBSE solution is uniquely collaborative (leveraging live, multi—user models), promotes architected re-use, and enables seamlessly integrated model-based product engineering (SysML and OVM).
 - No other vendor delivers even one of these capabilities
- **Cross-Discipline Collaboration Capabilities**
 - Dramatically increase collaboration between the disciplines focusing on removing communication and automation barriers between them
 - Give all roles the needed real-time visibility into status and release readiness
- **Full lifecycle traceability**
 - Spans discipline, domain and organizational boundaries
 - Meets the needs of the most stringent safety-critical compliance reporting.
- **Systems Engineering Thought Leadership**
 - Lead and participate in the standards bodies that are inventing the future of systems engineering



A family of **software and systems engineering** products that accelerate **product innovation**.

PTC Integrity enables a holistic software and systems engineering approach by improving collaboration, automation and reuse across teams and disciplines.



System Engineering Process Governance



Challenges

Documenting your organizational processes for easy access and deployment

Large effort to author and maintain the process content

Applying the processes during program and project management

Getting engineers to actually use the processes

Negative Consequences

High cost of developing and deploying organizational processes

Missed project deadlines due to high operational ambiguity

Poor productivity due to lack of formal training

Metrics

% of on-time project deliveries

Time it takes to start a new project

Lag time between tasks



After Scenario

Efficiently defined and managed organizational processes

Accessible process information across an organization

Relevant knowledge delivered to the right users anytime, anywhere

Positive Outcomes

Improved process quality by merging industry Best Practices and your experience

Improved productivity due to widely deployed processes

Reduced costs of process definition and rollout

Justification

Reduce your process maintenance costs by 30% and your training costs by up to 40%

System Engineering Process Governance Best Practices

- Standardized System Engineering Best Practices
- Tailored System Engineering Practice Application

Key System Engineering Process Governance Capabilities

Preloaded Process Libraries	Repurpose system engineering process templates based on industry standards and best practices	Accelerate process and project plan creation
Rapid and flexible process authoring	Create processes with built in text editors and graphical drag-&-drop process flow designer	Increase process author & project manager productivity
Auto-generated BPMN diagrams	Visualize your processes with auto-generated BPMN diagrams & relationship graphs	Faster creation of process deliverables
Multi-user process repository	Manage and improve valuable process assets with version and access controls	Continual updates for changes to standards and lessons learned
Customizable processes	Automatically tailor system engineering process to each specific project	Increased application to real life projects

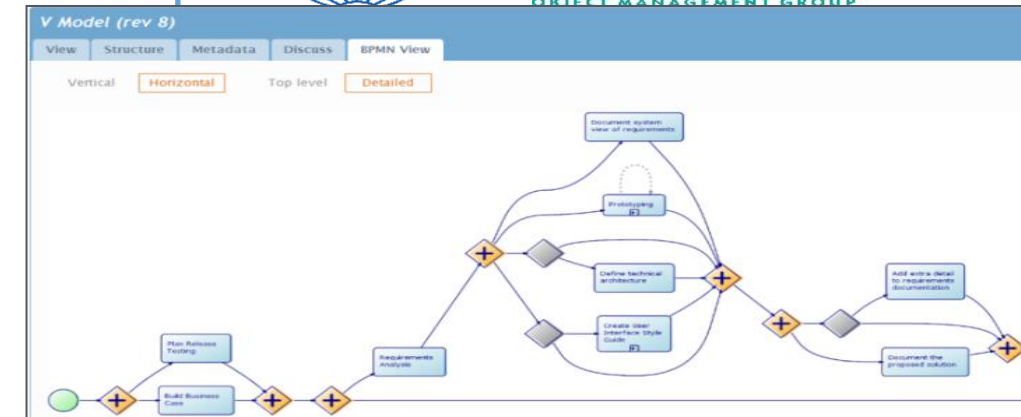
.....

Out-of-the box documentation for industry standard Software & Systems Engineering processes

Core processes enhanced with best practices, based on real life projects

Continual updates for changes to standards and lessons learned

PTC Integrity™ Process Perspective™



The screenshot shows the 'Analysis (rev 9)' process definition in the PTC Integrity software. The interface includes tabs for 'View', 'Relationships', 'Structure', 'Discuss', and 'BPMN View'. The main content area is titled 'Waterfall lifecycle' and contains the following text:

Entry conditions
Project concept known
The project concept has been proposed if not approved. The project concept will be refined during require definition and analysis. This has been updated.

Resources can be committed
Resources can be allocated to the project - i.e. they have been budgeted for and are available.

Exit conditions

Below the text, there is a BPMN diagram showing a flow from 'Design' to 'Construction' to 'System Testing'. The 'Design' task includes a note: 'The design of a module is to specify how it will be used and how it will work. This is a generic design process and by module we are referring to a subsystem, component, class, etc.'

.....

Modular processes authoring & editing

Automated process diagramming

Web-based and multi-device UI

Metrics gathering and process improvement feedback

Tailored process rules for projects initiation

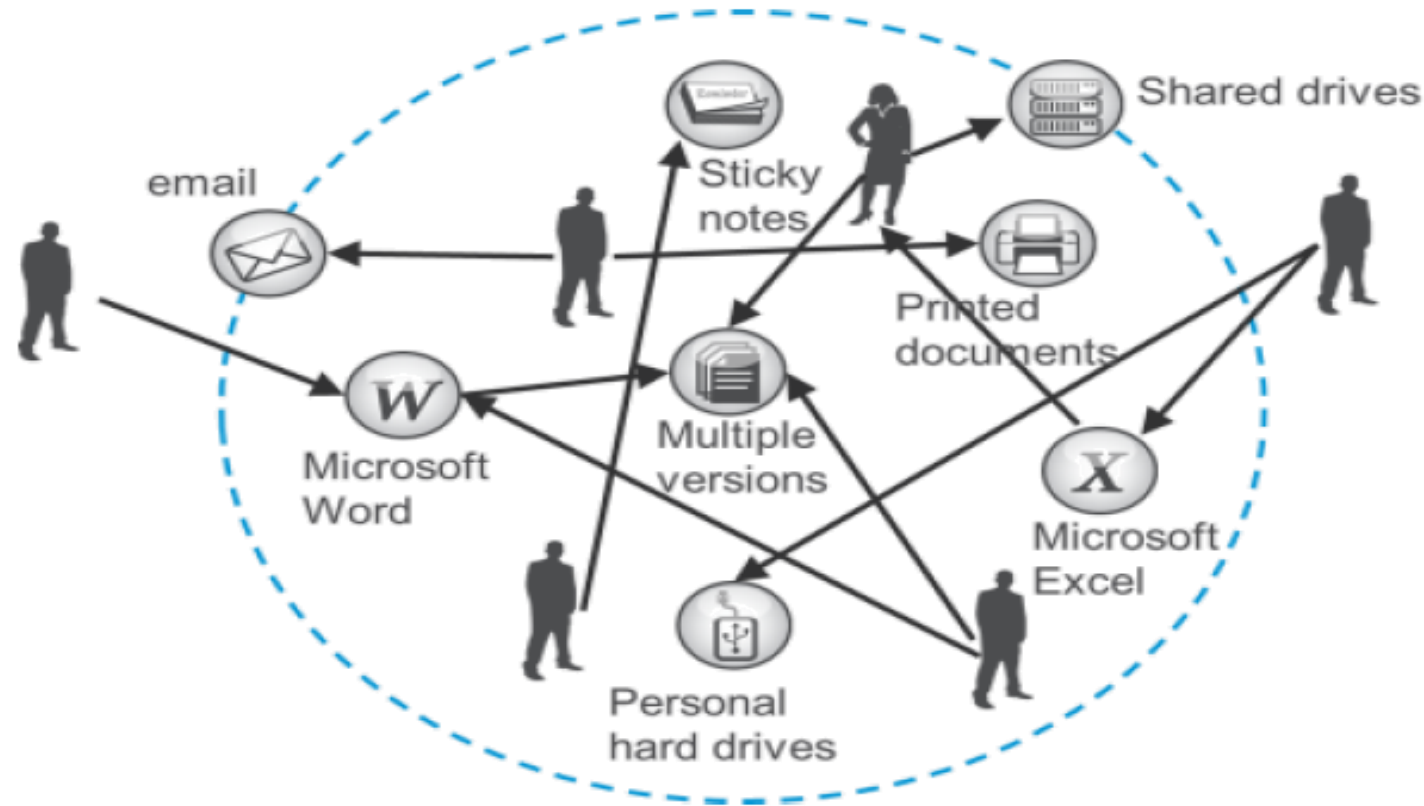
Tuesday, June 9
1:15 p.m – 2:00 p.m.

**Document, Manage and Deliver Your Processes with PTC
Integrity Process Director**

Jon Archer, PTC; Hedley Apperly, PTC

- All companies have business processes, but not all document utilizing industry standards in a reusable way that can be shared and delivered downstream. This presentation will demonstrate how PTC Integrity Process Director provides an easy to use, associative, industry standard process model to allow for ease of documentation and flexible delivery of companies unique processes.

Requirements Engineering



Challenges

.....
Unclear System Context

Requirements managed in many formats and systems

Inability to verify whether requirements are met

Limited or no systematic reuse of requirements

Negative Consequences

Inability to meet customer and market needs

Poor product quality

High development costs

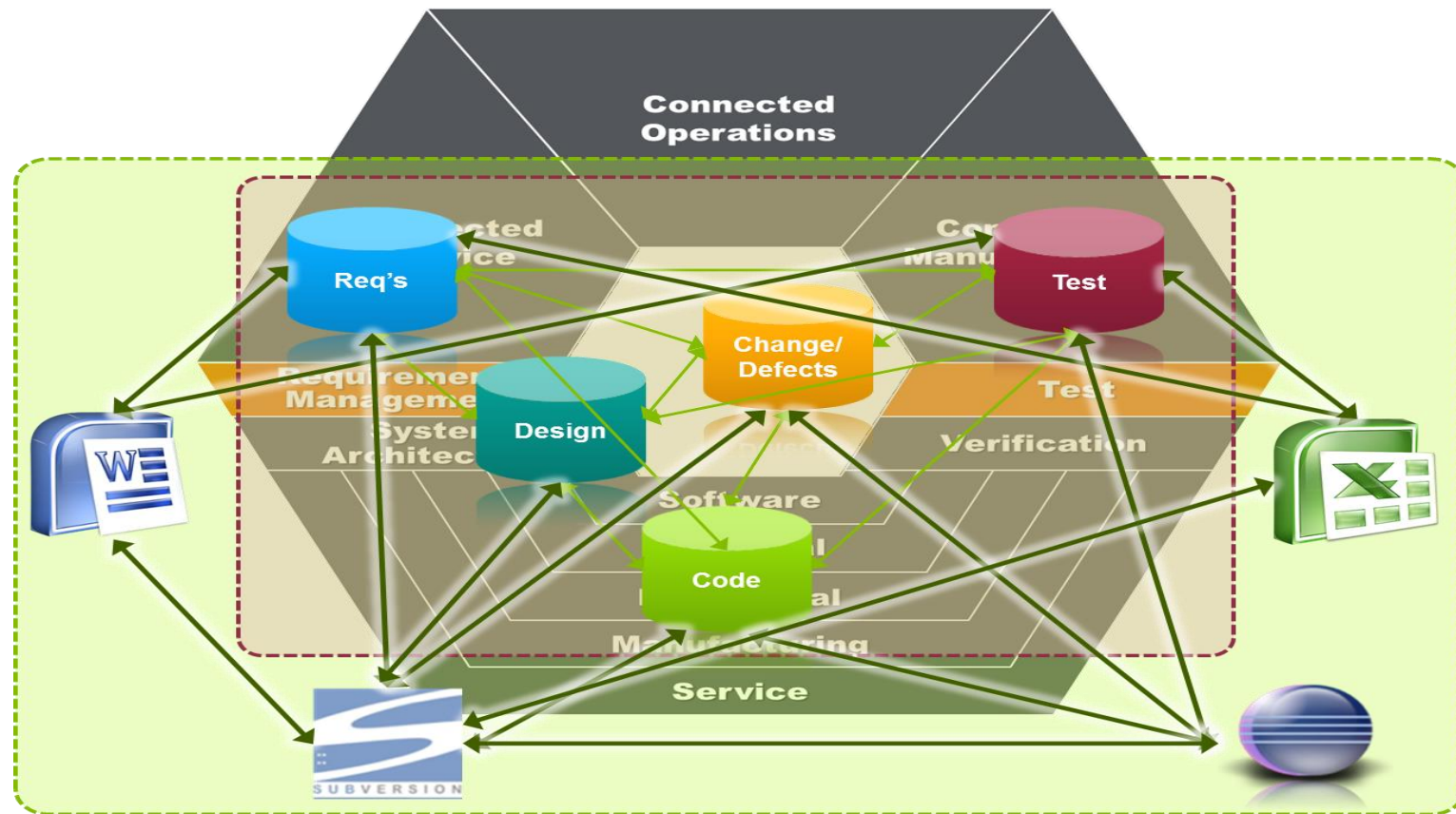
Less opportunity for innovation

Metrics

42-64% of defects originate from requirements errors

60-80% of project failures can be attributed to requirements errors

Over 20% of development cost is rework caused by ill defined and poorly managed requirements



After Scenario

Manage requirements cradle-to-grave
ensure customer needs and quality expectations are met

Address quality early and often
validate and verify requirements and designs

Positive Outcomes

- Reduce cost by reducing rework
- Respond quickly to market & customer needs
- Improved product quality

Justification

Accurate Requirements can reduce rework by 50% and reduce project costs by 15% or more.

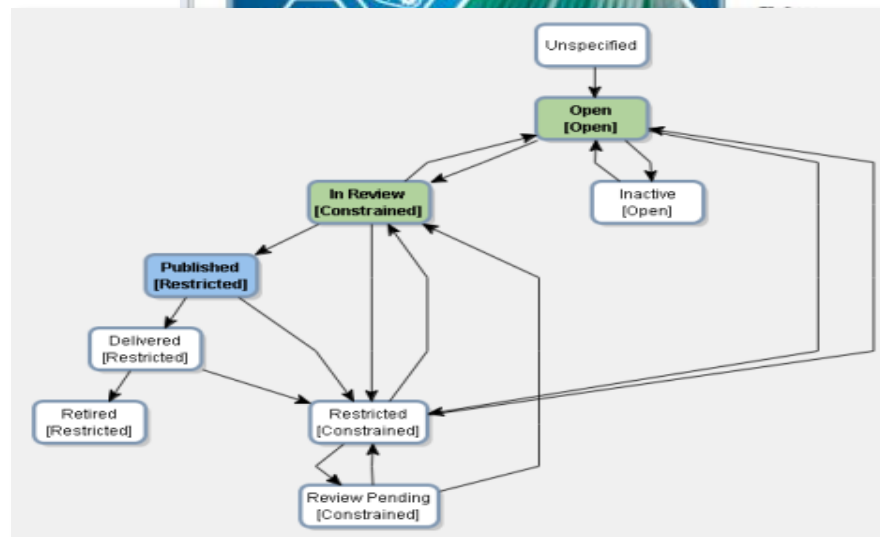
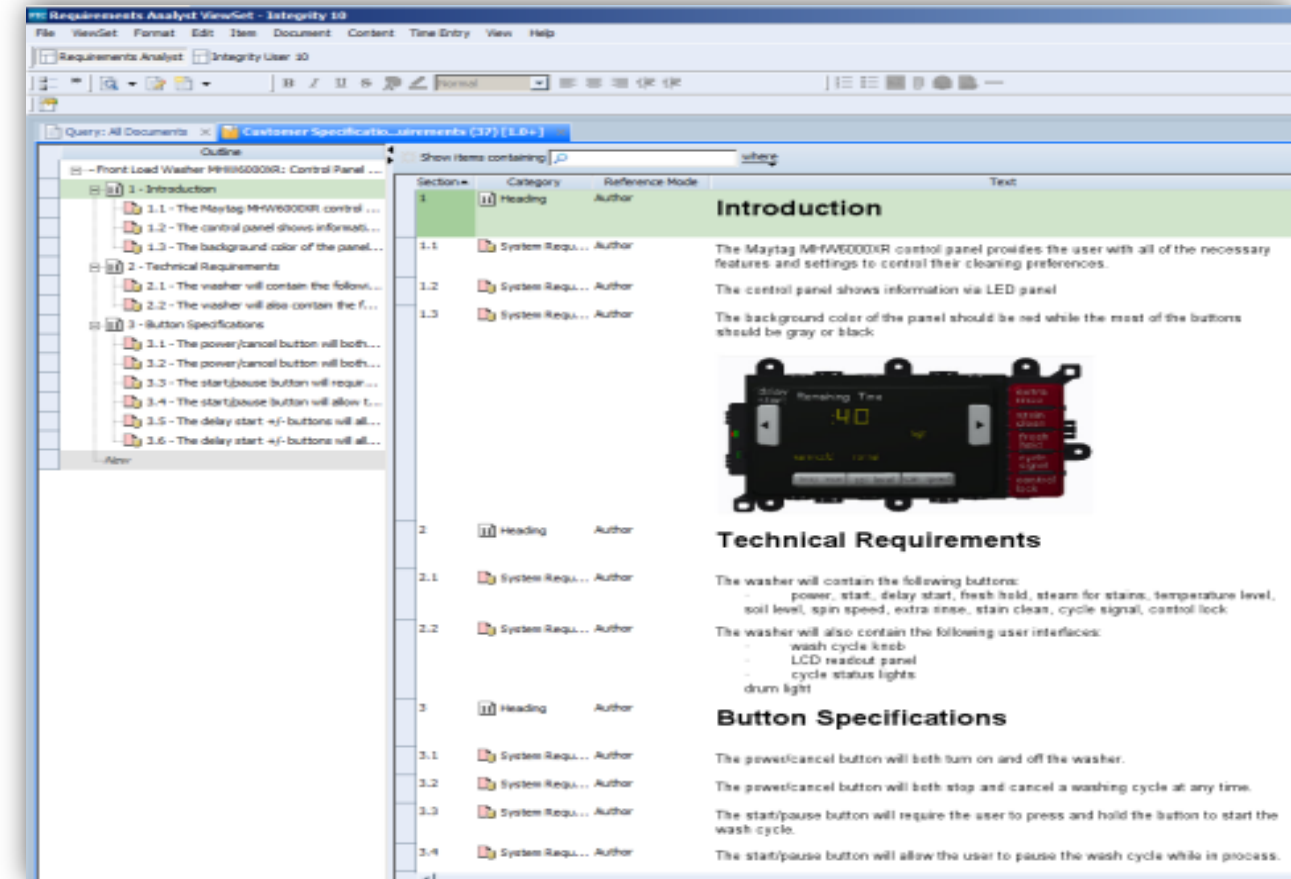
Requirements Engineering Best Practices

- Collaborative requirements definition
- Accurate requirements analysis and approval
- Comprehensive requirements traceability
- Standardized requirements and test change management

Key Requirements Engineering Capabilities

Requirements Definition	Collaboratively author all requirements in a single environment	Reduce number of duplicate, incomplete and conflicting requirements
Requirements Analysis and Approval	Process enforcement of requirements review, approval, and issue resolution	Improve product design with early resolution of requirement issues
Requirements Traceability	Traceability between customer needs, market requirements, technical specifications and designs	Reduce effort to verify that product satisfies requirements
Requirements & Test Change Management	Standardized change management process for all requirements and test artifacts	Eliminate uncontrolled requirements & test changes

Hierarchically arranged requirements
Requirement priorities and categories
Import from external sources



Process for requirements review, approval, and issue resolution

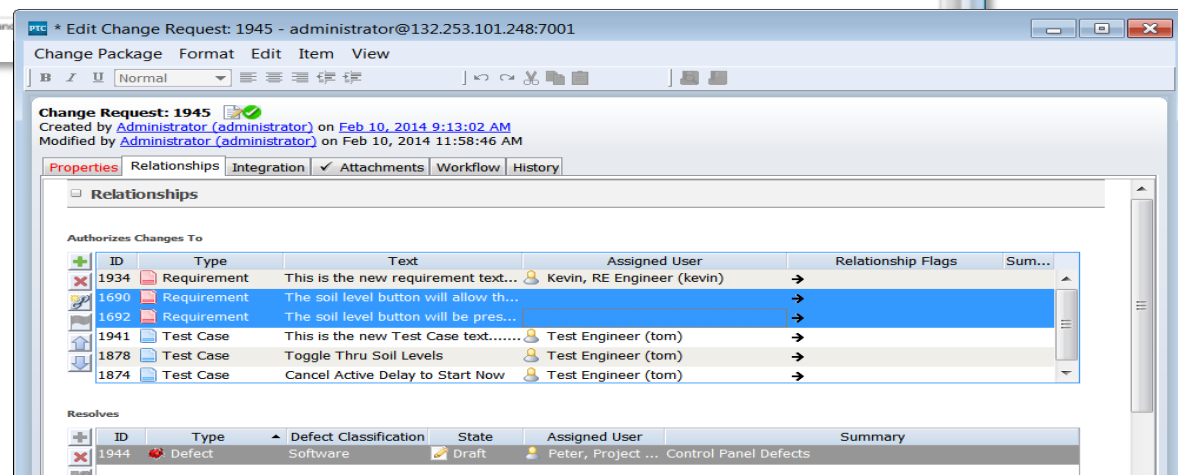
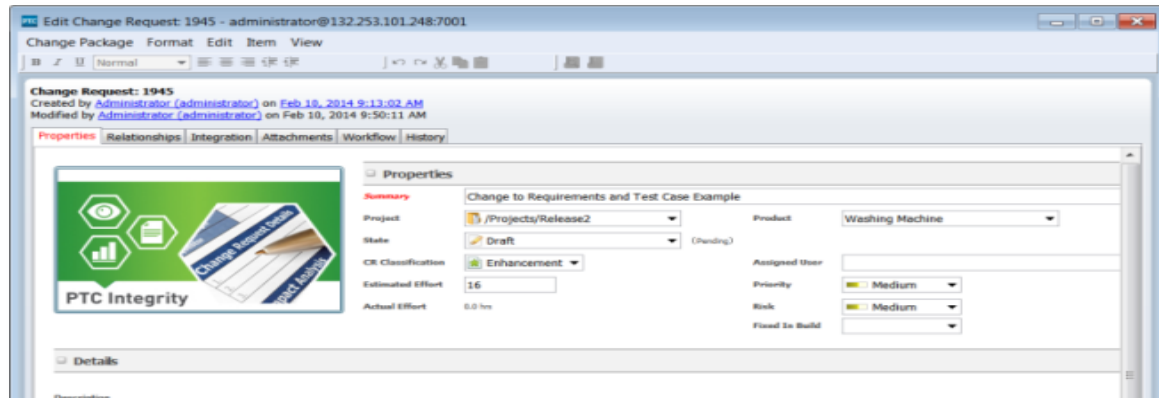
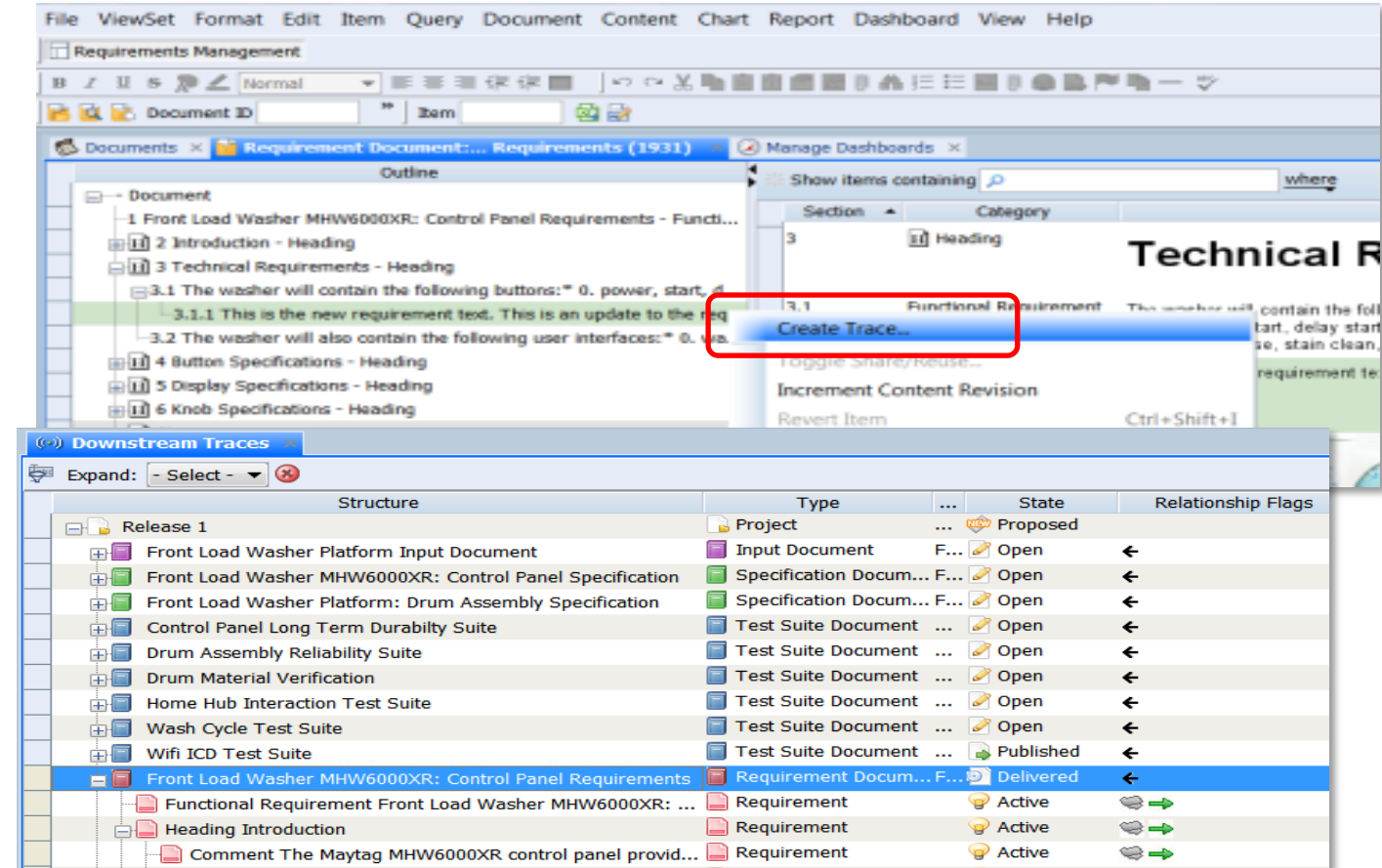
Requirements lifecycle management

Dashboards and reports on review, approval, and issue resolution status

Traceability between various requirements

Allocation to tests, product structures, and design data

Reporting to identify orphaned requirements

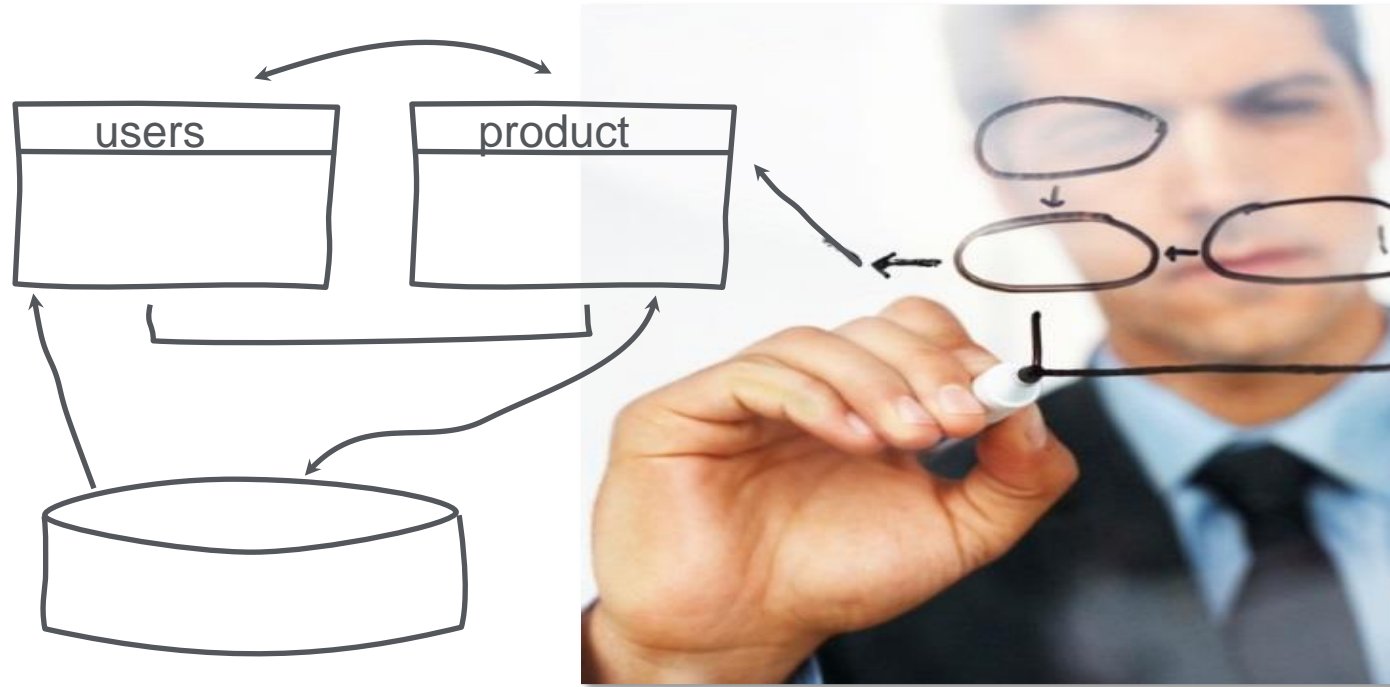


Manage and communicate change requests to requirements and tests

Trace relationships to assess change impact

Track the change history for requirements and tests

System Design



Challenges

Poor collaboration between disconnected engineering disciplines when designing complex systems

Labor intensive and error prone design reviews

Lack of early system design validation causing expensive, late stage rework

Negative Consequences

Higher risk of building the wrong system

Increased costs of later stage errors

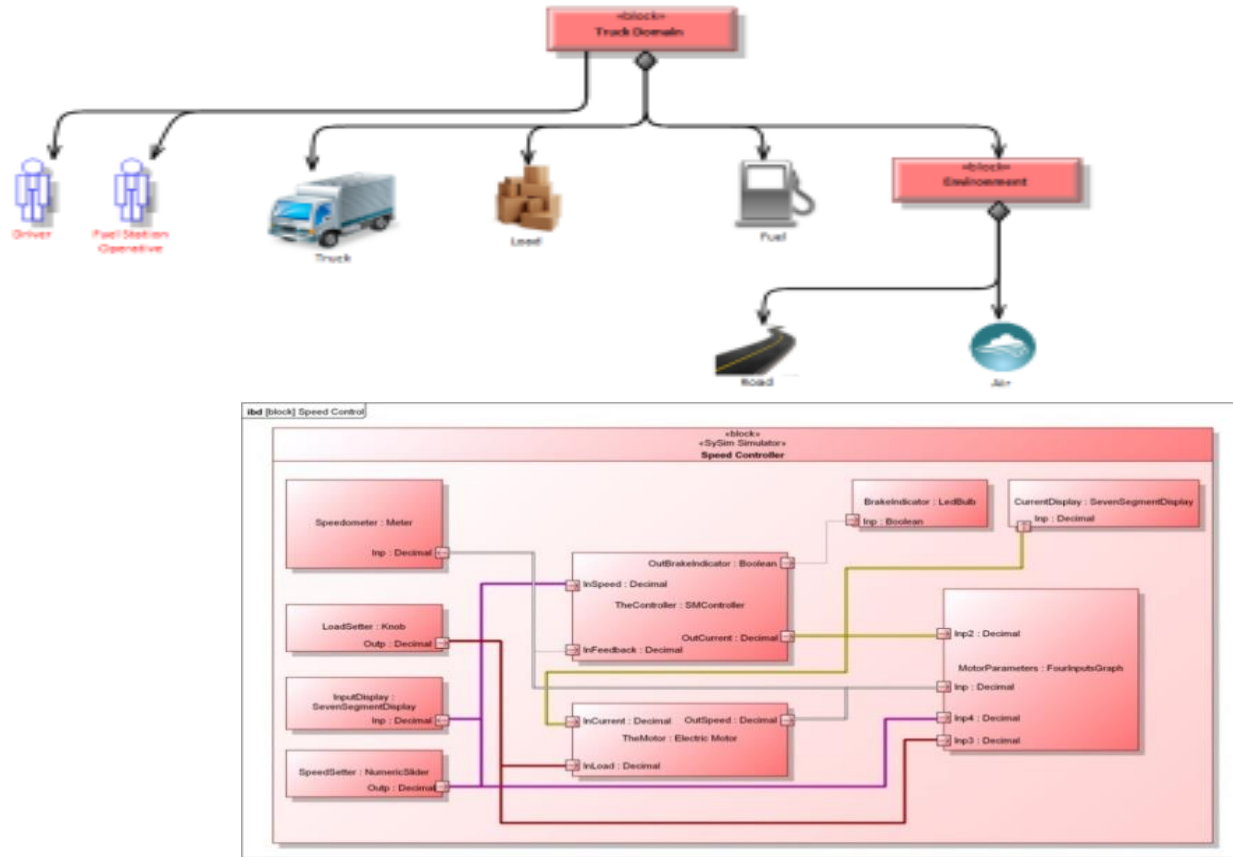
Rework delays time to market, increases development costs, and lowers product quality

Metrics

Time spent on rework in late development stages

Time spent understanding impact of change on complex systems

Number of unsatisfied requirements



After Scenario

Model-based Systems & Software Engineering (MBSE)

with standards based SysML/UML graphical modelling

More Efficient Engineering Process

with linked architecture, requirements, Systems of Systems, Sub-Systems & Integrations

Design & Build the Right Systems

with end to end traceability with automated reviews, simulation & trade studies

Positive Outcomes

Improved Quality through Early Design Review and Consistency

Bring Systems to Market Faster with Parallel Design Effort

Cost Reductions from Design and Development Automation

Justification

55% Reduction in Total Development Cost per Project with 5% improvement for on time delivery when implementing a Model-based Systems & Software Engineering (MBSE) solution

System Design Best Practices

- Model Based System Engineering
- Automated System Design Review
- System Level Simulation
- System Model Management
- System Model Publishing
- Domain Specific Languages Implementation

System Design Capabilities

UML, SysML based System Models

Consistent, high quality system models to communicate requirements and consider design alternatives

Reduce time and effort to design complex product systems

Automatic Model Review

Check models for completeness, correctness and consistency

Find errors in models early, to ensure robustness of design

System Model Behavioral Simulation

Simulation graphic driven behavioral simulation of complex system designs

Early detection and correction of errors, inconsistencies & unwanted side effects

System Model Publishing

Produce high quality System Model documentation, formatted to suit the audience

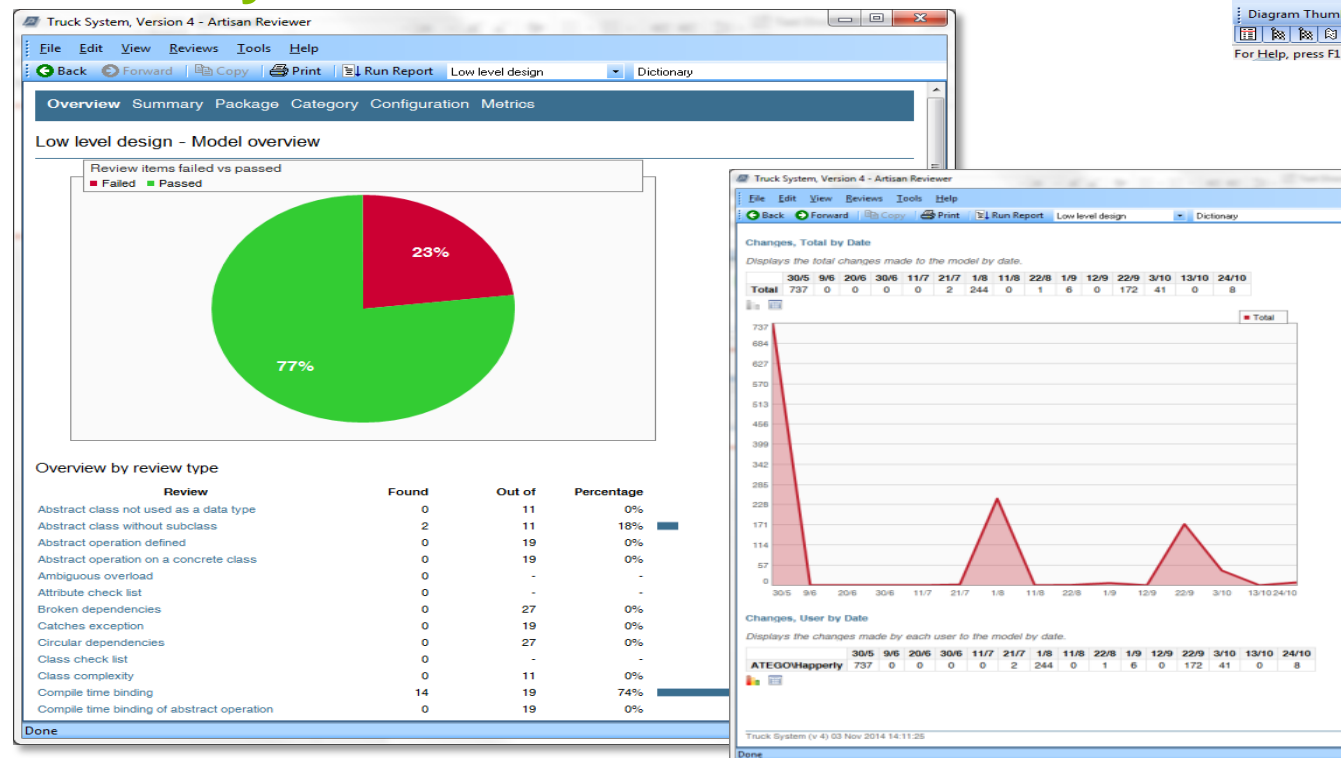
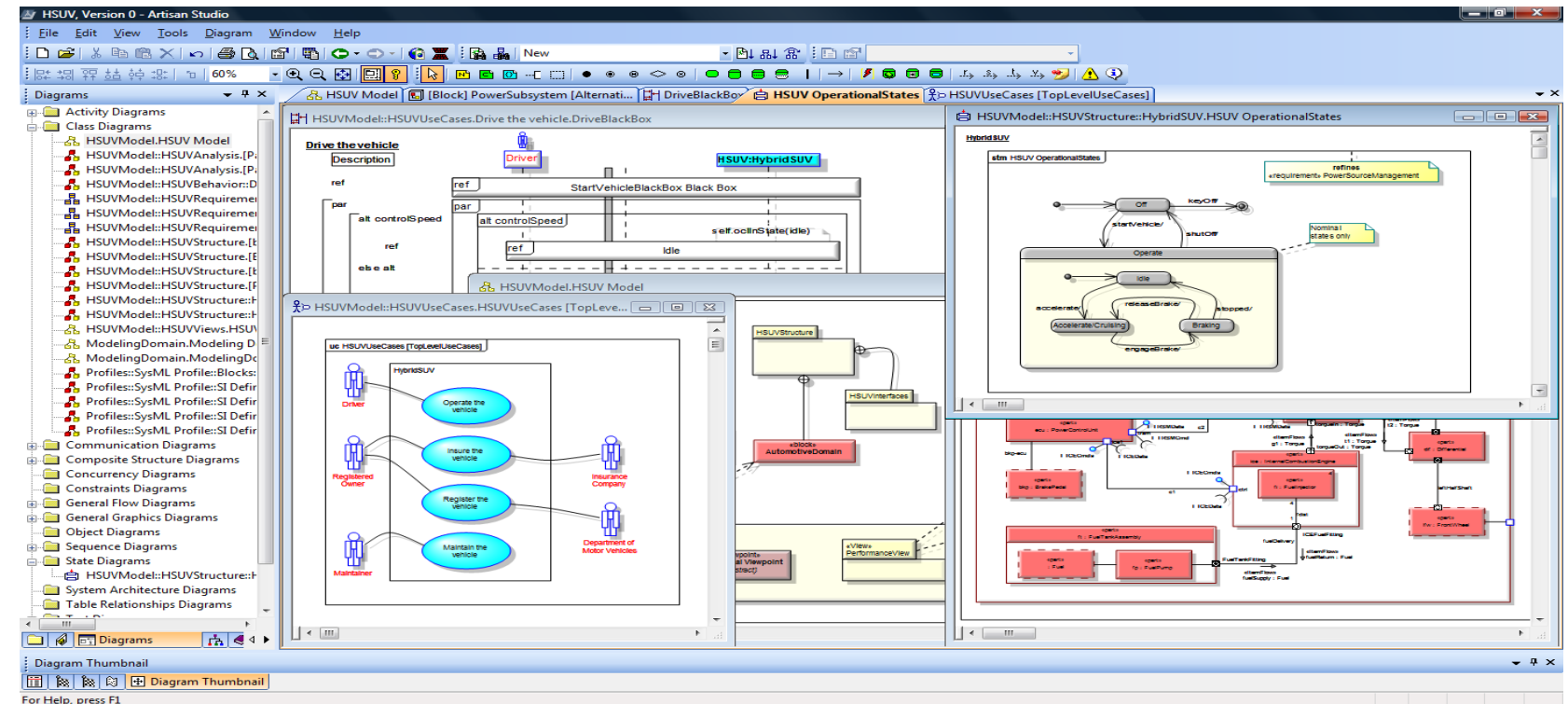
Once source of data drives multiple publication formats

Model-based Systems Engineering

**Graphically define product context
and stakeholders**

**Map and trace from requirements
to the design**

**Model architecture including
system functions and structure**



Automated System Design Review

Active design mentoring

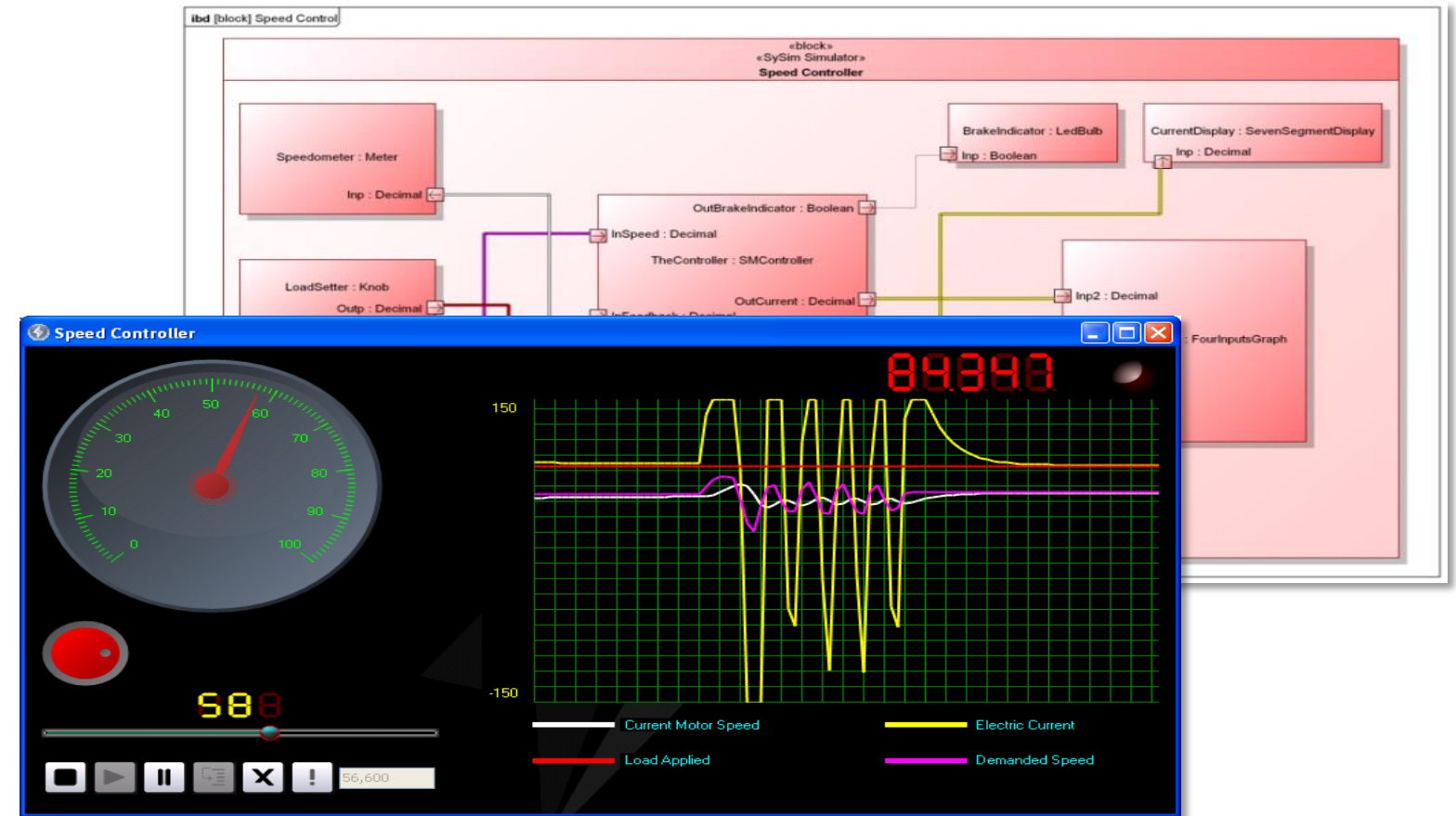
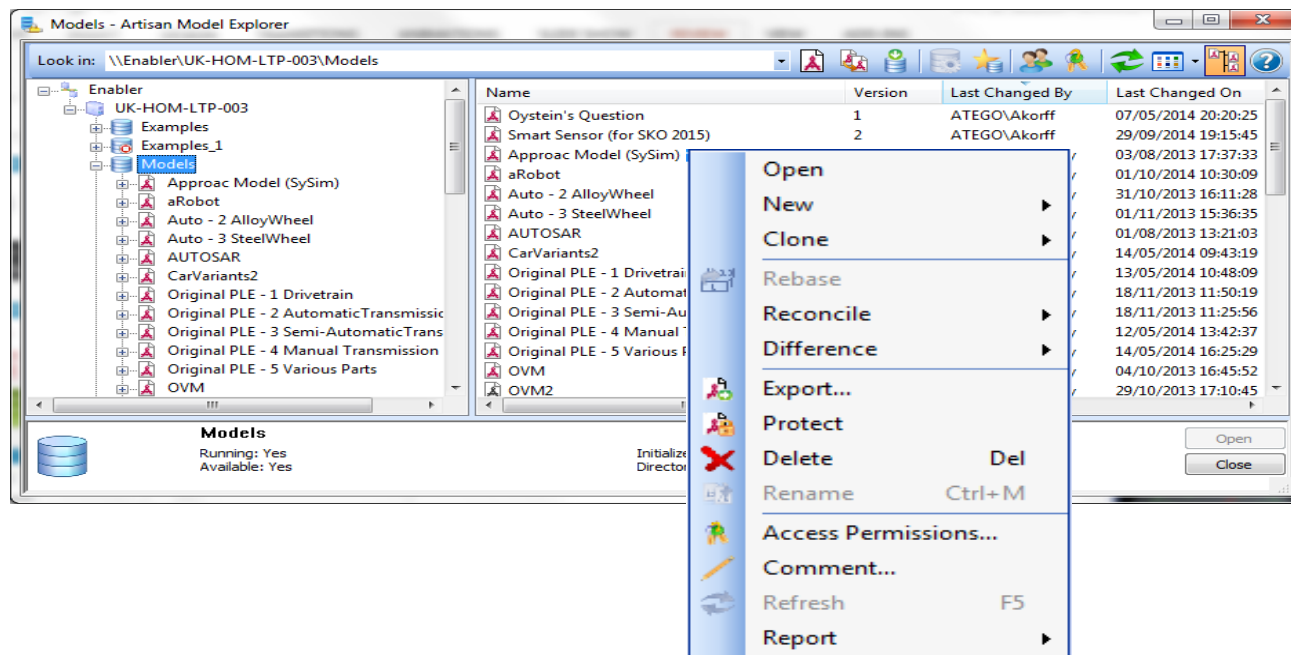
Model quality management dashboard

System Level Simulation

Validate early complex system
behavior

Graphically design drop and play
system simulation

Co-simulate between SysML models,
software code and third party
simulation tools



System Model Management

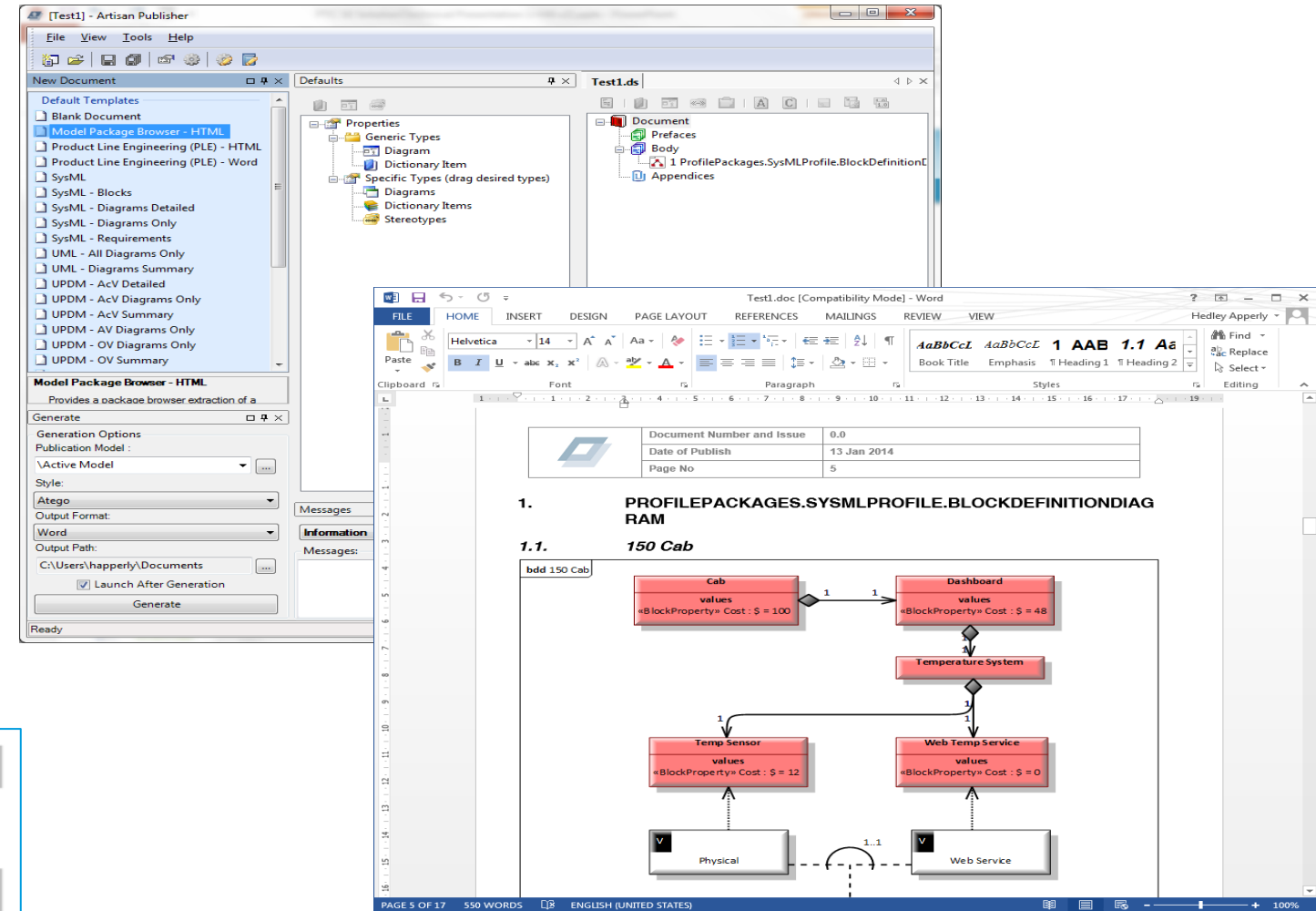
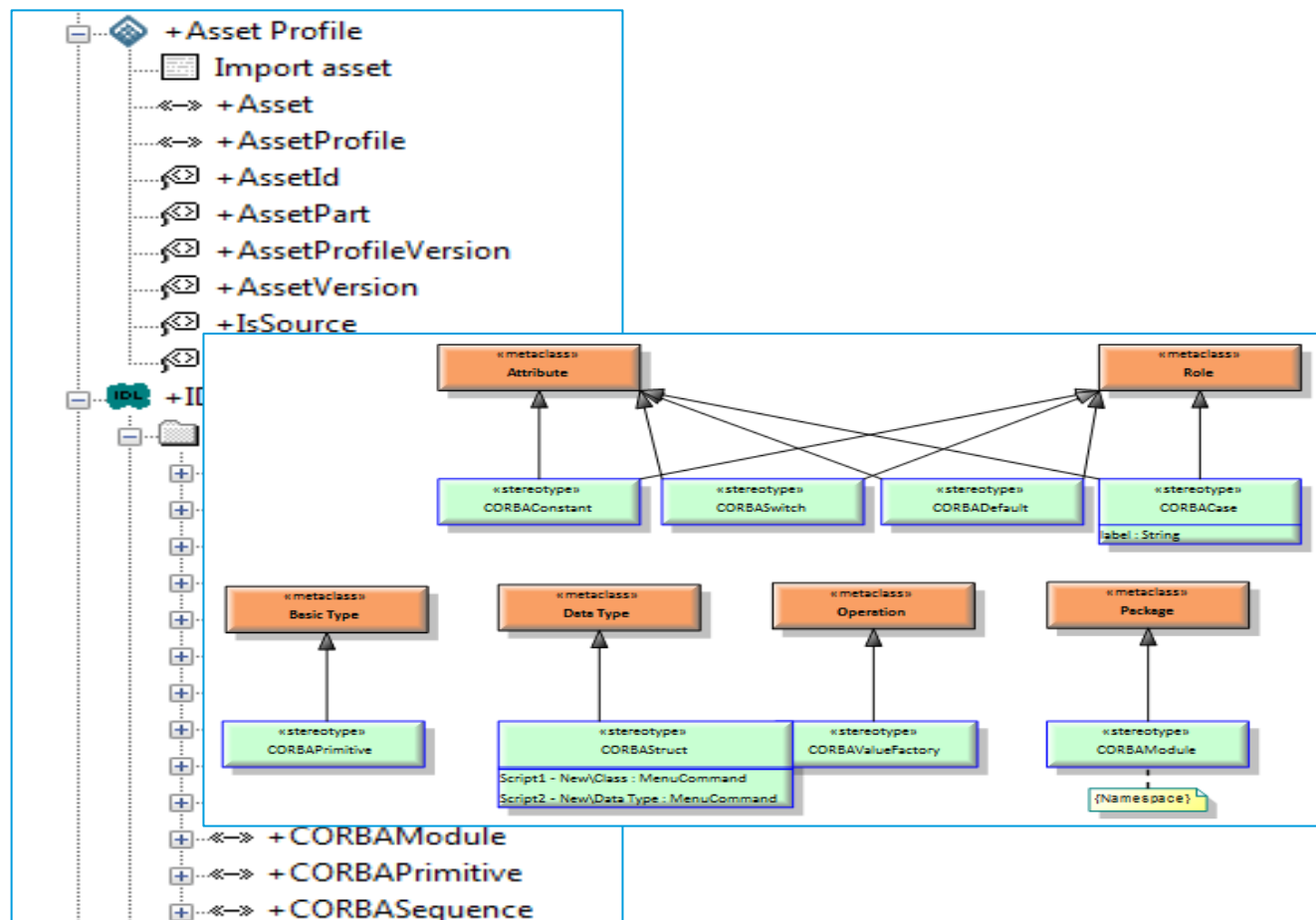
Multi-user database administration

Model configuration management

System Model Publishing

Automated Word & HTML model
documentation

Document template definition



Domain Specific Language (DSL) Implementation

UML & SysML extension Profiles

Modeler tool feature extensions

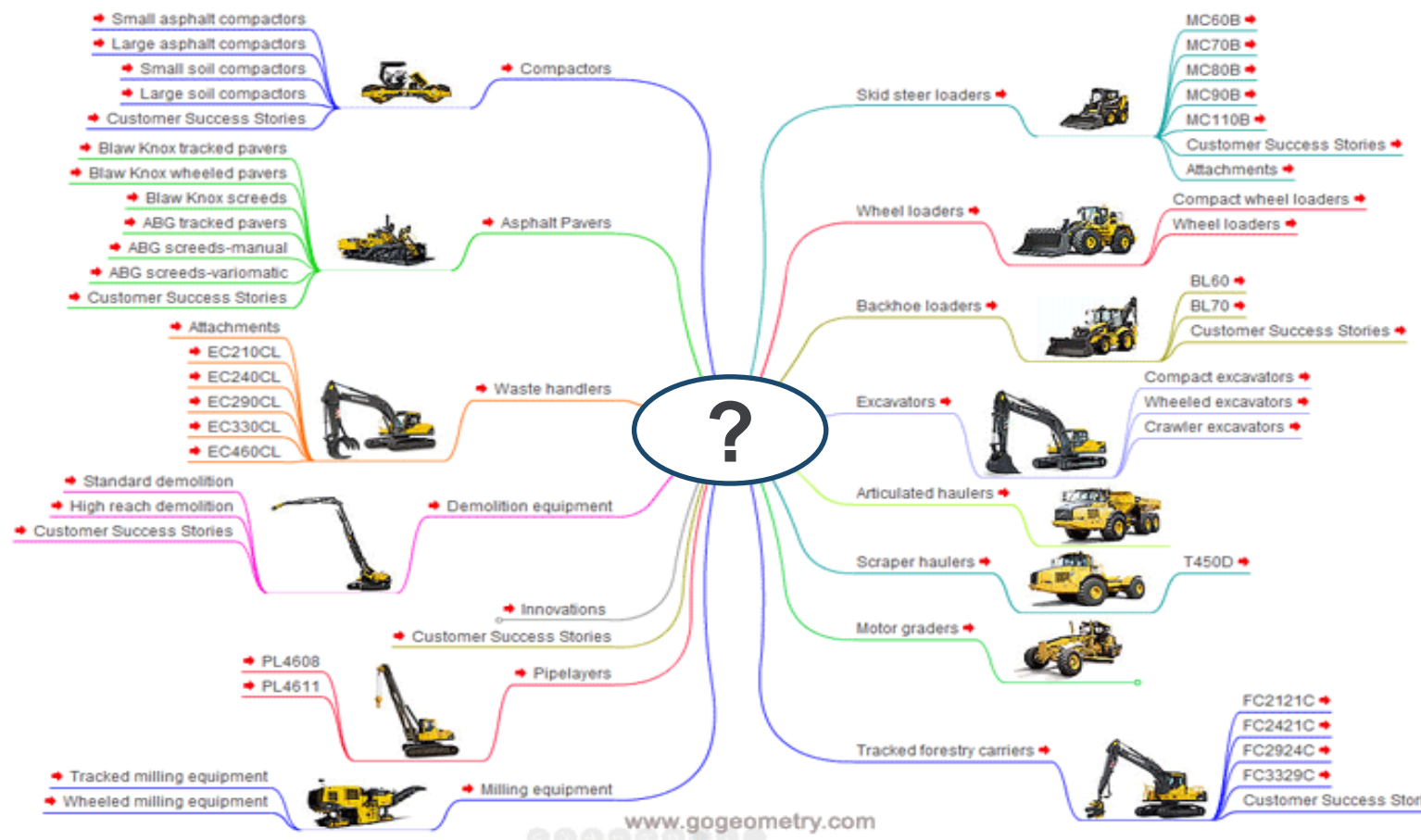
Tuesday, June 9
2:15 p.m – 3:00 p.m.

Introduction to Model-Based Systems Engineering

Hedley Apperly, PTC

- Studies show that a model-based systems engineering (MBSE) approach can help organizations reduce risk and improve business outcomes when compared with alternative approaches, delivering 23% more projects on time, at 62% lower cost, according to a survey of 667 engineering respondents. This presentation explores the vision and value of model-based approach, and includes a 10-minute demo of model-based systems design in action.

Product Line Design



Challenges

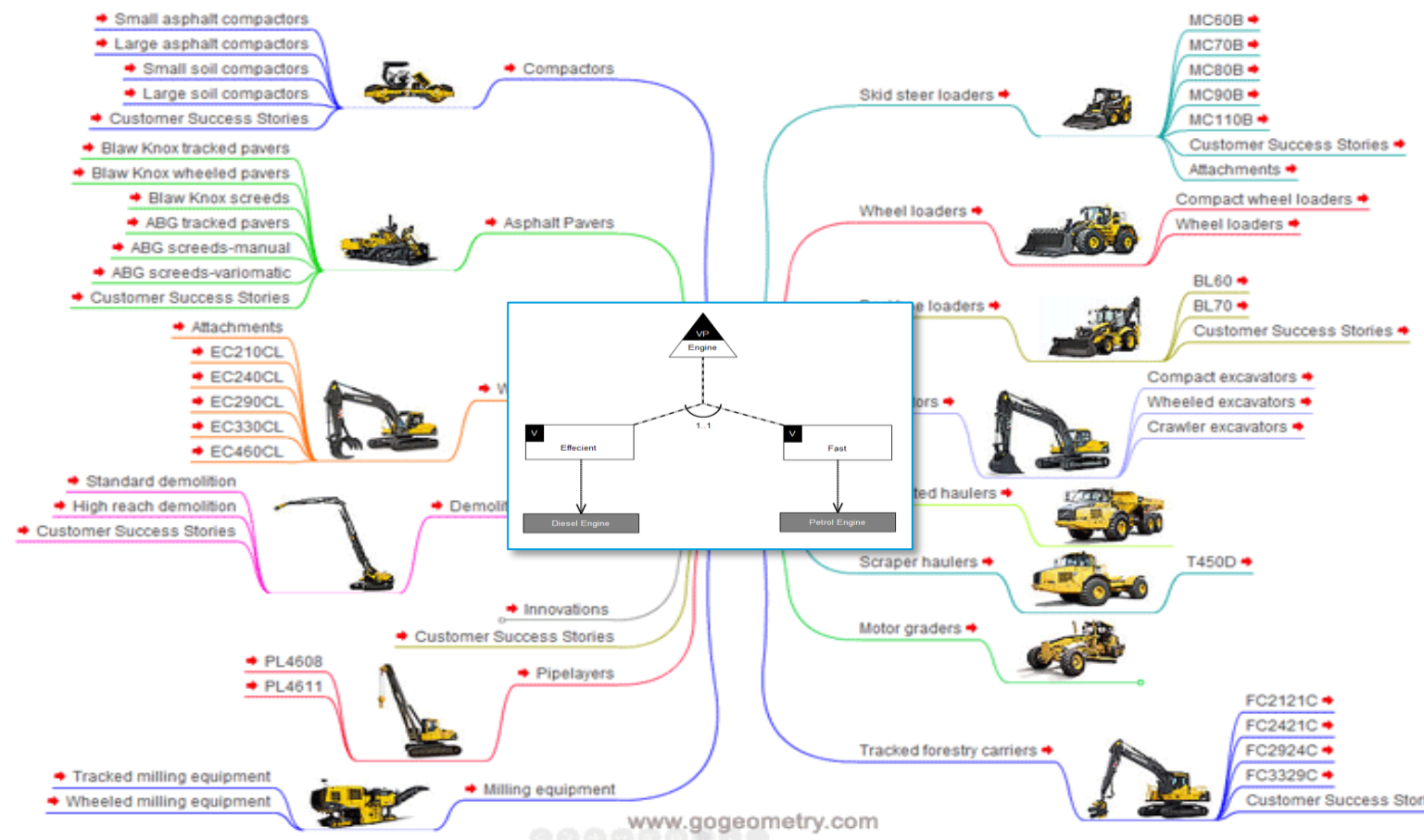
- Increasing number of product families
- Increasing number of products in families
- Understanding product similarity
- Understanding product variations

Negative Consequences

- Duplication of effort
- Lack of overall product family visibility
- Increased change effort due to unknown impact of change

Metrics

- Cost and time savings from reuse
- Trade analysis effort
- Time to develop product variations



After Scenario

Single system product line design instead of creating a multitude of products

Managed system commonality and variation

Model driven system trade-off analysis to inform business decision making

Positive Outcomes

Less duplicated effort with optimized reuse

More commonality between designs and implementations

Managed product line complexity

Justification

MB-PLC delivers 21% more projects on time compared to just SE and 62% reduction in total development cost per project.

(EMF 2013 Independent Survey Results from 667 Systems engineering respondents)

Product Line Design Best Practices

- Model Based Product Line Engineering
- Architected Modular Design
- Product Line Analysis and Validation

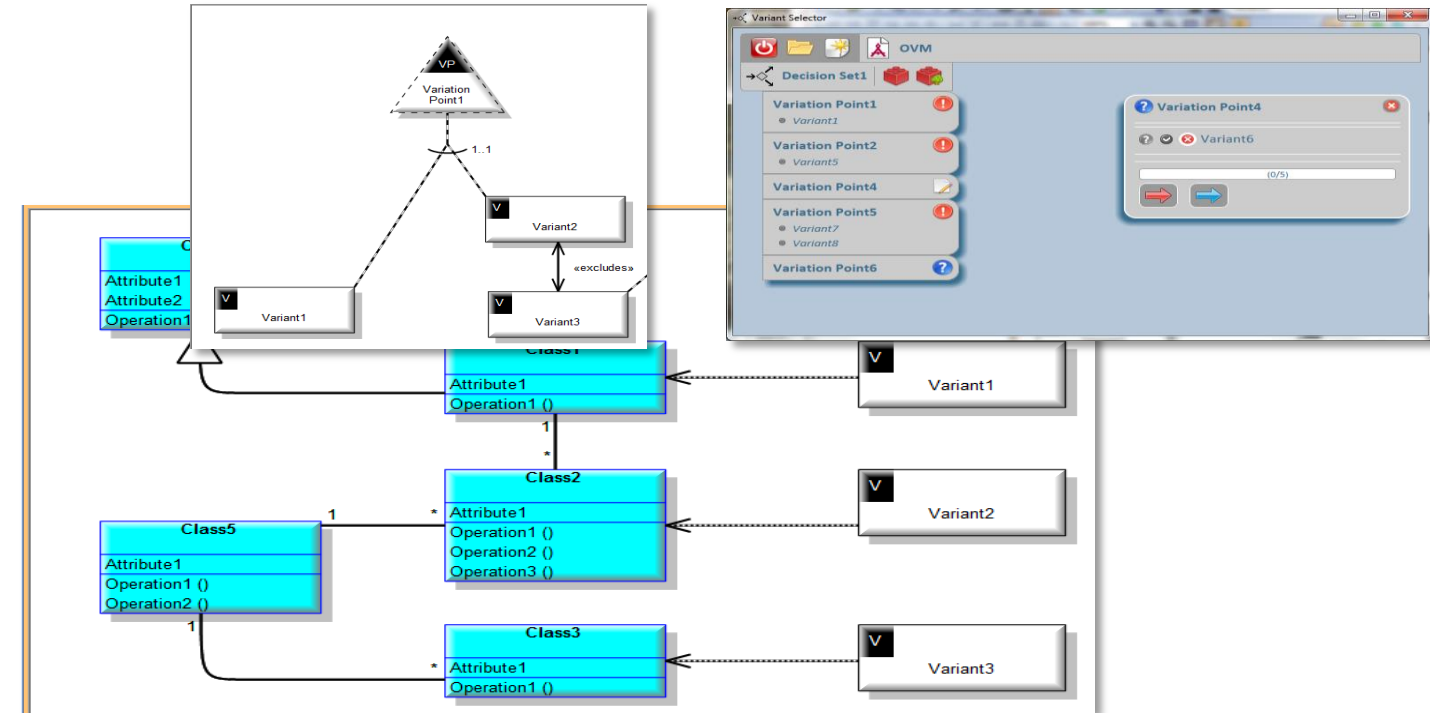
Key Product Line Design Capabilities

Single system model-based product lines	Products designed as a single system containing product variation and commonality	Reduced time to develop product families
Asset libraries with common modules	Scalable repository for asset publication, management and reuse	Accelerated projects by reusing pre-designed and implemented sub-systems, components and services
Model-driven system trade-off analyses	Product Line validation with static model reviews, and visual, dynamic system level simulation	Reduced cost with early functional, non-functional and cost comparison

Model Variability with Variation Points and Variants

Select Family features via Variant Selector or Decision Set Editor

Perform product trade studies using system analysis

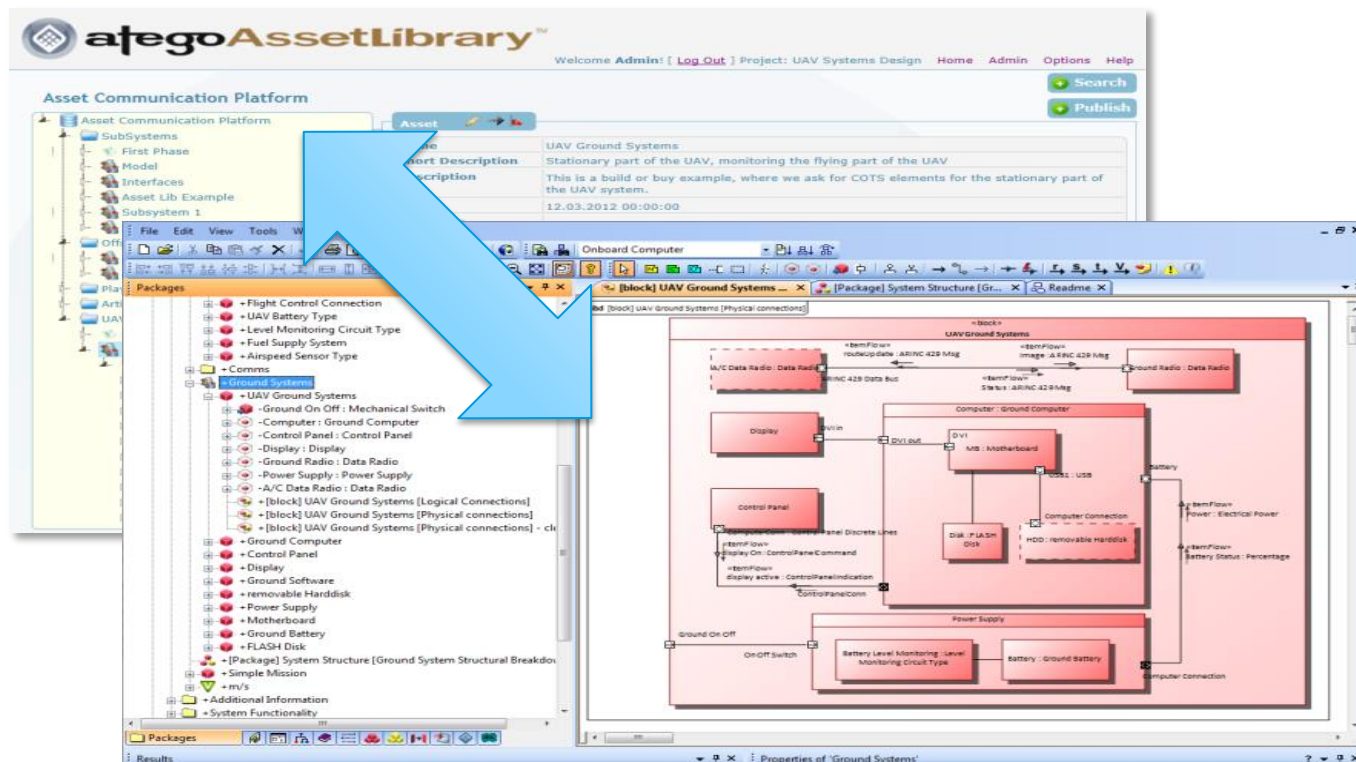


Publish assets into highly scalable repository for management and reuse

Publish common modules from Asset Library for design reuse

Connect system models using asset meta-data and traceable links

Publish and use variable assets in product line models

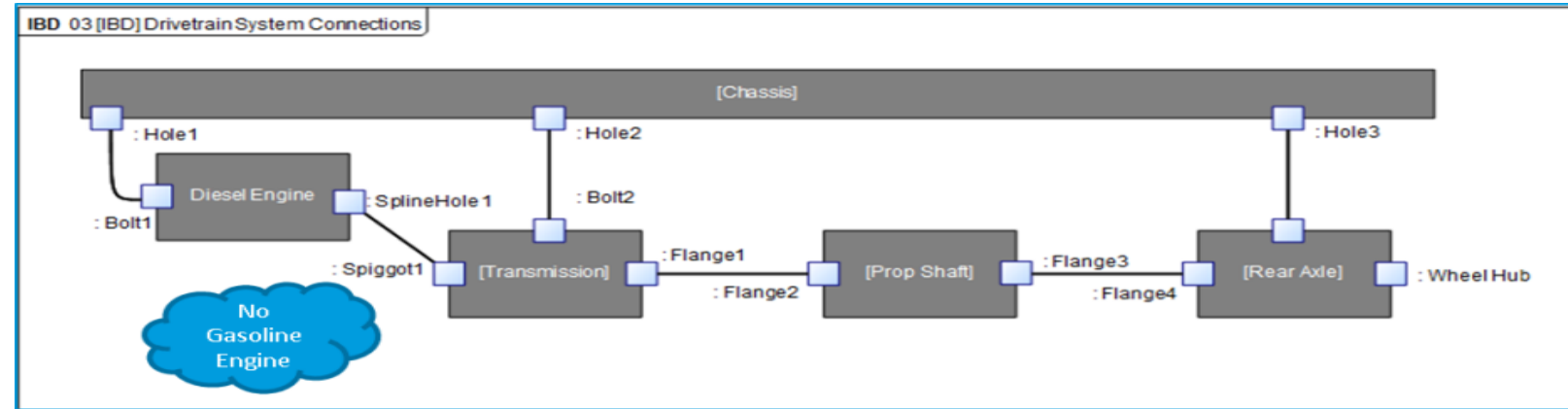


.....

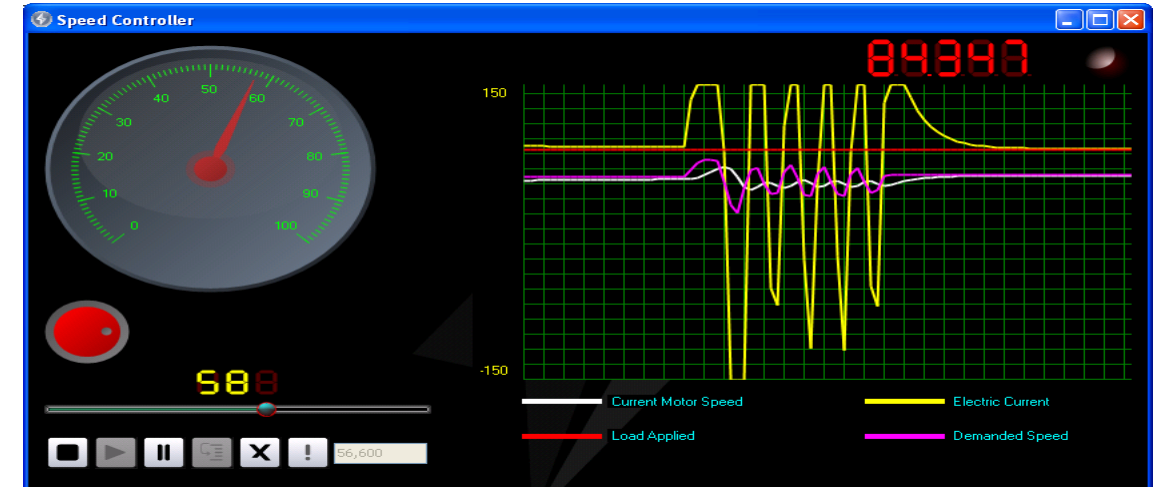
Check design quality and system interface consistency

Perform trade studies using parameterized budget reporting

Validate Product Line with visual, dynamic system level simulation



		A	B	C	D	E		F	G
		Name	Quantity	Estimate	Margin (%)	Mass (in kg)			
						No Margin	With Margin		Budget
1		Ava 500 (Ava 500)	n/a	0	0.00%	99	99		100
2									
3		Body (Ava 500::Ava 500)	1	12	0.00%	27	27		27
4		Dr Screen (Ava 500::Ava 500)	1	15	0.00%	15	15		15
5		Bumper Sensor (Ava 500::Ava 500)	1	2	0.00%	2	2		2
6		Microphone (Ava 500::Ava 500)	1	2	0.00%	2	2		2
7		Speaker (Ava 500::Ava 500)	1	2	0.00%	2	2		2
8		Drive System (Ava 500::Ava 500)	1	0	0.00%	66	66		66
9		Motor (Ava 500::Ava 500)	2	15	0.00%	15	15		15
10		Battery Pack (Ava 500::Ava 500)	3	0	0.00%	12	12		12
11		Battery (Ava 500::Ava 500)	1	12	0.00%	12	12		12
12									



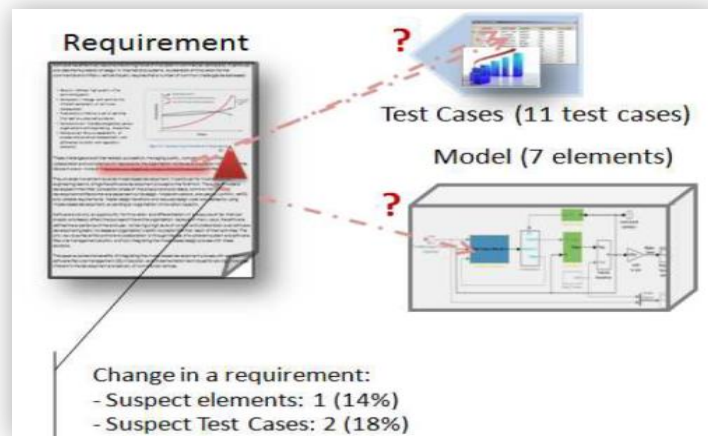
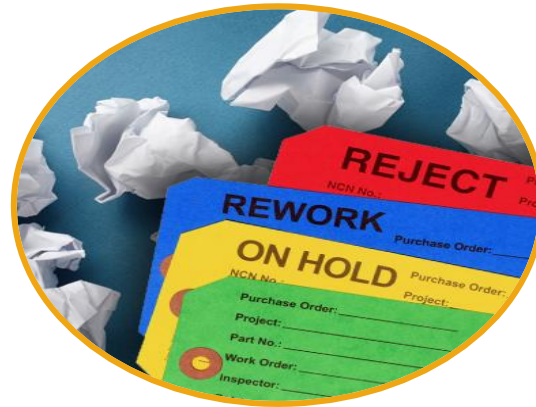
Tuesday, June 9
5:00 p.m – 5:45 p.m.

**Design Better Platforms, Product Lines and Variants with
PTC Integrity Modeler Product Line Engineering**

Hedley Apperly, PTC

- Today's manufacturers have transitioned their focus from individual products to platforms and variants that give them speed and agility. In this presentation, we explore new approaches to design reuse that help you make more informed product line and variant design decisions. Learn how to assess product line alternatives and make more informed design decisions.

Test Management



Challenges

Test planning disconnected from requirements and product design

No traceability to verify test coverage

Uncontrolled requirements & test changes

Disjointed systems and processes for managing tests, results and defects

Negative Consequences

Requirements errors create late cycle rework

Uncoordinated change creates waste and inefficiency

Rework causes delayed time to market, increased development costs, and lower product quality

Metrics

% of engineering time spent on rework

% of on-time deliverables

% of engineering time on compliance tasks

% of requirements and test reuse



After Scenario

Complete Test Coverage
via integrated test management system

Quick Impact Assessment
with full traceability between requirements, test cases, defects

Full visibility into test planning & execution
via powerful querying, charting and reporting

Positive Outcomes

Reduce rework by up to 50%

Reduce change order cycle time and cost associated with change orders

Reduced time to market

Lower development costs

Justification

Ingersoll Rand selected the PTC System Requirements and Validation Solution as the single source of truth for product requirements and test data. By eliminating several other tools and redundancies, they compressed their development processes by 30%, directly supporting their business goals.

Test Management Best Practices

- Comprehensive test traceability and coverage
- Collaborative test definition and authoring
- Managed test execution and results analysis
- Integrated test planning
- Standardized requirements and test change management

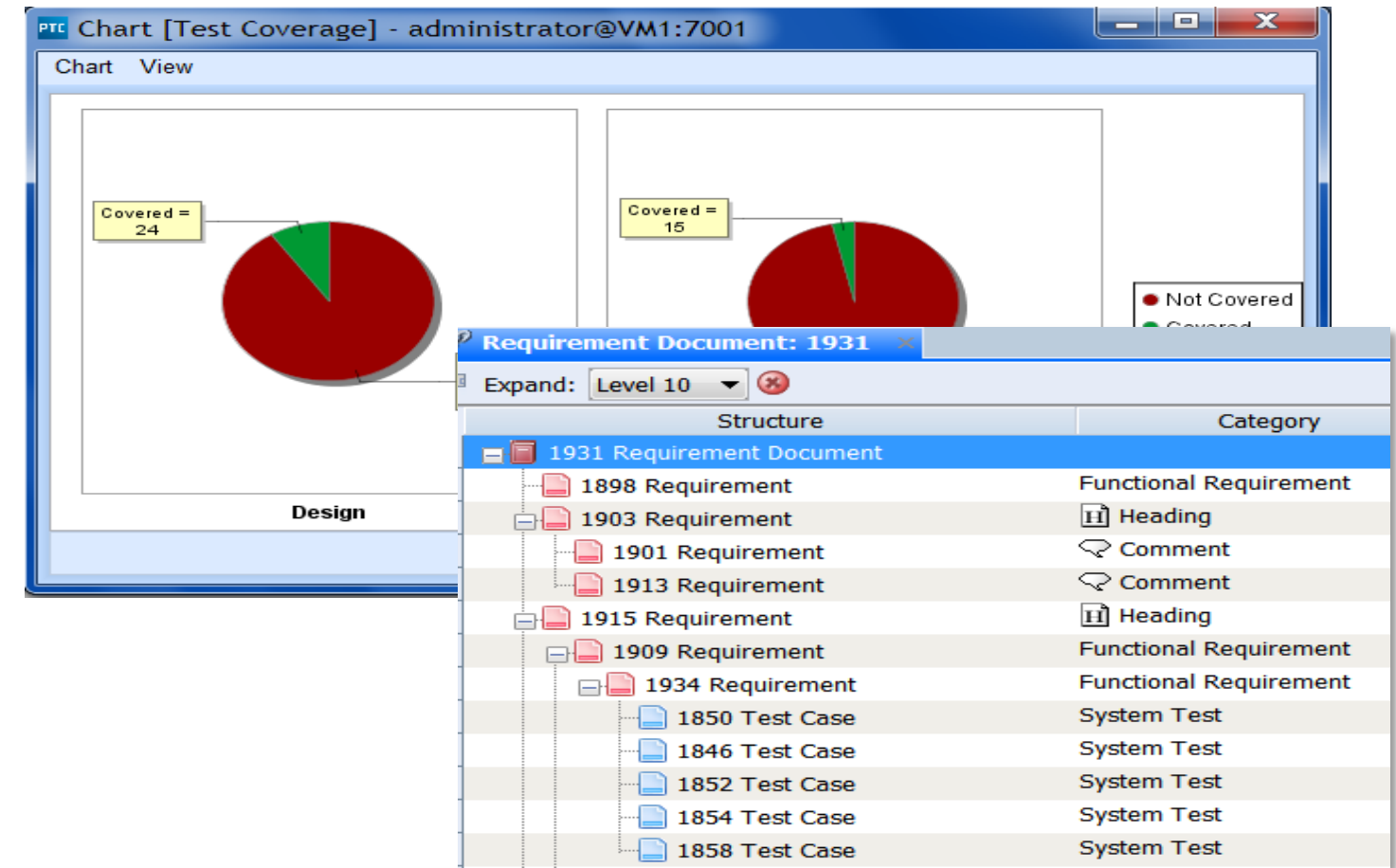
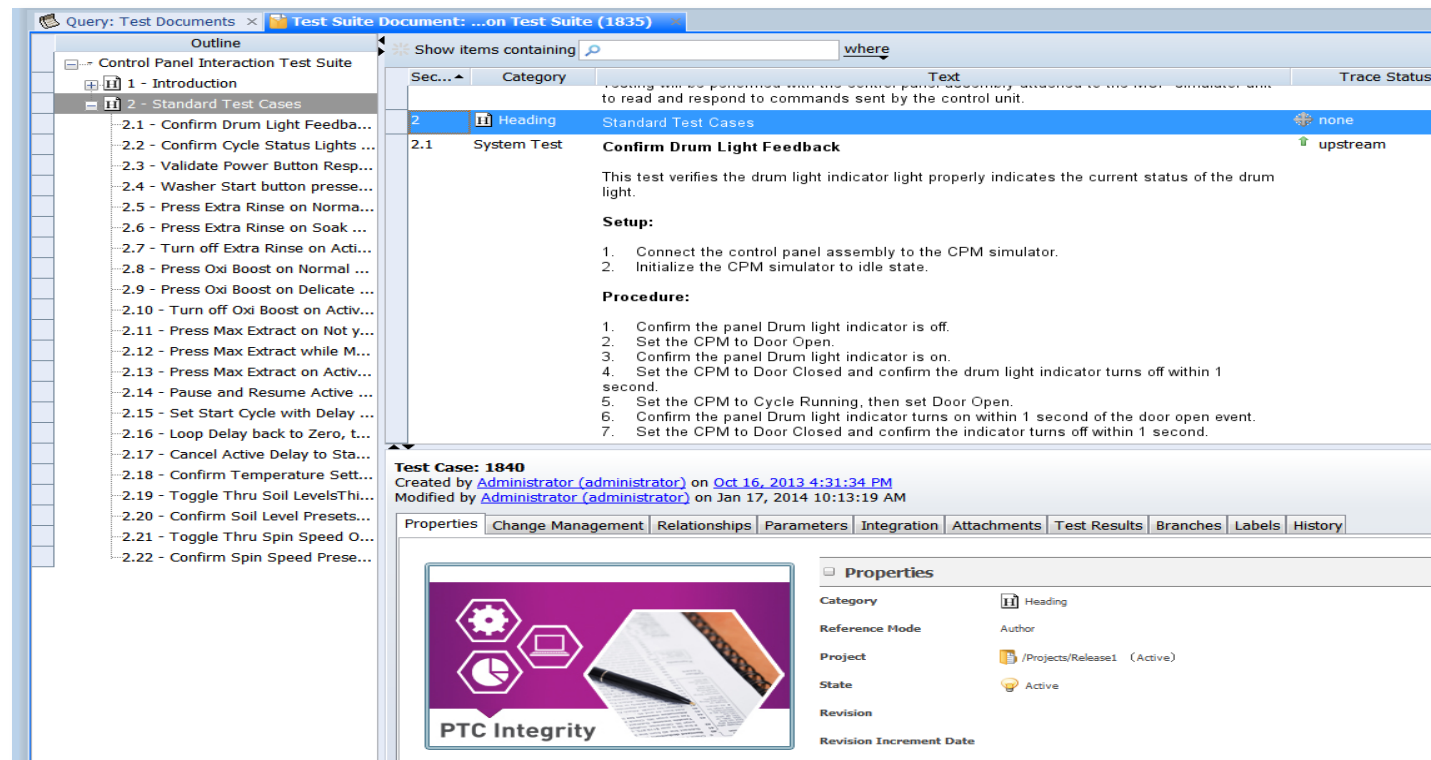
Key Test Management Capabilities

Test traceability and coverage	Establish traceability between test cases and requirements	Reduce unnecessary testing resulting from changed requirements
Test definition and authoring	Author and capture all test cases and plans in a single environment	Reduce the number of duplicate, incomplete, and conflicting test cases
Test execution and results analysis	Execute tests and capture results centrally for results analysis	Reduce time and effort to evaluate product readiness and quality
Test planning	Author test plans to reflect test strategy	Improve effectiveness of test resources by performing correct tests
Requirements & Test Change Management	Standardized change management process for all requirements and test artifacts	Eliminate uncontrolled requirements & test changes

Test cases allocated to requirements

Test coverage charts and reports identify completeness

Suspect flags automatically notify of requirements change



Hierarchies of test cases

Assigned test case parameters

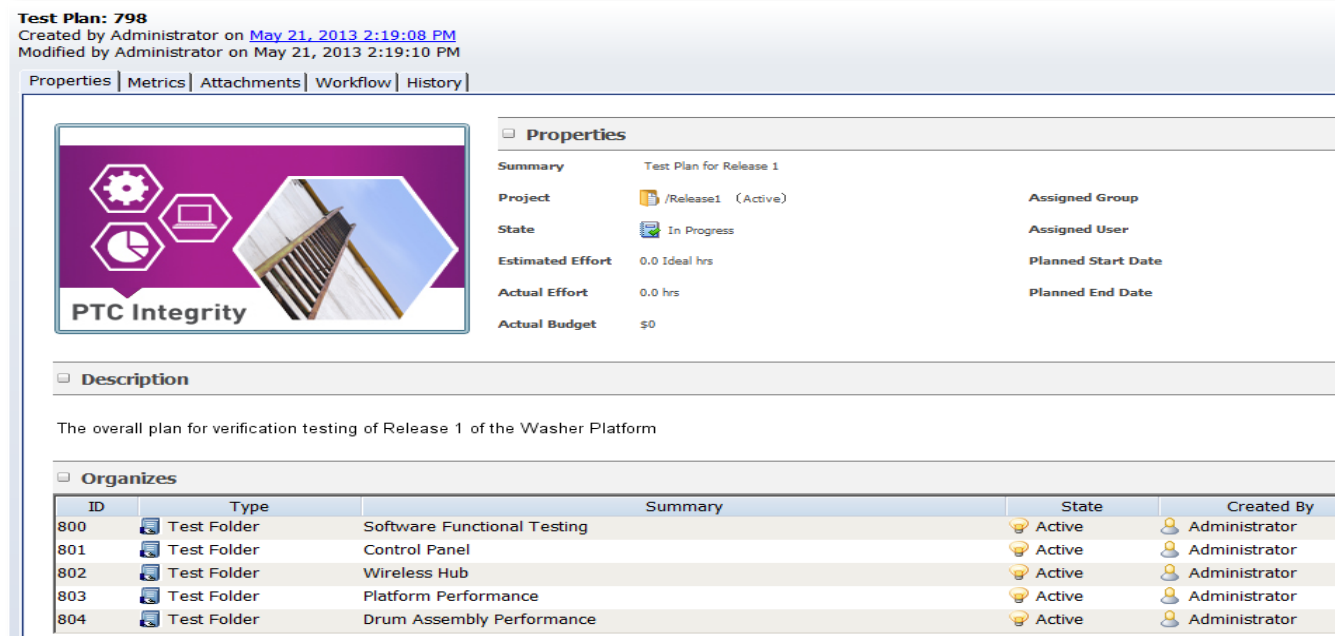
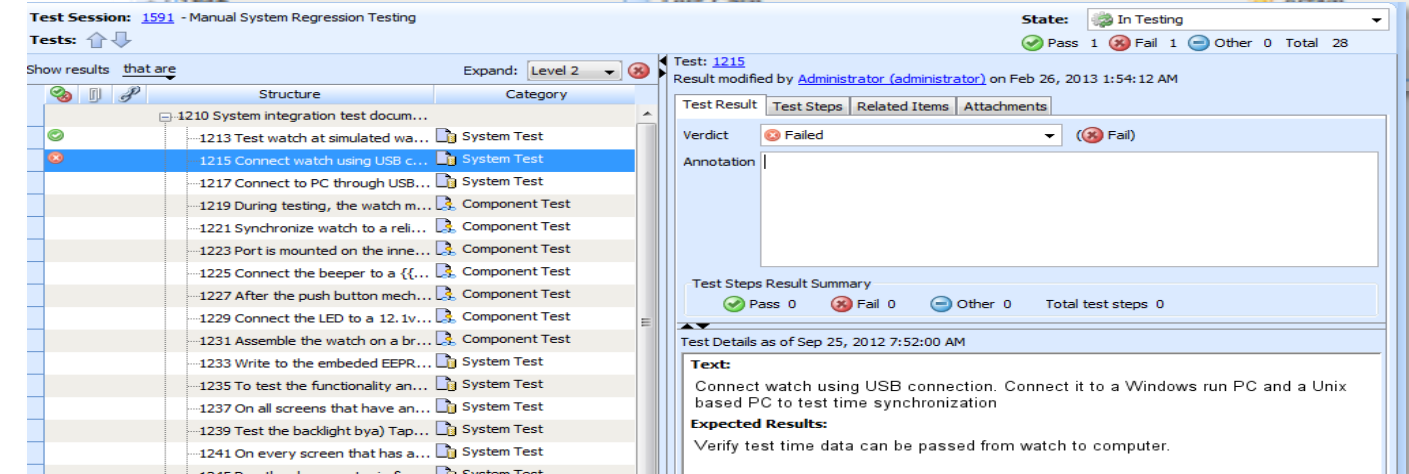
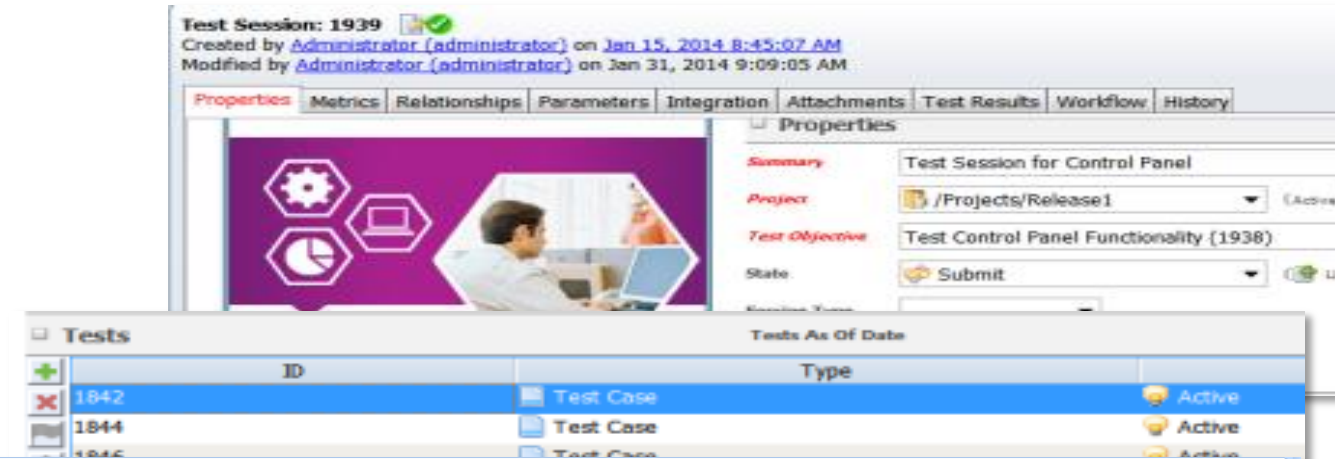
Test case priorities and categories

Imported test cases

Persistent record of manual and automated test results

Change management process tracks defects and resolutions

Test sessions capture multiple test cases



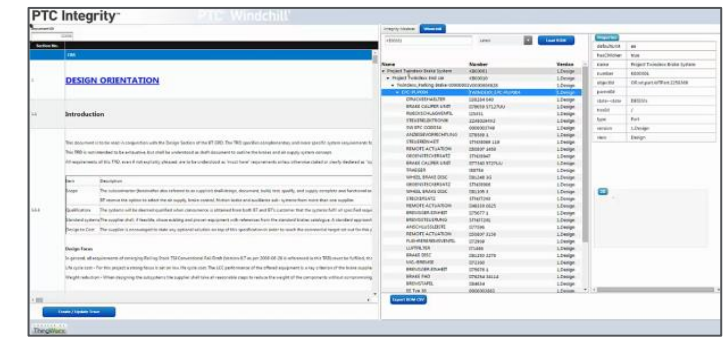
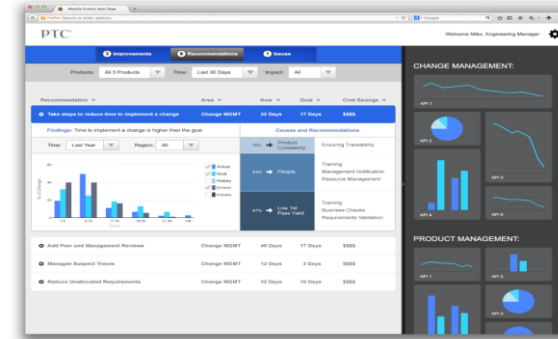
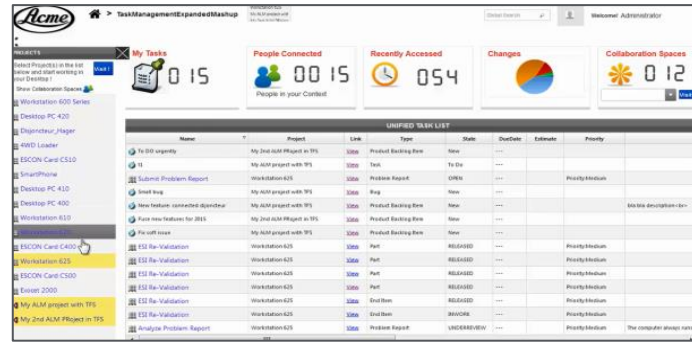
Test objectives guide and manage test activities

Test suites organize test cases

Test metrics measure progress against test objectives

- Importance of Systems Engineering
- PTC's Systems Engineering Solution
 - Process Governance
 - Requirements Engineering
 - System Design
 - Product Line Design
 - Test Management
- Systems Engineering Integration Roadmap





- **CONNECT** Engineering with the connected product in support of Closed Loop Lifecycle Management
- **ENABLE** composite PTC solutions further through ThingWorx based Task Based applications
- **EMBRACE** a marketplace for third party packaged and built applications

TASK BASED APPS

ALM – PLM Application Framework



Application Enablement Platform



**Jun
2015**

**Mar
2016**

**Sep
2016**

RELEASE

1. Synch Reqts and Models

- Trace Integrity Requirements to Integrity Modeler models

RELEASE

1. Link Requirements (1)

- Trace Integrity Requirements to Windchill Parts, from Windchill
- Flag change suspects in Windchill

RELEASE

1. Link Requirements (2)

- Trace Integrity Requirements to Windchill Parts, from Integrity
- Automate trace relationships for Save As and Branching
- User Authentication sharing

2. Link Change (1)

- Link Integrity and Windchill change process
- Visualize Integrity change status within Windchill

FUTURE

1. Functional Structure Mgt

- Create functions/system structure from Integrity Modeler or Windchill UI

2. Link Change (2)

- Initiate connected change process from wither Windchill or Integrity

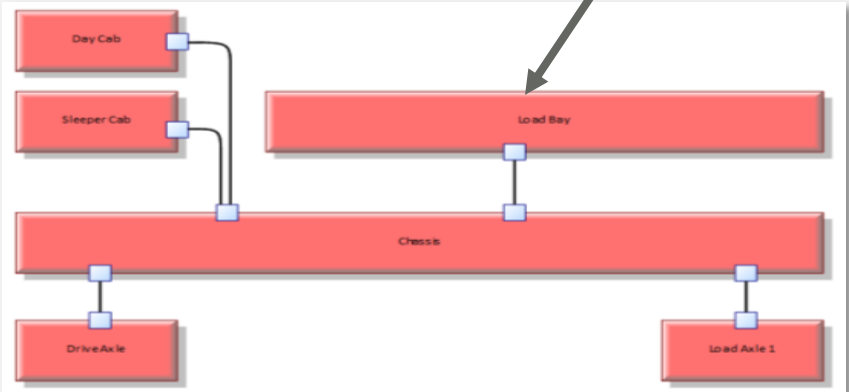
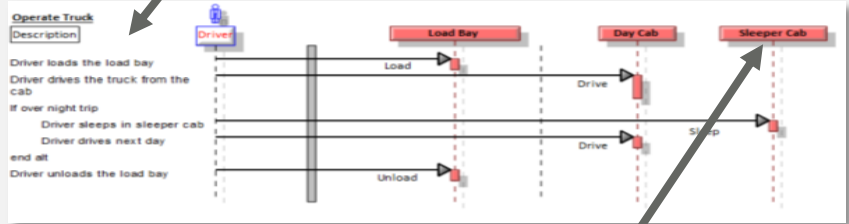
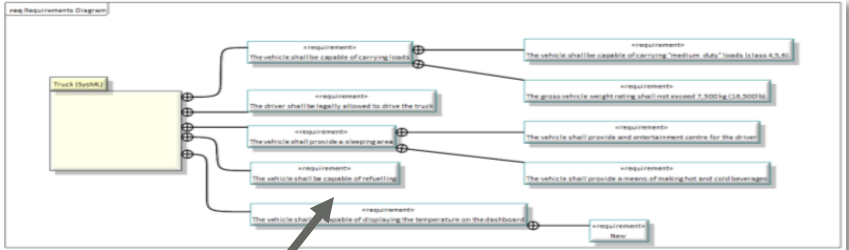
3. Relationship Navigation

- Visualize from within Integrity or Windchill

4. Link Configurations

- Support software release process to BOM
- Link baseline configurations

PTC Integrity™ Modeler™



PTC Integrity Lifecycle Manager Synchronizer

Requirements

Additional Model Elements

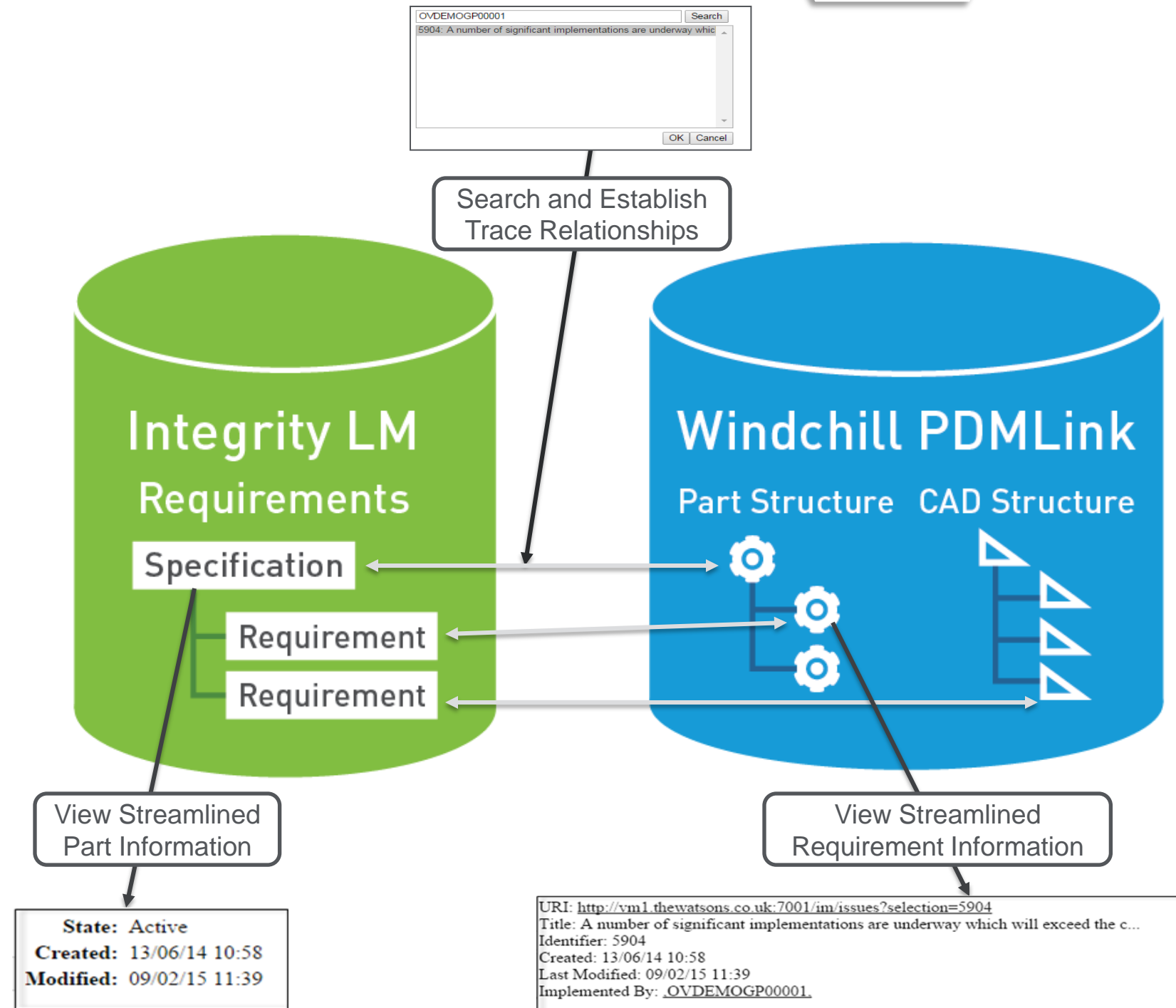
Model Trace Links

PTC Integrity Lifecycle Manager

Automatic Functions	Aggregate Functions	External Information Functions
abs (numeric-expression)	sum (numeric-expression)	averystats (date-range)
floor (numeric-expression)	avg (numeric-expression)	relCount (relationship-field)
isEmpty (expression)	max (numeric-expression)	aggregate (relationship-field, expression)
dateDiff (start-time, end-time)	count ()	sumTimeEntry (start-date, end-date)

Requirements Traceability to Product Information

- Trace Integrity Lifecycle Manager requirements to PTC Windchill PDMLink objects
 - Selectable type of traces
 - Initiated from Integrity LM or PTC Windchill PDMLink
- Visualize trace relationships from either Integrity LM or PTC Windchill PDMLink
 - View traced requirements in PTC Windchill PDMLink
 - View traced part in Integrity LM
- Streamlined view of linked information
 - Hover over link for preview information
- Capture suspect flag in PTC Windchill PDMLink when traced requirement changes
- Invoke application for more detailed access
- Report on traceability information



Actions Part - 000002116, DRUM ASM - COMPLETE, A.6 (Design) Released

Details Structure Related Objects Changes History Where Used Traceability Relationship Explorer

Editing: Insert Existing, Remove, Insert New, Edit

Check Out/In: Check Out, Revise, Check In, My Checkouts

Clipboard: Paste, Copy

Viewing: Show/Hide, Views, Display

New/Add To: New, Add to

Filter: Edit Filter, Current Filter, Saved Filters

Tools: Compare, Open in

Reports: Reports, Export

Find in Structure

Identity

- 000002116, DRUM ASM - COMPLETE, A.6 (Design)
 - 000002117, LIFTER - DRUM, A.2 (Design)
 - 000002627, DRUM ASM - BALL BALANCERS, A.5 (Design)
 - 000002201, BALL BALANCER ASM, A.5 (Design)
 - 000002289, 10-14 X .750 TRW SC TLR SS, A.1 (Design)
 - 000002439, DRUM ASSEMBLY - CROSSPIECE, A.2 (Design)
 - 000001910, CROSSPIECE ASM - COMPL, A.5 (Design)
 - 000001998, CROSSPIECE 5.0 - MACHINED, A.2 (Design)
 - 000002402, CROSSPIECE ASM, A.2 (Design)
 - 000001946, SHAFT - CROSSPIECE, A.1 (Design)
 - 000002454, CROSSPIECE, A.1 (Design)
 - 000002227, SLEEVE - SHAFT, A.1 (Design)
 - 000002596, SEAL - CROSSPIECE, A.1 (Design)
 - 000002468, DRUM ASM, A.2 (Design)
 - 000001955, COVER - FRONT, A.1 (Design)
 - 000002164, COVER - REAR, A.1 (Design)
 - 000002541, WRAPPER - DRUM, A.1 (Design)
 - 000002559, M6-1 X 25 OV 6L PP LC SS, A.1 (Design)

(18 objects)

Attributes Uses Occurrences Supersedes

Part Attributes

Number: 000002116

Name: DRUM ASM - COMPLETE

Version: A.6 (Design)

State: In Work - Released - Canceled

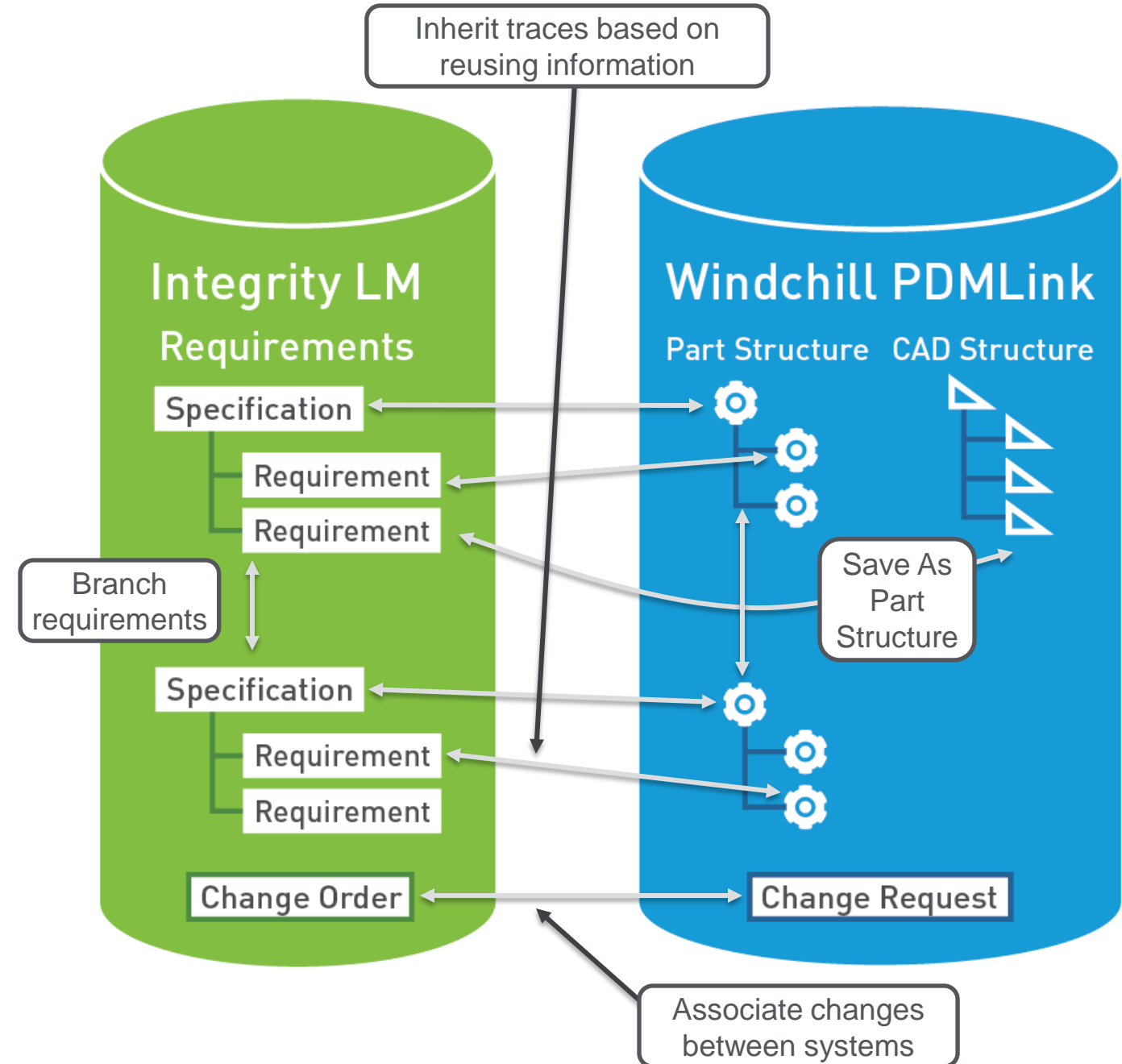
Status: Checked in

Modified By: Site, Administrator

Last Modified: 2015-05-28 10:43 EDT

Improve Requirements Traceability and Coordinated Change Management

- Support traces to Integrity LM versioned and branched requirements
 - Windchill PDMLink Save-as action
 - Branched requirements in Integrity LM
- Automate establishing trace relationships
 - Windchill PDMLink Save-as action
 - Branched requirements in Integrity LM
- Associate change processed and objects between Integrity LM and PTC Windchill PDMLink
 - Execution of the change exists in both systems
- Visualize information about Integrity LM changes within a PTC Windchill PDMLink change process
 - Streamlined view of linked information
 - Invoke application for more detailed access
- Improve user authentication passing



Advanced registration is strongly suggested, but not required.*

Tuesday, June 9 Training Seminars

10:00 a.m. – 12:00 p.m.	SysML Modeling with PTC Integrity Modeler <i>Matthew C. Hause, PTC</i>
1:00 p.m. – 3:00 p.m.	Requirements Engineering with PTC Integrity Lifecycle Manager <i>Sheilah Fournier, PTC</i>
4:00 p.m. – 6:00 p.m.	Create and Automate Any Process with PTC Integrity Process Director <i>Jonathan Archer, PTC</i>

Wednesday, June 10 Training Seminars

7:45 a.m. – 9:45 a.m.	SysML Modeling with PTC Integrity Modeler Matthew C. Hause, PTC
10:00 a.m. – 12:00 p.m.	Requirements Engineering with PTC Integrity Lifecycle Manager Sheilah Fournier, PTC

*Each session has room for 24 attendees to sit and share 12 computers.
Anyone above the 24 cap will be an observer.

- Your feedback is valuable
- Don't miss out on the chance to provide your feedback
- Gain a chance to win an instant prize!
- Complete your session evaluation now

PTC[®] Live Global