Integrated Requirements Engineering and Development of Specifications

Vector Technology Days 2019
Agenda

- PREEvision at a Glance
  - Integrated Requirements Engineering: Integration Levels
  - Out of The Box Functionalities and Processes
  - Coordination Process with Internal Experts or External Partners
Modellbasierte E/E-Entwicklung mit PREEvision

Vom Architekturentwurf bis zur Serienreife
Challenges in E/E Development: Complexity/Capability Gap

- Number, complexity and interaction of functions
- Increasing **authority** over control of the vehicle
- Increasing **requirements** on certification

- Number and rate of change of functions
- Inappropriate processes and tools

- **High Cost**
- **Poor Quality**


- **Complexity**
- **Capability**
Challenges in E/E Development: Complexity/Capability Gap

- Product Line engineering to maximize reuse
- Consideration of the relationships between various views on the system architecture
  - Functional, Hardware, Software, Network, Topology, ...
- Early validation of system concepts based on automated evaluation
- Exchange of concepts with suppliers in early design phases

- Quality
- Cost
- Complexity
- Capability
Best in Class Approach vs. Integrated Tool Support

**Best-in-Class-Approach means:**

- Many interfaces for data exchange
- Customer is system integrator
- Different GUIs
- Complex (multi-site) operation
- No team collaboration support
- Isolated process management data and engineering data
- No (easy) traceability between engineering data
Integrated Tool Support means:

- Replacement of many authoring tools
- Model based (all in one) instead of in worst case a document based working with many files
- Common repository, problems like tool borders or broken links simply do not exist. Organization has full overview over the E/E artifacts and their correlations
- Comprehensive data model by well defined language (metamodel)
  - Consistent data (correct by construction)
  - Bidirectional links
  - No redundancy
- Common GUI and Usability
- Seamless traceability
- Reduce costs & faster time to market
Supported E/E Engineering Use Cases

- Architecture Design
- Function Driven Design System Design
- Requirements Engineering and Requirements Management
- AUTOSAR System, Service and Software Design
- Communication Design (AUTOSAR and Legacy Formats)
- Hardware Component Development
- Wiring Harness Design
- Test Engineering and Test Management
- Change and Release Management
- Design of Safety Relevant Systems
- E/E Backbone, Collaboration Platform, File Management
- Product Line Engineering and Variants
- Wiring Harness Design
- Change and Release Management
PREEvision at a Glance

PREEvision Modeling Layers and Process & Team Support
Model Based Development in Large Groups

- Multi User – Single Source
  - Multiple users work collaboratively, supported by Lock&Commit, Life Cycles, Rights&Role Management and Ticket System

- Multiple projects are handled in one system, supported by Productline Management, Variant Management

- Import and export with industry standard exchange formats (AUTOSAR, DBC, LDF, FIBEX, RIF, ReqIF, KBL, CSV)
Model Based Development – Single Point of Truth

- Domain specific language and data model
- Single source model across all Automotive E/E development Use Cases:
  - The model is the Single Point of Truth
  - Mappings ensure full traceability
  - The model can be analyzed by metrics
- All data objects have a semantic meaning
  - Base for various model checks e.g. for Correctness, Completeness, Consistency
- Automated algorithms for synthetization, scheduling, signal routing, etc.
Agenda

PREEvision at a Glance

- **Integrated Requirements Engineering: Integration Levels**
  - Out of The Box Functionalities and Processes
  - Coordination Process with Internal Experts or External Partners
Evolution of Requirement Engineering Tools

Integrated Requirements Engineering: Integration Levels

File based
- Purely textual
- No Traceability

Standalone Tool
- Structured Requirements
- Internal Linking & Tracing
- Linking to other tools or databases not reliable and not possible on atomic basis

Tool Environment
- Linking & Tracing to e.g. Test tools
- Integration via OLSC possible but relations across database boundaries cannot be versioned and archived.

Fully Integrated
- Common database allows links on atomic basis
- No tool borders
- Versionable and reliable
- Consistent traceability
Integrated Requirements Engineering: Integration Levels

Level 1: Purely Textual

Purely textual in PREEvision?

- Model based
Level 1: Purely Textual

Purely textual in PREEvision?

- Model based
  - Formulized language for describing data (Metamodel), information is clear, structured and consistent
  - Different Requirement Types (Definition, Information, ...)

15
### Level 1: Purely Textual

<table>
<thead>
<tr>
<th>LEVEL</th>
<th>ID</th>
<th>Hierarch.</th>
<th>Text</th>
<th>Type</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td></td>
<td></td>
<td>1 TechnicalRequirements</td>
<td>Requirement Package</td>
<td>F</td>
</tr>
<tr>
<td>Δ 2-2</td>
<td>REQ_72223</td>
<td></td>
<td>1.1 Weight</td>
<td>Heading</td>
<td>F</td>
</tr>
<tr>
<td>3-3</td>
<td>REQ_635</td>
<td></td>
<td>Center of mass should be at center of the vehicle (top view) and 40 cm from bottom.</td>
<td>Requirement (shall)</td>
<td>F</td>
</tr>
<tr>
<td>4-3</td>
<td>REQ_10088</td>
<td></td>
<td>The maximum weight of the vehicle is not to exceed 1.5 tons.</td>
<td>Definition (must)</td>
<td>F</td>
</tr>
<tr>
<td>Δ 5-2</td>
<td></td>
<td></td>
<td>1.2 Miscellaneous</td>
<td>Heading</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 LegalRequirements</td>
<td>Requirement Package</td>
<td>F</td>
</tr>
<tr>
<td>Δ 7-3</td>
<td>REQ_11261</td>
<td></td>
<td>Regard ISO</td>
<td>Information (can)</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>The vehicle has to be watertight against liquids (diving not included)</td>
<td>Requirement (shall)</td>
<td>F</td>
</tr>
<tr>
<td>8-3</td>
<td>REQ_9465</td>
<td></td>
<td></td>
<td>Safety Goal</td>
<td>F</td>
</tr>
<tr>
<td>9-1</td>
<td></td>
<td></td>
<td>2.1 Emissions</td>
<td>Heading</td>
<td>F</td>
</tr>
<tr>
<td>10-2</td>
<td>REQ_72224</td>
<td></td>
<td></td>
<td>Requirement Package</td>
<td>F</td>
</tr>
</tbody>
</table>
Level 1: Purely Textual

Purely textual in PREEvision?

- Model based
  - Formulized language for describing data (Metamodel), information is clear, structured and consistent
  - Different Requirement Types (Definition, Information, ...)
- Consistency and Online Checks
- Lifecycles
- Version Management
- Reuses
- Roles and Rights Management
- LibO Formatting & Reporting
- Use Case Diagrams
Level 2: Usage of Requirements Links

- Link requirements among each other
  - Different Link Types (trace, refine, ...)
  - Better Traceability
  - Better Understanding

Natural border of standalone tools
Level 3: Usage of Mappings

Linking the Model

- Type safe
- Basis for analysis, e.g. coverage analysis
- Consistent traceability through the model
- Simple navigation
- Simplification of understanding, e.g. by usage in diagrams
- Easy creation via drag&drop in Model View, diagrams or in Table Editors
Integrated Requirements Engineering: Integration Levels

Level 3: Usage of Mappings
Level 4: Usage of Variants

Make it context dependent

- Determine variants, e.g. relevant series for a requirement directly in the Requirement Text Editor.
Level 4: Usage of Variants
Level 4: Usage of Variants

Make it context dependent

- Determine variants, e.g. relevant series for a requirement directly in the Requirement Text Editor
- No need to create an independent requirement for each variant as information is separated
- Activation of a variant highlights requirements associated to the variant in the specification editor
- Generation of variant sensitive reports and specifications
Reference the model

- Referencing of any **attribute** of any artifact in the model via drag&drop into a rich text.
- Referencing of **diagrams, charts** and **tables**
- Placeholders are kept always up to date
- Relevant attributes can be defined for a perspective
- Copy and paste support with automatic merge of already existing placeholders
- Change marker indication if placeholder content has been changed
Level 5: Usage of Placeholders

- Referencing any attribute of any artifact in the model via drag&drop into a rich text.
  - DnD support from Model View, Tables and Diagrams
- Changed placeholder content can be identified via change marker indication in front of the referenced attribute:

  ![Change marker example](image)

  Indication of change marker is propagated additionally in
  - Containing attribute (e.g. description)
  - Table Editors
  - ModelView
  - Property Editor
Integrated Requirements Engineering: Integration Levels

Level 6: PREEvision

The model is the requirement specification

- Get along without requirement artifacts at all – use instead the model as single point of truth!
- Via the ReqIF Export any artifact with any attribute from any layer can be used for the specification document.

Goal might not be completely reachable due to standards, laws, company internal or non functional requirements used in the specification
Higher Integration Level leads to

- Deeper traceability
- Better data consistency
- Easier managing of complexity
Integrated Requirements Engineering: Integration Levels

Summary

Data Management, Collaboration, Platform Management, Safety, Change Management

Requirements Engineering and Management

- Functional Architecture Design
- SOA Design
- Software Design
- Physical Architecture Design
- Physical System Design
- COM Design
- WH Architecture Design
- WH Series Design

REQ-X Mapping

Test Engineering and Management

Subject under Test
Specified Artifact under Test
Agenda

- PREEvision at a Glance
- Integrated Requirements Engineering: Integration Levels
- Out of The Box Functionalities and Processes
  - Coordination Process with Internal Experts or External Partners
Out of The Box Functionalities and Processes

Processes

Data Management, Collaboration, Platform Management, Safety, Change Management

Requirements Engineering and Management

- Template Definition
- System Specification
- Component Specification
- Agreement with Supplier

Test Engineering and Management

- Specification of Tests
- Design & Implementation of Tests
- Planning of Tests
- Execution & Documentation of Tests
- Evaluation & Analysis of Test Results
Test Engineering: Process, Roles and Import & Export Interfaces

- Specification of Tests
- Design & Implementation of Tests
- Planning of Tests
- Execution & Documentation of Tests
- Evaluation & Analysis of Test Results

**Requirements**
- Requirements Engineer
- Test Engineer
- Test Manager

**Test Specification**
- Specification Heading
- Item Item Step
- Library
- Manual test sequence
- Test script

**Test Implementation**
- Test Engineer
- Test Design & Implementation Tool
  - Vector vTESTstudio
  - 3rd party
- Synchronize Test Scripts
- Parse for Verdicts and Requirements / Test Specification Associations

**Test Planning & Tasks**
- Test Manager
- Requirements Source
  - RIF, ReqIF
  - XLS
- TestItems Source
  - ReqIF
- Traceability Information
- Synchronize Test Scripts

**Test Execution**
- Test Engineer
- Test Execution Tool
  - Vector CANoe
  - 3rd party
- Verdict
- Report

**Test Analysis**
- Test Manager
- Evaluation & Analysis of Test Results
Out of The Box Functionalities and Processes

Requirements Management: Involved Roles

Legend:
- **Responsible Role**
- **Supporting Role**
Out of The Box Functionalities and Processes

Integrated Requirements Process
Out of The Box Functionalities and Processes

Version management

**CHECK OUT**
- User 1
- User 2
- User 3

- Check out (or create a new) artifact creates a new revision starting at zero.
- A new revision can be edited by all users (only at one time - lock)

**LOCK**
- User 1
- User 2
- User 3

- While one user has locked an artifact no other user can edit the artifact at the same time.

**COMMIT**
- User 1

- Commit to store the changes in the system (database) and make it available for other users

**CHECK IN**
- User 1

- Check in “freezes” a specific state of the artifact which is not changeable anymore in this revision.
- Check ins could be performed by any authorized person.

**BRANCHES**
- User

- Branches allow to check out older revisions
- Branches are visible in:
  - Model View
  - Version History
  - Property Editor
Integrated Requirements Process

Out of The Box Functionalities and Processes

Template

- V1
  - Structure
  - Attributes

Instantiation

System specification

- V1
- V2

Component specification A

- V1
- V2

Component specification B

Reuse & Copy

System provides content
Reuse & Checkin

Merge & Update
keep links

Changes

Linking between Component and System
Out of The Box Functionalities and Processes

Merge

Merge Options:
- Identity of resulting artifact
  - Source
  - Target
- Priority in case of conflicts:
  - Source
  - Target
  - Replace
- Children are merged with the same
  - Identity
  - Name

> All external relations are merged and connected

Source context

Door New (ECU)
UUID = 123
Code = A
Ignition = not set

Material PP

Req. A

Target context

Door Left (ECU)
UUID = xyz
Code = B
Ignition = HW

Material PP

Req. B

Merge 2 artifacts of the same class

Merge 2 artifacts of the same class
The Merge Block introduced with PREEvision 8.5 can be used for an automatic merge of 2 artifacts of the same Meta Class. Merge options are available in Merge Block:

- **Identity** allows to define if either the source or the target artifact shall keep their identity
- **Priority** defines if either the data of the source or of the target artifact shall have priority in case of conflicts. With “Replace” all data of the source is taken over and the data of the target is discarded.
Integrated Requirements Process

Out of The Box Functionalities and Processes

- Template
  - V1
    - Structure
    - Attributes

- System specification
  - V1
  - V2

- Component specification A
  - V1
  - V2

- Component specification B
  - V1

Processes:
- Instantiation
- Reuse & Copy
- Merge & Update (keep links, set Lifecycle State)
- Change details

Changes:  
- Green
- Blue
Recognition of an changed artifact either by
- revision number of an updated requirement or a
- lifecycle state or by
- change marker symbol $\triangle$
  (Model View, Table Editor or Property Editor)

Change details for an artifact can be accessed on level of attribute changes in the Change History View
Change Logs

Support for recording, analysis and display of relevant changes

- Changes are documented via a ticket
- Relevant changes can be defined by profiles (Meta Classes and Meta Attributes)
- On commit of an relevant artifact a ticket can be selected, the changed artifacts are associated, change log is created
- Change History allows to view the recorded changes regarding
  - change interval
  - relevant attributes
- Change logs can be accessed and can be displayed in Table Editors for showing relevant changes (from viewpoint of change order and viewpoint of changed artifact)

Deliver relevant change details to colleagues or suppliers
Out of The Box Functionalities and Processes

Lifetime of specifications

- Specification versions
  - V1
  - V2
  - V3
  - V4

- Released Specification

- Test Engineer
  - In test

- Component Responsible
  - In work

- Suppliers
Agenda

- PREEvision at a Glance
- Integrated Requirements Engineering: Integration Levels
- Out of The Box Functionalities and Processes
  - Coordination Process with Internal Experts or External Partners
Coordination Process with Internal Experts or External Partners

Review & Vote

System Responsible
Component Responsible

Internal Review

Component A

System provides content

Component Responsible
Supplier

Agreements by voting between OEM and Supplier(s)

Component A

System specification
V1

Component specification A
V1

Changes

Component A

V2

V2

Component A

V2

V2

V2

V2

V2

V2

V2

V2
Review and Vote

**Review**
- Participants
- Artifacts to review
- Evaluation by comments
- Comment and Chat View

**Vote**
- Participants
- Artifacts to review
- Voting type
  - Majority vote
  - Consensuses vote
  - Mandatory or optional vote
- Voting result analysis

Combination of Review & Vote is also possible for an internal review.
Review specifications

- Review instantiation via wizard
- Personal Task Tables for Review participants

Component Responsible

Task Overview Table
- Agreement Task: Component 4
- Agreement Task: Component 9

System Responsible

Task Overview Table
- Agreement Task: System 1
- Agreement Task: Component 4
- Agreement Task: Component 9

Coordination Process with Internal Experts or External Partners
Agreements between OEM and Suppliers

Coordination Process with Internal Experts or External Partners

The ReqIF Agreement Export puts all the relevant information in one ReqIF file.

- E.g. the Requirements Specification, where you have initiated the „Specification Review“.
- The information about the provided status.
- The agreement information for the review.
Coordination Process with Internal Experts or External Partners

Agreements between OEM and Suppliers

- **Initiate Review**
  - OEM

- **Export the Specification prepared for review**
  - OEM

- **Review**
  - Spec
  - Supplier

- **Import ReqIF file with review content**
  - OEM

**Agreement Export via ReqIF**

- **Supplier add his comments and set his status e.g. in 3rd party tool**

**Agreement Import via ReqIF**

- **Only Supplier Status and Comments are imported**

**OEM can perform an assessment in PREEvision of the feedback given by the supplier**
Exchange of Agreements between OEM and Supplier via ReqIF

Coordination Process with Internal Experts or External Partners

ReqIF Import
- Import Automation Support

ReqIF Export
- Export Automation Support

PREEvision

Customer Features
- Feature
- Requirement

Requirements
- REQ
- REG
- Safety REQ

Test Data
- Test Item
- Test Case

Variant Information
- Safety Requirement
- Deriv.: SLV
- Convertible

Requirements Subtypes
- 11 Requirement (Is Headline)
- Requirement (shall)
- Information (can)

Editor
- Description
- Picture
- Table

Agreements between OEM and Suppliers (Review and Vote)

Agreement A
- Agreement Task: Supplier 1
- Agreement Task: OEM

Agreement B
- Agreement Task: Supplier 2
- Agreement Task: OEM

Requirement Interchange Format

Only export supported
Coordination Process with Internal Experts or External Partners

Small Example

Initiate Review → Spec → Export the Specification prepared for review → OEM → Review → Spec → Supplier → Import ReqIF file with review content
For more information about Vector and our products please visit

www.vector.com

Author:
Robert Rotter
Vector Germany