PREEvision
Model-Based Electric/Electronic Development
From Architecture Design to Series Production
For modeling the entirety of electric/electronic systems, from requirements to software and hardware architectures to the wiring harness, PREEvision provides a comprehensive data model with dedicated abstraction layers.

Requirements
System designers develop customer features and corresponding requirements. Via links and mappings, the implementation of customer features and requirements can be traced at any time.

Logical Function Architecture
In the logical function architecture, the requirements are implemented by logical functions that are connected via ports and interfaces. The resulting function network is the basis for the technical development of hardware and software.

Software/Service Architecture
In the system and software architecture, software components and their interfaces are modeled. A service-oriented modeling is possible. The software architecture with libraries supports the AUTOSAR method. Additionally, all implementation artifacts can be managed.

Communication
On the communication layer, one defines how software components exchange data across hardware borders. PREEvision supports all relevant network technologies including CAN, CAN FD, LIN, FlexRay and Ethernet.

Hardware Architecture
On the hardware layer, ECUs, sensors and actuators, their networking via bus systems as well as the power supply and hardwired connections are modeled.

Electric Circuit Diagram (Electronics Logic)
In the electric circuit diagram, the electric characteristics of the components and their interconnections are defined. Also, the internal electrical design of components with fuses and resistors can be modeled.

Wiring Harness
On the wiring harness layer, one defines the physical details of the wiring harness including pins, connectors, cables, inline connectors and splices.

Geometry
In the vehicle geometry, installation spaces and locations are defined or imported via 3D KBL data. Then routing paths via topology segments including inline connectors are modeled.
**Intuitive User Interface**
The state-of-the-art user interface supports graphical diagrams, table editors, and forms for modeling the various different layers.

**Integrated Tool**
Mappings and links relate artifacts and the various different abstraction layers with one another, thus making changes and developments in the model traceable.

**Reuse**
The reuse of development artifacts ensures efficient development, including across product lines. In addition to individual artifacts, entire system sections can be reused.

**Industry-Driven Data Model**
The consistent domain language as a technical semantic means of expression for model description was developed and tested together with leading automobile manufacturers.

**Transparency in Development**
All development artifacts are available in their current versions to those involved. Coordination processes, such as release and finalization of versions, are performed with tool support and with the assistance of integrated version, life cycle, and change management, which also enables the reproduction of historical working versions.

**Openness**
Standard interfaces such as AUTOSAR, KBL, and ReqIF and scriptable interfaces enable available data to be quickly merged to a common model.

**Planning and Traceability in Development**
Product and release management, as well as change management, simplify planning of development steps. Change markers, release processes, and the finalization of working versions make all progress transparent and traceable.

**Risk Reduction in Development**
Automatic validation and consistency checks ensure quality of designs in early development phases and detect undesirable developments and rule violations.

**Multi-User Platform**
“Lock & commit” mechanisms in PREEvision enable conflict-free parallel working across locations.

PREEvision is the premier tool for model-based development of distributed, embedded systems in the automotive industry and related fields. This engineering environment supports the entire technical development process in a single integrated application.

PREEvision offers comprehensive functions for both classic and service-oriented architecture development, requirements management, communication design, safety-related system design, AUTOSAR system and software design as well as wiring harness development. PREEvision supports the tried-and-tested system engineering principles of abstraction, decomposition, and reuse throughout.

The integrated and model-based approach enables:
- Early evaluation of E/E architectures
- Consistent requirements and test management
- Function-driven development
- Service-oriented architectures (SOA)
- Software and communication design according to AUTOSAR
- Model-based wiring harness development
- Efficiency through concepts such as reuse and product line/variant management
- Consistent design in a single tool
- Parallel work on a common database from multiple locations (engineering backbone)

PREEvision helps to complete complex tasks, such as the development of driver assistance systems, hybrid drives and end-to-end architectures including backend systems, to remain straightforward and controllable.

### PREEvision Use Cases

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design, evaluation and optimization of electric/electronic architectures</td>
<td>4</td>
</tr>
<tr>
<td>Product line approach</td>
<td>6</td>
</tr>
<tr>
<td>Integrated requirements management</td>
<td>8</td>
</tr>
<tr>
<td>AUTOSAR system and software design</td>
<td>10</td>
</tr>
<tr>
<td>AUTOSAR communication design</td>
<td>12</td>
</tr>
<tr>
<td>AUTOSAR Ethernet and SOA</td>
<td>14</td>
</tr>
<tr>
<td>Wiring harness development</td>
<td>16</td>
</tr>
</tbody>
</table>

### PREEvision Process and Team Support

<table>
<thead>
<tr>
<th>Support</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development of safety-related systems according to ISO 26262</td>
<td>18</td>
</tr>
<tr>
<td>Test engineering and test management</td>
<td>20</td>
</tr>
<tr>
<td>Collaboration, multi-user operation and parallel work from different locations</td>
<td>22</td>
</tr>
<tr>
<td>Project, change and release management</td>
<td>24</td>
</tr>
</tbody>
</table>
PREEvision offers comprehensive functions for the model-based design of electric/electronic architectures and the fast design and evaluation of architecture alternatives.

**PREEvision Advantages**
- Architecture designs verified early on
- Reduced expenses and short development duration
- High initial quality for series development
- Consistent and integrated modeling, from systems design to wiring harness design
- Multi-dimensional comparison of architectures according to customizable criteria, such as cost, weight, installation space, bus-load, etc.

**Factors for Success at a Glance**
To develop a cost-optimized and robust E/E system, PREEvision enables the easy evaluation of architecture alternatives, even during early development phases. The success factors of an architecture may be determined by architects themselves. In addition to important optimizations for the overall vehicle, such as weight, costs, installation space, cable length and power consumption, PREEvision also takes E/E system-relevant targets into account, such as the bus-load and timing or security requirements.

It is also possible to compare an architecture in different vehicle variants of a product line. Finally, the architect decides on which factors for success are taken into account.

**Multi-Dimensional Decision Network**
A multi-dimensional decision network can be laid out for evaluation of the architectural design, and thus an architecture can be evaluated in comparison to other alternatives. The data model holds information required for quantitatively evaluate and from this weight or bus-load, for example, may be calculated.

**Metric Framework**
The parameters for benchmarking of alternative architecture designs are calculated with metrics. PREEvision features a host of included metrics which can be enhanced by the customer. The implementation of the metrics is in the Java programming language. Metric blocks are a component of the graphical metric language. Metrics used within the metric framework are always run against the current model data. They can be calculated using the overall model or a previously selected variant.

The results of the performance comparison provide architects with a basis for making decisions on determining the architectural design. They provide important information for the refinement of the architecture during the development process. Thanks to the comprehensive data model, the finalized architecture model can be used as the basis for further development of the E/E system.
The product line approach of PREEvision ensures efficiency in E/E development with libraries and reuse of designs. Model series or individual vehicles can be derived from product lines using variant management.

**PREEvision Advantages**
- Comprehensive product line concept
- Centralized and product line-based libraries as basis of development
- Reuse of associated artifacts
- Abstraction concept for dependencies between development branches
- Flexible amalgamation of artifacts and model parts
- Variant management from product line to concrete vehicle
- Feature-Oriented Domain Analysis (FODA)

PREEvision helps to avoid redundant development work and facilitates the reuse of solutions that have already been developed. This saves development time initially, and also in later stages of projects. In case of changes a reused design has to be changed only once. All reuses in the model are updated accordingly.

PREEvision allows the reuse of individual artifacts including the corresponding hierarchy, for example, an ECU with all its components and connectors. Furthermore, artifacts located in various abstraction layers can be assembled in building blocks or sets and then be reused as a whole. A building block comprised of hardware and software components and the corresponding requirements, for example, for engine control, can be constructed and in turn be used as a template in various product lines. Whole product lines can be further developed as development branches and then combined later on.

**Libraries**
Libraries in which standard modules and components are managed serve as the basis for structured and efficient development without redundancy. In PREEvision, a dedicated product line can serve as a library. The description of all aspects of the E/E architecture using graphical or table-supported editors results in a 150% model containing all the installable E/E equipment features of a vehicle series. These include different and mutually-exclusive drive concepts such as diesel or gasoline-operated engines.

**Variant Management**
From a 150% model of a product line, it is possible to derive model series (120% model) from which concrete, fully configured vehicles can be derived. PREEvision supports two methods for variant management: Variants can be derived based on feature models (Feature-Oriented Domain Analysis, or FODA). Or they can be modeled with equipment templates and their alternatives.

**Design Tool**
With a design tool for the quick implementation of new ideas and solutions, the so-called Scratchpad, each user has a secure working environment outside the main model context at their disposal. Model parts can be further developed here and later integrated into the model.

**Merging of Data**
For further development of individual artifacts, entire components or product lines in development branches, as well as the integration of imported data, PREEvision provides a detailed comparison view. The controlled merging of data is also possible. This enables detailed determination of which data is merged and which is overwritten. Historical versions can also be compared and restored.

**If new product lines are created for teams with their own specific tasks through copying, the artifacts remain connected to one another thanks to the abstraction concept of PREEvision. Cross-product line development can be traced in this way.**
PREEvision supports requirements management in such a way that requirements do not have to exist separately, but can instead seamlessly interact with all development artifacts in the model.

With its integrated approach, PREEvision extends far beyond the functions of a classic tool for requirements management. Thanks to its consistent data model, development artifacts such as software and hardware components can be seen as requirements and used in system specifications. The artifacts may be linked to requirements with mappings or are embedded directly in requirements using placeholders.

PREEvision Advantages

- Editing of requirements in tables or via rich text editor with graphical support
- Early validation and consistency testing
- High initial quality of requirements
- Version and life cycle management
- Use case diagrams
- Linkability and traceability
- Change marking and history
- Requirements import (RIF, ReqIF, Excel)
- Easy report creation
- File attachments management
- Integration with test engineering and management

Classic Requirements Management

For classic requirements management, requirements and customer features with formatted text, graphics or tables can be developed and specified. Requirements are hierarchically structured and classified using a unique ID. In the requirement editor, all information (such as description, attribute values and mappings to other artifacts) can be conveniently entered and edited. Additional attributes can be freely defined.

Requirements Engineering and Management

PREEvision Use Case

In addition to simply being recorded, requirements may also be developed based on models and through the use of diagrams like the UML-based use case diagram. RIF, ReqIF and Excel imports enable the connection of additional tools.

Integrated Requirements Engineering

In PREEvision, requirements do not exist separately: They may be reused or associated with other requirements and development artifacts. Requirements can be mapped to one another or to other development artifacts. Via placeholders, attributes of development artifacts, for example, the price of an ECU, can be included in requirement descriptions.

The UML-based use case diagram represents the anticipated behavior of the system from the user’s viewpoint.

Development and Modeling of Requirements

PREEvision provides a large number of functions for developing and managing requirements after creation. The “lock & commit” concept prevents data collisions during multi-user operation, and a freely designable life cycle model creates transparency in the development process. Requirements are also subject to versioning. As a result, requirement development remains traceable over time.

Validating and Verifying Requirements

With the help of metrics and live checks, mechanisms and validators are available for automatically checking requirements and their attributes. PREEvision holds a large number of predefined tests that can be expanded with user-specific checks. Correct implementation of requirements is ensured through test engineering and test management.

Reports and Specifications

Reports such as system specifications are automatically generated based on templates. Placeholders in templates are replaced by current model data when the report is created. Placeholders can be entire requirement texts, diagrams, tables or individual values. The placeholder principle ensures that current values are always taken into consideration in the dynamic development process. Links to the PREEvision model from the created document are possible.

Software or hardware components are seen as requirements in the model and used in system specifications. Reports are automatically generated based on templates. During creation, the respective current model data replaces placeholders in the template. Placeholders can be entire requirement texts, diagrams, tables or even individual values.
With PREEvision, consistent development of software and hardware architectures using many AUTOSAR concepts is possible.

PREEvision supports both an abstract system description as a network of logical functions with their ports and connections as well as an AUTOSAR compliant modeling of software components, ports and interfaces, i.e. the description of the AUTOSAR virtual function bus. In both approaches, the logical functions or the software components with their interfaces and ports can be modeled in table editors or graphically in various diagrams. Already specified hardware and software components can be reused across product lines or integrated into the system via import.

**PREEvision Advantages**

- Graphical modeling of software architectures
- Support of the AUTOSAR software component templates
- Import and Export of AUTOSAR 4
- Automatic synchronization of software types, prototypes and instances
- Library concept for management of software components, interfaces and data types
- Consistency checks for verifying AUTOSAR compliant modeling
- Functions for combination, relocation and division of software components and automatic connection of ports

PREEvision supports the type-prototype-instance concept of AUTOSAR. Whether types are first created in the library and prototypes derived from that or instances are used directly for modeling is not important in PREEvision. Extensive automatic synchronization ensures a consistent model at all times. Automatic consistency checks help here as well.

**AUTOSAR Software Component Template**

The data model of PREEvision supports all important aspects of the AUTOSAR software component template. Software components can be hierarchically structured, and modeling the internal behavior of software components, triggers, parameters and administrative data is supported. Special requirements can be fulfilled with self-defined attributes. AUTOSAR Special Data Groups (SDGs) can be implemented with them as well.

**Software-Hardware Mapping**

Software components can then be allocated to hardware components. This defines which part of the data exchange between software components is happening across ECU borders via a physical bus system. The flexible distribution of software components across the hardware network offers possibilities for system-wide optimization.

**Code Storage**

Thanks to the integrated SVN server of the Collaboration Platform, PREEvision can also manage the implementation artifacts of the software components. They are then subject to version and release management.

**AUTOSAR Import and Export**

Descriptions of, for example, software components or a complete ECU extract for a supplier can be generated from PREEvision. PREEvision supports the import and export of the most important AUTOSAR versions. Starting with AUTOSAR version 4, exporting of the system description, software component description, ECU extract and system extract are possible. PREEvision also allows the configuration of your own extracts.

**PREEvision supports a consistent AUTOSAR design from software and hardware design to mapping to communication design.**

Software components as well as their ports and connections can graphically be modeled in software diagrams.
PREEvision supports the AUTOSAR compliant communication design of all data elements transferred over the bus, regardless of whether via CAN, CAN FD, LIN, FlexRay or Ethernet.

**PREEvision Advantages**
- Support for CAN, CAN FD, LIN, FlexRay and Ethernet
- System signal/data mapping of data elements to system signals
- Automatic creation of signals, data mappings and signal routes
- Support for the standard formats LDF, DBC and FIBEX
- Support for J1939, including ISO 11783 (ISOBUS)
- Integration with other Vector tools

On the communications layer, one defines how software components exchange data across hardware borders via physical signals. For this purpose, PREEvision supports CAN, CAN FD, LIN, FlexRay and Ethernet networks, which can also be modeled in mixed topologies in the main network diagram. On the communication layer, the signals, PDUs, frames and schedules are specified. Table editors and automations such as the signal router are available for this purpose. For service-oriented architectures PREEvision supports Ethernet as network technology.

**Signal Router**
The signal router calculates the best route for the signals with the data elements to be transferred. In addition to many routing options and routing via delta states, all routed signals can also be displayed graphically in networking diagrams. The signal router also supports the completion of the communication matrix by automatically generating many of the communication artifacts needed.

**CAN, CAN FD, LIN and FlexRay**
In order to transmit system signals over the bus, additional details must be defined. Each bus segment is configured dependent on its technology here. One component of this configuration is the PDU layout and frame layout. In this phase, signals are assigned to PDUs, which are in turn assigned to frames. Other communication attributes are also specified, such as the data type, send mode, initial value, scheduling, network management, transport protocol etc. A special editor is available for creating and editing frames, PDUs and signals. Protocol-specific editors for CAN, CAN FD, LIN and FlexRay are also available. With gateways, signals can be transmitted from one bus system to another. Regardless of the network technology used, the routing of the signals can be traced from sender to receiver. For integration with additional tools, PREEvision supports the LDF, DBC and FIBEX formats in addition to AUTOSAR.

**Partial Networking**
With partial networking, parts of architectures can switch off temporarily to conserve available resources. For this purpose, systems are subdivided into clusters. Via control and status ports, the components of a cluster are mutually woken up, kept in stand-by-mode or shut down.

**J1939**
PREEvision also supports the CAN network protocol SAE J1939 for commercial vehicles, including ISO 11783 (ISOBUS) for agricultural machines. ISOBUS enables uniform and easy control of different add-on equipment from the cab of the tractor. Expanded CAN multiplexing is supported.

**AUTOSAR Toolchain**
PREEvision is part of the Vector AUTOSAR tool chain and works with CANoe, DaVinci Developer and DaVinci Configurator Pro.
PREEvision supports developers integrating Ethernet networks into vehicles as well as the setup of a service-oriented architectures (SOA).

With PREEvision, Ethernet networks and clusters can be easily described from scratch. A dedicated user interface, the SOA & Ethernet Explorer, guides the user step by step through the design process.

### PREEvision Advantages
- Dedicated user interface for Ethernet design from scratch
- Automations for system and communication design
- Diagram for modeling of switch configurations
- SOA and package diagrams
- AUTOSAR import and export

### SOA & Ethernet Explorer
The SOA & Ethernet Explorer helps the developer to design service-oriented architectures as well as their implementation and to model the Ethernet-based communication. Automations assist the user and reduce development time. Based on a given service design PREEvision creates the respective software types as well as the software interfaces and connects the services automatically. Many communication artifacts are also created automatically. PREEvision checks for completeness of the model and hereby ensures an AUTOSAR-compliant description.

### QoS Switch Diagram
To divide physical Ethernet networks, so-called VLANs may be used for security reasons and to define different quality of service levels. In PREEvision, the quality of service mechanisms of switches can be modeled in a diagram. With the QoS switch diagram you can set up and configure FIFOs, shapers and schedulers for each port of a switch. In general, PREEvision offers many graphical editors which help to see the big picture and keep track of the relations within the model. In the network diagram, for example, configured VLANs can be highlighted for easy identification.

### SOA
When restructuring the software architecture from a signal-based to a service-oriented design, PREEvision supports the developer with comfortable tables and diagrams:

<table>
<thead>
<tr>
<th>Service Provider</th>
<th>Service Consumer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calculate Speed</td>
<td>Current Speed</td>
</tr>
<tr>
<td>Update Speed</td>
<td></td>
</tr>
<tr>
<td>Speed Limit Exceeded</td>
<td></td>
</tr>
</tbody>
</table>

The UML-based SOA diagram helps to define use cases and to derive services, service roles and interfaces. To understand the bigger picture, the UML-based graphical notation of package diagrams visualizes the larger contexts and relations within the model. In upcoming PREEvision versions further UML-based diagrams will be supported.

### AUTOSAR Classic and AUTOSAR Adaptive
PREEvision supports service-oriented architectures and Ethernet design for AUTOSAR Classic and AUTOSAR Adaptive (Adaptable starting from version R17-10) including configuration support of SOME/IP for both platforms.
As one of the most expensive and heaviest single parts of an E/E architecture, the wiring harness offers enormous potential for optimization and cost savings. PREEvision supports development engineers in finding cost-optimized designs and working out all the details.

PREEvision Advantages
- Consistency from initial design to production order
- Precise representation of model data
- Easy exchange of designs and geometry data via KBL
- Automation such as wiring harness router and ground spot optimization
- Automatic calculation of cable lengths, weight, costs, etc.
- Easy creation of diagrams, including plugs and pin assignments

PREEvision supports a comprehensive development from architecture design to series production. Due to the shared database in one integrated model, wiring harness engineers can use the data of the architecture design to elaborate the wiring harness in detail. In return, architects can use wiring harnesses of the series production for optimization. For the architecture and wiring harness design, PREEvision provides the modeling layers of the hardware network topology, the electric circuit diagram, the wiring harness and the vehicle geometry. Additionally, PREEvision provides special tools such as a wiring harness router, import and export functions via KBL or the power calculation that is important for architects.

Hardware Network Topology
For the basic hardware architecture, components such as ECUs, sensors and actuators can be modeled as well as the connections between the hardware components.

Electric Circuit Diagram
In the electric circuit diagram, hardware components and their connections are refined through the description of the electrical characteristics and the definition of power supply and ground. Integrated circuit synthesis automatically generates schematic connections and spares the developer from having to carry out error-prone activities. PREEvision provides hardware modules and components such as fuses and resistors for modeling the internal electrical design of components. The power calculation of PREEvision determines the static power consumption of a component network in various operating states. The required diameter and materials for cables and pins, for example, can be determined based on this.

Vehicle Geometry
The vehicle geometry can be modeled or imported from a 3D KBL file. In the geometry diagram, possible routing paths, installation locations for components and connector locations can be defined.

Geometry Mappings
In PREEvision, installation locations and connector locations must be connected via so-called mappings with E/E components and wiring harness connectors. This design step connects the wiring harness layer and the geometry layer and is prerequisite to use the wiring harness router.

Wiring Design
Based on an electric circuit diagram, PREEvision can synthesize a wiring diagram with connections and pins which can be refined with details such as leads and cables. The wiring harness layer contains all wire connections, connectors, splices, isolation points and pins.

Wiring Harness Routing and Terminal Determination
If the connectors of the wiring diagram are mapped to connector locations in the geometry, the wiring harness router can calculate the optimal routing via wiring segments as well as the needed isolation points. As final steps terminal determination and adaption of the wiring harness follow for production-readiness.

Data Exchange
For the series production of wiring harnesses PREEvision supplies typical wiring harness diagrams as a PDF or image file for series development of the cable tree. PREEvision also supports KBL (VDA "Kabelbaumliste" format) for exchanging manufacturing master data, circuitry information and geometry data. Additional information for smooth cooperation between the OEM and supplier can be exchanged as reports.

The diagrams for wiring harness design are based on the linked model data. Thus, editing diagrams means editing model data.
To minimize the effort for design, development and maintenance of safety-related systems as per ISO 26262, PREEvision offers consistent development support.

The ISO 26262 standard for functional safety of road vehicles places considerable requirements on the development of safety-related systems. The system must achieve the required Automotive Safety Integrity Level (ASIL) of the safety-relevant functions. This also applies to all hardware and software components that contribute to the execution of a safety-relevant function. Requirements of ISO 26262 include responsibilities, development processes, documentation and technologies for the development of safety-relevant systems. PREEvision supports the entire safety process, from system design to safety case.

### PREEvision Advantages
- Integrated from system design, through HARA, FMEA and FTA, to the safety case
- Consistent modeling of all design artifacts in a single tool
- Transparency and traceability for all stakeholders
- Automatic consistency checks for libraries for functions, malfunctions, operating situations and modes
- Generation of reports for the safety case
- Adjustable templates for the safety plan

### Hazard Analysis and Risk Assessment
Hazard Analysis and Risk Assessment (HARA) identifies and classifies the hazards that can potentially come from an item. Another objective is the formulation of safety goals for prevention or reduction of hazardous situations in order to rule out excessively high risks. A powerful editor and libraries for functions, malfunctions, operating situations and operating modes ensure efficiency here.

### Quantitative and Qualitative Safety Analysis
PREEvision supports FMEA (Failure Mode and Effects Analysis), as well as FTA (Fault Tree Analysis). PREEvision can use the existing system design for analysis and optimization here. This integrated approach reduces the effort for the implementation and maintenance of a consistent analysis. PREEvision provides test engineering and test management for verification and validation.

### Functional and Technical Safety Concept
Safety goals can be refined into functional safety requirements for development of the functional safety concept. For the technical safety concept, they are in turn implemented by technical safety requirements.

### Safety Cases, the Safety Plan and Reports
PREEvision provides a powerful report generator. A consistent safety case can be created based on the work results created by the safety manager without excessive effort. This also makes the creation of the development interface agreement possible. The predefined report templates can be used directly, but can also be adapted to special requirements at the company.
PREEvision supports test engineering, control of tests and management of accumulated test data across the entire E/E development process.

**PREEvision Advantages**

- Traceability from requirements to test report
- Easy integration into existing test tool landscapes
- Optimized for vehicle development
- Adaptable report templates
- Optimized for vehicle development
- Adaptable report templates
- Optimized for vehicle development
- Adaptable report templates

**Requirements Management**

Requirements management helps all project partners achieve a common understanding of the product to be developed and its quality. To test requirements as efficiently as possible, test cases can be assigned to requirements with PREEvision.

**Test Engineering**

During test engineering, test scenarios covering all product requirements are developed. For this purpose, test specifications and test cases are created and managed in a structured way in PREEvision and then serve as the template for subsequent test implementation.

**Test Data Management**

PREEvision manages file-based test data, such as test scripts and logging files. The widely used Subversion tool (SVN) is utilized in the PREEvision Collaboration Platform for configuration, change and version management.

**Automatic and Manual Tests**

PREEvision manages and visualizes automatic tests with the corresponding test scenarios. The actual test case is initialized in a test execution tool such as CANoe. If test automation is not beneficial, inefficient or even impossible, manual tests make sense. Manual test cases are described and managed. Test scenarios are assigned in individual test steps in PREEvision. The results of the automatic test cases are imported from test reports of the execution environment. With manual tests, results are entered directly in PREEvision. In this way, every test run can be evaluated later on.

**Monitoring and Trend Analysis**

The cockpit view provides test managers with the current status of all test projects. The most important information is collected and visualized in the main window. Charts provide a well-laid-out and quick-to-understand representation of key information, which serves as the basis for management of all test activities in a targeted way. The cockpit view can be completely adapted to the needs of individual organizations thanks to its editor. Using trend analysis, PREEvision maps the development of key figures (metrics) over time. A variety of different diagrams are available for visualization of metrics and results.
PREEvision supports cross-location cooperation in E/E projects. The PREEvision Collaboration Platform product option provides project and development teams with common access to required data without conflicts.

PREEvision is designed for carrying out complex E/E projects in large organizations and can serve as a central data backbone for all E/E development. The effort required to merge datasets is minimized. At the same time import and export functions enable easy integration into existing tool and system landscapes.

PREEvision Advantages
> Common access to a central database (single source)
> Conflict-free cooperation thanks to "lock & commit" concepts
> CODification of development status through version management
> Freely definable role and rights management
> Configurable life cycle management
> Change history and change marker
> Integrated file management with Subversion (SVN)
> Customizable views on the model data (scopes)

“Lock & Commit” Concept
In PREEvision, a "lock & commit" concept prevents data collisions, and ensures conflict-free work and efficient cooperation. All operations can be executed directly on development artifacts, regardless of whether they are hardware or software components. The lock mechanism is implemented automatically. Changes to artifacts are easy to identify thanks to change markers.

Scopes
Scopes provide dedicated views of the model, that only show the contents that are relevant for a use-case. Scopes can be provided centrally or they can individually be defined by the user.

Version Management
Integrated version management enables the management of all development artifacts in versions and is implemented with check-in and check-out mechanisms. Versioning not only occurs for the entire product line model, for individual model layers or sub-packages, but also in a fine-grained way for any individual artifact. Development artifacts can be defined as a version individually or in packages. Using the version history, development can be traced and tracked over time.

Role and Rights Management
Access to E/E projects, views and perspectives, artifact classes and even specific actions is controlled explicitly through role and rights management. Individual roles and rights can be defined for each project. Read/write or read-only rights can be assigned even on the level of individual components with PREEvision. The roles model can be expanded as desired.

Life Cycle Management
Life cycles in PREEvision enable the management of work flows which are attuned to specific organizations. A state in the life cycle of an artifact reflects the current development status and degree of maturity and can influence certain properties of the artifact. This makes the development of artifacts traceable over their life cycles. Life cycles may be freely defined here.

Integrated File Management
Using the PREEvision Collaboration Platform, integrated management of files is also possible. Files can also be versioned in revisions and branches, in a similar way to data. Using the integrated Subversion server (SVN), complete directory structures are managed. The files in PREEvision can also be accessed with external SVN clients.

Change marking (delta sign) indicates even tiny changes on the attribute level. The life cycle (color bar) indicates the development state of the artifact. Check-in allows development states to be codified, and development can be traced in detail using the change history and comparison functions.
To steer E/E projects and to carry out changes during a project in a controlled and comprehensible manner, PREEvision provides an integrated process support consisting of project, change and release management.

**PREEvision Advantages**

> Integrated project and release management
> Ticketing for change requests
> Change history and change marker
> Roles and rights management
> Customizable life cycles
> Reports, tables and cockpit views for project controlling

Besides typical project planning functions, PREEvision offers a ticketing system to handle and resolve defects and provides dedicated features to steer, trace and test changes. Due to the integrated approach of PREEvision, change requests in the form of tickets can be directly linked to development artifacts. Therefore, changes remain traceable at any time.

**Project and Release Management**

PREEvision provides the functionality of classic project management tools: All tasks within the project can be organized in sub-projects or work packages. The work packages may include features, requirements, tickets or test tasks and can be assigned to resources. With the support of individual working hour models for each resource, PREEvision provides flexible working time models. With milestones and release dates, working packages and sub-projects are distributed along the timeline. The progress and the actual state of each development artifact can be tracked via a life cycle management.

**Change Management**

With the integrated change management, modifications of development artifacts can be carried out in a controlled and coordinated manner. PREEvision offers multiple functions for this: A simple change marker indicates any changes even on the level of attributes and makes them easy to find. The change history tracks the changes during time. Also an extensive ticket system supports the processing of defects and change requests.

**Ticket Management**

Change requests and defects are processed as tickets and can directly be linked to development artifacts like functions, for example. Tickets can be categorized by severity, occurrence, priority and safety relevance. Comments and file attachments facilitate discussions and documentation. To resolve tickets in a structured manner, they may be combined in working packages or ticket sets and be classified as duplicates or parts of existing tickets. If a ticket is relevant for more than one release, merge tasks ensure the implementation in multiple development branches.

For execution and monitoring of tickets, PREEvision provides tailored tables.
With PREEvision products, you can compose the optimum configuration for your use cases: PREEvision Architect supports all modeling layers for product lines; PREEvision Function Designer and PREEvision Electric Designer support subareas. The PREEvision Collaboration Platform is the add-on for team operation, change management and product and release management.

PREEvision is the engineering platform for the E/E development process and includes the following products:

- PREEvision Architect
- PREEvision Function Designer
- PREEvision Electric Designer

As well as the following product option:

- PREEvision Collaboration Platform

### PREEvision Architect
PREEvision Architect provides the integrated scope of functions for designing an E/E architecture, ranging from requirements management, logical architecture, AUTOSAR system design, software and hardware architecture, to wiring harness and vehicle geometry. Model evaluation, optimization and documentation round out the design process. Also, development to series production readiness is supported in the following areas: consistent description of requirements, specification of functions, components, networks, and definition and maintenance of logical and physical interfaces. Development support for the design of safety-related systems according to ISO 26262 as well as test engineering and test management are also included.

### PREEvision Function Designer
PREEvision Function Designer fully supports the AUTOSAR system design process alongside the description of requirements, design of logical functions, software and hardware components, network architecture and communication design. Development support for the design of safety-related systems according to ISO 26262 as well as test engineering and test management are also included.

### PREEvision Electric Designer
PREEvision Electric Designer supports requirements management, hardware architecture design, circuit diagrams, wiring harness design, and vehicle geometry. Development support for the design of safety-related systems according to ISO 26262 as well as test engineering and test management are also included.

### PREEvision Collaboration Platform
The server-based PREEvision Collaboration Platform enables team operation of PREEvision and includes product and release management as well as change and configuration management.