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 signal-based and service-oriented architecture development, requirements engineering, 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
- Description of diagnostic functions
- Model-based wiring harness development
- Efficiency through concepts such as reuse and product line and 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.

**Intuitive User Interface**
The highly customizable 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 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 modules or systems can be reused.

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

**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 process, 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 unwanted developments and rule violations.

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

**PREEvision Use Cases**
- Design, evaluation, and optimization of electric/electronic architectures ................................... 4
- Product line approach ..................................................... 6
- Integrated requirements engineering .......................... 8
- AUTOSAR system and software design ................... 10
- AUTOSAR communication design .......................... 12
- AUTOSAR Ethernet and SOA ..................................... 14
- AUTOSAR Adaptive design .......................................... 16
- Diagnostics ...................................................................... 18
- Wiring harness development ..................................... 20
- Test engineering and test management .................. 22
- Collaboration, multi-user operation, and parallel work from different locations ............... 24
- Development of safety-related systems according to ISO 26262 ................................................. 26
- Project, change and release management ............... 28
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 use cases, 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, their behavior, 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.

**Diagnostics**
On the diagnostics layer, diagnostic objects can be described and linked to the application software. This ensures consistency between diagnostics and its realization in software.

**Hardware Architecture**
On the hardware layer, ECUs, high-performance computer, sensors, and actuators, their networking via bus systems as well as the power supply are modeled.

**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.

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

**Geometry and Harness Design**
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.
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.
- Traceability from requirements through all layers including the wiring harness

PREEvision supports the design of E/E architectures across all relevant technical layers. The abstraction layers extend from the logical architecture, through the software architecture and hardware architecture, to the hardware geometry. Data available from other tools can be imported and thus easily amalgamated in a model. Signal and wiring harness routers and automatic consistency checks accelerate development and ensure consistent modeling.

**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 can 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 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 quantitative evaluation and from this, weight or bus-load, for example, may be calculated.

PREEvision provides the UML-based SOA diagram for service-oriented architecture development.

**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 metrics are implemented with 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.
Product Lines PREEvision

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 concept for single artifacts, modules, and systems
- Version management
- Abstraction concept for dependencies between development branches
- Variant management according to AUTOSAR
- Feature-Oriented Domain Analysis (FODA)

**Variant Management**
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 electric or gasoline-operated engines. 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 provides a powerful variant management according to AUTOSAR.

**Reuses**
PREEvision offers a powerful reuse concept to minimize redundant development effort. This saves time initially, but also in later stages of projects. In case of changes, a reused design must be changed only once. All reuses in downstream levels can be updated automatically. Even continuous release cycles in the development process can be realized. The version management of PREEvision allows to freeze revisions with check-in mechanisms. You can later decide if you want to continue working with a new revision or if you want to open a branch for parallel development. Version management not only applies to the entire product line, to individual model layers or systems, but also in a fine-grained way for any individual artifact.

**Libraries**
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 modules or systems and then be reused as a whole. A system 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. No matter if single artifacts or complete systems: A product line with all its contents can serve as a library for the downstream development levels.

**Merging of Data**
For further development in branches or the integration of imported data, PREEvision provides a detailed comparison view. This allows a controlled merging of data.

PREEvision supports an E/E development in multiple levels. Starting from a network architecture design for a complete product range, for example, the design is then concretized in several levels.
PREEvision supports requirements management in such a way that requirements are integrated into the model and therefore can seamlessly interact with all development artifacts.

In PREEvision, requirements do not exist separately. Thanks to its consistent data model, they are development artifacts themselves and can directly interact with, for example, software and hardware components. The artifacts may be linked to requirements with mappings or are embedded directly in requirements using placeholders. With the integration of requirements in the S/E development, PREEvision extends far beyond the functions of a classic tool for requirements management. All development artifacts can be seen as requirements and used in system specifications. Attributes like the price of a hardware component can be included via placeholders in requirement descriptions.

PREEvision Advantages
- Tables and rich text editor with graphical support for recording requirements
- 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 via ReqIF or Excel
- Easy report creation
- File attachments management
- Integration with test engineering and test management
- Support for reviews and votings

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.

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 many functions for developing and managing requirements after creation. The reuse of complete packages of requirements increases efficiency. 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 many predefined tests that can be expanded with user-specific checks. Correct implementation of requirements is ensured through test engineering and test management.

Changes & Releases
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.

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

Architecture Design
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, for both AUTOSAR Classic and AUTOSAR Adaptive. In all 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 a consistent AUTOSAR design from software and hardware design to communication design for all network technologies including Ethernet.

#### Type-Prototype-Instance Concept

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 4 versions. Exports of system descriptions, software component descriptions, ECU extracts and system extracts are possible. PREEvision also allows the export of Adaptive formats and the configuration of your own extracts.
PREEvision supports the AUTOSAR compliant communication design for all bus systems, 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
- Numerous routing options and delta state routing
- Support for the standard formats LDF, DBC, FIBEX, and ARXML
- Support for J1939, including ISO 11783 (ISOBUS)
- Support for SecOC and TLS
- Integration with other Vector tools

On the communications layer, one defines how software components exchange data across hardware borders via 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 if necessary are specified. Table editors and automations such as the signal router or the automatic generation of PDUs and frames are available for this purpose.

**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 be displayed graphically in the network diagram. 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 arranged in PDUs, which are 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, FlexRay, and Ethernet 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.

**Secured Communication**

Due to safety-critical functions like autonomous driving, security for onboard communication is growing more and more important. PREEvision supports solutions like Secured Onboard Communication (SecOC) und Transport Layer Security (TLS). This allows the direct implementation of the safety goals in the model.

**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 set-up of a service-oriented architectures (SOA).

With PREEvision, Ethernet clusters can be easily described from scratch, both for AUTOSAR Classic and AUTOSAR Adaptive. Dedicated user interfaces like the SOA & Ethernet Explorer guide the user step by step through the design process.

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. Further aspects like transformers (End-To-End, SOME/IP) can comfortably be modeled and reused for the generated signals. Also, standardized modelings like service discovery are generated with one click and can then be configured. Additionally, PREEvision checks for completeness of the model and hereby ensures an AUTOSAR-compliant description.

PREEvision Advantages
> Dedicated user interface for Ethernet design from scratch
> Automations for system and communication design
> Diagram for the modeling of switch configurations
> SOA and package diagrams
> Support for modeling transformers and service discovery
> AUTOSAR import and export

In the software architecture, services and interfaces are modeled and implemented by software components. The signal router follows the switch configuration defined in the communication layer.
AUTOSAR Adaptive addresses the complex and interdependent functionalities of modern vehicles and the resulting demand for high-performance computing, bandwidth and software updates over the air.

PREEvision offers multiple diagrams and a dedicated user interface to model AUTOSAR Adaptive components as well as combined systems of AUTOSAR Adaptive and AUTOSAR Classic. The AUTOSAR Adaptive Explorer provides tools like tables or data syntheses to create an Adaptive system from scratch.

PREEvision Advantages
- SOA as common basis for AUTOSAR Adaptive and Classic
- Modeling of combined systems of AUTOSAR Classic and Adaptive
- Dedicated user interface for the Adaptive design
- UML-based class diagrams and state chart diagrams
- Support for high-performance computers with multicore architectures and virtualization
- Import and Export of different AUTOSAR files

Service-Oriented Architectures (SOA)
Service is an abstract description for a set of software functionality with the purpose to be available and be reused by several different clients. The service-oriented architecture diagram in PREEvision helps to overview and understand complex systems by depicting the dependencies and relations between services. Services communicate via defined interfaces. These service interfaces define methods, fields, and events that can be accessed by the clients of the service. Service interfaces can be designed using UML-based class diagrams.

Adaptive Applications
On application or platform level, an Adaptive Application implements one or more services. It also groups the collection of executables that are the unit for the software delivery. For deploying an Adaptive Application, startup dependencies to other applications or to states of the machine must be defined. For this design step, PREEvision provides the Process Startup Dependency Editor. For designing the state machine of an application, the state chart diagram can be used.

Combined Architectures
To successfully introduce the new platform, hardware and software based on Adaptive must work well with the existing AUTOSAR Classic components. The service-oriented architectures with services and service interfaces can bridge the gap between the two worlds. In PREEvision, the Port Adapter helps to connect Classic to Adaptive software components. For the communication via Ethernet as exclusive bus technology for AUTOSAR Adaptive, PREEvision supports the SOME/IP protocol including synthesis functions for the binding of interfaces. The serialization of the service interfaces is defined by means of transformers. In PREEvision, transformers for SOME/IP and end-to-end protection are supported.

High-Performance Computers
In modern vehicles, complex ECUs and high-performance computers (HPCs) complement the simple ECUs of AUTOSAR Classic to fit the computing needs of functionalities like connectivity, electrification or autonomous driving. High-performance computers aggregate microcontrollers and microprocessors with multiple cores. Basic software and applications share these and other available resources (Memory, IO, communication channels, etc.) by using a Hypervisor. The combination of platforms (AUTOSAR Classic, AUTOSAR Adaptive, Android, etc.) on one hardware is possible.

Import and Export
PREEvision supports AUTOSAR 19-03 and the following design products:
> Service Interface Description
> Application Description
> Machine Manifest
> Service Instance Manifest

Service design as the basis for the architecture, Ethernet as the network technology and a hardware topology including high-performance computers are the characteristics of AUTOSAR Adaptive.
With PREEvision, consistent and AUTOSAR-compliant development of software and diagnostics as part of a contemporary E/E development process is possible.

**PREEvision Advantages**
- Model-based approach ensures consistency between diagnostic data and realization in software.
- Lower coordination effort and less manual effort.
- Reduction in potential sources of error.
- AUTOSAR-compliant modeling of diagnostics and application software.
- References between diagnostic objects and diagnostic ports are transferred over the entire tool chain and enable automatic allocation.

The specification of diagnostic data of an ECU and its software are typically realized independently of each other by different people in different roles in different organizations. In addition to that, the information is maintained in different authoring tools without a standardized data exchange. The consistency between the different authoring tools must be maintained manually.

This makes the process complicated and can lead to inconsistencies, for example, if details of the data type in the diagnostic specification do not match those of the implemented application software. Such inconsistencies are typically detected at a late point in time and are therefore expensive to correct. The overall coordination work leads to high manual effort and makes the development process error-prone.

**Diagnostic Data Exchange**
The diagnostic information modeled in PREEvision can be exported to the Vector tool CANelaStudio where it can be further refined and supplemented. The scope for the diagnostic export is based on the Software to Hardware Mapping, means the availability of diagnostic objects on a dedicated ECU. Changes made in CANelaStudio can be written back to PREEvision by data exchange, and the software details are updated automatically. It is no longer necessary to manually adapt the diagnostic data to the software.

So, PREEvision and CANelaStudio offer a bidirectional data exchange and support various cooperation models. The data exchange between both tools helps keeping internal and external views on diagnostics in sync and considers services and fault memory with all the details derived from the diagnostic ports.

**Diagnostic Specification**
In PREEvision, each ECU is represented by a diagnostic master in the diagnostics layer. The diagnostic objects are referencing software parts and data elements or interfaces. Details like data types, conversions, and intervals are directly derived from the software implementation. The structure and availability of services of a routine are derived from the assigned client server interface. This ensures consistencies between diagnostic information and software model.

**Diagnostics Modeling Layer**
PREEvision offers a dedicated modeling layer for diagnostic objects. In this diagnostics layer, the user can define diagnostic information like:
- Diagnostic data identifier (DID)
- Diagnostic controls (I/O Control)
- Diagnostic routines (Routine)
- Diagnostic events (Event)
- Diagnostic Trouble Codes (DTC)

Certain supplemental properties can also be described for these objects such as identifiers or the diagnostic services that are used.

Relevant parts of the AUTOSAR event handling can be modeled as well. Diagnostic objects can be linked directly to its realizing application software in the software layer. Further details of the diagnostic specification are refined and supplemented in authoring tools like CANelaStudio.

**Diagnostic Scope**
The relationships between software, hardware, and diagnostic design and how the information for diagnostics is derived from them.
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.

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 wires and cables. The wiring harness layer contains all wire connections, connectors, splices, isolation points and pins.

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.

Wiring Harness Routing and Terminal Determination
To connect the wiring design with the geometry in PREEvision, installation locations and connector locations must be connected via so-called mappings with E/E components and wiring harness connectors. 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. Via VEC information of the electric circuit and the wiring design including master data can be exported. 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.

PREEvision supports wiring harness design from architecture to series development.
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
- Libraries for functions, malfunctions, operating situations and modes
- Generation of reports for the safety case
- Adjustable templates for the safety plan

Item Definition
According to ISO 26262, item definition describes the vehicle system to be analyzed, and its interactions with the environment and other items. In PREEvision, the item can be described in text form and modeled graphically with system diagrams.

PREEvision supports the development of safety-relevant systems with multiple editors, diagrams, analyses, and reports.

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 definition 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 modes ensure efficiency here.

Functional and Technical Safety Concept
For development of the functional safety concept, you can refine safety goals into functional safety requirements and describe activity chains for individual architecture components. Subsequently, the technical safety requirements are derived from the functional safety requirements and the system design and the technical safety concept are developed. For the system design, multiple diagrams are available.

Quantitative and Qualitative Safety Analysis
PREEvision supports FMEA (Failure Mode and Effects Analysis), as well as FTA (Fault Tree Analysis). The safety engineer can use the existing system design in PREEvision for analysis and optimization. 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.

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 report templates in PREEvision can be used directly, but can also be adapted to special requirements at the company.

Events and logic gates are the basis of the Fault Tree Analysis (FTA) to determine the probability of malfunctions of components.
PREEvision supports test engineering and the management of test processes and accumulated test data across the entire E/E development process.

PREEvision Advantages
- Traceability from results to requirements
- Tracking of test execution progress and test specification coverage
- Trend analysis of test results and KPIs over time
- Optimized for vehicle development
- Adaptable report templates
- Management of test environments and test objects
- Version management for test data
- Multi-user platform including roles and rights administration
- Out-of-the-box interaction with Vector’s vTESTstudio and CANoe

Test Engineering
During the test engineering process, 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. They serve as the base for test implementation and execution. Additionally, frequently used test cases and test conditions can be stored in libraries and reused in test specifications. The test implementation can be generated automatically from the corresponding test specification.

Test Management
The tests that are required to ensure the product quality can be planned, executed, documented, and evaluated in PREEvision. This way, it is possible to make statements about the current test process and degree of maturity of the product at any point in the product development process. Additionally, PREEvision manages file-based test data, such as test scripts and logging files.

Automated and Manual Tests
PREEvision consolidates automated tests into test scenarios. The execution of automated tests is initialized in a test execution tool such as CANoe. If test automation is not beneficial, inefficient, or even impossible, manual tests can close the gap. Manual tests are described in individual test steps which are then managed in test scenarios. The results of the automated tests are imported from test protocols of the test execution tool, the results of manual tests are directly entered in PREEvision. Every result of a test run can be evaluated later with an additional verdict if necessary.

Monitoring and Trend Analysis
The cockpit view provides test managers with the current status of all test projects. The most important data is summarized and visualized. Charts provide a well-laid-out and quick-to-understand representation of key information. The displayed data serves as the basis for monitoring all test activities in a targeted manner. The cockpit view can be completely adapted to the individual needs of test managers. The trend analysis in PREEvision illustrates the progress of test results and KPIs (Key Performance Indicators) over time. A variety of diagrams is provided for visualization purposes.
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 Advantages**

- Common access to a central database (single source)
- Conflict-free cooperation implemented with “lock & commit” concepts
- Codification of development status through version management
- Freely definable roles and rights management
- Configurable life cycle management
- Change history and change marker
- 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.

**Roles and Rights Management**

Access to E/E projects, views and perspectives, artifact classes and even specific actions is controlled explicitly through roles 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 status 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.
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 Collaboration Platform is the add-on for team operation. The Server API allows the development of web applications.

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 Collaboration Platform enables team operation of PREEvision and includes product and release management as well as change and configuration management.

PREEvision Server API
The Server API enables the development of web applications to edit and visualize model contents. The REST API allows read and write access to artifacts including their life cycle state, rich texts, files, diagrams and metrics.
PREEvision
Model-Based Electric/Electronic Development
From Architecture Design to Series Production