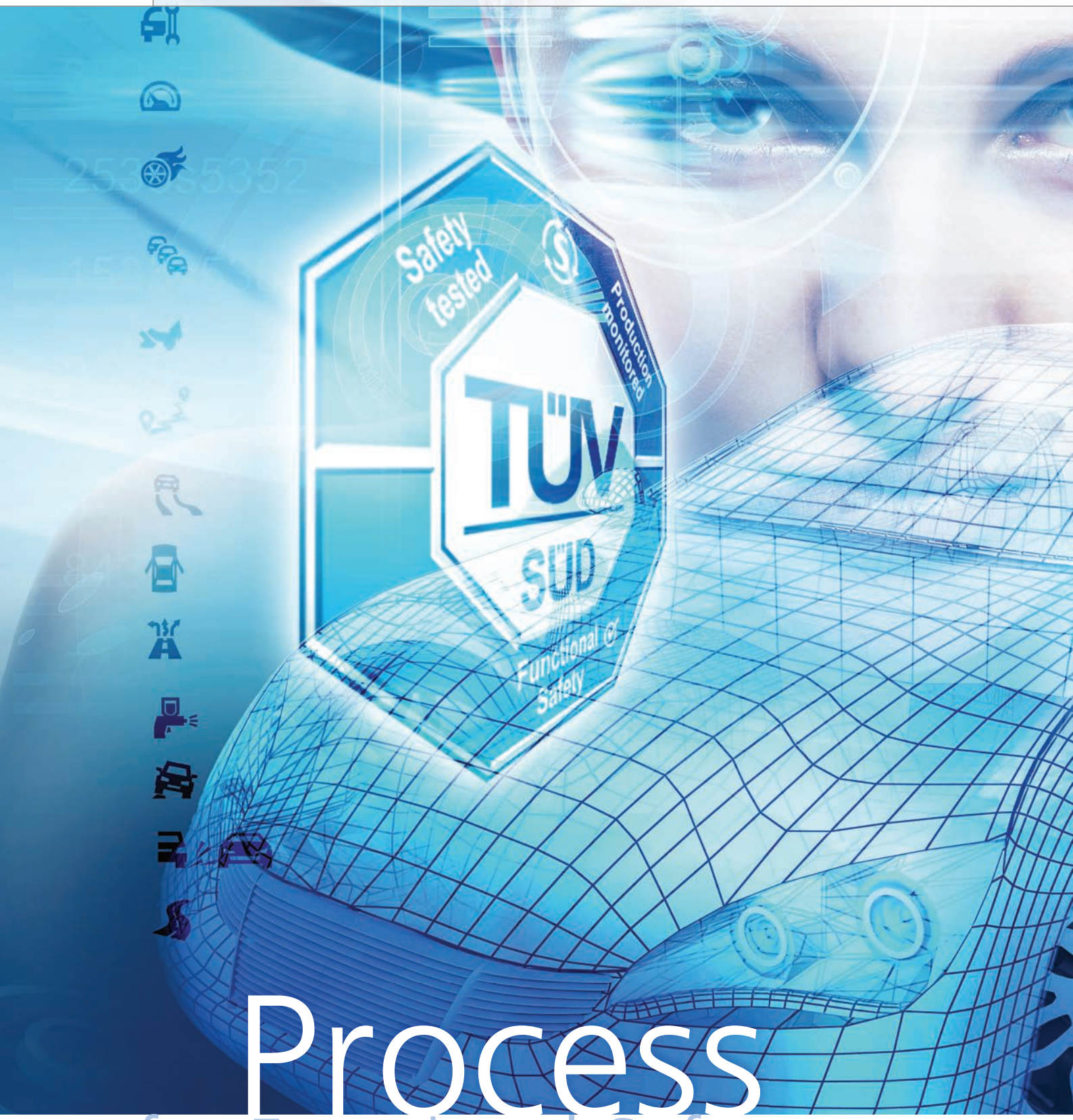


Process for Functional Safety

Model-based software development for electric drivetrains according to ISO 26262

The Siemens Drive Technologies Division has added model-based development to their classic software development process for implementing safety-critical vehicle functions. With the support of dSPACE TargetLink Strategic Partner, Model Engineering Solutions GmbH, Siemens Drive Technologies defined a process on the basis of dSPACE TargetLink that satisfies the requirements of ISO 26262 (Road Vehicles – Functional Safety).

Safety-Critical Software in a Vehicle

Many suppliers in the automotive industry are either already using model-based development for their safety-critical software, or facing the challenge of having to add model-based methods to their software development processes in the future. Model-based development brings numerous proven advantages, such as higher software quality and easier software maintenance. With the introduction of the new ISO 26262, there is now an international standard for the functional safety of in-vehicle electric/electronic systems. ISO 26262 is derived from the IEC 61508 safety standard, adapted to meet the specific conditions in a vehicle. Unlike the IEC standard, ISO 26262 explicitly applies to model-based development. This new standard clearly states *what* the development process for safety-critical, in-vehicle software must achieve to guarantee the software's functional safety with regard to the various Automotive Safety Integrity Levels (ASIL, classified in levels from A to D). However, the standard leaves the question of *how* this can be implemented largely unanswered. Practical experience shows that there is no universally accepted approach for applying the standard. Only project-specific solutions can prove successful, because they also take the company's or the department's existing processes and tool chains into consideration.

Electrical Vehicle Components from Siemens

Siemens Drive Technologies develops, produces and markets key components for e-drive vehicles. This is done from a project-based point of view, keeping in mind the specific requirements of the automotive industry customers in each project. Its portfolio covers a wide range from electric motors and power electronics to intelligent onboard charging technology (figure 1). Siemens has already established a classic development process for software functions, primarily oriented around the widespread V-cycle. This development process complies with the Automotive SPICE guidelines (aSPICE: a version of the international standard ISO/IEC 15504 'SPICE', specially adapted to automotive needs) and CMMI-Dev (Capability Maturity Model Integration for Development). Virtually all vehicle manufacturers demand compliance with these two maturity models, so compliance is absolutely essential.

Requirements for Model-Based Development of Safety-Critical Software

Siemens Drive Technologies wished to integrate model-based development into its existing classic development process because of the plan to develop a larger portion of safety-critical vehicle functions with the model-based methods of MATLAB®/Simulink®/Stateflow® and dSPACE TargetLink® in the future. To reach this goal, several criteria had to be met:

- Define an ISO 26262-compliant process for model-based development that integrates optimally into the existing process environment, considers the available tools and does not violate any of the required process characteristics (such as aSPICE conformity).
- Comply with all requirements from ASIL A – D, focusing on part 6 of the ISO standard (product development: software level).
- Define the procedures and design patterns for designing model architectures with Simulink and TargetLink within the context of ISO 26262.
- Ensure that all functional requirements can be traced back into the generated code (requirements traceability).
- Guarantee that it is possible to test against requirements already on model level, meaning even before the generated code is available.

Planning Project-Specific, ISO 26262-Compliant Implementation

To implement these requirements, Siemens Drive Technologies commissioned Model Engineering Solutions GmbH (MES) as a specialized partner. As a TargetLink Strategic Partner, MES offers industrial customers consulting on quality assurance for in-vehicle embedded software. MES provides dSPACE TargetLink Partner services that cover everything concerning ISO 26262-compliant development with dSPACE tools, including gap analysis (identifying strategic and

Figure 1: A permanently excited synchronous motor for automotive applications.

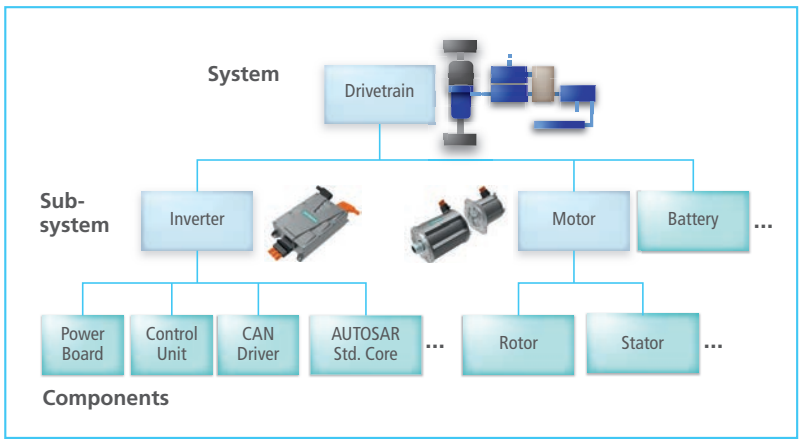


Figure 2: System partition and definition of the levels of an electrical drivetrain.

operative gaps), process modeling, process manuals and support for process implementation. Through numerous production projects, MES has acquired a comprehensive understanding of how safety-critical vehicle software is developed in Germany and how to best implement the requirements of ISO 26262, particularly in model-based software projects.

Model-Based Development Process at Siemens Drive Technologies

At Siemens Drive Technologies ISO 26262-compliant software development is performed in the field of electrified automotive drivetrains. A drivetrain is a complex system that can be subdivided into various subsystems and associated system levels (figure 2). On the subsystem level there is the inverter, which implements the electrical energy flow between the battery (direct current) and motor (alternating current), and the motor, which then converts the electric energy into mechanical energy. These subsystems consist of components themselves, such as the motor stator and the inverter's control electronics. The run-time software running on these control electronics is located on the level below the component level, called the module level. The software itself (not shown here) is divided into several software modules, each of which can be further subdivided into several software functions. In the future some of these functions will be developed with the model-based method, as described in the process presented here.

The Software Development Process

Code- and model-based software development follows the V-cycle (figure 3). At the starting point for module development there are the software module requirements such as the torque control in the electric drive. In the system development process, these requirements are created through step-by-step refinement of customer requirements up to the relevant system level (in this case, the module level). Each design phase has a corresponding test phase to validate the maturity of the functional models and software.

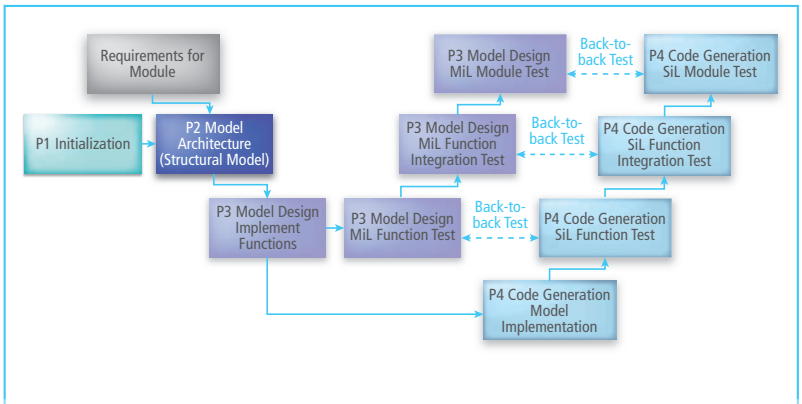
Phases of the Model-Based Development Process

The model-based, ISO 26262-compliant development process has four key phases:

Phase 1 – Initialization:
This phase covers all the preparatory activities performed before project inception, such as the kick-off meeting, setting up the development environment, specifying the requirements for the environment model, and defining and assigning roles and responsibilities. To the classic software development roles of software engineer, software architect and so on, new, additional roles for model-based software development are added, such as the model engineer who performs function modeling in Simulink. One of the requirements of the ISO standard is that this phase selects the project-relevant modeling guidelines (see ISO 26262-6, §5.4.7) that can be tested automatically with Model Examiner (MXAM).

Phase 2 – Model Architecture:
In this phase the requirements for the software module or software function to be developed are encapsulated in functionally cohesive units,

Figure 3: V-cycle for model-based software development at Siemens Drive Technologies (excerpt).



a suitable environment model is selected, and the test concept for the components is fine-tuned for model testing. The key task of this phase, though, is to have the modeler and the software architect define and implement the model architecture. The resulting model architecture, including its interfaces, is represented by the structure model, which consists of empty, interconnected Simulink subsystems. This structure model thus defines the basic framework for the software architecture, which is implemented later by TargetLink subsystems and functions. Some of the requirements the ISO standard places on the model architecture and the software architecture (see ISO 26262-6, §7.4, tab. 3) are:

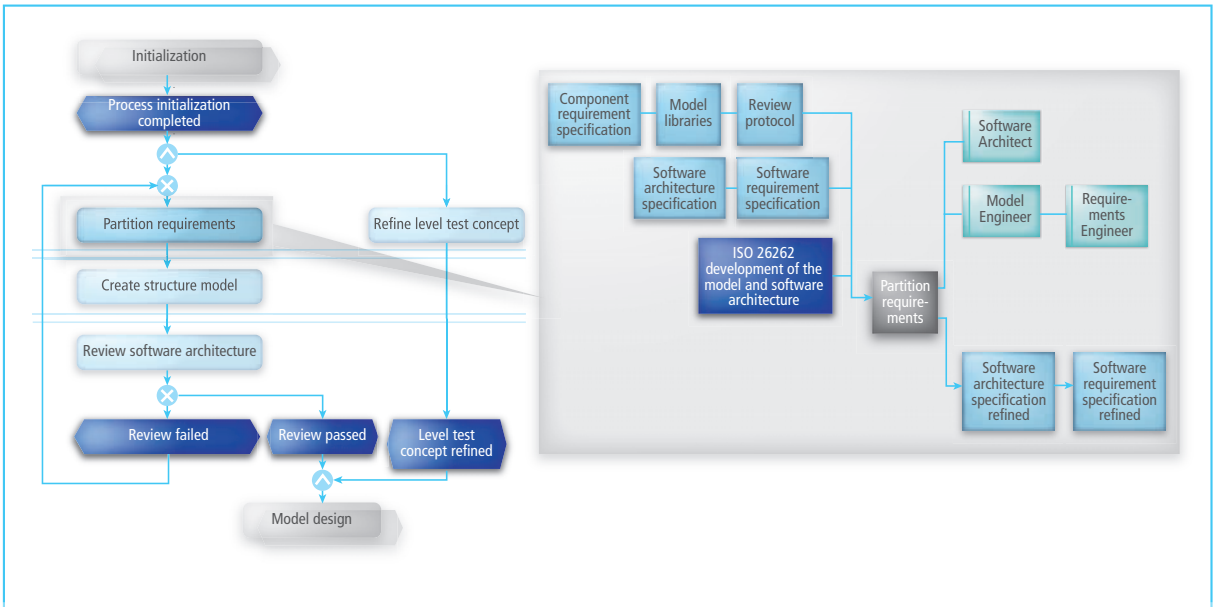
- Hierarchical structure and low software component complexity
- Small-sized interfaces with low complexity
- High cohesion inside software components
- Restricted software component coupling

These requirements are fulfilled by various methods such as (1) using design patterns for the model architecture, (2) reviewing the architecture, and (3) using the M-XRAY for measuring and evaluating the model complexity. ISO 26262 also defines requirements for testing the software modules and functions, such as requirements-based testing, interface testing and back-to-back tests between the model and code (see ISO 26262-6, tables 10 and 13). The test concept for the module or function is expanded in this phase as needed.

> "An integrated tool chain made of specialized development tools is absolutely necessary for a development process to be ISO 26262-compliant."
>
> *Dr. Ingo Stürmer, MES*

Phase 3 – Model Design:
In this phase, the functional requirements for the module and the functions are modeled in greater detail in the structure model, resulting in a functional model. In addition, the module is tested in accordance with the test concept in model-in-the-loop (MIL) mode as a Simulink simulation in double precision against the related requirements.
The advantage is that the model's compliance with the requirements can be verified at an abstract level – even before the code is available (the goal: early testing). The functional model is thus an executable specification of the functional requirements. The ISO standard especially defines the requirements for designing the software modules and functions, such as using a suitable notation. Using Simulink or TargetLink thus already fulfills the requirements for a semi-formal notation for ASIL B – D components (see ISO 26262-6, table 7).
In addition to these high-level notation requirements, the ISO standard also requires very concrete error prevention measures, such as avoid-

Figure 4: Example model of Phase 2, 'Model Architecture', with a detailed representation of the 'Partition Requirements' step.



ing implicit type conversions (see ISO 26262-6, table 8).

Phase 4 – Code Generation:
In the implementation phase, the functional model is given the information required for code generation by TargetLink (function partitioning, number representation, data types, local/global variables, etc.). Depending on the project, this phase might also have subphases. For example, if the functional model is purely a Simulink model, it must be converted into a TargetLink model first. The test results of the generated code are then compared with the results of the MIL test of the functional model as software-in-the-loop (SIL) simulation in back-to-back tests, in accordance with ISO 26262.
As soon as the generated code has been tested, the model-based software development tasks are combined with the code-based software development. This means, for example, that the test activities for the integration tests of several modules or functions once again follow the normal code verification procedure up to the system test. The test methods mentioned above (requirements-based testing, interface tests and back-to-back testing) come into play again and must meet specific integration test coverage criteria such as function coverage and call coverage (see ISO 26262-6, table 15).

Process Modeling for Model-Based Development According to ISO 26262

At Siemens Drive Technologies, development processes are modeled consistently with the tool support from ARIS. This guarantees that all the necessary process steps for the model-based development of program code are completely documented, in accordance with ISO 26262. The documentation is produced in several successive documentation stages that build on one another, as described briefly below. The development process's 'Model Architecture' phase (figure 4) can be represented concisely as an event-driven process chain (figure 4, left). A detailed representation (figure 4, right) shows which role on the basis of which document performs the process step (such as the 'Partitioning requirements' step) with regard to development results and the ISO 26262 requirements.

Process Manual for Model-Based Development

Siemens Drive Technologies provides a process manual that not only includes a graphical representation of the process steps of model-based software development but also explains the development process in detail. The manual contains all the defined process steps with detailed instructions for everyone involved. The manual also includes separate chapters on topics related to model-based development such as model-based testing, designing the model architecture, and the ISO 26262 requirements on the software process. The document structures the information for describing the different process steps in such a way that process users, and any other interested persons, can quickly get a general overview and find the information they need. The manual's chapter and subchapter structure contains five points for each process phase (figure 5). The example presented here is from the 'Model Architecture' phase:

1. Goals:
Defines the objectives to be achieved by the process phase. For the 'Model Architecture' phase this means the functional decomposition of requirements into parts that can be implemented by a hierarchy of software functions. This must also already consider the software architecture that will be defined later by the TargetLink model, to avoid time-consuming restructuring of the functional model into an implementation model. Another objective is to fine-tune the test concept for the module (figure 4).

2. Prerequisites & Inputs:
Defines which information and work products must be available when the activities are started. To reach the objectives of the 'Model Architecture' phase, the 'component requirement specification' created during the higher-level system development process and the initial version of the 'software requirement specification' must be present, for example.

3. Activities in Detail:
Describes the process steps in detail – who has to do what, which work

Figure 5: Structure of fundamental topics and steps for each phase of the ISO 26262-compliant development process at Siemens Drive Technologies.

products are required and which are created. As already explained with Figure 4, the descriptions of the individual activities must particularly refer to the ISO 26262-relevant requirements. These relate to the software architecture, the test methods, the review process, etc.

4. Work Products:
Contains a summary of all the work products resulting from the individual process phase steps, including the ones relevant for ISO 26262 such as the test level concept and review reports.

5. Success Criteria:
Lists the criteria that must be met for the process step to be declared 'completed successfully' and the next step to begin. For the model architecture, this means that it must have been implemented in the structure model and that the structure model has been successfully integrated with the integration model, to name two examples.
A glossary of all important terms, especially ones relevant for model-based development, makes it easier to understand the manual. Each chapter is a complete section in itself and does not need to be read together with the process description. Another important section of the manual explains how and with what tools the recommended methods must be implemented for model-based development.

ISO 26262-Compliant Process and Tool Chain

External support by MES played a central role in producing a practice-relevant definition of the process and implementing the process. The project partner provided valuable external know-how on what works and what does not. The experience from various Siemens projects was also a factor that contributed to success. The process modeling phase involved experts who design the overall system development process at Siemens. It became clear that the introduced processes can already fulfill the elementary requirements for an ISO 26262-compliant procedure when they are transparent in organization and optimized for safety-critical aspects. Similarly, the software tools being used must be suitable for guaranteeing high product quality and ISO compliance.
For example, the German certification organization TÜV SÜD has already certified the production code generator TargetLink for use in safety-critical development projects up to ASIL D. This qualification is based on a process's compliance with the TargetLink reference workflow, published by dSPACE, which describes the best practices for model and code validation for TargetLink. This workflow was integrated into the Siemens development process. MES used the Model Examiner (functional safety solution) to ensure that the models complied with guidelines, and M-XRAY for model metrics and handling the model complexity. Simply using these tools already fulfilled important ISO requirements for model-based development (complete coverage of Part 6, §5.4.7, table 1). The tool chain used by Siemens proved to be suitable for bringing the process defined in the standard to life in the projects. The tool chain developed at Siemens also ensures bidirectional traceability from the requirements, to the code generator/model, and to the tests.

> "The TargetLink reference workflow, approved by TÜV SÜD, made it easier for us to set up an ISO 26262-compliant, model-based development process based on dSPACE TargetLink."
>
> *Dr. Heiko Zatocil, Siemens*

Earlier Tests, and Greatly Improved Development Process Safety

The new ISO 26262 recognizes model-based development as a

high-quality approach for developing safety-critical software for vehicles. Extending the code-based process by adding model-based development brings considerable advantages because the software can be validated earlier and with the support of better methods and tools. To achieve this, the company's internal processes must be adapted to the standard's requirements. Siemens Drive Technologies has already successfully developed the first safety-critical software components in accordance with the new process model. Model-based development will play an increasingly important role next to classic development procedures and is already becoming more widespread because it follows ISO 26262-compliant process steps. ■

David Brothanek, Dr. Martin Jung, Verena Jung, Michael Krell, Reinhard Pfundt, Dr. Elke Salecker, Dr. Ingo Stürmer, Dr. Heiko Zatocil

David Brothanek
David Brothanek is Senior Manager and a specialist for the modeling, optimization and implementation of processes at Headframe IT GmbH in Essen, Germany.

Dr. Martin Jung
Dr. Martin Jung is Head of Software and System Development Consultation at develop group in Erlangen, Germany, and is also an instructor of software architecture at the Friedrich-Alexander University of Erlangen-Nürnberg, Germany.

Verena Jung
Verena Jung is a team leader in Integration and Test, and her responsibilities include coordinating test activities in software and component development at Siemens AG in Erlangen, Germany.

Michael Krell
Michael Krell is Functional Safety Manager at Siemens AG in Erlangen, Germany. He provides support for implementing ISO 26262 requirements in projects.

Reinhard Pfundt
Reinhard Pfundt is a software manager responsible for planning and coordinating the software for frequency converters, DC/DC converters and onboard charging devices at Siemens AG in Erlangen, Germany.

Dr. Elke Salecker
Dr. Elke Salecker is Senior Software Consultant at Model Engineering Solutions GmbH in Berlin, Germany. An expert on model-based software development in accordance with ISO 26262, she supports customers in process definition and implementation.

Dr. Ingo Stürmer
Dr. Ingo Stürmer is the founder and CEO of Model Engineering Solutions GmbH in Berlin, Germany. He is an acknowledged specialist in development processes with dSPACE TargetLink, and helps customers optimize their model-based development process and certify their company-specific tool chain in accordance with ISO 26262.

Dr. Heiko Zatocil
Dr. Heiko Zatocil is head of Function Development at Siemens AG in Erlangen, Germany, where he is playing a key role in advancing model-based software development. He initiated and coordinated the creation of the process described here.