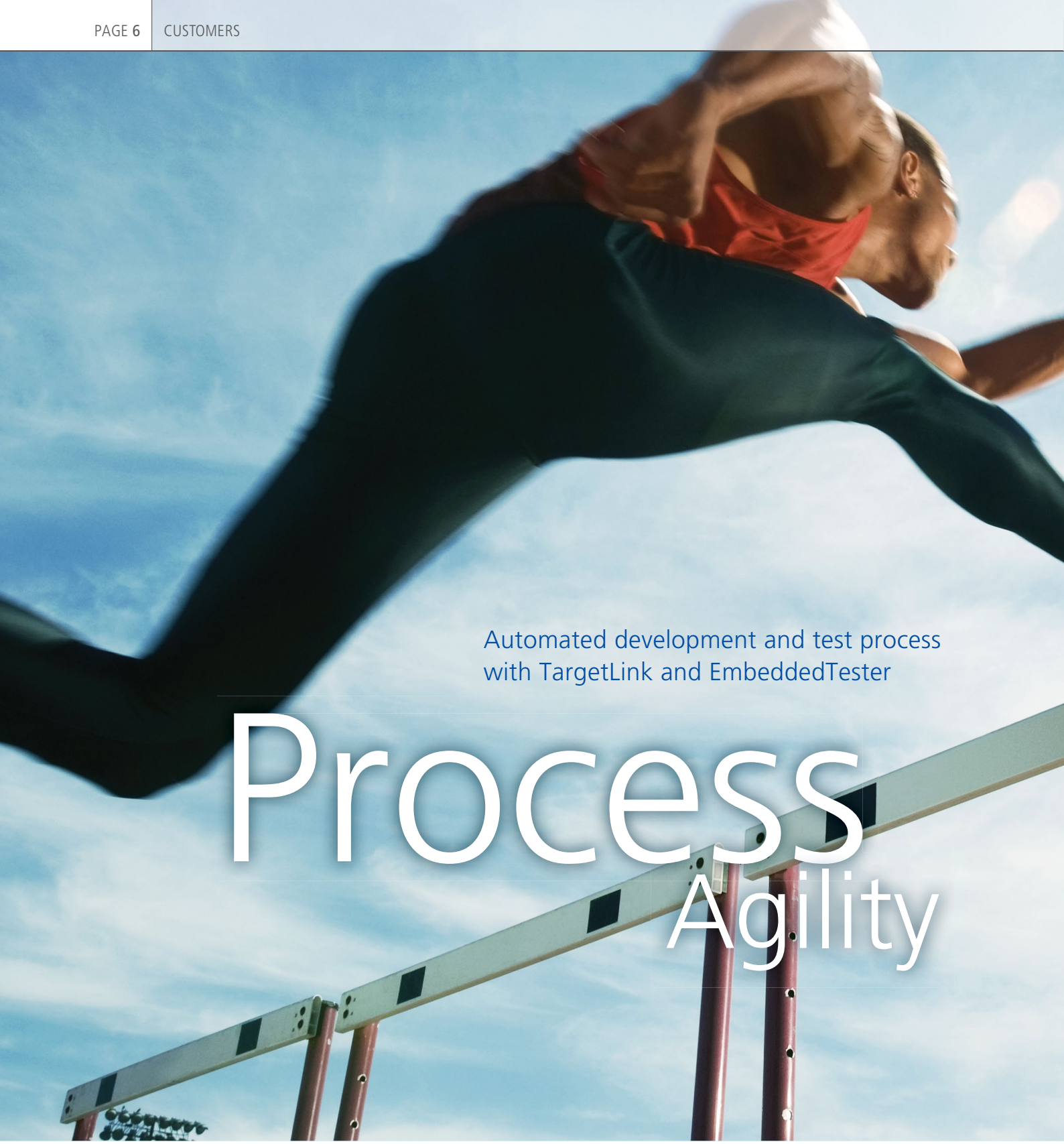


Automated development and test process with TargetLink and EmbeddedTester

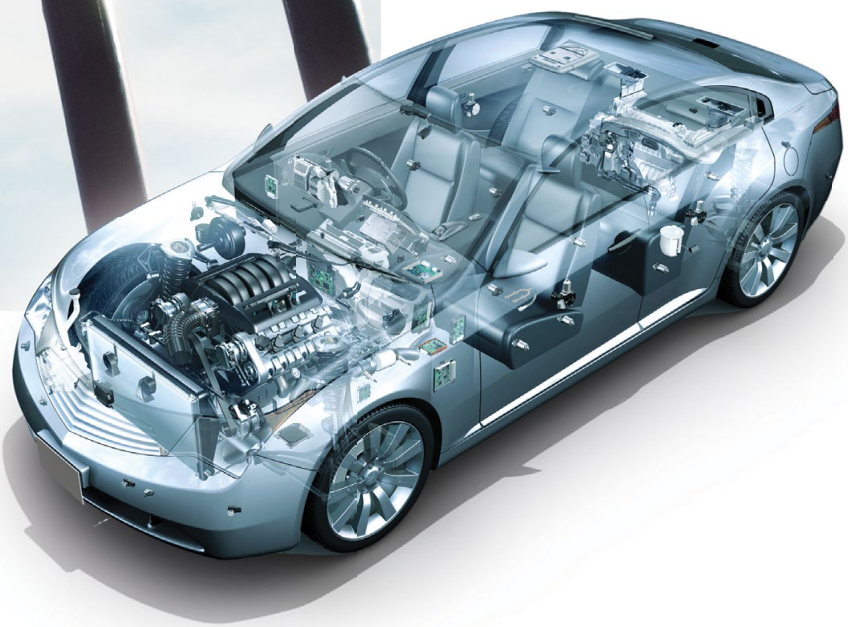
Process Agility

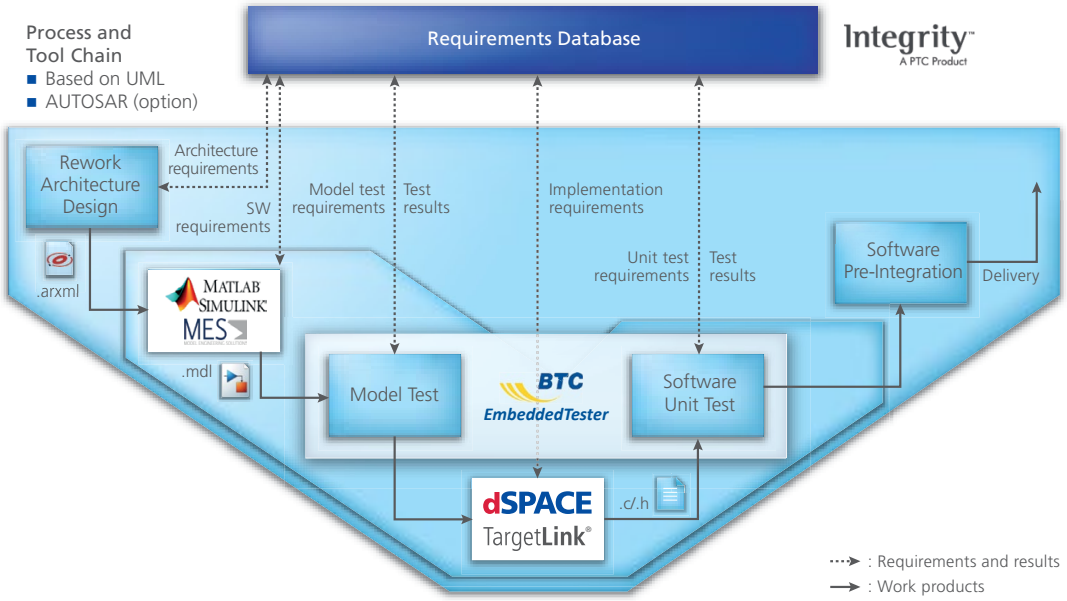
Friday afternoon. Urgent call to the supplier. The customer has a change request for their controller. Can the new feature be implemented in the scheduled time and with the full promise of quality?

The Aachen Engineering Center (AEC), a development center of DENSO AUTOMOTIVE Deutschland GmbH, has one clear goal: to stand by its European customers during product development and support them with a market-specific product range. The AEC uses the European market's high technological standard and innovative potential to boost development in engine control and powertrain, electrics and electronics, hybrid technologies, and information and safety.

>>

The vehicle functions are mainly determined by automotive ECUs.





A seamless tool chain that links the tools with the requirements in every process phase.

Defining the Development Process

Customer projects often call for model-based software development. A mature development process with a powerful tool chain is therefore the basis of every efficient project. The main goals of defining this process are:

- *Quality:*
  One of the highest goals is the quality of the software being developed. To avoid manual errors it is crucial to perform systematic tests as early as possible and to have a high degree of automation.
- *Agility:*
  Today's development projects often face great time and cost pressures. In addition, the number of changes made during a project is increasing. These changes have to be implemented and tested as quickly as possible. To stay competitive under these conditions, a highly efficient development process is required. DENSO has opted for an agile and iterative development process. The development tools are deeply integrated into this process so that new requirements can be included in the production code and tested as soon as possible.
- *Compliance:*
  Today's development processes for automotive software have to meet process standards, such as ISO 26262 and Automotive SPICE, and software standards, such as AUTOSAR and MISRA. It therefore makes sense to use development and verification tools that were developed with an eye to these standards.

Model-Based Development and Code Generation with TargetLink

A successful evaluation and DENSO's company-wide positive experience made TargetLink® a natural choice code generator for the development process. This decision was further strengthened by the high acceptance for TargetLink among OEMs, who have already used this

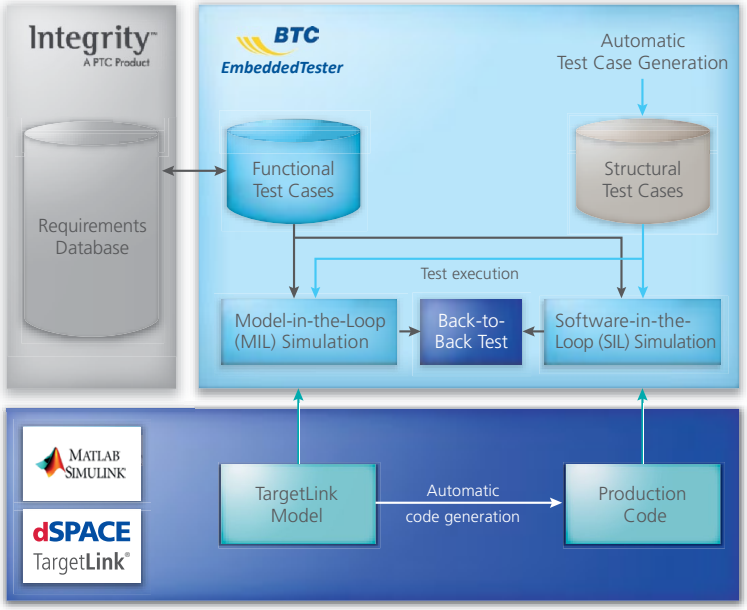
> "The high degree of automation and the efficient integration of tools such as TargetLink and BTC EmbeddedTester relieve our developers of unproductive work steps and let them focus on innovative development itself."
>
> *Samuel Gravez, DENSO*

automatic code generator in numerous automotive projects. One key feature of TargetLink is its strong AUTOSAR support. The dynamic architecture design is modeled in UML, the static design is described according to the AUTOSAR standard. The ARXML (AUTOSAR XML) file that is imported from this is used to generate the initial TargetLink frame model. Users do not have to deal with AUTOSAR details but can focus on their core tasks. The update functionality also makes it easier to adapt the model to changes in interface definitions which result from a newly imported ARXML file. Users can manage the individual AUTOSAR elements in the TargetLink Data Dictionary (DD). Multiple workspaces with diff&merge functionality and structuring, naming and copy functions for Data Dictionary elements facilitate the integration of TargetLink into the tool chain.

Qualified Configuration via Modeling Guidelines

For function modeling in TargetLink, a custom block library is used. This library contains blocks that are usable in TargetLink, with qualified and recommended configurations, and further, more complex functions. MES Model Examiner is used to monitor the compliance with established modeling guidelines such as the MathWorks Automotive Advisory Board (MAAB) guidelines or the TargetLink Modeling Guidelines. MES M-XRAY measures the complexity of the TargetLink model, lets the users know whether their model is sufficiently partitioned and provides additional information on the dimensions of the implemented functions.

If these checks are successful, C code is automatically generated from the functions that were modeled in TargetLink. The production code and the handcoded parts then undergo software unit tests.



Models, C code and test cases for software component testing.

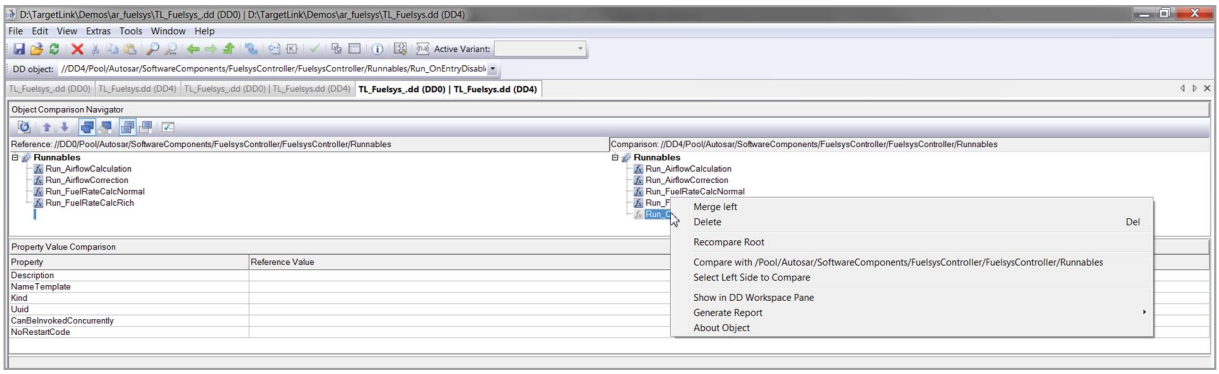
Verifying the Model and Code with BTC EmbeddedTester

To comply with ISO 26262, DENSO chose a combination of requirements-based testing and back-to-back testing for the unit test of the functions developed in Simulink/TargetLink. First, the model as an executable specification is tested against the textual requirements. In a second step, a fully automatic back-to-back test is performed to show whether the model was translated into C code completely and accurately. When it came to choosing the test tools, the most important criteria were their integration with TargetLink, the high degree of

>>

Samuel Gravez
Samuel Gravez is Manager for Software Pre-Development for Powertrain ECUs at DENSO in Wegberg, Germany.

Martin Prisching
Martin Prisching is Senior Engineer for Model-based Software Development at DENSO in Wegberg, Germany.



Comparing AUTOSAR elements of different Data Dictionary workspaces: old vs. new.

automation and the quality of the automatically generated test cases for back-to-back testing.

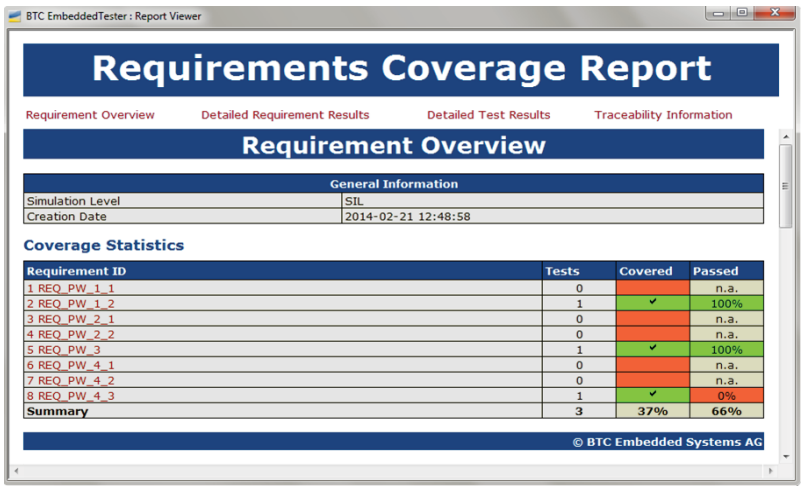
Requirements-Based Testing

Requirements-based testing is used to show whether the model completely and correctly fulfills the textual requirements. To meet the goals of efficiency and agility, the tools have to be deeply integrated. DENSO uses PTC Integrity for requirements management, TargetLink for modeling and code generation, and BTC EmbeddedTester for testing and verification. First, a project is created in BTC EmbeddedTester that automatically extracts all relevant information about the model and code. This includes interface variables, calibration parameters, data types and value ranges. Then, the developers import the requirements from PTC Integrity to ensure traceability between the requirements and test implementation.

> "TargetLink is very suitable for ISO 26262- and AUTOSAR-compliant development and offers powerful analysis tools."
>
> *Martin Prisching, DENSO*

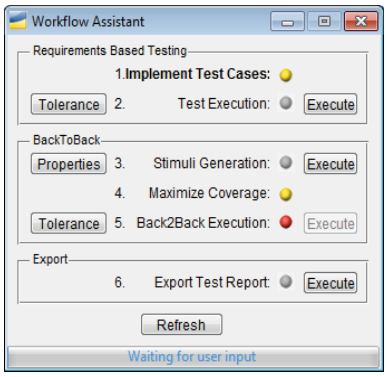
For each requirement, the developers then create one or more test cases with the integrated test vector editor from BTC. These test cases describe both the system input and the expected output behavior. When all test cases have been automatically executed, BTC EmbeddedTester generates a test report and a requirements coverage report. The test results are then written back to PTC Integrity to give the user an immediate overview of the current quality of all models directly in the requirements management. If a requirement changes, the linked test implementations are automatically highlighted in BTC EmbeddedTester so they can be checked and adapted if needed. It is also possible to automatically generate a debug environment for the model or code for failed tests to analyze incorrect implementations.

A report displays the requirements coverage.



Back-to-Back Testing

The subsequent back-to-back testing helps to verify that the model was translated correctly and com-



Dialog of the workflow assistant that guides the user through the verification process.

pletely during code generation. Since the models work with floating-point data types, scaling the fixed-point representation in the production code in particular can cause differences between model and code.

Since the coverage goals defined by ISO 26262, such as modified condition/decision coverage (MC/DC), can usually not be achieved with functional tests, further test cases are generated automatically in BTC EmbeddedTester. This ensures the complete coverage of the model and the code.

In addition, robustness criteria such as downcasts, divisions by 0 or range violations are checked.

If not all coverage goals are met, the tool automatically provides a mathematical proof. Here, BTC EmbeddedTester particularly excels with its analytical approach using model checking technology and with the quality of the generated test cases. In the next step, all test cases are executed at model and code level and the simulation results are compared automatically. If the model and code for a particular test behave differently and are outside of the customer-defined tolerance, a debug environment is extracted to resolve the problem. This can be done by adapting the scaling, for example.

Process Integration and Automation

To make the test process even more intuitive and efficient, DENSO has implemented a workflow assistant together with BTC. This workflow assistant is based on the open plug-in concept of BTC EmbeddedTester and, as a wizard, guides the user through the individual steps of the verification process. The assistant lists the process steps one by one. A kind of traffic light system indicates the completed and uncompleted steps. Users can then execute the uncompleted steps, such as test case generation or test execution, directly from the assistant. The assistant also provides links to the reports for test results and requirements coverage.

Summary and Outlook

This tool chain gave DENSO the means to define an efficient and agile process for model-based development and to put it to practical use successfully. Thanks to a high degree of automation and the seamless tool integration, changed requirements can be integrated into the running project. A great asset of this tool chain is that the tools meet the requirements of the automotive industry and fully support standards such as ISO 26262 and AUTOSAR.

After a successful test phase in predevelopment, DENSO will now introduce this development process to production development at the AEC. The freed capacity is reinvested in activities with a higher value generation and innovative power. This lets DENSO manage the increasing complexity and acceleration of the development process. Early and continuous integration, frequent customer feedback, and fast reaction times also yield greater customer satisfaction. ■

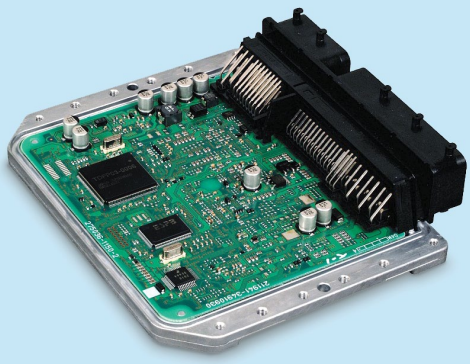
Samuel Gravez,
Martin Prisching,
DENSO AUTOMOTIVE Deutschland

Process at DENSO

Challenge:
To guarantee the high quality of the software to be developed, despite the adverse development conditions: time and cost pressure, changing requirements during the development, and complying with standards such as AUTOSAR, ISO 26262, MISRA, or SPICE.

Solution:
DENSO's development process is based on a seamless, mostly automated tool chain. Everything from test case generation to test reports is automated. This makes it possible to propagate to production code very early on. The tools, such as TargetLink, are qualified for the important standards.

Advantage:
Fast reaction times generate greater customer satisfaction.



Automotive ECU development benefits from agile processes at DENSO.