

Sample use case

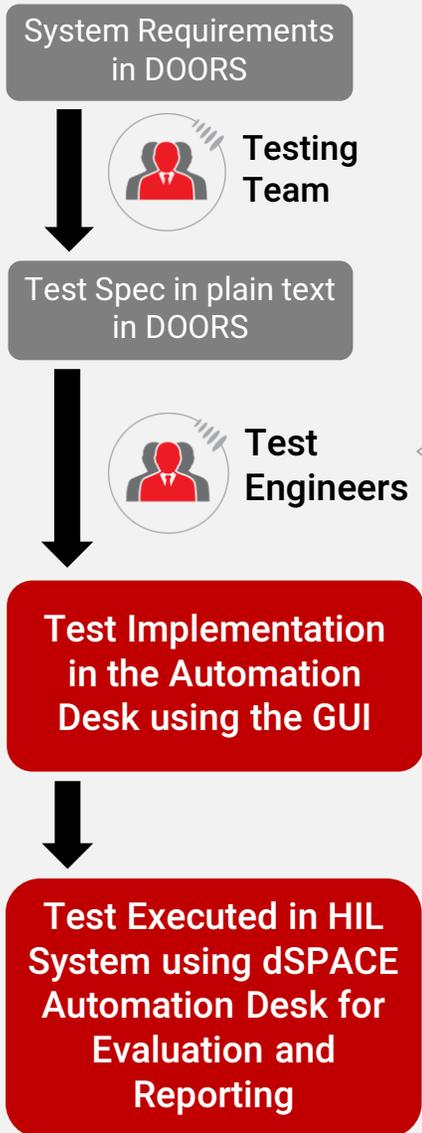
Advantage of ABot in automotive testing



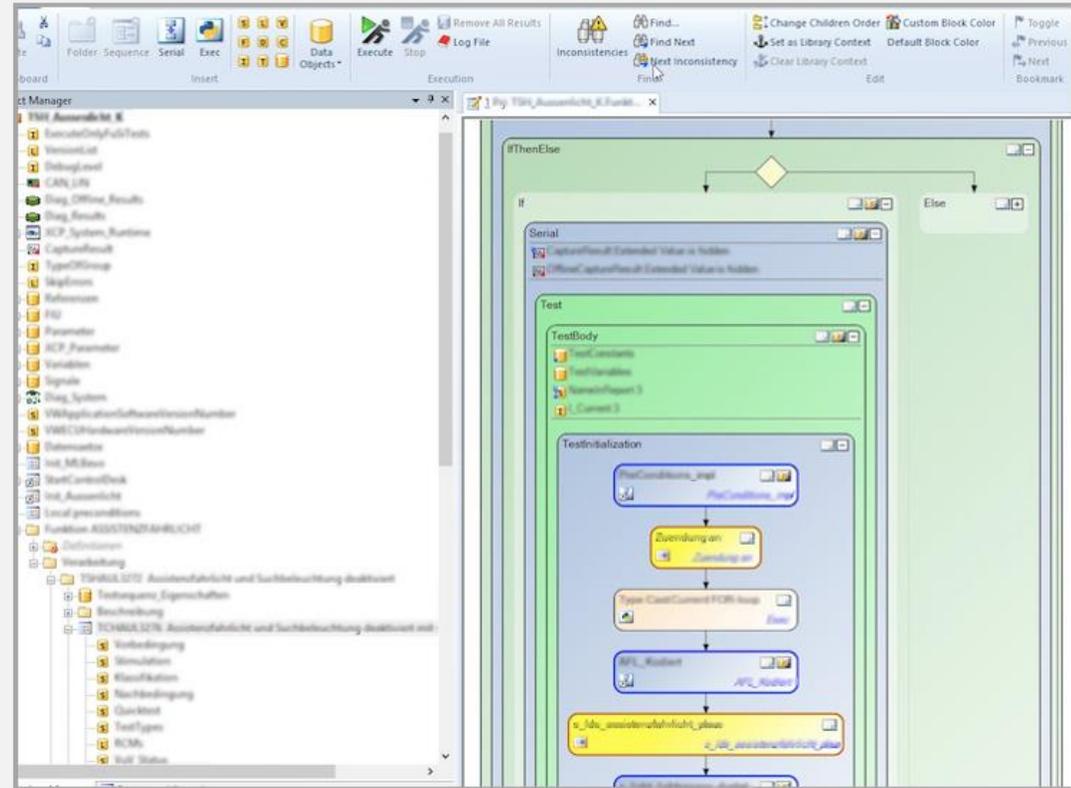
Welcome to the BCM Test Automation Demo

We are going to walk you through the current workflow of the BCM Testing, critical bottlenecks of the current testing process and a solution which can make the testing of BCM and other Bosch ECUs more optimised and productive. We will also talk about the current status and the next steps.

The Current Workflow of Testing



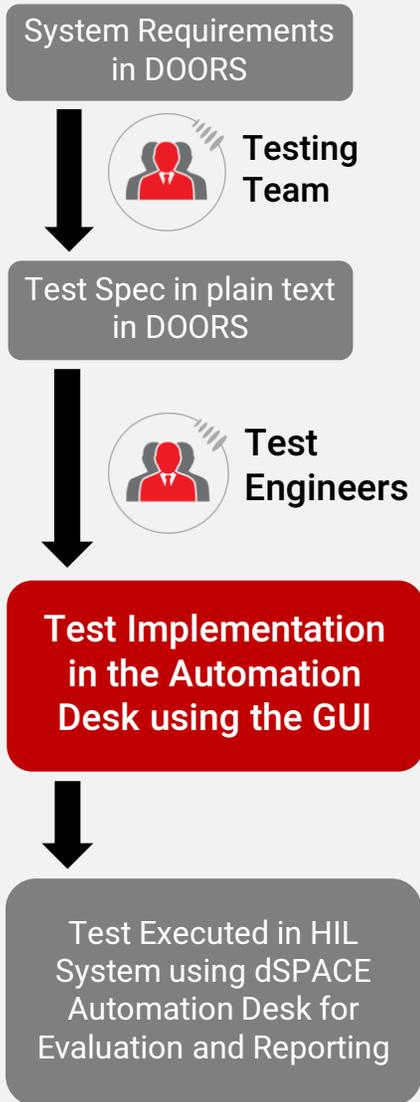
Next, the test engineers implement the tests as per the specifications. For test implementations the Bosch Test Engineers use a tool called Automation Desk which is from dSPACE. This is a complex process since there are hundreds of libraries and widgets that need to be carefully utilised and managed, making the Test Implementation the most elaborate phase in the Testing workflow.



Once the tests are implemented, the test engineers execute the newly implemented test cases in the HIL System.

After the execution is complete, the results are evaluated and subsequent reporting is done through the Automation Desk.

The Current Workflow of Testing



As you can see, the Automation Desk tool plays a vital role into the entire testing workflow. However, due to the nature of the tool, it makes the test authoring process quite complex.

- Test Implementation using GUI driven Automation Desk takes a huge amount of time
- Steep learning curve for the dSPACE Automation Desk tool
- Reusing test case implementations across projects is a challenge
- Majority of time in the Test Authoring Workflow is consumed by this step

The Solution

Let us now look into a solution that can optimise the workflow. We are going to propose only 2 changes in the current workflow.

- Firstly, the detailed test specification will be created using the **Behaviour Driven Development**, or **BDD** principles as opposed to plain text.
- And secondly, an **automation framework** will execute these BDD based test specifications and automatically generate the Automation Desk project.

What is Behaviour Driven Development (BDD) ?

- BDD is the new industry standard of writing Product specifications today. It is neither a document nor a source code, rather a new way of authoring the requirements and the acceptance criteria of a functionality.
- BDD has internationalisation support for which the specifications can be written using both English and German languages.

Scenario: All doors lock automatically when the car drives at 20 km/h

Given the battery has charge
And the doors are closed
And the gears are engaged

Context

When I start the engine
And I drive the car at 20 km/h

Events

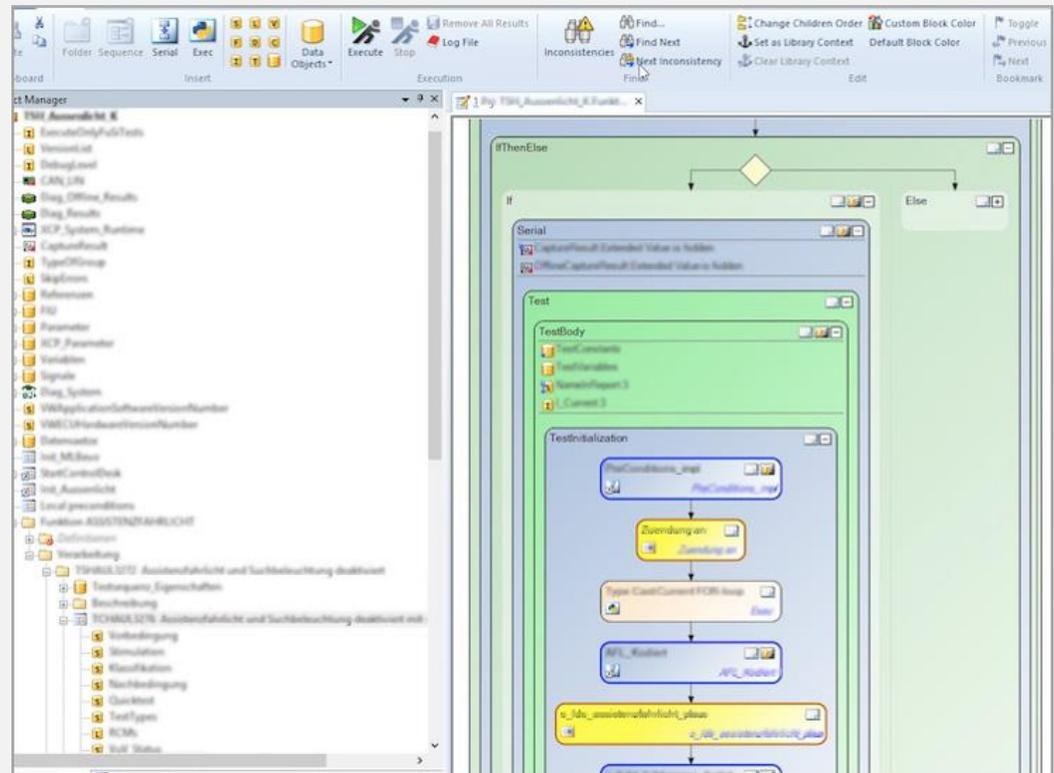
Then the doors should be locked

Outcome

Sample BDD test scenario, which checks that all doors of the car gets locked automatically when the car is driven at 20 kilometre per hour.

What is Behaviour Driven Development (BDD) ?

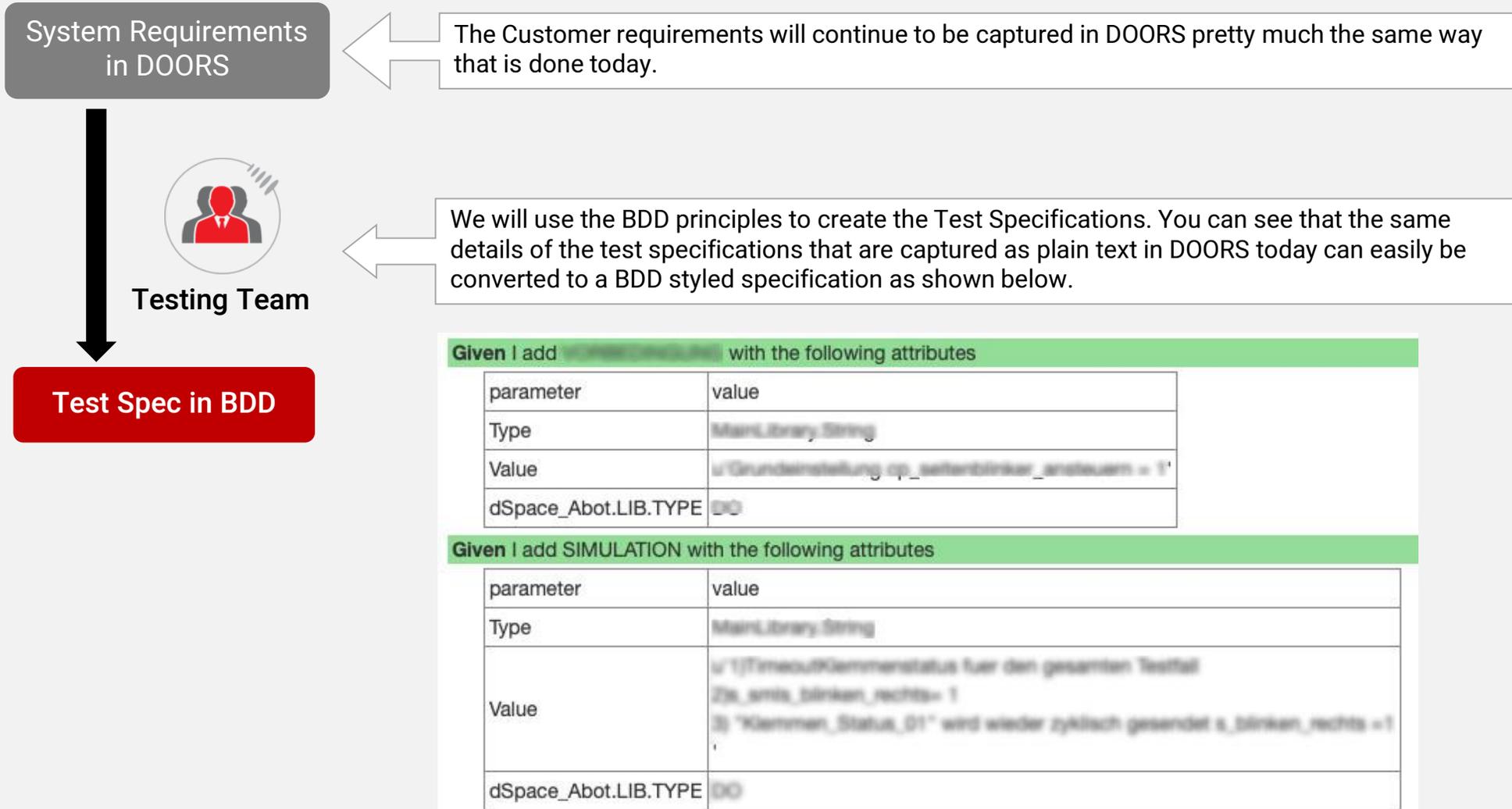
- BDD is the new industry standard of writing Product specifications today. It is neither a document nor a source code, rather a new way of authoring the requirements and the acceptance criteria of a functionality.
- BDD has internationalisation support for which the specifications can be written using both English and German languages.
- BDD specifications can be **executed** through an automation framework
- On execution, the test specification **automatically generates** the Test Implementation in the dSPACE Automation Desk



Automation Desk Project

The Proposed Workflow of Testing

Let us now talk about the proposed workflow.



The Proposed Workflow of Testing

System Requirements in DOORS



Testing Team

Test Spec in BDD

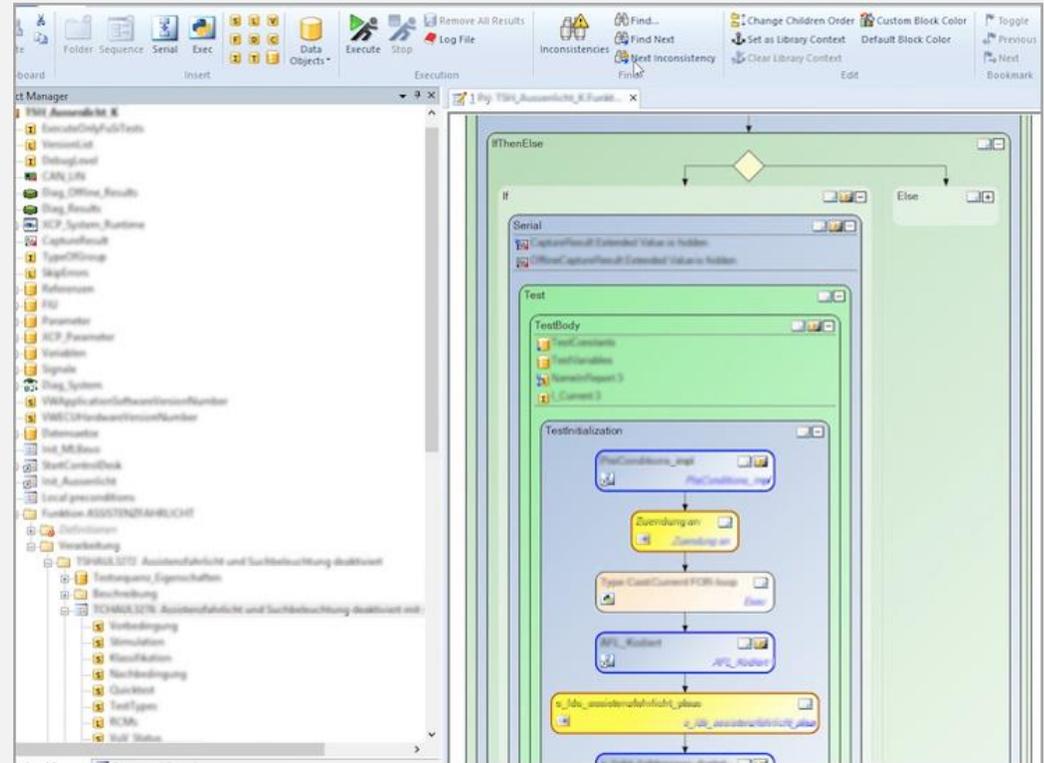


Automation Framework

Auto Generate Test Implementation for the Automation Desk

Test Executed in HIL System using dSPACE Automation Desk for Evaluation and Reporting

These BDD based test specification will then be picked up by the Automation Framework and executed.



The Automation framework on executing the specifications will create the test implementations in the dSPACE Automation Desk. This way the entire bottleneck of creating the test in the GUI based Automation Desk will be avoided without disrupting the following phases in the testing workflow.

The Auto generated test implementations can now be opened using the Automation Desk and hence, the test execution, evaluation and reporting will continue in the same way as it does today.

Advantages of Transitioning to BDD from the GUI based approach

- The BDD approach saves substantial time by automatically generating the GUI based Automation Desk projects.
- It is much easier to author new tests as well as maintain them. Given its simplicity, it will be easy to onboard any new test engineer.
- These test cases are highly reusable. The test engineers can actually copy from previous implementation, create central repository of common implementations, import parts from other projects and so on.
- Better configuration management, source control and reusability will result into increased efficiency of the tests and optimise the precious lab usages.
- Finally, by generating the Automation Desk implementations automatically, the BDD Automation Framework gets the workflow back into the existing track which makes the integration with all current infrastructure absolutely seamless.