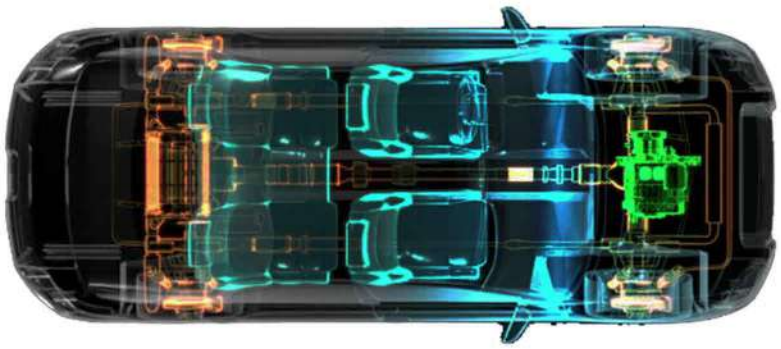
embitel
delivered with passion

AUTOMOTIVE SOFTWARE TESTING AS PER ISO 26262 STANDARD

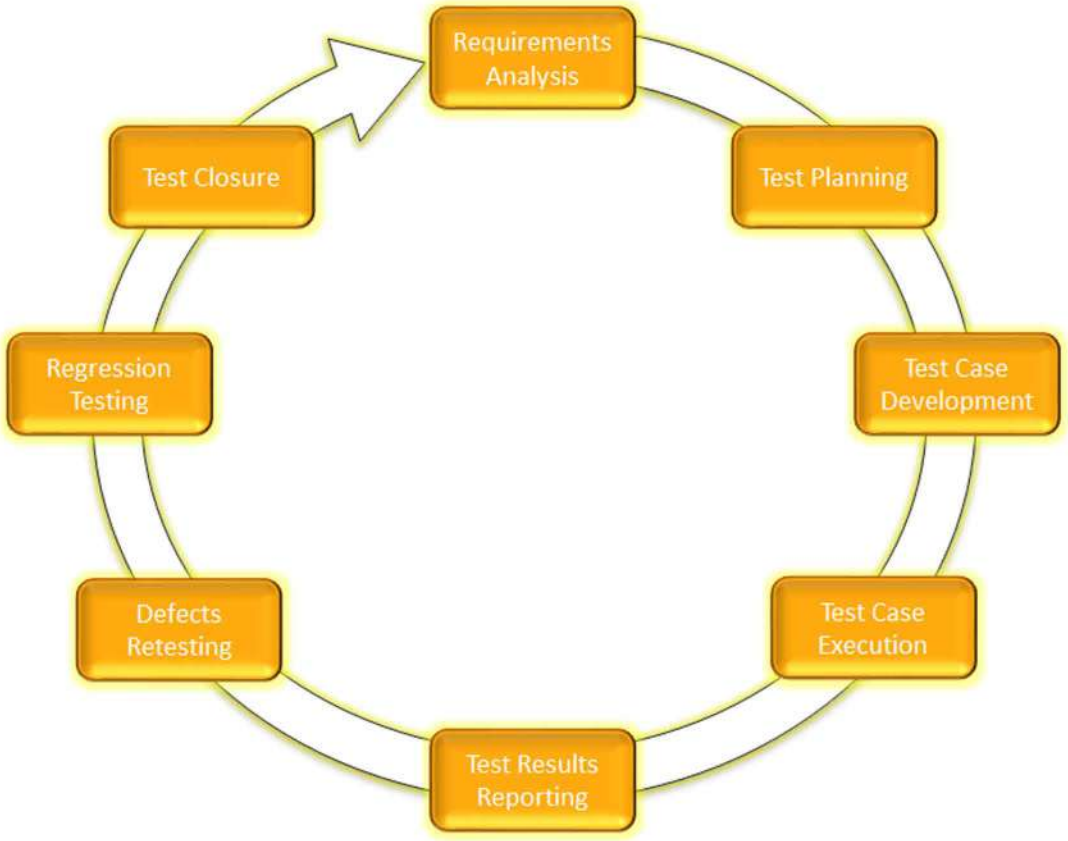
Functional Safety | ISO 26262 | ASIL | CANTATA | RTRT | TESSY



A Handbook on | ISO 26262 compliant Automotive Software Testing

PROCESSES INVOLVED IN ISO 26262 COMPLIANT TESTING OF AUTOMOTIVE APPLICATIONS

TESTING CYCLE: UNIT TESTING | INTEGRATION TESTING | SYSTEM/FUNCTIONAL TESTING.

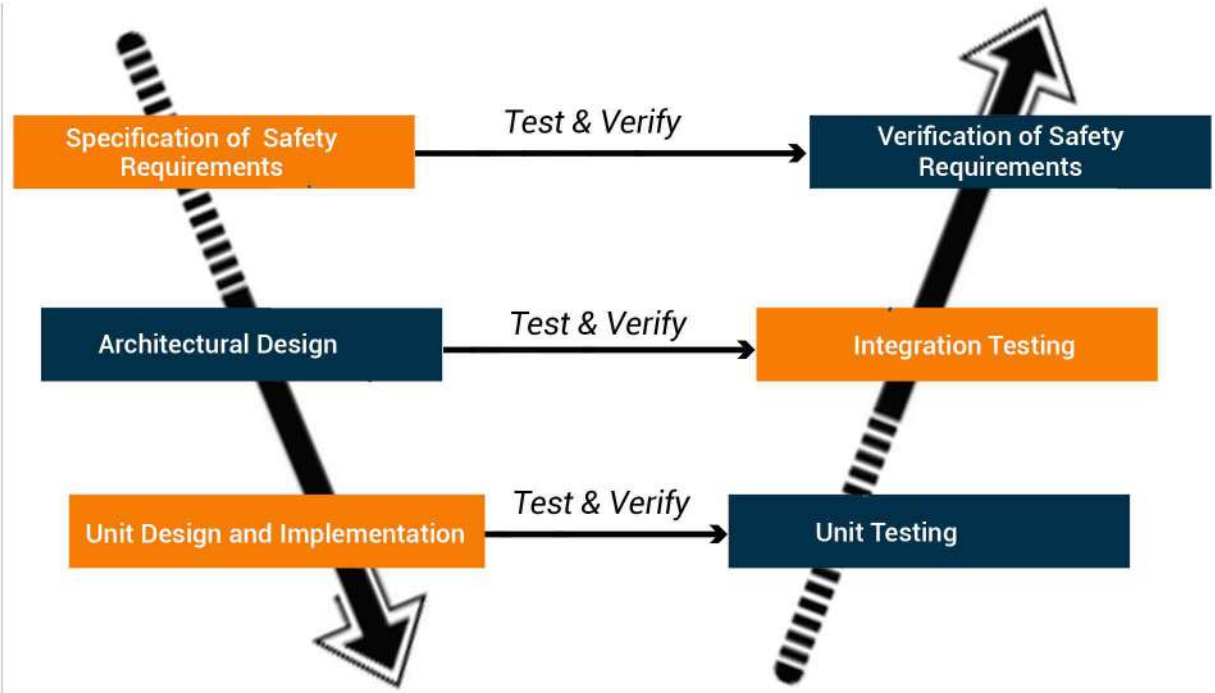


FOLLOWING THE ISO 26262 VERIFICATION AND VALIDATION GUIDELINES

ISO 26262 standard lays down a set of guidelines for ASIL compliant software development. Requirements pertaining to testing and verification are part of these guidelines. Part 6 of the standard document enlists the test methods to be implemented during software development. Part 8 of the document also has guidelines for verification processes.

These test methods must be followed during the unit, integration and validation testing of the automotive software.

Figure 3: The V-model used in ISO 26262 software development



SOFTWARE UNIT TESTING GUIDELINES AS PER ISO 26262 STANDARD

ISO 26262 standard clearly specifies the methods and techniques for both static and dynamic verification of the software unit. It begins with static verification of software unit design. Third-party software tools are required to perform majority of the verification techniques mentioned in the table.

Static analysis of the code is followed by the dynamic testing of the software unit. It is performed by executing the software in a simulated environment. The following methods of software unit testing are specified in part-6 of the ISO 26262 standard.

Methods for software unit verification

	Methods	ASIL			
		A	B	C	D
1a	Walk-through[a]	++	+	o	o
1b	Pair-programming[a]	+	+	+	+
1c	Inspection[a]	+	++	++	++
1d	Semi-formal verification	+	+	++	++
1e	Formal verification	o	o	+	+
1f	Control flow analysis[b, c]	+	+	++	++
1g	Data flow analysis[b, c]	+	+	++	++
1h	Static code analysis[d]	++	++	++	++
1i	Static analyses based on abstract interpretation[e]	+	+	+	+
1j	Requirements-based test[f]	++	++	++	++
1k	Interface test[g]	++	++	++	++

1l	Fault injection test[h]	+	+	+	++
1m	Resource usage evaluation[i]	+	+	+	++
1n	Back-to-back comparison test between model and code, if applicable[j]	+	+	++	++

INTEGRATION TESTING GUIDELINES AS PER ISO 26262 STANDARD

Integration testing is performed at the software integration stage. At this stage, different software units are brought together to work as one system. Integration testing verifies that the resulting system performs as intended.

Software integration testing is planned, specified and executed in accordance with Clause 9 of part 8 of the ISO 26262 standard document.

THE FOLLOWING METHODS ARE SPECIFIED FOR THE INTEGRATION TESTING:

Table 10 — Methods for verification of software integration

Methods		ASIL			
		A	B	C	D
1a	Requirements-based test[a]	++	++	++	++
1b	Interface test	++	++	++	++
1c	Fault injection test[b]	+	+	++	++
1d	Resource usage evaluation[c, d]	++	++	++	++
1e	Back-to-back comparison test between model and code, if applicable[e]	+	+	++	++
1f	Verification of the control flow and data flow	+	+	++	++
1g	Static code analysis[f]	++	++	++	++
1h	Static analyses based on abstract interpretation[g]	+	+	+	+

Verification of Software Safety Requirement

The objective of this phase is to verify that the software fulfills its safety requirements in the target environment.

As per the ISO 26262 standard, the test shall be performed under the following test conditions.

Table 13 — Test environments for conducting the software testing

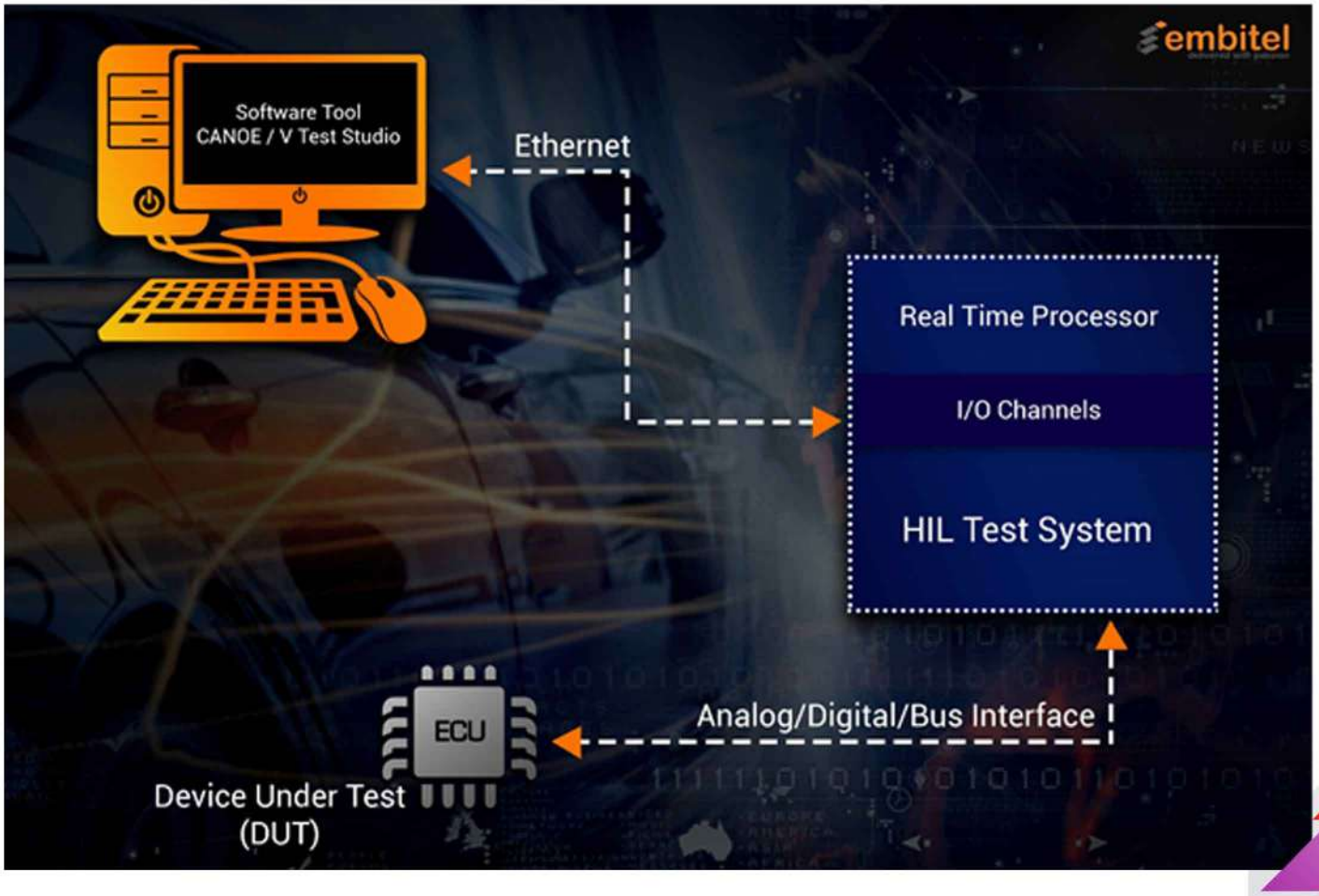
Methods		ASIL			
		A	B	C	D
1a	Hardware-in-the-loop	++	++	++	++
1b	Electronic control unit network environments[a]	++	++	++	++
1c	Vehicles	+	+	++	++

[a] Examples include test benches partially or fully integrating the electrical systems of a vehicle, "lab-cars" or "mule" vehicles, and "rest of the bus" simulations.

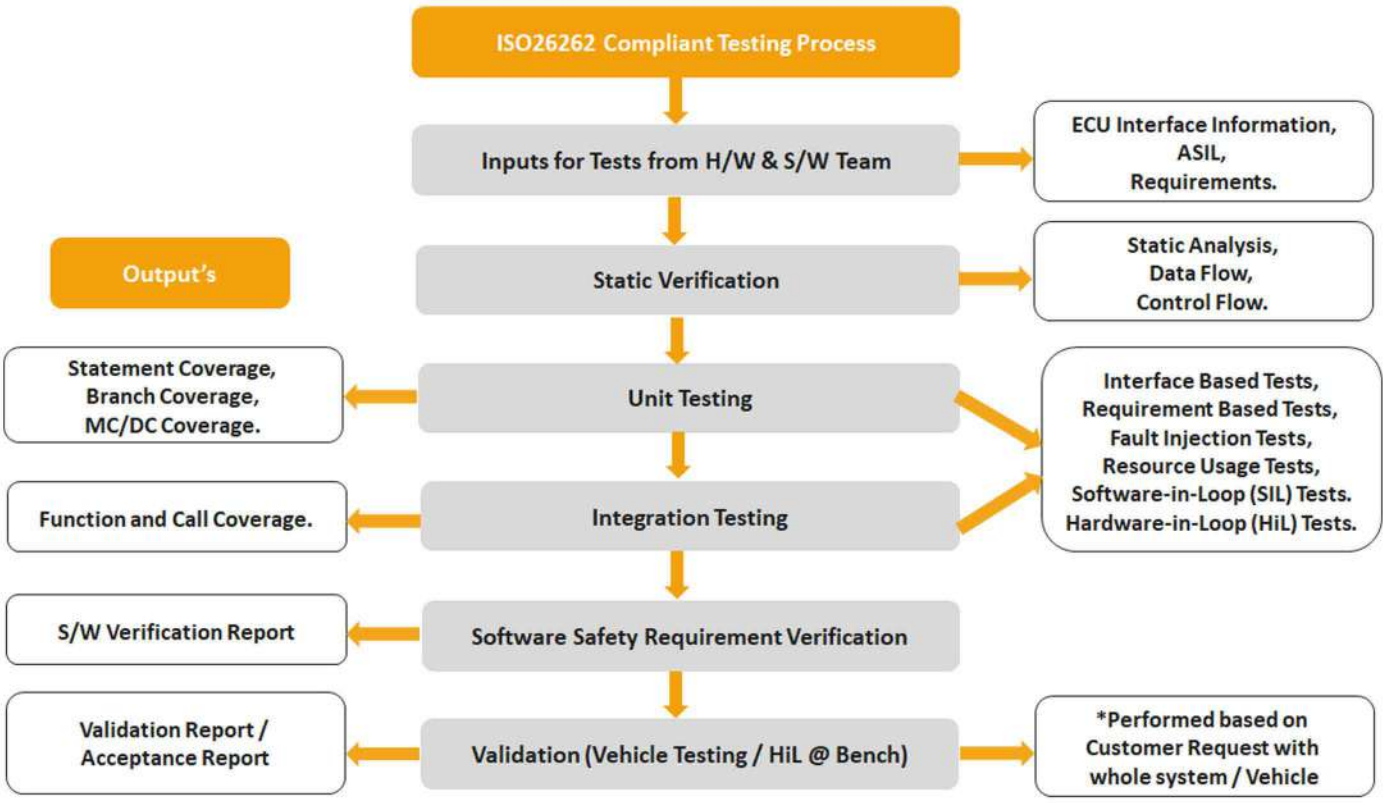
Hardware-in-Loop Testing (HIL @Bench)

HIL Testing needs a special mention as it marks a very crucial form of functional testing. During HIL Testing, the ECU is connected to a Test System that mimics an assembled product (an engine, a body control module or even the entire vehicle system). It interacts with the input and the output of the ECU under test, as if it were an actual vehicle.

Using a HIL Testing software tool, different test scenarios can be created, and the output validated. Depending on the software requirements, the test coverage can be expanded without worrying about the cost and physical risk to the system.



FLOWCHART SHOWING THE VERIFICATION AND VALIDATION PROCESS FOLLOWED AT EMBITEL



WHY SHOULD EMBITEL BE YOUR PREFERRED PARTNER FOR ISO 26262 COMPLIANT PROJECTS?

Expertise in Unit, Integration and Functional Testing Tools Qualified by ISO 26262

- **CANTATA :** An ISO 26262 qualified tool, ideal to automate unit and integration testing and ISO 26262 verification of the code.

- **RTRT :** An automation tool for unit testing, developed by IBM. It is a widely used testing tool and an ISO 26262 qualified tool.

- **PolySpace :** A widely used tool for static verification of the code. It checks for security vulnerability, coding guidelines and other classes of bugs.

- **MX-Suite :** This is one of the most comprehensive automated testing tools for software verification of embedded systems.

- **Tessy Tool:** Tessy automates the whole unit test cycle for an Embedded software development project including regression testing.

Testing Team Comprises ISO 26262 Experts

- The core testing team comprises Functional Safety Consultants, Project Managers and Testing Engineers.
- Every member of the testing team has hands-on experience in ISO 26262 projects.
- Our engineers may also travel to the customer's location to perform software testing based on an onsite/offshore business model.

CUSTOMER SUCCESS STORIES THAT SPEAK FOR THEMSELVES

1. Functional Testing of a Heated/Cooled Seat ECU

Customer

An automotive Tier-1 supplier from United States

Project Scope

- Functional testing of Heated/Cooled Seat ECU for Heating and Venting as per functional specification with Memory Seat Features
- Diagnostics and Fault Handling

Our Solution

We performed the HIL Testing of the following modules:

i. Memory Seat Features like easy entry/exit of driver, favorite memory position, soft stop of motors, etc.
ii. Heating and venting of the seats on various modes
iii. DTC handling for each motor, short to battery and short to ground
iv. Parameter configuration with DID Read/Write

Tools and Technology

- Vector Canoe with VT System
- DOORS
- Quality Manager
- Jazz Server
- Vector Canalyzer
- PIC, Renesas tool chains

2. Mutation Testing of an Automotive EPS for ISO 26262 (ASIL D) Compliance

Customer

An Automotive Engineering Company, specializing in Powertrain and Body Electronics Development

Project Scope

- Mutation testing of an Electronic Power Steering ECU by creating Mutant code and testing it with a specific test-case that makes the mutant code fail
- Test-case to be modified till it kills the mutant code

Our Solution

Our testing team worked with the QA team to identify test cases for 100% coverage.

Our automotive ECU software testers analyzed mutant test results to identify 'acceptable' and 'not acceptable' mutants (mutant codes that were not killed).

Test scripts were modified to kill all "not acceptable" mutants.

CONNECT WITH US

INDIA : +91 80 41694200 **USA :** +1-248-385-2017

GERMANY: +49 711-60 17 47-789 **UK :** +49 170 1688028

EMAIL : sales@embitel.com