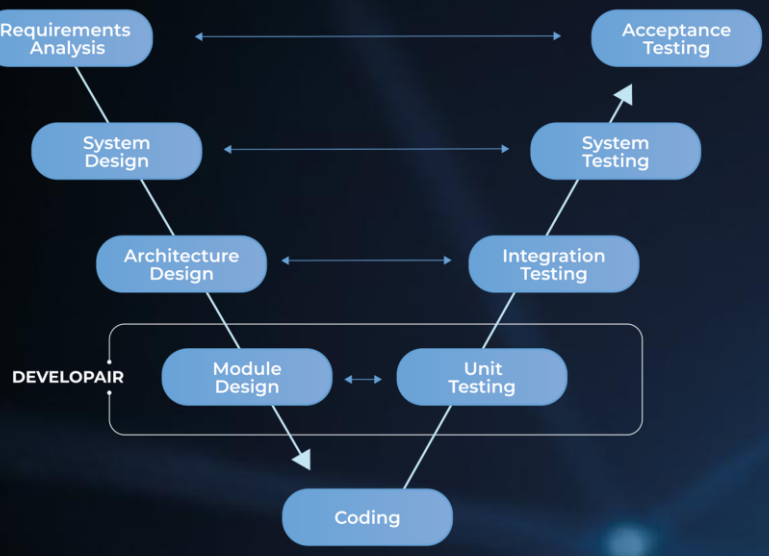
AI BASED TOOLS FOR VERIFICATION & VALIDATION

The fastest way to build reliable embedded software

DEVELOPAIR

TÜV SÜD

Challenges embedded SW development



Cost and development time pressure: high dedication in the verification and validation phases.

Regulatory compliance: quality and reliability issues.

Opportunity cost: high dedication to low value-added tasks like unit testing.

Attracting talent: difficulty in finding embedded SW developers.

Areas of application

For companies aiming at improving their software development processes or at fulfilling functional safety regulations in an efficient way. We can support:

Industry: IEC 61508

Railway: EN50128 and EN50657

Road vehicles: ISO 26262

Medical devices: IEC 62304

Aviation: DO-178C

E.g. companies that develop SW that controls the brakes of a train, industrial machinery, power electronics, the ABS of a car of even a pacemaker.

The certified tool fulfils the requirements for support tools classified T2 according to IEC 61508-3 and EN 50128. The tool is qualified to be used in safety-related software development according to IEC 61508, EN 50128 and ISO 26262. It is suitably validated for use in safety-related development according to IEC 62304.

Our solution

A software tool that automates Verification & Validation activities of embedded software.

Based on advanced mathematical models and AI techniques.

Features

Smart Editor: GenAI-powered to define unambiguous and standardized requirements.

Automatic Verification of requirements to detect errors, inconsistencies, conflicts, etc.

Automatic Test Generation to generate all the functional test cases and scripts needed for standard compliance.

Seamless Integration with requirements and tests management tools.

Benefits

30-50% reduction in SW development costs.

Reduction of SW development times **by up to 50%.**

Reduction of nonconformity **costs and regulatory** compliance issues.

Reduction of **opportunity cost:** 3 out of 10 engineers reassigned to high value-added tasks.

SMART EDITOR AND VERIFICATION

An easy-to-use editor to ensure the quality of software requirements. With support of natural language patterns and graphical specification models, like finite state machines (FSM), providing:

- **Reduced ambiguity the requirements.**
- **Greater standardization.**

Other problems associated with non-restricted natural language requirements are:

- **Vagueness**
- **Complexity**
- **Untestability**
- **Omission**
- **Duplication**
- **Wordiness**
- **Inappropriate implementation**
- **Conflicts**

The editor includes a smart assistant that automatically generates requirements in Rely from a description in natural language or requirements in other formats.

AUTOMATIC TEST GENERATION

Business pain:

Manual generation of software unit tests is a repetitive, costly, time-consuming and low value-added process.

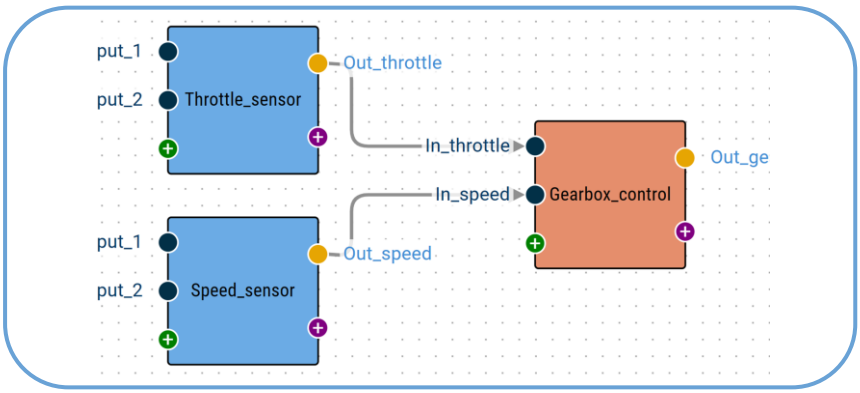
Solution:

With the Automatic Test Generation module, it is possible to auto-generate the tests from the requirements.

Benefits:

This improves test quality by avoiding human errors, reduces time and cost, and allows engineers to focus on high value-added tasks. The lower cost per test also allows more tests to be performed, increasing safety and reliability.

Architecture editor

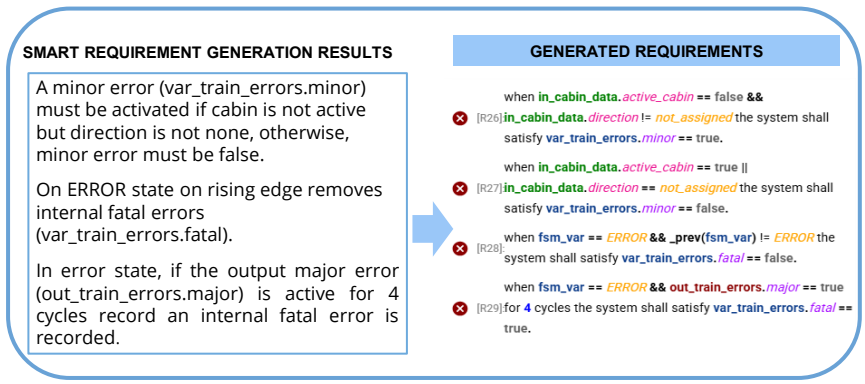


Architecture editor: it supports multilevel approaches and allows you to graphically define system components and interfaces.

Smart requirement editor

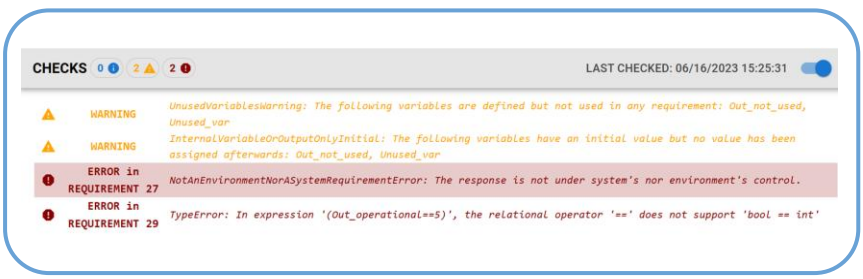
Rely: Developair provides a specification language that allows you to **easily define requirements** in **restricted natural language** and **favouring standardization**.

Smart assistant based on Generative AI



Smart assistant for defining requirements: the generative AI-based assistant can automatically generate requirements in Rely from requirements in other formats or a description in natural language.
Besides creating requirements, the assistant can also detect if new elements such a data types, variables, inputs or outputs need to be created.

Automatic verification of requirements



Smart error detection: it provides double-stage feedback by detecting in real time simple errors, such as undefined variables or incompatible data types, and complex errors such as conflicts between requirements or potentially undesired behaviours, when launching the verification.

Testing workflow with Developair

Thanks to our advanced algorithms test are automatically generated with the push of a button.

Seamless integration with requirements management tools and architecture definition tools.

Developair includes connectors to tools like **Polarion**, **IBM DOORS** or **DNG** connectors for other tools, including proprietary tools, are available on request.

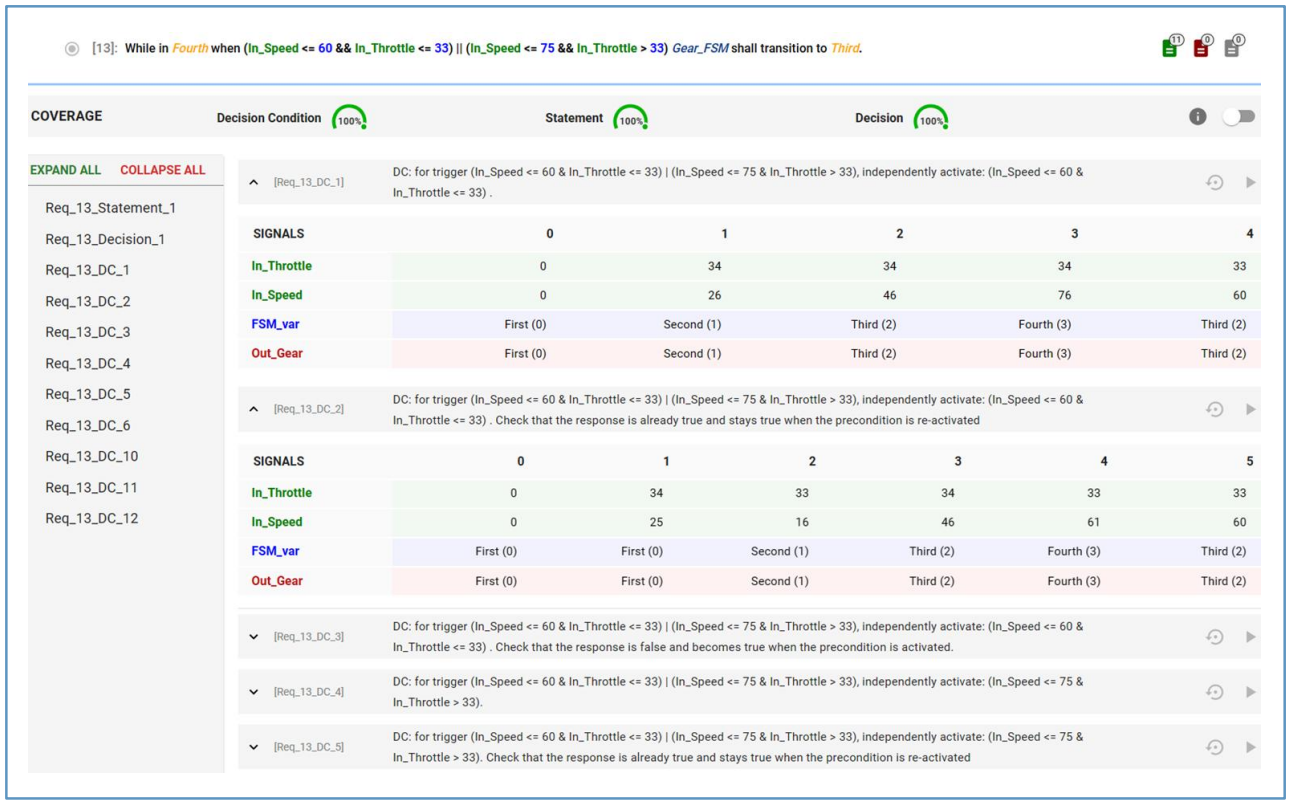
Example of the automatic test generation

In this example we can see the values of the tests automatically generated by the platform for the following requirement:

> "**While in** Fourth, **when** In_Speed<=60 && In_Throttle>33
> **for** 3 **cycles** Gear_FSM **shall transition to** Third"

This **requirement describes** when a gearbox should shift from fourth to third gear depending on the speed and pressure on the throttle.

These tests have been generated for **Statement coverage** (SC), **Decision coverage** (DG) and **Decision/Condition Coverage** (DCC). **Condition Coverage** (CC), **Modified Condition/Decision Coverage** (MCDC) and **Boundary-value analysis** are also available. In this view you can see each testcase per each requirement.



Seamless Integration with test execution tools.

With our connectors it is possible to execute the tests generated in our platform in testing tools such as **Simulink Test, CxxTest, or Ansys Scade** with just a few clicks. Connectors for other testing tools, including in-house tools, are available on request.

DEVELOPAIR

DEVELOPAIR TECHNOLOGIES

Paseo de Miramón, 170

20014 • Donostia-San Sebastián

Gipuzkoa (Spain)

+34 637 22 88 12 info@developair.es www.developair.tech