Toolchain Development
CI & CD

SOKRATEL
Automotive | Automation
Green Energy

MathWorks®
Partner

Automating processes in your company comes with a range of advantages:
- Obtaining a **high and reproducible quality** for each product lifecycle
- **Minimizing administration tasks** of developers
- Improving **productivity, scalability, and flexibility** of projects

Sokratel's team of systems engineering experts offers **development of effective concepts, quick production of functional prototypes** as well as support with further developments.

Toolchain Development

Process automation requires a tool that executes and controls all steps in your product lifecycle. These programs use **continuous integration (CI)** and **continuous delivery/ continuous deployment (CD)** processes. CI focuses on testing and building artifacts after each new commit to ensure application stability. CD goes even further and includes the steps of deployment to production and release to the customer.

Due to its flexibility, we use **Jenkins®** to automate processes for on-premises servers. For **cloud-based solutions** we use Microsoft's **Azure DevOps** and its CI/CD functionality to manage software projects remotely.
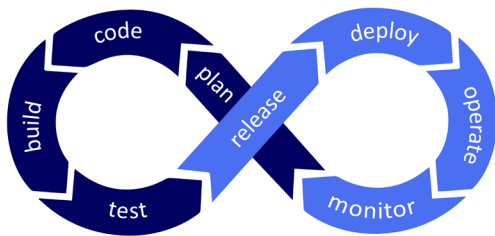
Continuous Integration

The CI process can handle the organisation and automation of test processes, including the report of results to project management software and the generation of artefacts from code. By linking all processes we obtain a fully automated **toolchain**, which **tests, reports and builds**. The automation minimizes bugs in your application and testing costs reduced drastically as the CI solution can run hundreds of tests at once.

Static Code Analysis

A static test does not execute the implemented code itself, it **checks the structure of the code** to improve **program performance** (e.g. can we reach each decision of a query or is there some dead code?). Dynamical tests, which execute the code, can hardly detect such issues.

How our team of experts can support you:
- Configuring a **server running Teamscale application**, which analyses many different high-level programming languages as well as Simulink models
- Configuring **projects and analysis profiles**
- Implementing **self-developed checks**
- **Simulink® Model Advisor toolbox:** checking Simulink models according to industrial standards
- **Experienced advice** on which checks are useful for your application

Dynamical Testing

At every stage of the development cycle, testing is fundamental to delivering an **exceptional, high-quality product**. With automated testing, the **reproducibility of any test type** can be performed **more frequently** and **faster**, allowing the QA team to focus on significant improvements. We provide support in each test environment:

- **Model-in-the-Loop:** support with initializing test frameworks & test execution in Simulink® Test
- **Software-in-the-Loop:** support with implementing tests in GoogleTest, JUnit, C# Unit Tests or Boost
- **Processor-in-the-Loop:** we are specialised on Xilinx MPSoCs & realizing PiL tests with the Embedded Coder by Simulink®
- **Hardware-in-the-Loop:** support with automating your HiL testing using ECU Test, EXAM or ControlDesk (managable with CI Tools due to their REST API capability)
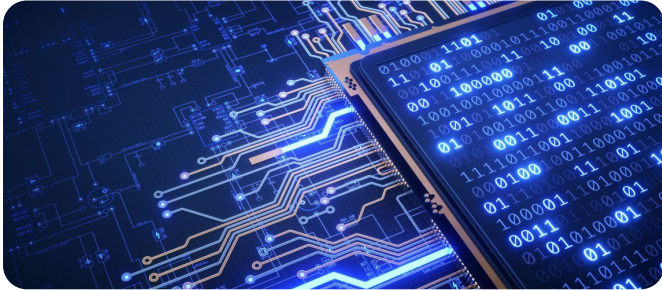
Turning visions into reality

Toolchain Development
CI & CD

Finally, the generated test suite results are forwarded to any other project management or issue-tracking tool like Jira resp. Jira Xray.

Continuous Deployment

After successful testing, the developed code must be compiled to a machine-readable language, no matter if it is a Simulink® model which needs to be compiled using **Simulink® Coder** or a C/C++ code using **CMake**. We can give you advice configuring the process to obtain an executable file running on the target of your choice.

To organize your dependencies and versioning during a build process, we use Build Management Tools like **Maven** or **Gradle** to realize this.

After building the application, it can be deployed to various artefact repositories, e.g. **Nexus repository, Azure artefacts** or **Azure blob storage** or various source code management systems such as Git or SVN. Even deploying directly to some Hardware is possible e.g. **Phoenix Contact** or **Bachmann** PLC's.

Tim Krause
Sokratel GmbH
Head of Division Automation

Thomas Frei
Sokratel GmbH
Managing Director

Contact: contact-automation@sokratel.com

You want to learn more?
Scan the QR-Code to explore our website or contact us!