Developing safety applications in accordance with EN ISO 13849 and EN IEC 62061

TM530

Perfection in Automation

www.br-automation.com

Requirements

Training modules:	TM500 – Basics of Integrated Safety Technology
	TM510 – ASiST SafeDESIGNER
Software:	None
Hardware:	None

Table of contents

1. INTRODUCTION

This training module covers the development of safety applications in accordance with EN ISO 13849 and EN IEC 62061

This course will explore the development process in greater detail, defining explicit responsibilities along the way.

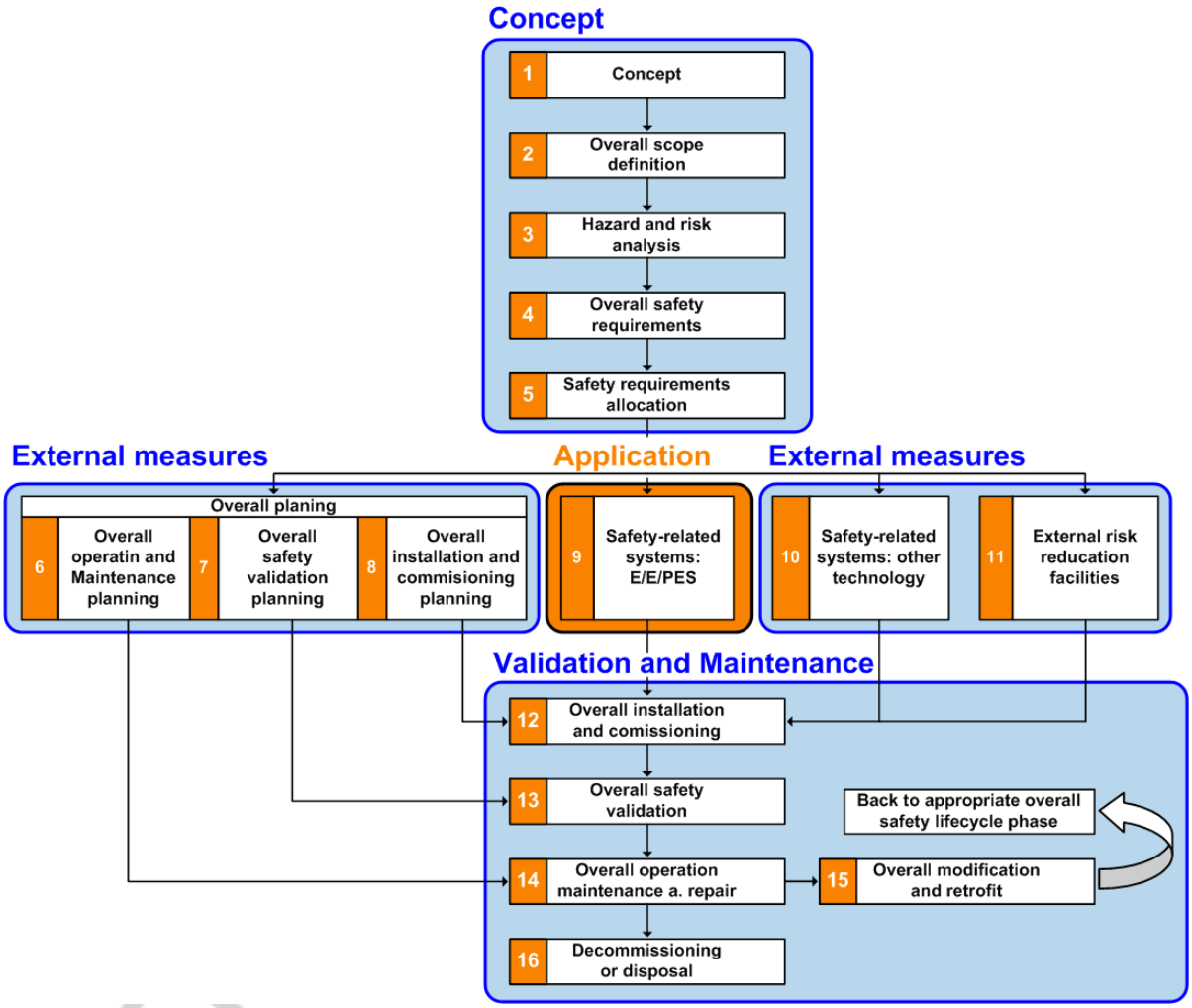


Fig. 1: Safety lifecycle

The safety lifecycle of a "safe machine" involves several phases, however this training module will only cover the area of software development (item 9 in the safety lifecycle).

The separate sections of the safety lifecycle are specified in the safety plan. This training module will not examine the functionality and method of operation of this safety plan.

Note:

More information about the safety plan and functional safety management (FSM) can be found in the standards EN ISO 13849, EN IEC 62061 and EN IEC 61508.

Caution:

This training module and the services provided by B&R in regard to safety-related software development are limited solely to item 9 of the lifecycle model.

1.1 Objective

The goal of this training module is to become familiar with the development process for creating a safety application, in particular item 9 of a lifecycle and the phases contained within.

After completing this training, you will also be familiar with the areas of responsibility and the tasks necessary for creating a safety application.

You will become familiar with how project change management works and learn how to apply these changes to your projects in the best possible manner.

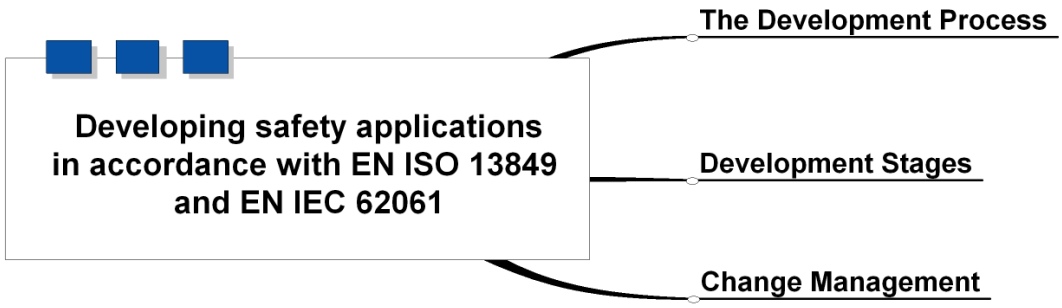


Fig. 2: Overview

2. THE DEVELOPMENT PROCESS

This part of the document provides an overview of the development process for a safety application.

This process involves various people and roles, which will be described in greater detail throughout the document. Each of the persons involved will then be assigned explicit responsibilities.

Interfaces will also develop between the different roles and stages throughout the development.

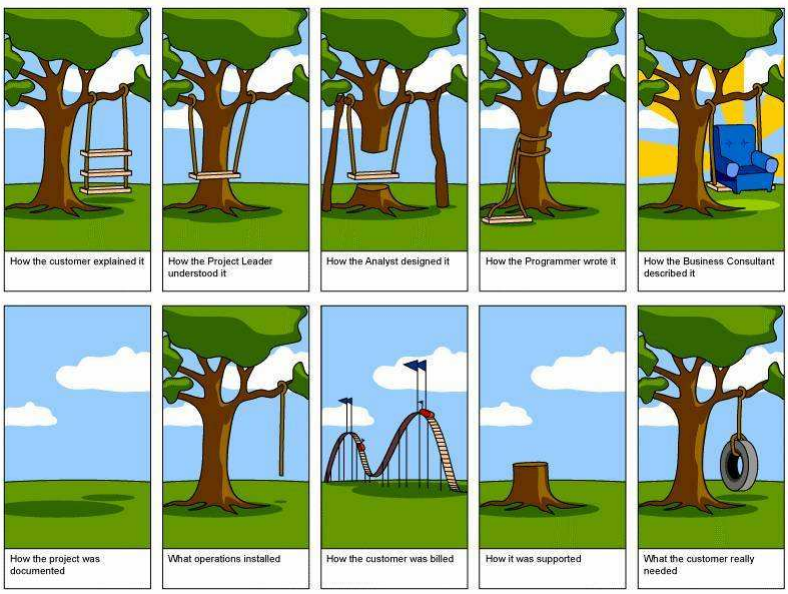


Fig. 3: Example of a negative development process

The image shown above illustrates a negative development process. Adherence to the individual phases contributes significantly to the overall success and helps to avoid negative results.

Caution:

Procedures and decisions must be documented accordingly throughout the entire development process.

2.1 Responsibilities and roles

Projects generally consist of several phases or project sections. These will be assigned to individuals or groups, depending on their size and scope.

The safety lifecycle can be divided into the following sections and associated responsibilities:

Phase (see Fig. 1)	Customer	Safety team	Functional application engineer
Concept	❶	❸	❸
External measures	❶	❸	❸
Application	❶	❶	❷
Validation and maintenance	❶	❸	❸

❶ → Takes over responsibility for safety
❷ → Works with
❸ → No relevance

2.1.1 Customer

The customer assumes full responsibility throughout the entire safety lifecycle. Project communication is also controlled by the customer.

Note:

In certain cases, the customer is not able to perform some of the described tasks and responsibilities on his/her own. In such cases, additional external companies or service providers will be brought onboard.

2.1.2 Safety team

The safety team consists of the safety application engineer (SAE) and the safety test engineer (STE). Each is assigned tasks related exclusively to the creation of the safety application.

Every employee in the safety team must fulfill the following qualifications:

- Experienced application engineer – at least 3 years experience
- Knowledge of the tools being used
  - Automation Studio
  - SafeDESIGNER
- Knowledge of functional safety and the relevant standards
  - Special safety training from TÜV or other institutions or
  - A long history of safety-related experience

Caution:

The selected person must fulfill the requirements mentioned above.

Caution:

The safety application engineer and the safety test engineer can be the same person in order to fulfill up to PL c or SIL 1. However, these positions must be occupied by two different people in order to fulfill anything higher than PL c or SIL 1.

Safety Application Engineer (SAE)
The safety application engineer is a member of the safety team.

The SAE is primarily tasked with creating the safety application in accordance with the customer's specifications – based on the Safety Requirement Specification (SRS).

Smaller projects often have just one safety application engineer, while larger applications/projects involve a safety application team.

Safety Test Engineer/Team (STE)
The safety test engineer is a member of the safety team.

The STE is primarily tasked with creating the test specification, as well as testing, verifying and validating the safety application created by the SAE.

Smaller projects often have just one safety test engineer, while larger applications/projects involve a safety test team.

2.1.3 Functional Application Engineer/Team (FAE)

The functional application must not be forgotten when looking at a safety application. The functional application is the counterpart to the safety application and involves meeting the functional requirements.

Both applications must be adapted to one another when necessary. The FAE is responsible for the implementation.

Note:

An example of a function / parameter that applies to both applications is the approval principle for the outputs, as well as diagnostics data that must be exchanged between the applications.
More detailed information regarding the functional application can be found in TM120 – Application Project Management.

Note:

The roles of functional application engineer and safety application engineer can also be covered by one person.

2.2 Interfaces

The previously defined responsibilities result in interfaces between the individual people/teams. We will now take a closer look at these interfaces.

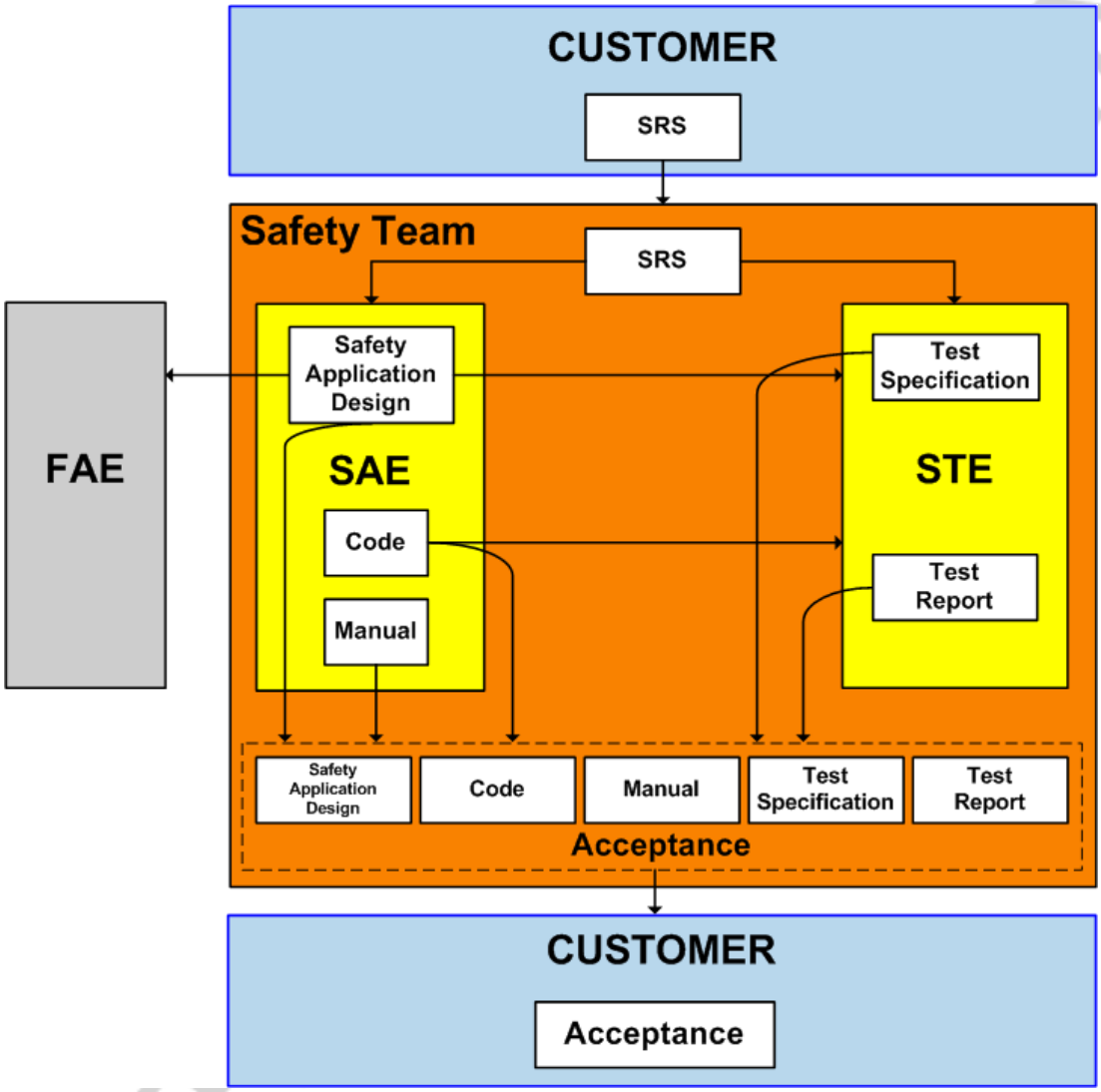


Fig. 4: Interfaces

Note:

Documents and templates are also used for communication and data exchanged between the interfaces / people. These documents and templates will be described in further detail in the appendix of this training module.

2.2.1 Customer and safety team

One of the interfaces consists of the communication and data exchange between the customer and safety team.

The customer provides the safety team with the specifications for the safety functions (SRS), which specifies the individual safety functions in great detail. The safety team uses this document as the basis for developing the safety application and the test specification.

> **Note:**
>
> The content or information contained in the SRS is described in the standard EN IEC 62061 and will not be discussed further in this training module.

When handing over the safety application (at the end of the development process), the safety team provides the customer with complete project documentation, which was validated by the customer when accepting the project, and consists of:

- Safety application design
- Code (incl. SafeDESIGNER documentation)
- Manual (if necessary)
- Test specification
- Test report

2.2.2 Safety Application Engineer (SAE) & Functional Application Engineer (FAE)

Another interface exists between the functional application and the safety application (i.e. between the SAE and the FAE).

Configuring the safety system requires certain tasks to be completed (such as creating Automation Studio projects, creating hardware configurations, etc) which can be done either by the SAE or FAE. Special focus should be given to hardware configuration and setting the parameters for these modules in Automation Studio. Data transfer between the functional and safety application has to be defined.

The SAE gives his safety application design to the FAE, in which the hardware structure is written down from a safety point of view (assignment of safety modules to a SafeLOGIC). Furthermore, a description of the communication channels between the functional and safety applications is also provided and their usage is specified.

2.2.3 Safety Application Engineer (SAE) & Safety Test Engineer (STE)

Another interface exists between the SAE and the STE.

The SAE gives his safety application design to the STE, which the STE then uses (together with the SRS) to create the test specification and to specify corresponding tests.

Once finished, the SAE also provides the STE with the safety application code for performing the specified tests.

If any errors are found during tests, then the safety team holds a meeting to analyze the problems and to work out suggested solutions or modifications.

3. DEVELOPMENT STAGES

Now that you are familiar with the individual roles and corresponding responsibilities, we will take a closer look at the individual development stages for the safety application.

3.1 Introduction

A simplified V-model of the software lifecycle is defined in EN ISO 13849. The main goal of the activities in the lifecycle is to have software that can be read, understood, tested and maintained.

The safety functions specification (i.e. the SRS) represents the initial data required for the lifecycle. The validated software is the resulting output at the end.

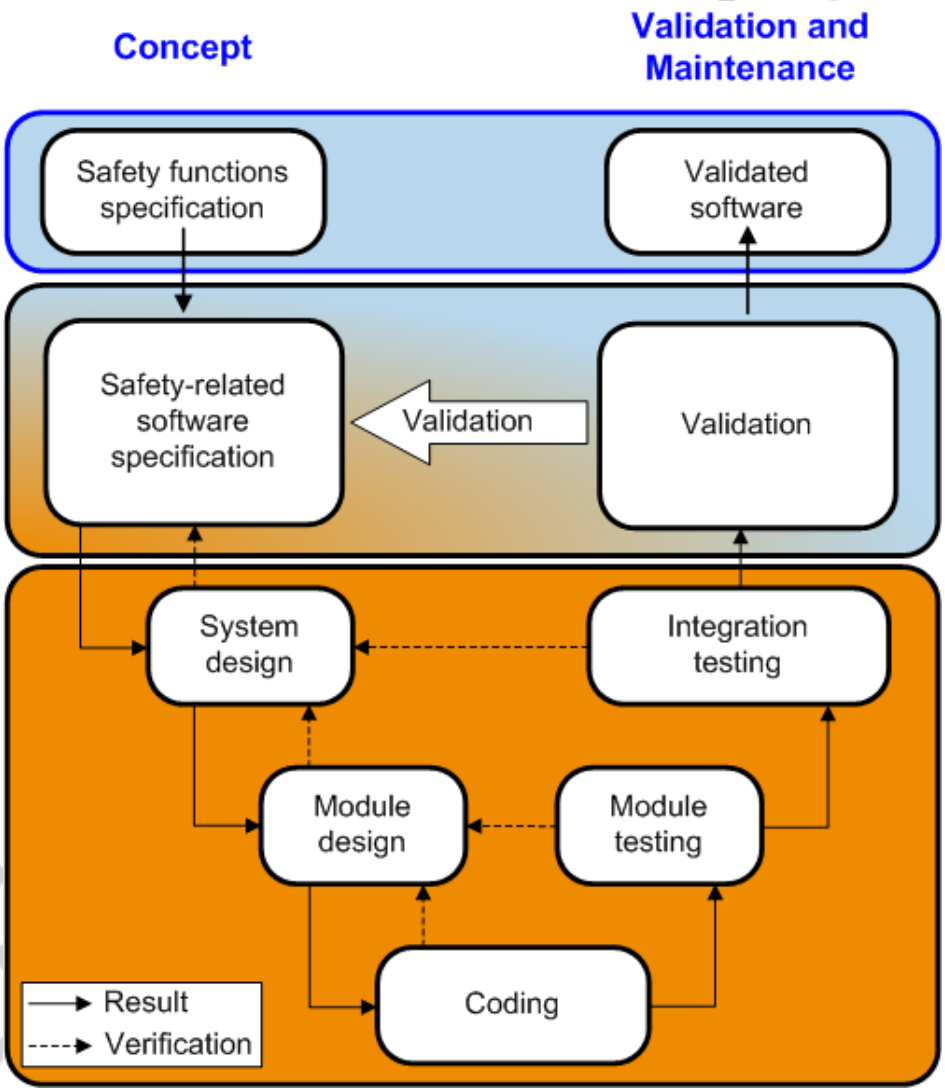


Fig. 5: V-model of the software lifecycle according to EN ISO 13849

The following table lists an overview of the development stages within phase 9 and the corresponding responsibilities.

Phase 9	Customer	SAE	STE	FAE	Document
Safety-related software specification	❶	❶	❷	❷	Safety Application Design (SAD)
System design	❸	❶	❸	❸	
Module design	❸	❶	❸	❸	
Coding	❸	❶	❸	❸	SafeDESIGNER documentation
Module testing	❸	❶	❸	❸	
Integration testing	❸	❷	❶	❸	Test specification and test report
Validation	❶	❷	❷	❷	Acceptance report

❶ → Takes over responsibility for safety
❷ → Works with
❸ → No relevance

Note:

The full lifecycle of a safety application involves more items than those listed here. This document only deals with the sections that are relevant to software development. This does not include things such as risk analysis, safety assessment, etc.

3.2 Safety-related software specification

This phase represents the beginning of the software lifecycle. Basic requirements are defined for further development of the software.

Caution:

The safety application design version must be based on a defined, approved version of the SRS.

The specifications for the safety functions, which are documented in the SRS, are the basis for this development stage. This information is now added in the first stage to the template provided by B&R (the safety application design).

3.2.1 Check the information

A checklist is included at the beginning of the template, which helps the user to check the information that is available or still needed. This list is used to check whether the SRS contains all of the required data or not.

Caution:

The information listed in the checklist must already be available in or provided in the SRS. Assumptions about any missing parameters (filter times, cycle times, etc.) are **NOT** permitted to be made. The customer must be contacted if the necessary information cannot be found.

Responsibility	Safety application engineer
Tasks	- Check contents of the SRS with the help of the checklist
Documents	- Safety requirement specification - Safety application design
Tools and aids	- Graphic and text processing - Version control system
Result	The information content of the SRS was checked and the checklist filled out accordingly in the safety application design

3.2.2 Safety application parameters

The necessary settings (e.g. for Automation Studio Version, Automation Runtime, cycle times, etc.) must be adhered to and documented.

Responsibility	Safety application engineer
Tasks	- Identify and determine parameters - Calculate worst case response time based on cycle times
Documents	- Safety requirement specification - Safety application design
Tools and aids	- Graphic and text processing - SafeDESIGNER - Version control system
Result	The cycle times needed for achieving the reaction time are specified in the safety application design.

3.2.3 Identify safety functions

The safety functions defined in the SRS are identified and the corresponding information is applied to the safety application design.

Responsibility	Safety application engineer
Tasks	- Identify safety functions - Enter information about the safety function in the prepared table(s)
Documents	- Safety requirement specification - Safety application design
Tools and aids	- Graphic and text processing - Version control system
Result	The individual safety functions have been identified and the necessary information about each safety function has been documented in the safety application design.

Caution:

Before proceeding to the next stage, the safety application design must first be approved by the customer in order to ensure that the information has been understood correctly.

3.3 System design and module design

Designing of the system can start after the SRS has been applied to the template, the safety application parameters have been defined and the individual safety functions were identified.

The first step when designing the system involves creating a block diagram with a schematic structure of the safety application. This block diagram illustrates the basic functional requirements of the SRS in regard to the safety application.

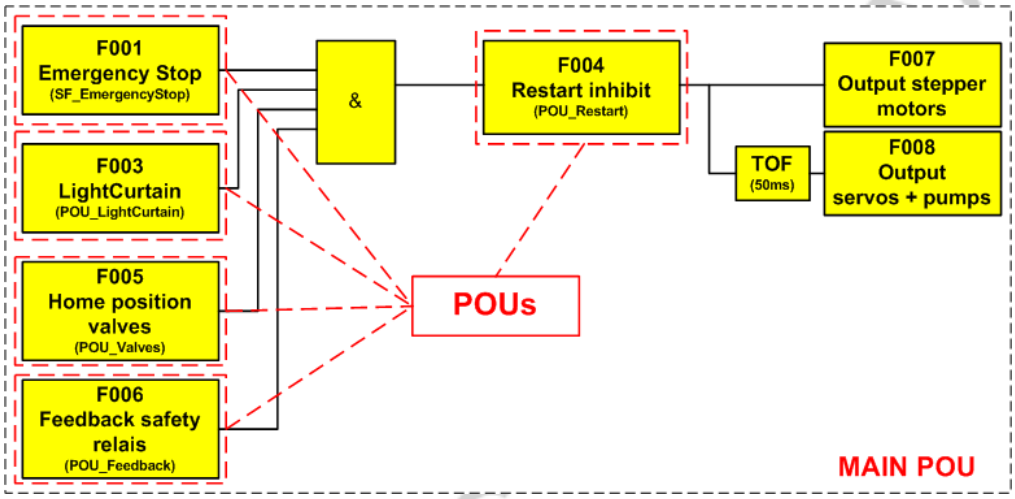


Fig. 6: Example of a block diagram

Note:

In accordance with IEC 61131-3, program organizational units (abbreviated POUs) are the language elements of a PLC program. They are small, independent software units containing program code.

Already in the block diagram, the individual POUs are given names that indicate a reference to the corresponding requirement in the SRS. For example, POU_LightCurtain would be the POU name, while the corresponding requirement from the SRS would be F003.

If the functionality in the block diagram involves a more complex relationship between multiple functions, then greater consideration must be taken and a more detailed description (module design) is necessary.

This module design consists of a verbal description as well as a block diagram for each module in order to clarify the implementation.

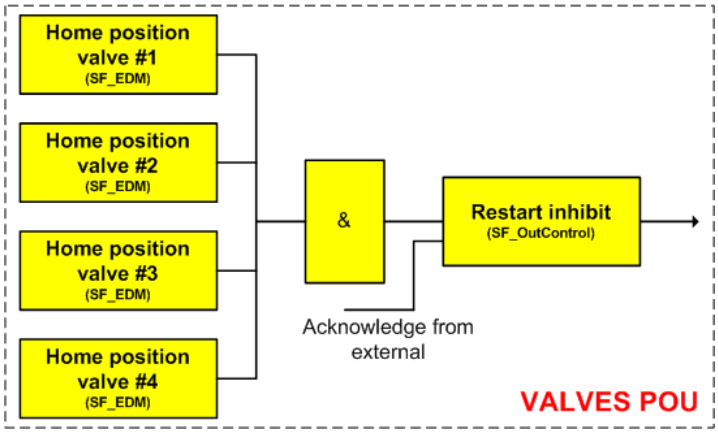


Fig. 7: Example of a POU block diagram

Responsibility	Safety application engineer
Tasks	- Analyze functionality of the application - Which functions are required? - Which functions work together? - How should the functions be connected? - Create block diagram - Take a closer look at more complex blocks
Documents	- Safety application design
Tools and aids	- Graphic and text processing - SafeDESIGNER - Version control system
Result	The safety application design contains a logically structured block diagram for illustrating the implementation. More complex blocks require more detailed descriptions and also have a separate block diagram.

Note:

At the end of this stage, the safety application design is handed over to the safety test engineer. The safety test engineer uses the information from the SRS and the safety application design to create the test specification.

3.3.1 Test specification

Caution:

The test specification must be based on a defined, approved version of the safety application design and the SRS.

The safety application must be accordingly tested to determine whether it is functioning properly or not. These tests must prove that the safety application fulfills the safety-related functionality specified in the SRS and that it is not affected by faults (e.g. short-circuits).

ID	Feature description
O.001.1	
O.002.1	

ID	Feature description
M001.1	
M002.1	

ID	Feature description
F001.1	The Emergency Stop stops the Machine in all operating modes.
F002.1	Changing between two modes is done by the mode selector
F003.1	The light curtain protects the user only in the automatic mode.
F004.1	The button starts and stops the machine in automatic mode.
F004.2	The button acknowledges all safety function requests.

Fig. 8: Test specification

Furthermore, the testing environment (e.g. simulation in SafeDESIGNER) must also be defined in order to enable the most effective testing.

A black box test is then performed at the POU level, based on the test specification. The test specification is the basis for the phase integration testing. The results are then entered in the test report.

Note:

The customer, safety application engineer and safety test engineer perform a review of the test specification in order to check for completeness.

Responsibility	Safety test engineer
Tasks	- Analyze application requirements - Analyze implementation (block diagram) - Specify tests and determine test environment
Documents	- Test specification
Tools and aids	- Graphic and text processing - Version control system
Result	All of the tests necessary for fulfilling the requirements from the SRS are documented. Potential faults (e.g. short-circuits) are also taken into account.

3.4 Coding and module testing

When programming and setting parameters, make sure that the following steps are executed or taken into consideration in SafeDESIGNER as well as Automation Studio.

3.4.1 Hardware configuration and interfaces

The functional application engineer and safety application engineer must agree upon who will perform the hardware configuration and parameter settings for the SafeLOGIC and SafeI/Os in Automation Studio.

The interface (communication channels) between the functional application and the safety application must also be taken into consideration.

Responsibility	Safety application engineer, Functional application engineer
Tasks	- Adding the SafeLOGIC and SafeI/Os to Automation Studio - Setting the parameters for the SafeLOGIC and SafeI/Os - Communication channels
Documents	- Safety application design
Tools and aids	- Automation Studio - SafeDESIGNER - Version control system
Result	An Automation Studio project with configured hardware and communication channels.

3.4.2 Coding

The SAE creates the safety application in accordance with B&R safety programming guidelines.

Note:

Further information about the programming guidelines can be found in the appendix of this training module.

Responsibility	Safety application engineer
Tasks	- Programming the safety application according to the software specifications - Adhering to the programming guidelines
Documents	- Safety application design
Tools and aids	- SafeDESIGNER - Version control system
Result	Implementation of the safety application in SafeDESIGNER is performed according to the system design or module design.

3.4.3 Module testing

This phase involves looking at the POUs and testing the base functionality of the created application. The test specification as well as the test results are recorded in the SafeDESIGNER documentation.

These tests can be performed either using the simulation environment in SafeDESIGNER or right on the machine.

Some areas cannot be tested immediately and must be reviewed, such as:

- Variables
  - Local
  - Global
- Device parameter

Information is also added to the SafeDESIGNER documentation during this phase:

- Machine manufacturer
- Project participants
- ...

Responsibility	Safety application engineer
Tasks	- Test the individual modules - Review - Document tests (specification & results) in the SafeDESIGNER documentation - Create SafeDESIGNER documentation
Documents	- SafeDESIGNER documentation
Tools and aids	- SafeDESIGNER - Version control system
Result	The performed tests and results are recorded in the SafeDESIGNER documentation. The SafeDESIGNER documentation contains all of the necessary information.

Note:

At the end, the safety application and the SafeDESIGNER documentation are given to the safety test engineer to begin testing.

3.5 Integration testing

This phase involves an overall test of the safety application. This means that the entire safety application is tested instead of individual POUs. The tests defined in the test specification are performed in this phase and the results are documented in a test report.

Note:

Black box tests are performed on the safety application.

In addition, the SafeDESIGNER documentation is then used to carry out a review, particularly of the following areas:

- Variables
  - Local
  - Global
- Device parameter

Responsibility	Safety test engineer
Tasks	- Test overall functionality - Review parameters, I/Os - Review SafeDESIGNER documentation - Document test results
Documents	- Test specification - Test report
Tools and aids	- Graphic and text processing - Version control system
Result	The results of the tests are recorded in the test report and then used to determine whether to proceed to the next phase or to make any necessary changes (→ change management).

3.6 Validation

Validation is an evaluated test (including analysis and testing) of the safety functions. The purpose of the validation procedure is to confirm the specification and the conformity of the design of the controller's safety-related parts.

Note:

More information can be found in the standards EN 62061 and EN ISO 13849-2.

Caution:

The customer is responsible for documenting the validation as well as checking for completeness.

These are black box tests that are performed at the system level.

The following basic tests must be performed:

- Restart behavior
- Cross-circuit on I/O channels against external signals (at least 24V / GND)
- Wiring tests

Responsibility	Customer
Tasks	- See standard

3.7 Acceptance

The acceptance represents the end of the development process. All documents and software are given to the customer. The acceptance report is also filled out at this point.

The acceptance report contains the versions of the individual documents as well as the CRC of the safety application.

Caution:

The acceptance report must be signed by all persons involved, the customer, the safety application engineer and the safety test engineer.

Responsibility	Customer, safety team
Tasks	- Application and all documents are handed over - Acceptance report is filled out
Documents	- Acceptance report
Tools and aids	- Graphic and text processing - Version control system
Result	Development process complete and acceptance report filled out with the signatures of all those involved.

3.8 Documentation

3.8.1 Safety application

After all phases have been completed, the following documents are provided as documentation for the safety application:

- Safety application design
- Test specification
- Test report
- Code (incl. SafeDESIGNER documentation)

3.8.2 Manual

Any special user notices required due to complexity of the safety application or special safety functions must be documented in a manual.

If there is no manual, then this information must be documented in the acceptance report.

3.8.3 Acceptance report

The acceptance report contains the following information:

- Document versions
  - Safety application design
  - Code (incl. SafeDESIGNER documentation)
  - Manual
  - Test specification
  - Test report
- CRC number of the SafeDESIGNER application

4. CHANGE MANAGEMENT

At some point during the project, new requirements or adjustments will arise that require changes to be made to the documents or software sections. These changes can cover a broad scope and can range from modified specifications to programming or testing errors.

In such cases, it is important that any potential effects of the changes are examined as closely as possible. These changed areas then also affect other stages of development.

Responsibility	**Safety team**
Tasks	- Analyze the problem - Why does something have to be changed? - Where did the problem occur? - What happened? - What are the solutions? - Analysis of effects - What are the consequences of this problem? - Which tests must be carried out again? - Which tests must be adjusted? - Which documents have to be revised? - Who is responsible for which changes?
Documents	- Safety application design
Tools and aids	- Graphic and text processing - SafeDESIGNER - Version control system
Result	The changes section of the safety application design contains corresponding information as well as the decision to implement the change.

Caution:

The respective changes to the SRS must be made by the customer and will not be covered any further in this training module.

Archiving

Note:

The customer is responsible for archiving the project and all corresponding documents and records.

Archiving of the safety application and the respective documents is a very important aspect in regard to change management. This data is archived by the customer as well as the safety team. This involves the items listed in the Documentation section.

A comparison of the project CRC is first made to determine if any changes to software or any document are required. This means that the CRC from the customer must always match that from the safety team. If this is not the case, then the reason why the versions or CRCs do not match must be determined.

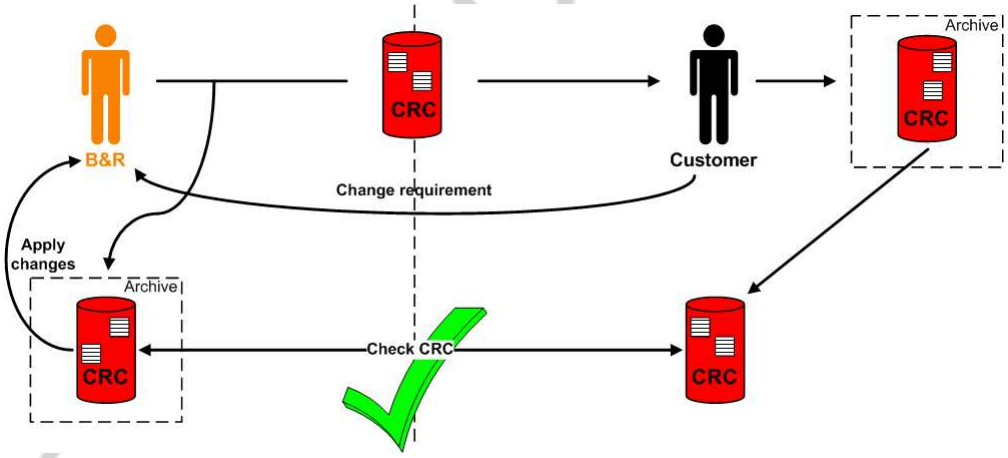


Fig. 9: Change management

Note:

The safety team will not make any changes if the CRCs differ from one another. Changes can only be made if the CRC archived by the customer matches that from B&R.

After a change has been successfully implemented, the data (both software and document) will be re-archived by the customer and the safety team.

5. SUMMARY

You have now become familiar with the responsibilities, roles and interfaces during the development of a safety application.

You are also familiar with the development process and its individual phases for creating safety applications in accordance with EN ISO 13849 and EN IEC 62061

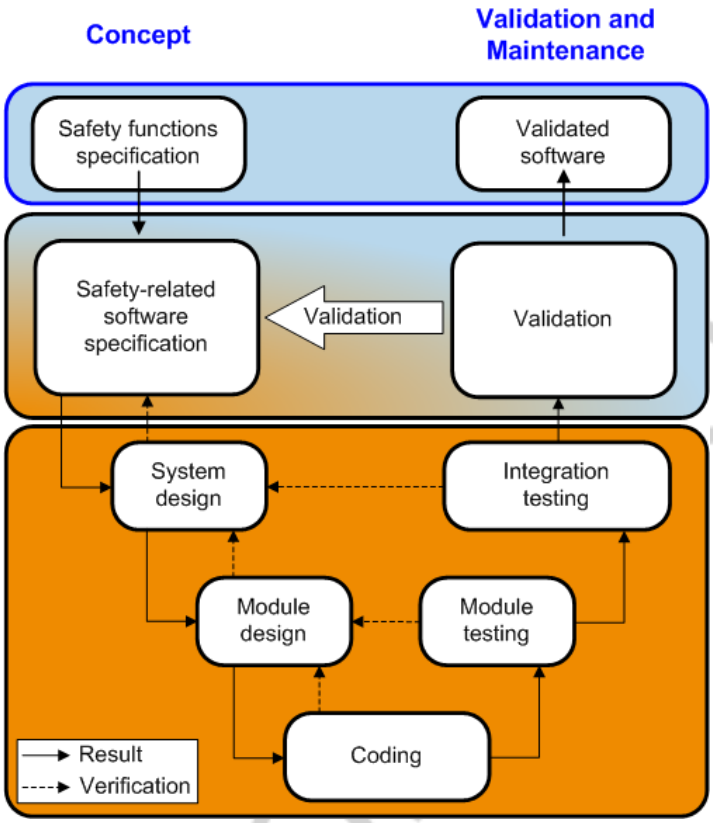


Fig. 10: Development process

You are familiar with the tasks and activities involved in change management.

6. APPENDIX

6.1 Programming guidelines

Programming guidelines are an integral part of a safety application. These must ensure that a person who is not familiar with the application will be able to interpret the code and the variables.

Variable names
Variable names that are easy to understand and that indicate the respective variable type should be used. For example; I/O variables, local variables, etc.

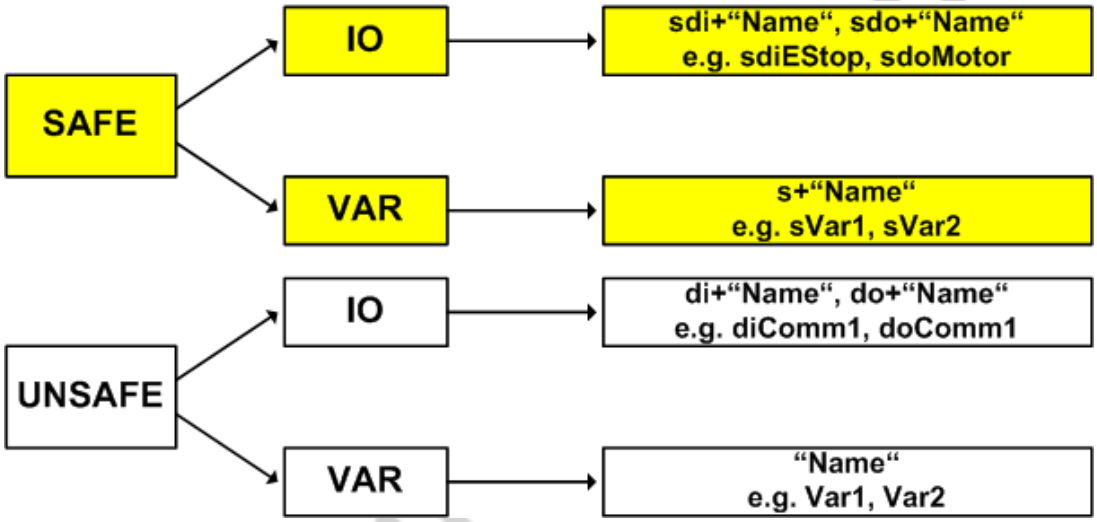


Fig. 11: Variable names

Note:

Information about variable declaration can be found in TM230 – Structured Software Generation.

Programming
The application should be divided into function blocks if more than one page is needed for a printout when using POUs. This is necessary in order to maintain a clear and organized overview.

Defensive programming
Attention is paid to integrating plausibility checks for conditions and value ranges in the software into the safety application. For example; value range for input signals, only one input can be active, etc.

Comments

Comments are used in the application to improve readability of the SafeDESIGNER application.

The SafeDESIGNER variable declaration contains a column for entering information and comments about the corresponding variable. This possibility should be used for improving readability.

Header

Both the worksheet and all created function blocks must have a header in the following format:

```
(*********************************************************************)
(* COPYRIGHT - B&R Industrial Automation                             *)
(*********************************************************************)
(* Application: Safety Training                                      *)
(* Function block: -                                                 *)
(* File: DemoProject                                                 *)
(* History: 17.06.2008 Created first release version                 *)
(*          19.06.2008 Changed the release button                    *)
(*          20.06.2008 Created second release version                *)
(*********************************************************************)
```

Fig. 12: Header

Architecture on 3 levels

The program and the function blocks must be designed and programmed based on the following schema.

Fig. 13: Safety application architecture

Type conversion

SafeDESIGNER has a function for converting safe data types to non-safe data types and vice-versa.

Great caution must be taken when making such conversions. Particularly when converting a non-safe data type to a safe data type. This sort of conversion requires an explanation to be provided right in the software.

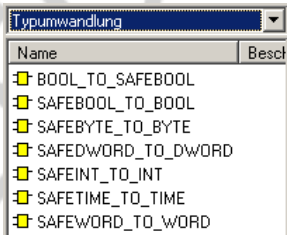


Fig. 14: Type conversion

6.2 Documents and templates

Note:

This section will only cover documents that are needed for the development process.
For example, this includes the SRS, but **NOT** the risk analysis.

6.2.1 SRS

The safety requirement specification describes the overall safety system in an understandable, structured manner and identifies the individual components.

Note:

More information can be found in the standards EN ISO 13849 and EN IEC 62061.

Caution:

There is no template for this document. This document **MUST** be provided and maintained by the **customer**.

6.2.2 Safety application design

The safety application design describes the structure and the implementation of the requirements from the SRS.

Furthermore, a checklist is used to check the SRS for completeness and clarity. If any problems are found, then the customer will be contacted and the open points will be clarified.

All of the data and information needed for the safety application is taken from the SRS and entered in this document.

Caution:

A template is available for this document. The document is the **responsibility** of the **safety application engineer**.
This document **must** be based on a specific version of the SRS.

6.2.3 Test specification

The necessary tests are specified in this document to verify proper functionality of the safety application.

Caution:

A template is available for this document. The document is the **responsibility** of the **safety test engineer**.
This document **must** be based on a specific version of the safety application design.

6.2.4 Test report

This document deals with the results of the tests that are listed in the test specification.

Caution:

A template is available for this document. The document is the **responsibility** of the **safety test engineer**.
This document **must** be based on a specific version of the test specification.

6.2.5 Acceptance document

This document contains the individual versions of the documents as well as the CRC of the safety application.

This document is signed by the customer as well as the safety team when the project is handed over to the customer.

Caution:

A template is available for this document. The document is the **responsibility** of the **safety team**.
This document **must** be based on a specific version of the documents.

6.3 Glossary

Term	Description
EN ISO 13849	Safety of machinery – Safety-related parts of control systems
EN IEC 62061	Safety of machinery – Functional safety of safety-related electrical, electronic and programmable electronic control systems
EN IEC 61508	Functional safety of electrical / electronic / programmable electronic safety-related systems
PL	Performance Level
SIL	Safety Integrity Level
POU	Program Organization Unit
SRS	Safety requirement specification
SAD	Safety application design
AG	Customer (German: Auftraggeber)
SAE	Safety application engineer
STE	Safety test engineer
FAE	Functional application engineer

Notes

Overview of training modules

TM200 – B&R Company Presentation **
TM201 – B&R Product Spectrum **
TM210 – The Basics of Automation Studio
TM211 – Automation Studio Online Communication
TM212 – Automation Target **
TM213 – Automation Runtime
TM220 – The Service Technician on the Job
TM223 – Automation Studio Diagnostics
TM230 – Structured Software Generation
TM240 – Ladder Diagram (LAD)
TM241 – Function Block Diagram (FBD)
TM246 – Structured Text (ST)
TM247 – Automation Basic (AB)
TM248 – ANSI C
TM250 – Memory Management and Data Storage
TM260 – Automation Studio Libraries I
TM261 – Closed Loop Control with LOOPCONR

TM400 – The Basics of Motion Control
TM410 – The Basics of ASiM
TM440 – ASiM Basic Functions
TM441 – ASiM Multi-Axis Functions
TM445 – ACOPOS ACP10 Software
TM446 - ACOPOS Smart Process Technology
TM450 – ACOPOS Control Concept and Adjustment
TM460 – Starting up Motors
TM480 – Hydraulic Drive Control

TM500 – The Basics of Integrated Safety Technology
TM510 – ASiST SafeDESIGNER

TM600 – The Basics of Visualization
TM610 – The Basics of ASiV
TM630 – Visualization Programming Guide
TM640 – ASiV Alarm System
TM650 – ASiV Internationalization
TM660 – ASiV Remote
TM670 – ASiV Advanced

TM700 – Automation Net PVI
TM710 – PVI Communication
TM711 – PVI DLL Programming
TM712 – PVIServices
TM730 – PVI OPC

TM800 – APROL System Concept
TM810 – APROL Setup, Configuration and Recovery
TM811 – APROL Runtime System
TM812 – APROL Operator Management
TM813 – APROL XML Queries and Audit Trail
TM830 – APROL Project Engineering
TM840 – APROL Parameter Management and Recipes
TM850 – APROL Controller Configuration and INA
TM860 – APROL Library Engineering
TM865 – APROL Library Guide Book
TM870 – APROL Python Programming
TM890 – The Basics of LINUX

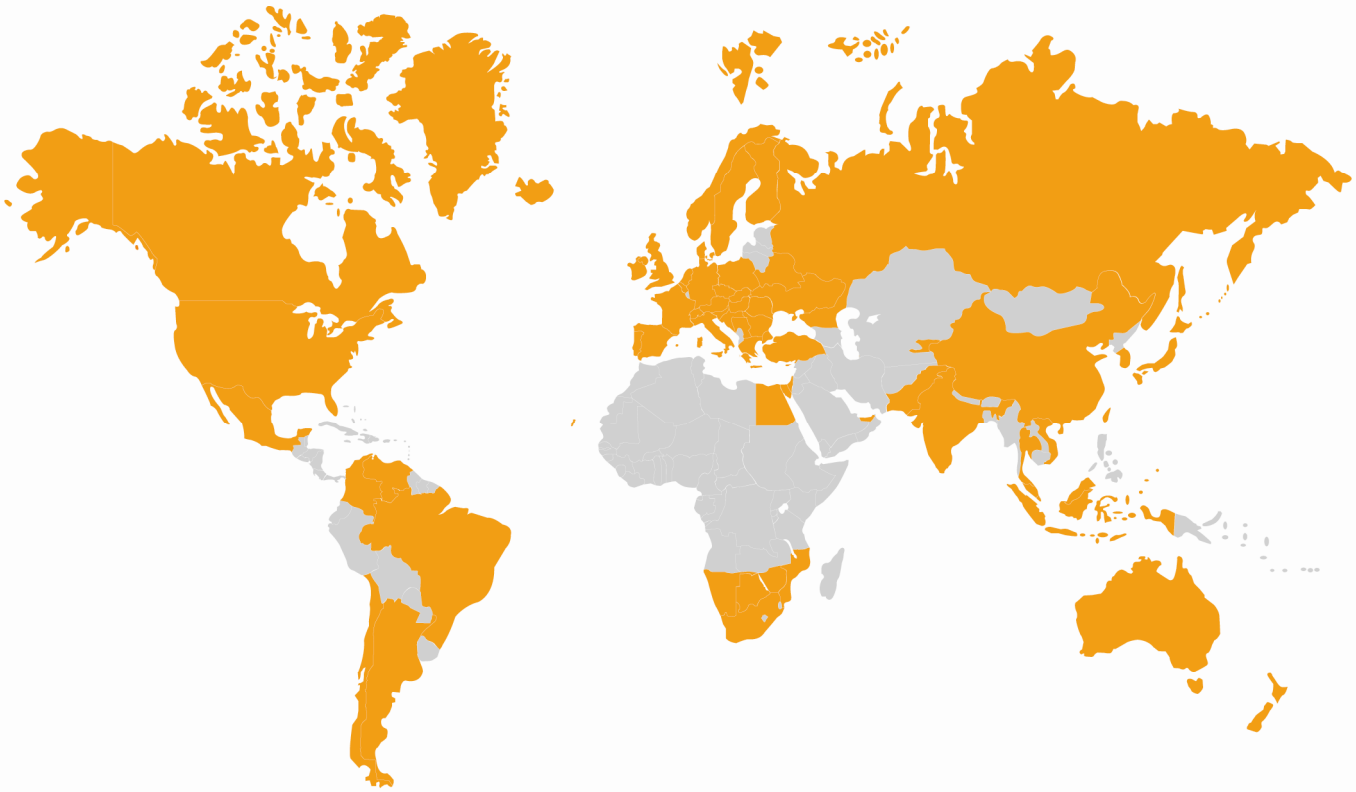
**) see Product Catalog

CORPORATE HEADQUARTERS

Bernecker + Rainer Industrie-Elektronik Ges.m.b.H.
B&R Strasse 1
5142 Eggelsberg
Austria
Tel.: +43 (0) 77 48 / 65 86 - 0
Fax: +43 (0) 77 48 / 65 86 - 26
info@br-automation.com
www.br-automation.com

TM530TRE.00-ENG 0509

155 offices in 60 countries - **www.br-automation.com/contact**



Argentina • Australia • Austria • Belarus • Belgium • Botswana • Brazil • Bulgaria • Canada • Chile • China • Colombia • Croatia • Cyprus
Czech Republic • Denmark • Egypt • Emirates • Finland • France • Germany • Greece • Hungary • India • Indonesia • Ireland
Israel • Italy • Japan • Korea • Luxemburg • Colombia • Kyrgyzstan • Malaysia • Mexico • Mozambique • Namibia • The Netherlands
New Zealand • Norway • Pakistan • Poland • Portugal • Romania • Russia • Serbia • Singapore • Slovakia • Slovenia • South Africa
Spain • Sweden • Switzerland • Taiwan • Thailand • Turkey • Ukraine • United Kingdom • USA • Venezuela • Vietnam • Zimbabwe