

Are **requirements** a power tool in your hands,
making your projects flourish
or are they holding you back?

Are you confident in the way how do you define
your **architectural elements**?

Do you understand what choices do you have
and are you fully aware of the consequences of
your decisions?

**Would you like to take advantage
of becoming fluent in this area?**

DE0214 Practical approach to requirements and architecture engineering

Requirements can either be your great advantage – gradually introducing further levels of details across the consecutive stages of development, clearly distributing the expectations between well defined architectural elements, making verification and validation simple and intuitive; or they can bring unnecessary chaos, delays and arguments between the teams.

Why not to have this area clockwork-organized?

General approach:

The *exida* approach is to provide a complete, practical guideline of development, architectural and requirement levels (according to ISO 26262 – but considering commonalities with other standards), relationships between them and understanding how do early design decisions impact late stages of the project.

DE0214 Practical approach to requirements and architecture engineering

Who should attend?

- ◆ Process/Methods Engineers
- ◆ System Safety Engineers
- ◆ Software Safety Engineers
- ◆ Hardware Safety Engineers
- ◆ Functional Safety Managers

Duration:

4 hours split as two 2-hours sessions

Language:

English

Location:

online

Certificate:

Each participant gets a letter of attendance.

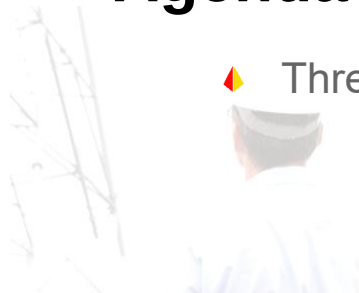
For more information, please contact:
Kerstin Tietel

☎ +49 89 44118232

✉ kerstin.tietel@exida.com

DE0214 Practical approach to requirements and architecture engineering

Agenda and Content

- 
- ◆ Three “pillars” of development defined in ISO 26262:
 - Horizontal layers – consecutive stages of development.
 - Vertical layers – relationships between requirements, design and verification.

- 
- ◆ Requirements engineering:
 - Requirements guidelines – useful, practical abstract from various standards.
 - Very important but often misunderstood: subjects, derivation and allocation.

- 
- ◆ Architectural engineering:
 - Defining architectural levels.
 - Consequences of different decisions.

- 
- ◆ Real-life examples:
 - What to do, what to avoid.