Reliable Tools for Functional Safety Development

Anders Holmberg
Agenda

- IAR Systems at a glance
- Functional Safety?
- Relevant standards
- What is tool qualification and why do I need it?
- How can I cope with the tool qualification?
- IAR Systems Solutions
- Q&A
Providing developers of embedded systems with world-leading software tools

11 offices worldwide with HQ in Uppsala, Sweden

Uppsala, Munich, Paris, Tokyo, Seoul, Shanghai, San Francisco, Dallas, Boston, Los Angeles, Sao Paulo

+ Distributor representation in 43 countries

Global professional technical support in 9 languages

Large ecosystem of partners

Listed on NASDAQ/Stockholm

33 years in the industry

32% of revenue invested in R&D
What is Functional Safety?

Definitions from IEC (http://www.iec.ch/functionalsafety/explained/)

• **Safety**
  Freedom from unacceptable risk of physical injury or of damage to the health of people, either directly, or indirectly as a result of damage to property or to the environment

• **Functional Safety**
  The *detection* of a potentially dangerous condition resulting in the *activation* of a protective or corrective device or mechanism to *prevent* hazardous events arising or providing *mitigation* to reduce the consequence of the hazardous event
Relevant Standards for Software in a Functional Safety Context

- **IEC 61508**
  *Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems*

- **IEC 62304**
  *Medical device software – Software life cycle processes*

- **EN 50128**
  *Railway applications - Communication, signalling and processing systems - Software for railway control and protection systems*

- **ISO 26262**
  *Road vehicles – Functional safety*

- **(DO178C)**
  *Software Considerations in Airborne Systems and Equipment Certification*
How am I affected by these standards?

• All these Functional Safety standards pose requirements on the development process!
  – Requirements gathering, implementation, testing, verification and validation of relevant safety functionality, programming language selection, etc, etc, etc…

• All these Functional Safety standards pose requirements on how to select development tools!
Tools Selection in a Functional Safety Context

- IEC 61508, IEC 62304, EN 50128
  - Very similar requirements (EN 50128 is based on IEC 61508)
  - Clause 7.4.4.10 of IEC 61508, part 3, states:
    
    To the extent required by the safety integrity level, the software or design representation (including a programming language) selected shall:

    a) have a translator which has been assessed for fitness for purpose including, where appropriate, assessment against the international or national standards;
    b) use only defined language features;
    c) match the characteristics of the application;
    d) contain features that facilitate the detection of design or programming mistakes;
    e) support features that match the design method.
Tools Selection in a Functional Safety Context

- IEC 61508, IEC 62304, EN 50128
  
  Clause 7.4.4.10 of IEC 61508, part 3, states:
  *To the extent required by the safety integrity level, the software or design representation (including a programming language) selected shall:

  a) have a translator which has been assessed for fitness for purpose including, where appropriate, assessment against the international or national standards;*
Tools Selection in a Functional Safety Context

- IEC 61508 – Part 3
  - 7.4.4.6 For each tool in class T3, evidence shall be available that the tool conforms to its specification or documentation. Evidence may be based on a suitable combination of history of successful use in similar environments and for similar applications (within the organisation or other organisations), and of tool validation as specified in 7.4.4.7.

  - 7.4.4.7 The results of tool validation shall be documented covering the following results:
    a) a chronological record of the validation activities;
    b) the version of the tool product manual being used;
    c) the tool functions being validated;
    d) tools and equipment used;
    e) the results of the validation activity; the documented results of validation shall state either that the software has passed the validation or the reasons for its failure;
    f) test cases and their results for subsequent analysis;
    g) discrepancies between expected and actual results.
## Tools Selection in a Functional Safety Context

### ISO 26262 Table 5 — Qualification of software tools classified TCL2

<table>
<thead>
<tr>
<th>Methods</th>
<th>ASIL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
</tr>
<tr>
<td>1a Increased confidence from use in accordance with 11.4.7</td>
<td>++</td>
</tr>
<tr>
<td>1b Evaluation of the tool development process in accordance with 11.4.8</td>
<td>++</td>
</tr>
<tr>
<td>1c Validation of the software tool in accordance with 11.4.9</td>
<td>+</td>
</tr>
<tr>
<td>1d Development in accordance with a safety standard(^a)</td>
<td>+</td>
</tr>
</tbody>
</table>

\(^a\) No safety standard is fully applicable to the development of software tools. Instead, a relevant subset of requirements of the safety standard can be selected.

**EXAMPLE** Development of the software tool in accordance with ISO 26262, IEC 61508 or RTCA DO-178
How can I cope with tools validation and justification?

Essentially two choices:

Do your own validation
  – Time and resource consuming
  – Compiler validation is outside the core competence for the absolute majority of tools users
    (If done fully in accordance with functional safety best practices...)
How can I cope with tools validation and justification?

Essentially two choices:

Do your own validation

- Project specific!
  Projects must be very similar to be able to reuse previous validation:
  - Hardware
  - Functional Safety requirements
  - Tool(s) options
  - Project staffing, etc, etc, etc...
How can I cope with tools validation and justification?

Essentially two choices:

- Rely on third-party validation/qualification evidence
  - Certification by e.g. TÜV
  - (Or testing by independent testing house)
The IAR Systems Solution

• Build chain that is certified by TÜV SÜD and complies with IEC 61508, ISO 26262, EN 50128, (IEC 62304)

• Certificate and report that accompanies the certificate, detailing the scope and validity of certification

• Full documentation, including Safety Guide and test reports

• Compiler that supports C89, C99, and C++ languages

• The certified version is frozen:
  – Prequalified service packs
  – Lifetime support of the FS version

• Regular updates on known issues

• Optional analysis tools available for static and dynamic analysis
The IAR Systems offering

The Functional Safety versions of IAR Embedded Workbench for ARM, Renesas RX and RL78 delivers:

**Simplified validation**
- Functional Safety certificate from TÜV SÜD
- Safety report from TÜV SÜD
- Safety guide

**Guaranteed support through the product life cycle**
- Prioritized support
- Validated service packs
- Regular reports of known problems

Validated EN 50128 (ARM)
IEC 61508
ISO 26262
IAR Systems Functional safety Solution

• Removes the requirement to prove that the toolchain complies with the safety standard.
• Software Development Lifecycle (SDLC) can focus on finding bugs instead of compiler issues
• Certified service packs and prioritized support help protect your tool investment
• Comes with optional static and dynamic analysis add-ons
Q&A
Thank you for your attention!
Take part of our Embedded World offer – 10% off on your code quality needs