Smart Tools for Ensuring Safety in Automotive Applications

IAR SYSTEMS

Shawn A. Prestridge

Key Takeaways from Session

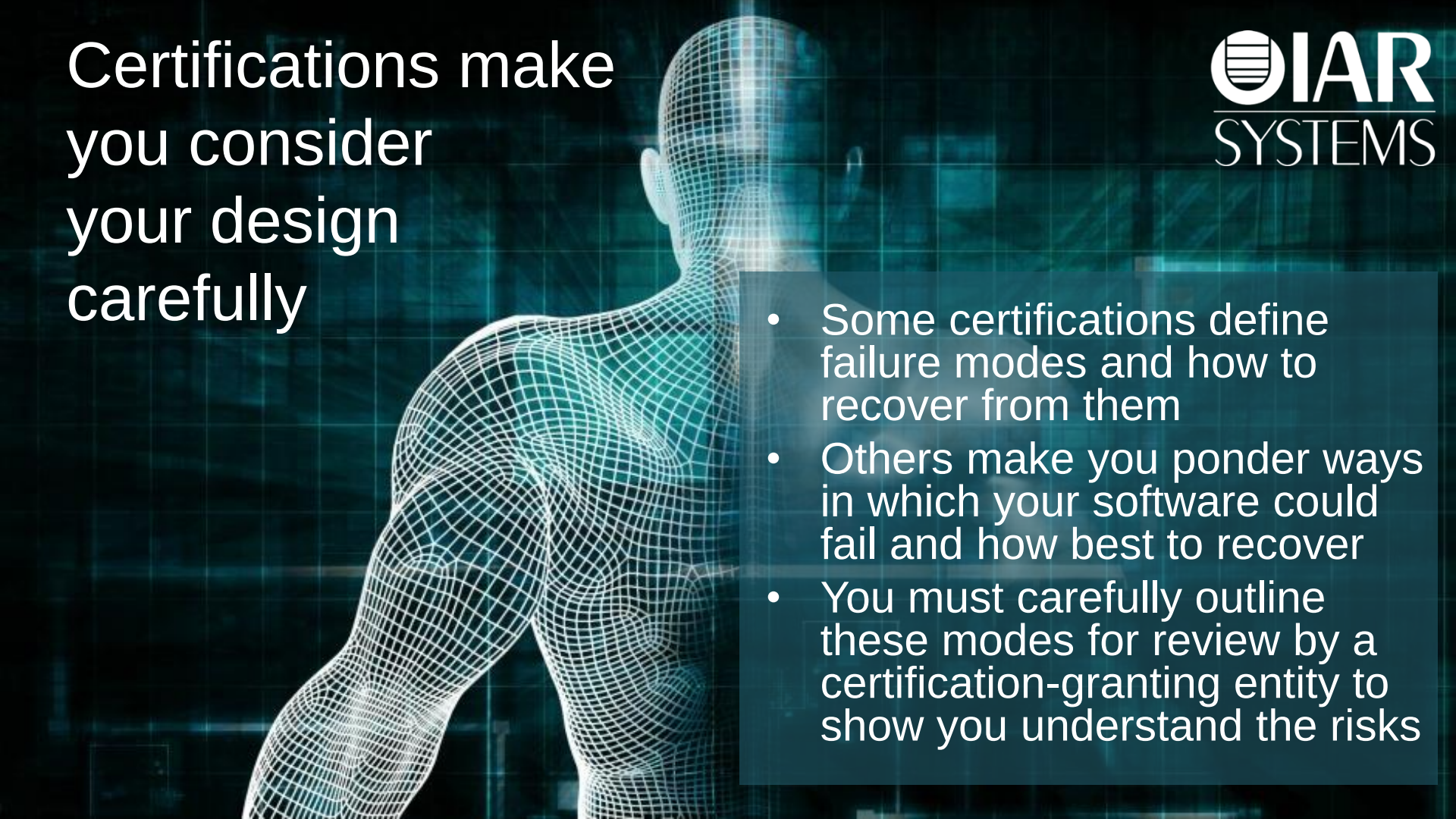
- What the benefits of Functional Safety are
- What the most popular safety certifications are
- Why you should care if your development tools are FS-certified?
- How development tools are FS-certified
- How FS-certified tools will speed your path to your own certification
- Which IAR Embedded Workbench has FS-certification
- What comes with the FS version of IAR Embedded Workbench
- Code analysis helps speed the path to certification

Benefits of Functional Safety

Why Should I Seek Certification?

- Reduce liability risks associated with your application
- Reduce odds of product recall
- Reduce number of firmware updates
- Ensure compliance with international standards and requirements
- Protects your company’s reputation
- Also protects your company’s bottom line

Certifications make you consider your design carefully



- Some certifications define failure modes and how to recover from them
- Others make you ponder ways in which your software could fail and how best to recover
- You must carefully outline these modes for review by a certification-granting entity to show you understand the risks

Functional Safety Certifications for Development Tools

What does it mean if my tool is FS-certified?

It means that your development tool has gone through a rigorous qualification process to ensure that it produces reliable and repeatable results when compiling your code. Additionally, it means:

- Development processes are in place to manage how the tool works with specific requirements put forth by different functional safety standards
- There are test and quality measures of the tool show validation of compliance with different language standards

What does it mean if my tool is FS-certified?

It also means:

- There are specific processes and metrics in place to handle issues reported from the field and how users are updated about known issues
- A safety manual is provided to show proof-of-compliance with standards and how to operate the development tool to comply with FS standards
- Assessment takes into consideration how many developers are using the toolchain to ensure it has a broad user-base

How do I certify my current toolchain?

The certification process is rather rigorous. The IEC 61508 standard details how support tools should be qualified in Section 7.4.4, but it is rather ambiguous on how a compiler should be qualified. Consider clause 7.4.4.10:

“The selected programming language shall have a translator which has been assessed for fitness for purpose including, where appropriate, assessment against the international or national standards.”

These and other stipulations make it difficult to certify a tool on your own and can result in significant work on your part to prove fitness and even more work to document why you think you have proven fitness! This only gets worse as you try to achieve higher and higher SIL safety levels.

I'll just use open-source software, thanks…

Not so fast! The common argument here is that if your project uses the same uncertified tool as another project that did eventually achieve certification, then you should be covered…right?

This is definitely not the case! You are still required to prove that:

- Your project is similar enough to the other project that you use the same functionality of the toolchain as the other project (impossible without source code-level access to the other project)
- You use the toolchain in a similar manner as the one that did achieve safety certification

You usually end up having to do the same work to requalify the tool!

How does using a Functional Safety-Certified tool speed my certification?

How does an FS-tool speed my certification?

- The simple answer is that it removes the requirement that you have to prove your toolchain complies with the safety standard.

- It also means that your test-and-fix phase of the Software Development Lifecycle (SDLC) can focus on finding bugs in your source code instead of wondering if a compiler issue is causing your problems

- Certified service packs mean that you don’t have to recertify the tool to get added functionality to your toolchain

How much time and money can it save me?

- Quite a bit! As we've seen previously, some of the requirements for tool certification can be a bit nebulous, so you avoid the back-and-forth with your certifying entity.
- Tool certifications can take up to 6-12 months of calendar time and occupy several employees, nominally 2-5
  - Also places extra testing requirements on each project using the tool
  - The actual numbers will depend on which SIL your project requires

- By using a tool that is FS-certified, you only need to certify your application which frees people from also needing to prove the development tools, which can save you upwards of $100k US.

What you get with an FS-certified version of IAR Embedded Workbench?

What do I get with my FS version?

The benefits of using the functional safety version are:

- A complete build chain that is certified by TŰV SŰD to comply with the requirements for tools selection in IEC 61508 and ISO 26262
- A report that accompanies the certificate t states under what circumstances the certificate is valid
- A test report that shows how the tool was tested to demonstrate compliance

What do I get with my FS version?

IAR SYSTEMS

You also get:

- A compiler that supports C89, C99, and C++ languages (Note that the safety standards do not recommend using exceptions and RTTI in C++)
- Prequalified service packs for your FS tool to maintain certification and support for the life of the FS version (as long as there are paying customers under support contract for that version)
- Regular updates on known issues

How do I certify my application?

General outline of FS requirements

- Identify what the safety functions are
- Identify risk reduction methods for the safety functions
- Verify safety function performs the way it’s supposed to
- Verify that system meets standards by rigorous testing
- Conduct Function Safety audits to assess testing evidence

Speeding the path to certifications

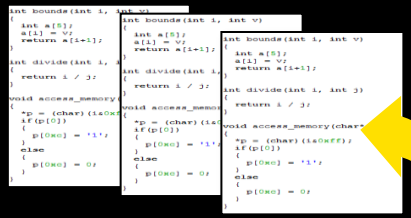
- Certifications are easier to achieve when you can prove that your code conforms to a coding standard (such as MISRA)
- Testing reports show that the overall number of defects in the software is low, despite many hours of testing and proves maturity of your development organization
- Code analysis also shows that your results are repeatable because you have a process in place to find and fix defects.

IAR SYSTEMS

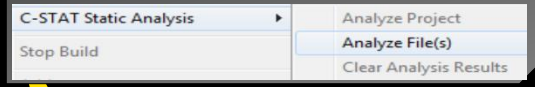
Using code analysis to help certification efforts

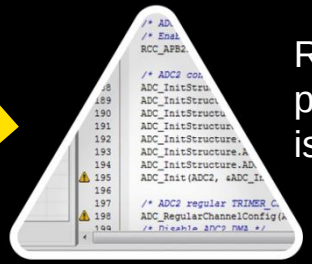
Static and Dynamic Analysis

Implement your design in code



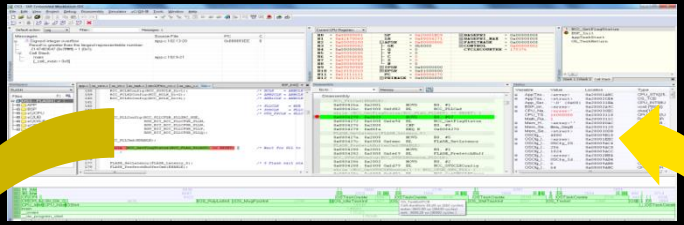
Let C-STAT analyze your code



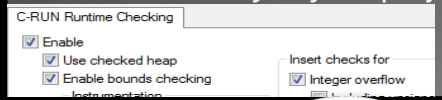


Review potential issues

Build and debug the application



Let C-RUN analyze your project

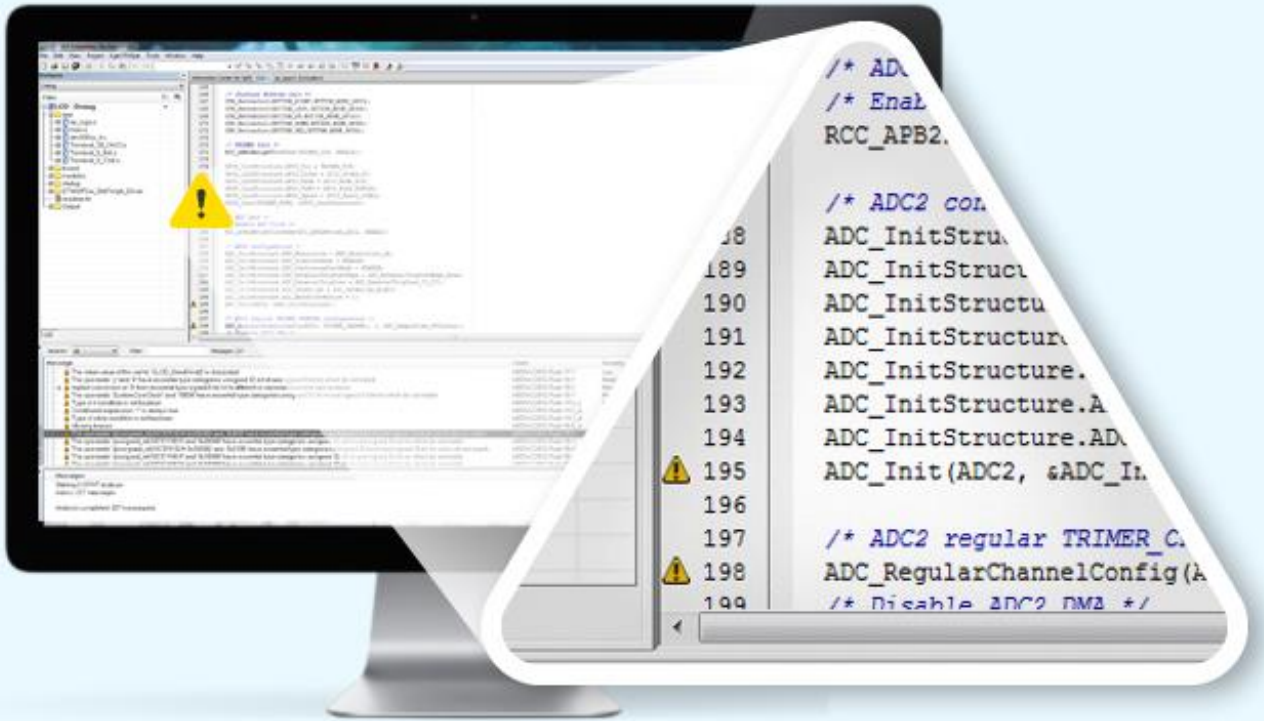


Investigate runtime errors

Requirements → Design → Implementation → Verification → Maintenance

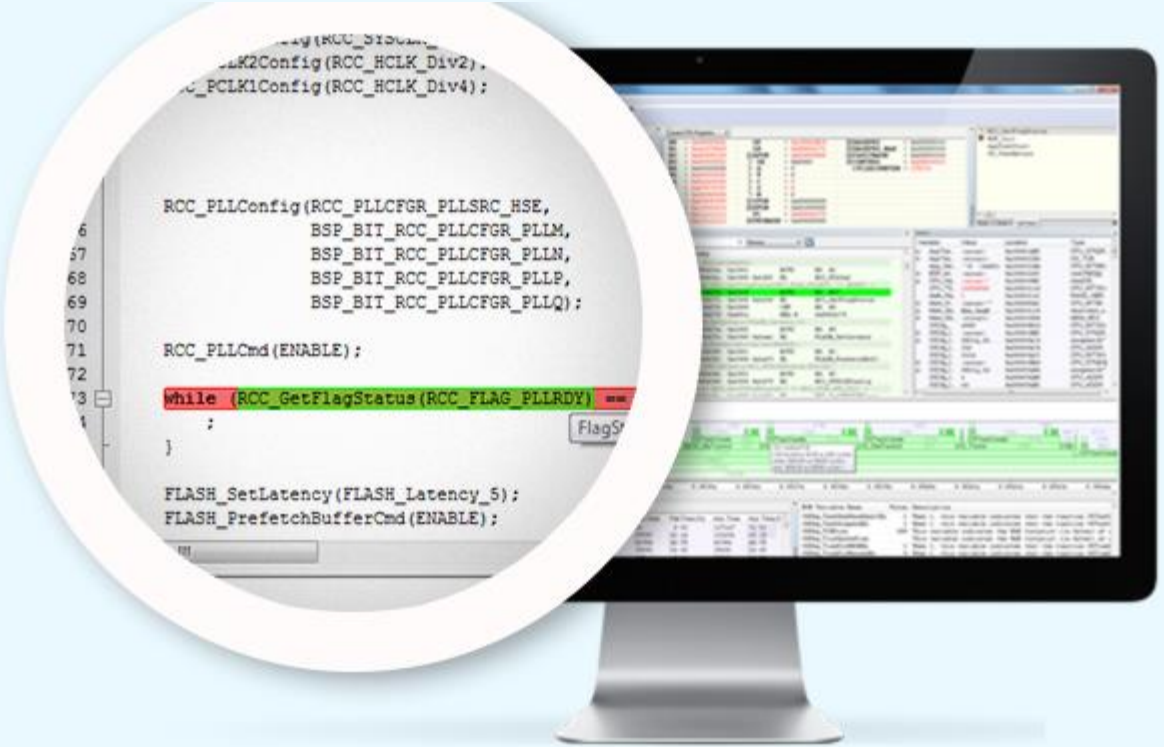
C-STAT Static Analysis

- Analysis of C and C++ code
- Intuitive and easy-to-use settings with flexible rule selection
- Checks compliance with rules as defined by MISRA C:2004, MISRA C++:2008 and MISRA C:2012
- Includes ~250 checks mapping to hundreds of issues covered by CWE and CERT C/C++
- Over 500 rules!



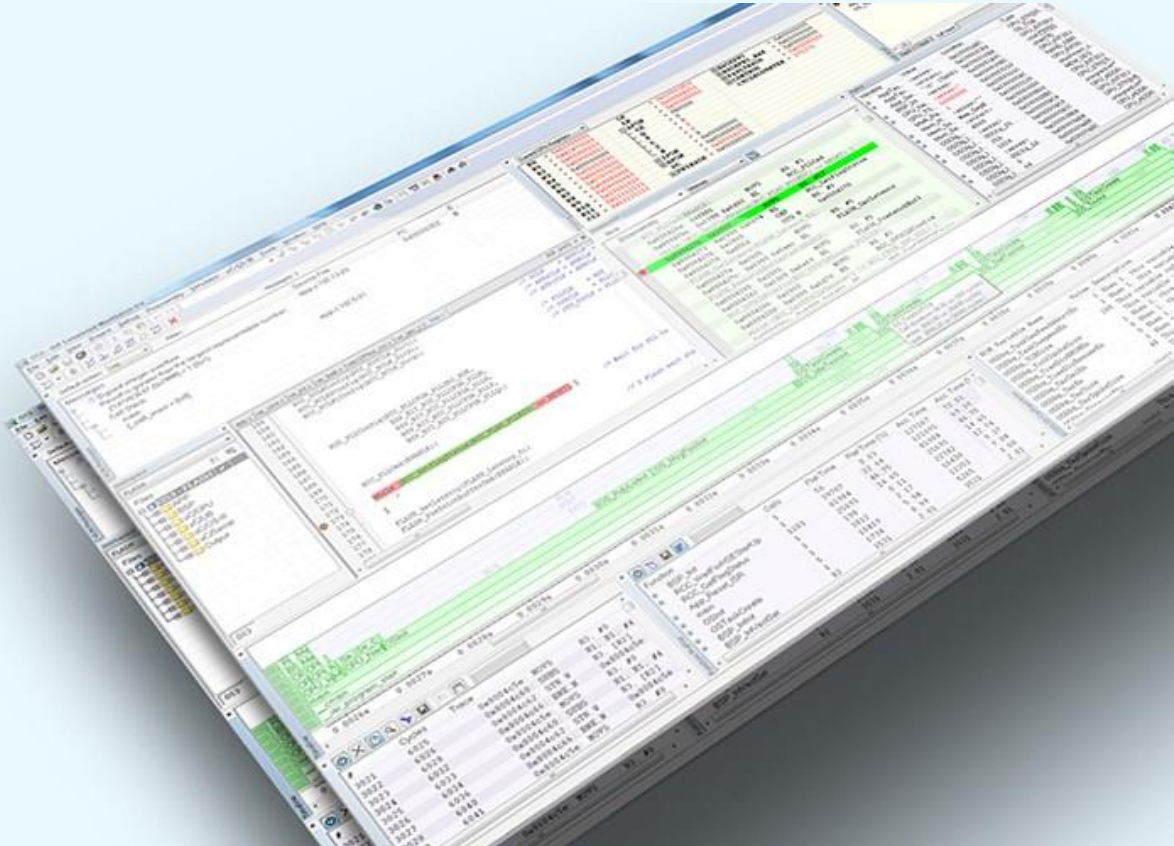
C-RUN Runtime Analysis

- Support for C and C++ code
- Intuitive and easy-to-use settings
- Code correlation and graphical feedback in editor
- Bounds checking to ensure accesses to arrays and other objects are within boundaries
- Heap and memory leaks checking
- Comprehensive and detailed runtime error information



C-STAT and C-RUN **DEMO**

IAR SYSTEMS



Comparing tools

Compared to other commercial offerings, C-STAT and C-RUN:

- Have a sizeable chunk of the functionality of other commercial suites
- Are only a fraction of the cost of other commercial suites
- Are intended to be used by the individual developer who is desk-checking their code (rather than by a QA professional)