Linaro connect
San Francisco 2017

Functional safety and development tools

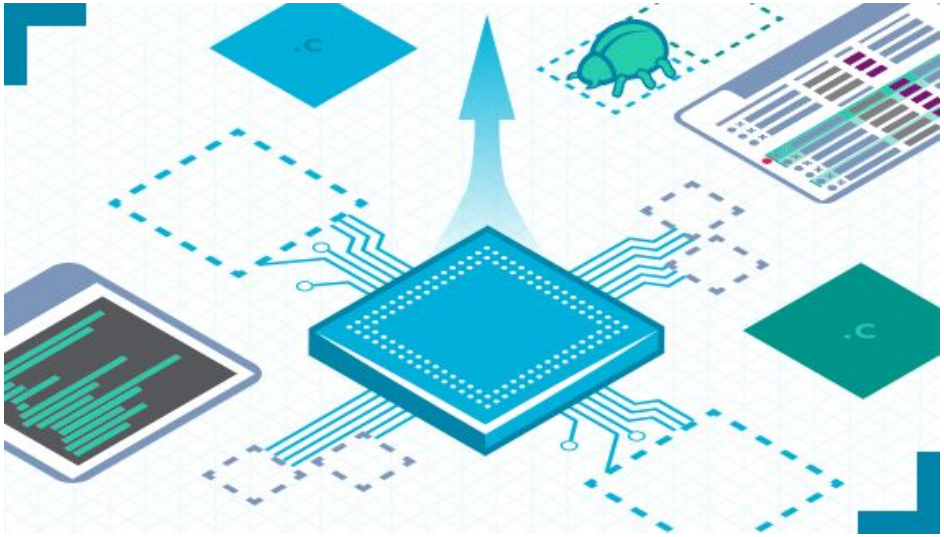
Peter Smith

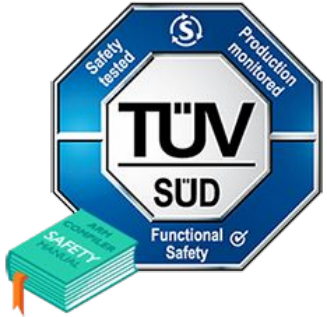
Objectives

- By the end of this presentation the aim is for you to understand
  - Basic concepts behind functional safety and functional safety standards
  - The steps a safety-related developer has to go through to use a development tool
  - What a tool developer can do to aid qualification and certification?
  - Challenges for open source tools

My background with functional safety

- Worked in the proprietary Arm Compiler Toolchain from 2000 - 2016
- Heavily involved in the production of the Qualification Kit and subsequent certification of Arm Compiler 5





Linaro
connect
San Francisco 2017

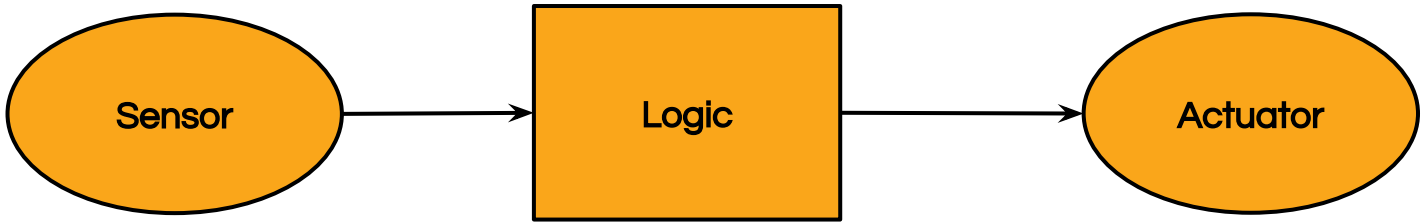
ENGINEERS
AND DEVICES
WORKING
TOGETHER

Functional Safety

- What is functional safety?
- How does it relate to software?
- What are functional safety standards?

What is functional safety?

- **Functional safety** is the part of the overall safety that depends on a system or equipment operating correctly in response to its inputs.



Functional Safety Concepts

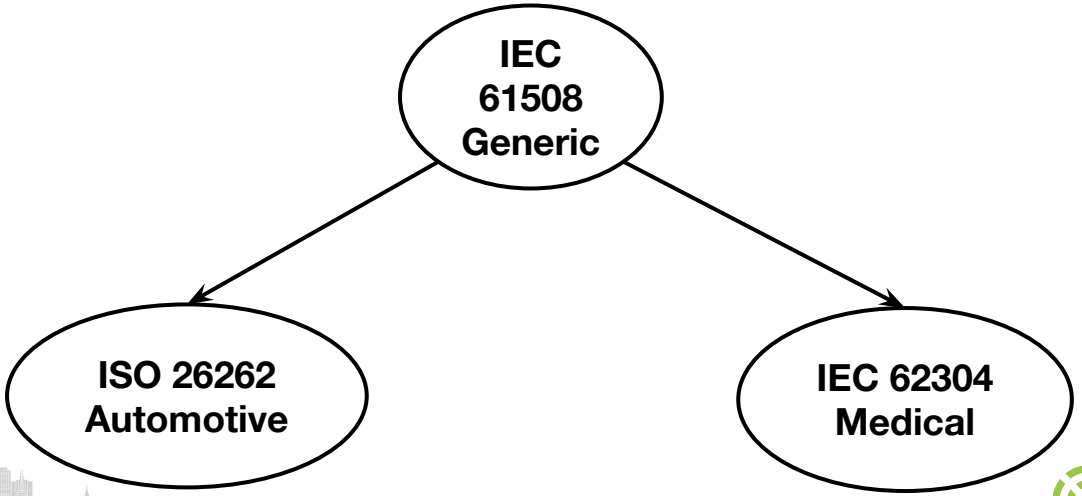
- Each system operating in a particular context has a failure rate that does not lead to unacceptable risk
- Functional safety does not compose
- Different parts of the system have different risk reduction targets
  - Often expressed as a safety integrity level such as SIL or ASIL
- Random hardware failures dealt with by detection and prevention
- Systemic failures (defects) reduced by rigorous development process
- Software in itself does not present a risk to humans, but it may control or monitor hardware that does

Functional Safety Standards

- Assumption that you are using some kind of development process
  - Adds extra functional safety requirements to the process
  - Usually structured as a v-model
- Cover entire lifecycle of product
  - Including planning, maintenance and tool usage
- Documentary evidence required for compliance
  - Traceability from requirement to implementation to validation
  - Map your process onto the reference v-model
- Intentionally vague, an argument must be made based on evidence
  - Widen applicability of the standard

Functional Safety Standard per industry

- “The nice thing about standards is that there are so many of them to choose from.” [Tanenbaum]
- Industry specific standards derive from a general standard
  - Risk factor
  - Accepted use cases



Linaro
connect
San Francisco 2017

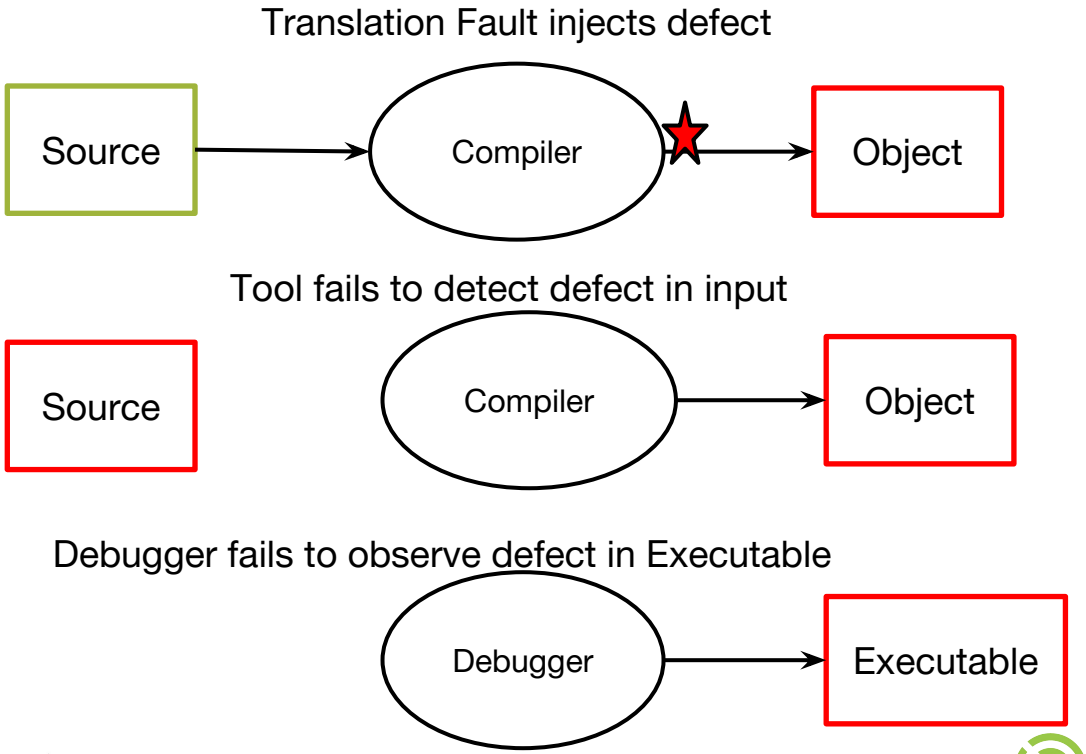
ENGINEERS
AND DEVICES
WORKING
TOGETHER

Development tools

- How does functional safety relate to development tools?
- What does a safety-related system developer need to do to assess their tools?
- What is tool qualification?
- What does tool certification mean?

Development Tools

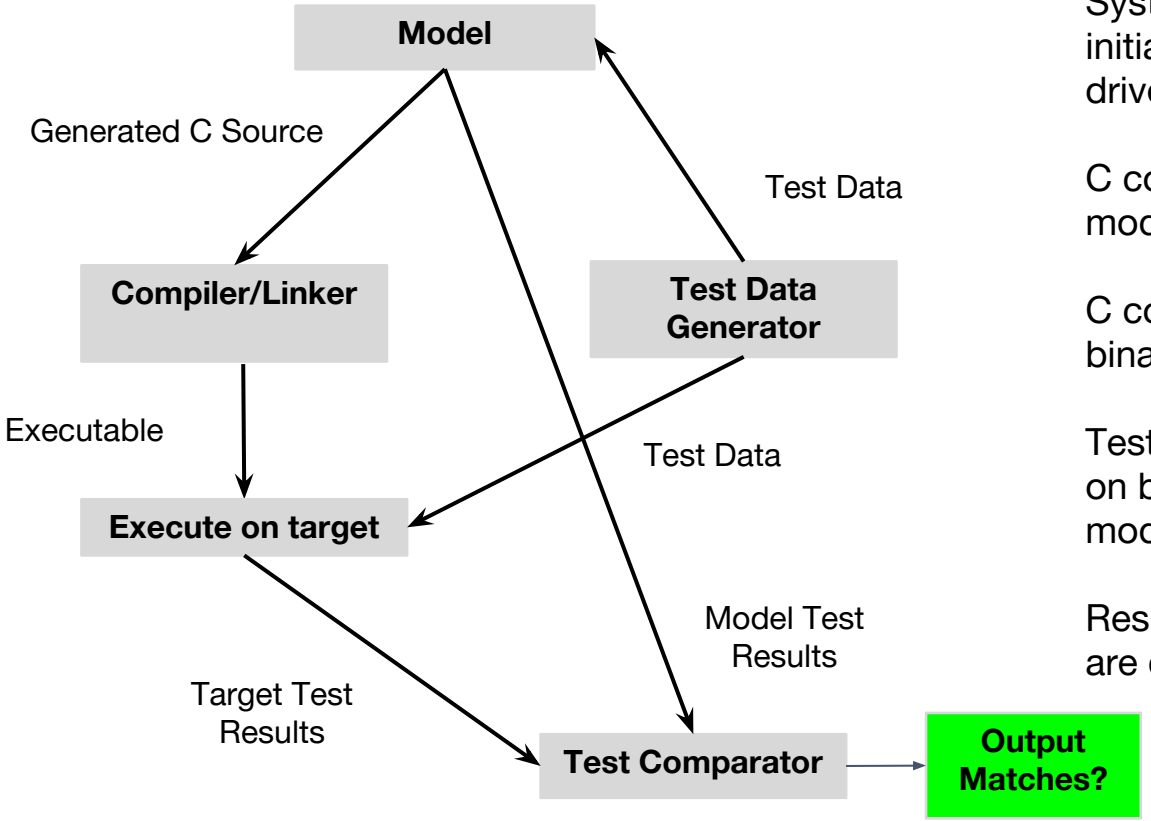
- Development tool failures may result in failures of the safety related system



Development tools

- All development tools must be evaluated and evidence presented that they are suitable for use
  - Includes all tools, such as the script used to generate test data
- Details vary between standards but in general:
  - Establish use case, required feature set and environment for tools
  - Determine the consequences of faults in the tools
  - Document steps taken to avoid known faults in tool
  - Establish the confidence level that faults in the tool will be detected
  - Take steps to mitigate tool failures
  - If lacking sufficient confidence that errors in tool won't affect safety-related system then the tool must be qualified
- Responsibility for qualification with the tool user and not the tool developer
- Evaluation is specific to tool version

Example Tool Flow



System is developed and initially tested using model driven development

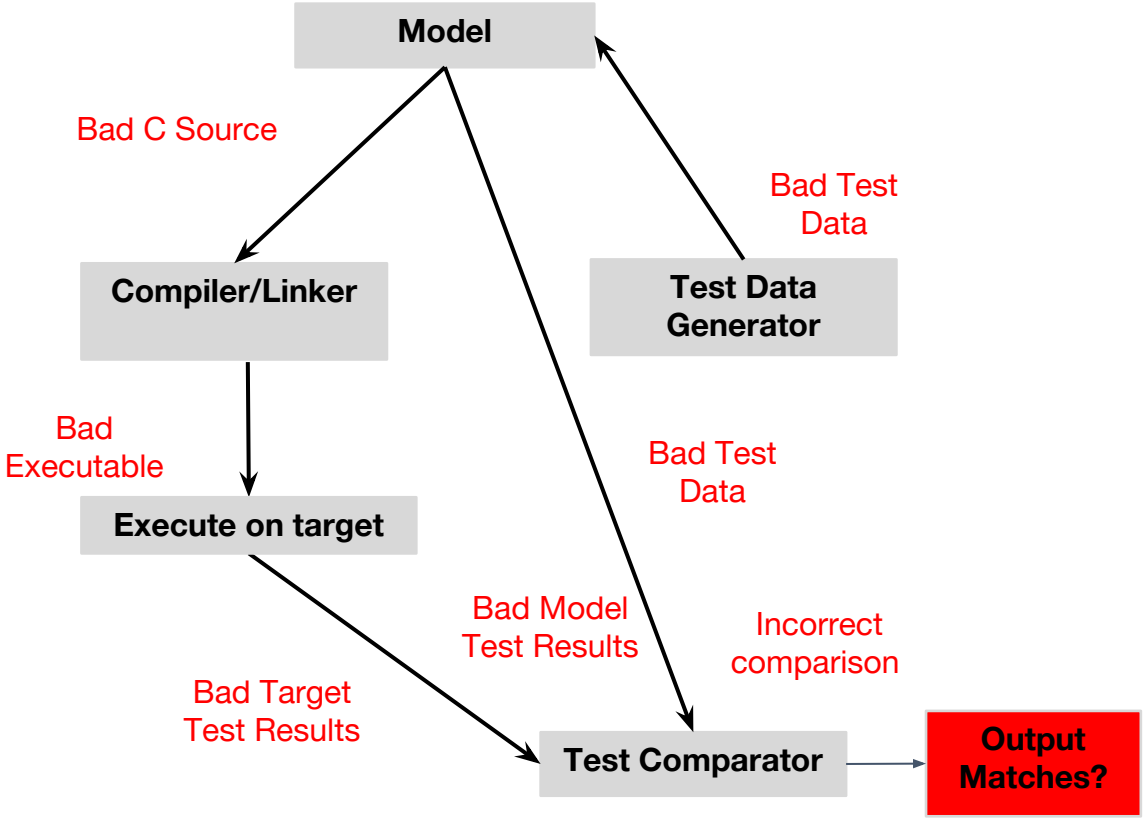
C code is generated by the model

C code is translated into a binary to run on the target

Test data is generated and run on both the target and the model

Results of model and target are compared

Example Tool Flow



Tool Failure modes and the consequences are examined

Output of one tool is the input to another. Downstream tool may be able to detect error in upstream tool

Inspection and review of inputs and outputs increase chances of error detection

Diversity approach such as cross-checking test results from model and system increases confidence in tools

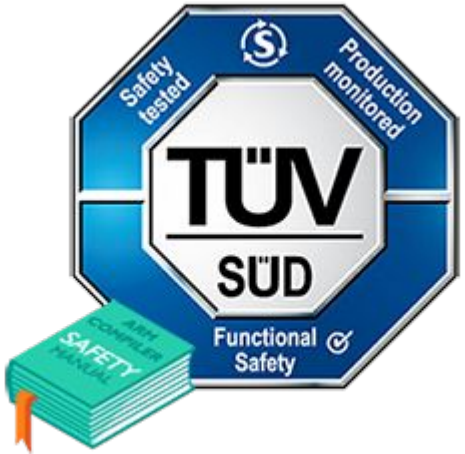
ISO 26262 Tool Qualification

- Qualification is an additional step to provide evidence that a tool can be used with confidence
- Evidence can be taken from any of the categories in the table

Method	Description	ASIL Highly Recommended
Increased confidence from use	Developer has used exact same version of tool in a similar safety-related situation.	A, B, C
Tool is developed according to a functional safety standard	Tool has gone through development using the same process as the safety-related system.	D
Evaluation of development process	Audit tools development process against some known standard such as automotive SPICE.	A, B, C
Validation of the software tool	Build validation suite to check tool meets requirements of safety related developer.	C, D

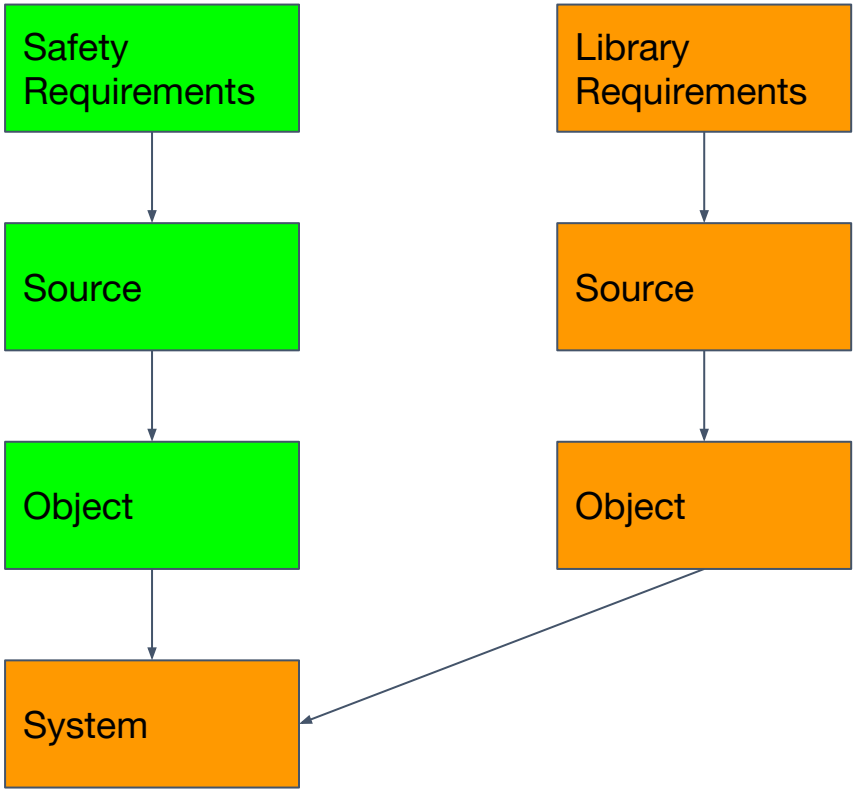
Qualification and certification

- The functional safety standards may require that a development tool or software component is qualified
- Safety-related system developer is responsible for the qualification
- Toolchain developer can obtain a certificate from a trusted third-party that can aid the qualification process
- A certificate states tool is “suitable for use” for “safety standard” provided the use case falls into these boundaries
- Obtaining Certificate often a commercial requirement



What about the libraries?

- Development tools like a compiler transform source code to binary code
  - Source code derived from safety-related requirements
- Library code is not derived from safety-related developers source code yet runs on the safety-related system
  - Can’t use toolchain qualification
- Options include
  - Safety Element out of Context
  - Qualification of software components



Toolchain Provider Assistance

- What can a provider do to assist qualification process?
- Arm's experience with qualification of Arm Compiler
  - Arm Compiler 5 entirely proprietary
  - Arm Compiler 6 includes open source components
- Lessons learned from process

What can a toolchain developer do to help?

- Toolchain provider cannot do the qualification for the safety-related system developer, but can provide vital information
  - Can obtain certification for common use cases
  - Subset of features that are covered by functional safety information
  - Provide extra information about using the tool for the common use cases
  - Provide known fault and mitigations
  - Can provide verification and validation results for standards conformance

Fault id 123
Fault description ...
Fault mitigation ...

Arm Compiler 5

- Proprietary toolchain targeting the embedded market
- Documented feature set
- Single issue tracking system for defects and requirements
- Long observed policy of citing issue number in commit messages
- Stable, and documentable, development and release process
- All committers Arm staff members that can be trained in functional safety
- Large internal test suite accumulated over time
- Use of externally recognized conformance test suites

Strategy:

- Focus on evaluation of development process
- Publish subset of defects
- Publish externally recognized test results
- Achieve certification from recognized 3rd party

ARM Compiler Safety Related Release

- A long-term support and maintenance release of toolchain chosen as safety-related release
  - Post-release only bug-fixes in updates
- Supported feature set emphasizes bare-metal C programming
  - Libraries excluded from the supported feature set
- Additional information for safety-related developers provided as a kit
  - Safety manual
  - Development process document
  - C90/C99 conformance test suite results
  - Defect report
- Qualification kit releases synchronized with toolchain release
  - Defect report updated with all safety-related defects known to affect any of the safety related releases

Cathedral and the Bazaar

- A tool being or containing open-source is not a problem in itself
- Difficult to map a stereotypical open-source development to the v-model
  - Specification of tool feature set
  - Comprehensive documentation
  - Traceability from requirements to source code
  - Number of committers and information known about them
  - Completeness of issue tracking data for known faults
  - Assessors not familiar with open-source development

Arm Compiler 6

- Compiler for Arm Compiler 6 armclang is derived from open source components from the llvm community
- Regular release of the toolchain tracks tip of trunk
  - Upstream commits vastly outnumber downstream
- Upstream follows a loose development process, with many committers interacting via mailing-list, bugzilla and IRC
- Clang features not fully documented

Strategy changes:

- Focus on validation of the software tool
- Snapshot upstream release
- Document supported feature set
- All future changes under Arm safety-related development process

Lessons learned

- Snapshotting a development tool, validating it then controlling updates is a viable approach to tool qualification
  - Build a cathedral on top of the bazaar
- Getting supported feature set right is most important up front decision
  - The more you support, the more documentation and testing you are going to provide
- Largest long-term post-release cost is writing up defects
  - A good write up can take as much time as a fix
- Automate as much of the information tracking as you can
  - Auto-generate documents from test and issue tracking systems
- Run-time libraries need a different approach
  - Less prior art about using open source for component qualification
- Get proof of concept approval from a certification authority as early as possible

Concluding thoughts for Linaro

- Safety-related developers have to justify their tool usage
  - Frequently requiring expensive tool qualification
- Open-source tools can be qualified by:
  - Snapshotting, documenting and validating a specific release
  - Providing tightly controlled updates post snapshot
  - Providing a known-faults service for the release
- Providing information for qualification is expensive
  - Up-front costs of producing documentation
  - Ongoing cost of service
  - Tracking contributions
- Can Linaro act as a coordinator for members to share the costs?
  - A service?
  - Members only?
  - Contribute back to community?

Linaro connect
San Francisco 2017

Thank You

#SFO17

SFO17 keynotes and videos on: connect.linaro.org

For further information: www.linaro.org