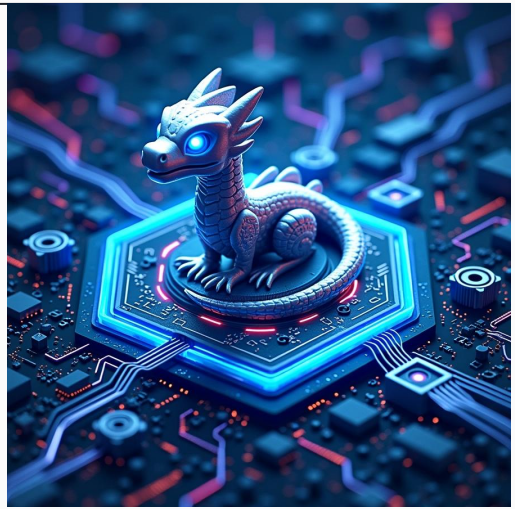
LLVM Qualification WG



Sync-up meeting #1

Focus: *Specifications*

July 2025

Contents

01
Welcome & Intros

02
Code of Conduct

03
Collaboration Format

04
Early Topics & Activities

05
Today’s Focus Discussion: Requirements & Traceability

06
Next Steps?

Welcome: Why We're Here

1

Shared interest in increasing trust, ensuring quality, and enabling *qualification* of LLVM components and LLVM-based toolchains for safety-critical *and* non-safety-critical developments

2

Shared vision of building a foundation for reusable and community-driven *qualification* artifacts upstream, through collaboration and transparency

Introductions

Quick round: name, affiliation, interest in this group.

One word you associate with *qualification*.

Code of Conduct

Short reminder (based on LLVM CoC): Respect, constructive discourse, inclusion

We listen to understand
We share, not sell
If issues arise, contact moderators (e.g. Wendi for now)

Link: Code of Conduct (for review by all members before publishing)

Collaboration Format

Support async input and real-time coordination equally

01

Discussion

LLVM Discourse under “Community”

Discord channel #fusa-qual-wg

02

Meetings

Monthly sync-ups
+ async follow-up via posts

Meeting minutes (Discourse thread)

03

Docs

GitHub (version-controlled, open to all)

Shared templates, references, process experiments

04

Other tools

Any suggestions?

01

Early Topics & Activities

Modular Breakdown

Split the qualification work across key components

01	02	03
Frontend	**Middle-end**	**Backend**
Language-specific aspects (e.g. Clang)	Optimizations, analysis, IR transformations	Target-specific code generation

Initial focus on Clang

C/C++ as a starting point
Broadly used in safety-critical applications

Other relevant tools and commands

`llvm-cov`, `clang-tidy`, `Autocheck`, etc

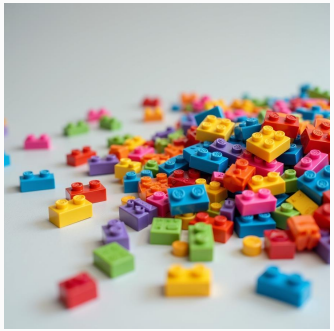
Pieces of the confidence-in-use puzzle

- **Specifications**
  Expected behavior, constraints
- **Testing**
  Coverage, regression, etc.
- **Formal Verification**
  Translation validation, equivalence checking
- **Sanitizer Use**
  e.g., RTSan, ASan, UBSan
- **Runtime Diagnostics**
  Error and Warning messages
- **Source Code Quality**
  Coding guidelines, autochecks
- **Known Issues Analysis**
  Workarounds and countermeasures
- **Documentation**
  User & safety manual, release notes, configuration options
- **Other pieces**
  Any other suggestions?

02

First piece: Specifications

How to define clear and structured specifications as a foundation for testing, traceability, and reusable qualification artifacts



01

Balancing specification clarity with upstream feasibility

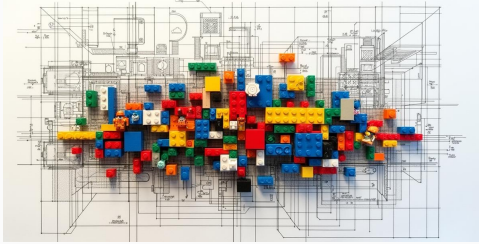
02

Enabling traceability between specifications and verification means

03

Defining scope, ownership, and sustainability

Facing the challenges



	Challenge	Key considerations
01	How to define and structure requirements in a way that supports qualification, without placing undue burden on the upstream OSS community	Minimal overhead for contributors Leverage existing artifacts where possible Align with community practices (e.g. formatting, annotations) Use a free and OSS Requirements Management Tool? e.g. (1) and (2)
02	Establish mechanisms for bidirectional traceability between specs and verification/validation elements (e.g. tests, static checks, sanitizer results)	How and where to annotate tests with requirements Manual vs automated traceability Prototyping with a well-scoped target
03	Determine what should be specified, by whom, and how it can be maintained across language targets, components, and organizations	Different needs for Frontend, Middlend, Backend, Linker Specs vs implementations (e.g. language standards vs behavior) Toolchain coverage, automation, and potential for a Safety Manual Reusability by downstream users (e.g. adding their own constraints)

Next Steps?

We're here!	July 2025	Specifications	Let's continue the conversation on Discourse & Discord
	Aug–Sep 2025	Summary of ideas?	Proposed solutions
	Oct–Nov 2025	Initial prototyping?	Test with a well-scoped target

Open Discussion

Let's discuss any doubts or concerns.
If something comes up later, contact the Working Group on Discourse or Discord.