Safety-Critical Rust Adoption

Adoption of Safety-Critical Rust

This survey is being administered by the Rust Safety-Critical Consortium, a part of the Rust Foundation. The goal of the consortium is to advance adoption of the Rust programming language in industries like automotive, aerospace, industrial, medical, and others. The consortium provides a forum for collaboration between safety-critical software developers, Rust community members, and software development tooling providers.

The goal of the survey is to understand the state of adoption of the Rust language in these industries and identify gaps in tooling, community, ecosystem, or language features. Any and all engineers or managers working in safety-critical industries are invited to respond and share your thoughts!

To view the activities of the Rust Safety-Critical Consortium or get involved visit us at our GitHub repo:
https://github.com/rustfoundation/safety-critical-rust-consortium

Prelude

Do you work in a safety critical industry? (Automotive, aerospace, medical, robotics, etc)

○ Yes

○ No

What industry do you work in?

You can select multiple options.

□ Automotive

□ Aerospace

□ Medical

□ Industrial

□ Robotics

□ Defense

□ Nuclear

□ Rail

□ Other

What is the size of the company that you work for?

○ < 10

○ 10 - 49

○ 50 - 249

○ 250 - 1000

○ > 1000

Approximately how many people actively write Rust at your company?

○ < 10

○ 10 - 19

○ 20 - 49

○ 50 - 200

○ > 200

What is your primary role or responsibility related to software development or safety within your organization?

You can select multiple options.

□ Software Engineer

□ Safety Engineer

□ Quality Assurance Engineer

□ Project Manager

□ R&D Engineer

□ Engineering Management

□ Startup Owner/Founder

□ CTO

□ CEO

□ Other

Languages

What language(s) do you use in your safety critical role?

You can select multiple options.

□ Rust

□ C

□ C++

□ Ada/Spark

□ Python

□ Java

□ Other

Do you currently use Rust?

You can select multiple options.

□ Yes, in my safety critical role

□ Yes, in non-safety critical production

□ Yes, as a hobby/in my free time

□ No

Would you be interested in using Rust in your safety-critical role?

○ Yes, I already am

○ Yes, but not I'm not yet using it

○ No, I'm not interested in using Rust

○ Not Sure Yet

Rust Users

What were your/your company's primary reasons to switch to Rust?

You can select multiple options.

- □ Memory safety
- □ Tooling (cargo, crates.io, clippy, etc)
- □ Language features (type system, matching, etc)
- □ Specific library/crate/framework
- □ Hiring advantages (interest in Rust/quality of Rust engineers)
- □ Regulatory requirement
- □ Cybersecurity advantages
- □ Other

Interested in Rust

What advantages are you interested in using Rust for?

You can select multiple options.

□ Memory safety

□ Tooling (cargo, crates.io, clippy, etc)

□ Language features (type system, matching, etc)

□ Specific library/crate/framework

□ Hiring advantages (interest in Rust/quality of Rust engineers)

□ Regulatory requirement

□ Cybersecurity advantages

□ Other

What are the primary blockers to using Rust in your safety critical role?

You can select multiple options.

□ Lack of tooling

□ Standards

□ Hiring engineers

□ Hardware support

□ Ecosystem (libraries, drivers, etc)

□ Regulatory hurdles

□ Legacy codebase

□ Training

□ Other

Not Interested in Rust

What are the primary disadvantages to using Rust in your safety-critical role?

You can select multiple options.

□ Doesn't offer any/enough advantage over existing languages

□ Too difficult to learn

□ Doesn't support my hardware target(s)

□ Not certifiable

□ Not enough hirable Rust engineers

□ Doesn't integrate with my existing codebase

□ Software supply chain concern (open-source)

□ Don't know enough about it

□ Other

Tooling

What types of safety critical code tools does your work require?

You can select multiple options.

- □ Certified compiler
- □ Formal verification
- □ Code coverage analysis
- □ Code-requirements traceability
- □ Static analysis
- □ Code metrics - cyclomatic complexity
- □ Linting tools (naming contenvtions, style guides, etc)
- □ Automated testing (fuzz, prop-based, etc)
- □ Other

Standards/Guidelines

What standard(s) do you work with in your safety critical work?

You can select multiple options.

□ ISO-26262

□ DO-178

□ IEC-61508

□ IEC-62304

□ Other

What levels of ISO-26262 do you work with?

You can select multiple options.

□ QM

□ ASIL-A

□ ASIL-B

□ ASIL-C

□ ASIL-D

What levels of IEC-61508 do you work with?

You can select multiple options.

□ SIL-1

□ SIL-2

□ SIL-3

□ SIL-4

What levels of DO-178 do you work with?

You can select multiple options.

□ DAL A

□ DAL B

□ DAL C

□ DAL D

□ DAL E

What levels of IEC-62304 do you work with?

You can select multiple options.

□ Class A

□ Class B

□ Class C

What coding guidelines do you work with in your safety critical work?

You can select multiple options.

□ MISRA

□ AEC

□ Other

Hardware/Environments

To what environments do you deploy your safety critical code?

You can select multiple options.

- □ Cloud
- □ Native applications
- □ Embedded Linux
- □ Embedded with a hypervisor (RTOS, event-driven architecture, etc)
- □ Bare metal embedded
- □ Other

What chip architectures do you work with?

You can select multiple options.

- □ ARM Cortex
- □ Infineon Tricore
- □ x86-x64
- □ PowerPC
- □ Renesas
- □ RISC-V
- □ Other

Open Ended Feedback

What kind of support or resources from the Rust community or the Safety-Critical Rust Consortium would be most helpful in enabling Rust adoption in your industry?

What steps would your organization need to take to seriously consider or adopt Rust for safety-critical projects?

Are there "best-in-class" libraries or solutions in another language which have features lacking within the Rust ecosystem for your use cases?

Are there any libraries that are not written in Rust that you would like to use in the Rust ecosystem?