Lighthouse SIG

WG SIG Annual Updates 2026

Feb 12th, 16:20-16:45 UTC

Philipp Ahmann, ETAS GmbH & the SIG community

ELISA
Enabling Linux in Safety Applications

Aerospace · Automotive · Linux Features
OS Engineering Process · Safety Architecture · Systems · Tools
Lighthouse · Space Grade Linux

Lighthouse OSS – Setting the scene

- Traditional software development methods were **built for proprietary software**
- No **existing quality standard** matches established open source development practices
- But OSS Communities have demonstrated **mature processes and consistent quality**
- OSS is a viable choice for regulated industries, **but there is a lack of a matching standard**

Photo by William Bout on Unsplash

Goal

The goal of this project is to evaluate and document established open source development best practices

&

to provide an assessment guide for the user to rate the quality of open source projects.

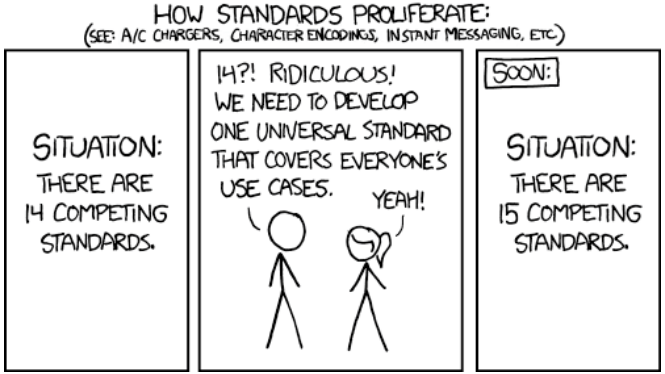
Photo by Paul Skorupskas on Unsplash

Open Source Good Practices - Yet Another Standard

- Standards are based on v-model
- No OSS project is strictly following v-model. 😱

- ASPICE or CMMI are main argumentation for quality management in Automotive
- Safety standards require a quality baseline to support safety claims.

- No existing quality standard matches established open source development practices!
  (code-centric, CI driven and agile focus)



https://xkcd.com/927/

Open Source Good Practices – Overview

Phases in 2025

1. Determination of status quo ✅
2. Definition of practices ✅
3. Assessment of pilot projects (ongoing)

Additional phase for 2026

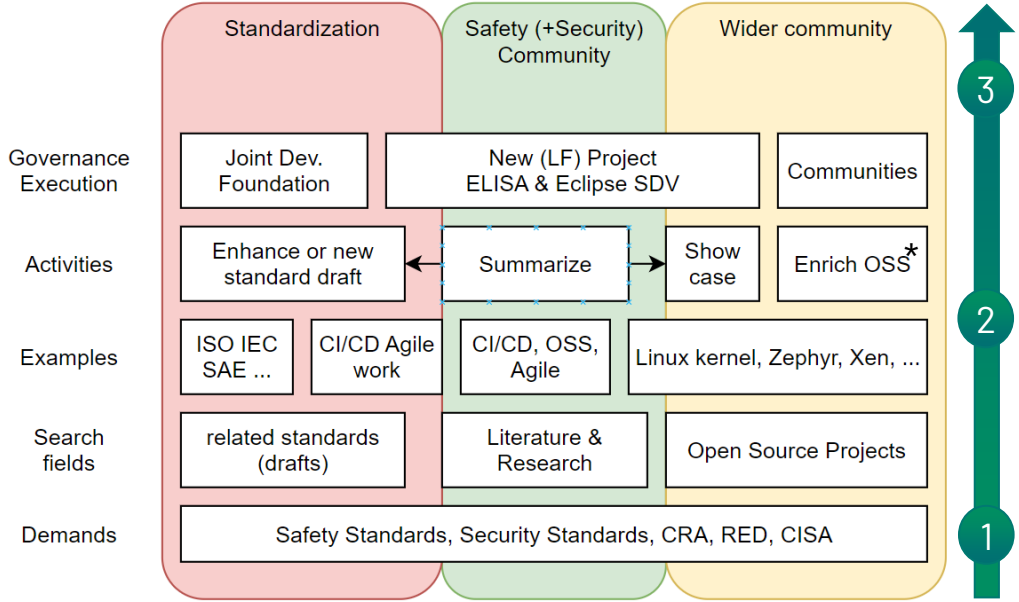
4. Prepare towards a standard

Pillars in 2025

1. Standards
2. Literature & Research
3. Open Source projects

Additional pillar in 2026

4. Existing best practices records & frameworks



** Can include existing frameworks*

OpenSSF · LFX Insights · CHAOSS · Joint Development Foundation · OpenChain

ELISA
Enabling Linux in Safety Applications

Achievements 2025

Aerospace · Automotive · Linux Features · OS Engineering Process · Safety Architecture · Systems · Tools · Lighthouse · Space Grade Linux

Photo by Michael Skok on Unsplash

What we did...

- Survey conducted together with LF Europe to check the demand
- Presented the survey result
- Started the SIG based on survey feedback

- Interacted with Eclipse Automotive Process SIG:
  Prevent duplicated work, confusion, and conflicting interest.

- Prepared template to evaluate projects based on e.g. literature information (more in following slides)

Challenge the idea and collect feedback

- Feb: UnConference session at CHAOSSCon
- May: Presentation of concept at ELISA WS in Lund to collect feedback
- Jun: Presentation at OSS NA:
  Developing a Community-Driven Standard for OSS Quality
- Nov: First presentation to community at ELISA WS in Munich by Simone
- Dec: Presentation at Eclipse SDV Meetup in Japan around LLVM and Lighthouse work by Wendi

ELISA
Enabling Linux in Safety Applications

Impressions

Aerospace · Automotive · Linux Features · OS Engineering Process · Safety Architecture · Systems · Tools · Lighthouse · Space Grade Linux

Start with a standard to come to categories

- Based on the quality aspects of ISO 9001
- Areas:
  - Risk
  - Stakeholder
  - Community
  - Resource
  - Compliance
  - People/team
  - Program
  - Quality Engineering
- For all QM areas, defined their aspects and create requirements (→ See it here)

Relevant aspects of the QMS		
QM Area	**Aspect**	**Description**
Risk	Risk Management processes	Based on the types / categories of ri define the steps to manage the risk,
Risk	Business Controls and Internal Audit	The goal is to define internal mecha
Stakeholder	Stakeholder Satisfaction, Feedback (Measured metrics)	The goal is to define criteria to meas
Stakeholder	Interactions & Progress reporting	The goal is to define the means to ir
Stakeholder	Quality control	The goal is to define acceptance crit
Stakeholder	Communication	The goal is to communicate achieve
Community	Involvement of organizations	The goal is to describe how to build
Resource	-	The goal is to determine and provide implementation, maintenance and c
Compliance	Processes & products evaluated against industry & government standards	The goal is to ensure the processes
Compliance	Contribution to standards, communities, regulations & laws	The goal is to define who and how w member list, process for submitting /
People/team	Roles & responsibilities assignment	The goal is to define a profile descri with role, assigned responsibilities,
People/team	Management of the skills including training, coaching & proven competencies	The goal is to check the competenc areas for improvements and the list program)
People/team	Recording platform	The goal is to automate the process
People/team	Learning product & service	The goal is to provide "learning" as
Program	Strategy	The goal is to define a guideline to n
Program	Project management: Backlog grooming, development / stabilization sprints	The goal is to detail the way to mana Definition of Done (DoD)...
Program	Problem Resolution & Process improvement	The goal is to define escalation proc future improvements)
Quality Engineering	Quality engineering strategy & evidence traceability	The goal is to define a strategy to er before the release of a product
Quality Engineering	Management of artifacts	The goal is to define the expected b and to set aligned internal processes Examples: documentation (template baseline acceptance / sign-off
Quality Engineering	Testing & gating strategy	The goal is to define a gudeline to d anomalies...
Quality Engineering	Tooling (including metrics & reporting)	The goal is to define metrics & thres
Quality Engineering	Release Readiness Report	The goal is to define the way to repo changes, verification and validation compliance with applicable standard

Look into existing evidences in literature

Examples of literature which was checked for Open Source practices:

- 2017 - Evaluation of open-source OS for Safety-Critical Applications (P. Berntsson)
- 2020 OSS Development Process Systematic Review (B. Napoleao, F. Petrillo, S. Hallé)
- 2025 - Towards a more sustainable and secure software tooling in free/libre open source software environments (S. Tatschner)

Mapped in both directions to create an evaluation template to rate OSS projects practices

Check: Process robustness & Evidence Confidence

Step 1: Assess "Process Robustness" (horizontal dimension)
How formalized and consistent is this practice? To what extent is this practice explicitly defined, owned, and embedded in the project's governance?
- Is the practice explicitly defined?
- Are expectations clearly stated?
- Are scope and applicability unambiguous?
- Is ownership or authority identifiable?
- Is the rule stable over time?

This is about rules, expectations, structure. It does not mean that people follow it, that it is enforced, or that evidence exists. That's Step 2's job.

Level	Description	Observable indicators	Comments
0 - Not defined	The practice does not exist as an explicit rule or expectation	No written rule, policy, or guidance No mention in governance, documentation, or contribution materials Any occurrence would rely entirely on individual initiative	Example: The practice isn't mentioned in the project's README, wiki, or workflow No shared understanding Any occurrence is accidental The practice is invisible
1 - Implicit	The practice exists as an unwritten norm or shared understanding	Expectations are implied through discussions or community culture Knowledge is transmitted socially (e.g., "this is how we usually do it") No authoritative reference that defines the practice New contributors would struggle to discover it independently	Stage of pre-formalization of a practice The practice is emergent, but not yet established in the rules by the community The practice has a name (even if informal) Contributors reference it in discussion Decisions occasionally invoke it
2 - Explicitly defined	The practice is formally described and publicly accessible	Written description exists (README, CONTRIBUTING, governance doc, wiki) Scope and intent of the practice are stated Applies consistently to a defined set of contributors or artifacts Language is normative ("must", "should", "expected")	This is the first level where the practice is "established" At this level, the practice is written down, contributors can look it up, and expectations are explicit But the practice still depends on specific individuals remembering to apply it, informal coordination, tribal knowledge filling the gaps Enough to state intent Typical failure modes: A key maintainer leaves, then the rule quietly stops being applied New contributors misunderstand scope or exceptions The rule exists, but is inconsistently interpreted
3 - Institutionalized	The practice is embedded into the project's governance and workflows	Defined ownership or authority (e.g., maintainers, reviewers, WG) Clear entry points in lifecycle documents (e.g., PR process, release rules) Exceptions, escalation paths, or decision rights are described Changes to the practice follow a defined governance process	At this level, the practice is embedded into governance, there is explicit ownership, the rule is connected to decision points (PRs, releases, approvals) Needed to survive growth and turnover No requirement yet that it is enforced or proven, only that the institutional hooks exist
4 - Systematized (or Optimized? or Audited?)	The practice is fully integrated as a first-class element of the project system	Process continuously improved and monitored Treated as part of the project's "way of working" Explicitly connected to other practices Versioned, maintained, and reviewed like code or specifications Designed to scale with project growth and contributor turnover	Most OSS projects might stop at writing the rulebook (Level 2) Here, the rules evolve through a defined process when the game changes Needed to survive time The rule is not a special case, but infrastructure

Step 2: Assess "Evidence Confidence" (vertical dimension)
Given the practice as defined (Step 1), how confident are we that it is actually followed in reality?
Can we see that enforcement actually happens?
Think evidence existence, observable traces, evidence quality, practice verifiably applied, observable application, trace shows rule execution
This step assumes the practice exists (or not) per Step 1, but does not care how good the rule is (part of Step 1's work?)

Level	Description	Observable indicators	
0 - No evidence	Nothing observable supports that the practice is applied	No evidence available No public trace, unverifiable claims No references, documents, or community trace	Example: The practice is not verified
1 - Minimal evidence	Isolated examples, self-assertions, or unstructured confirmation exist	One example, informal confirmation Minimal evidence: may rely on self-reporting or single instances Mailing list mentions, occasional manual actions, sparse documentation	Example: The practice was done once or twice, but it's not repeatable or verified
2 - Consistent evidence	Repeated traces over time exist, with a consistent observable behavior that might still be dependent on human interpretation	Multiple verifiable examples across time Moderate confidence: evidence from multiple data points, but not independently verified Repeated activity in commits/issues; clear policy or CONTRIBUTING.md; not yet automated	Example: The project regularly performs the practice and the info isn't outdated
3 - Comprehensive evidence	Evidence is structured and traceable, with (partial) automation that helps reduce observer bias	Evidence from multiple independent data sources (issues, commits, CI logs) High confidence: evidence is traceable, up-to-date, and partly automated Bot enforcement, dashboards, or automated gating; evidence stored in CI/CD log	Example: The practice is executed consistently and monitored (e.g., required review checks)
4 - Independent or automated verification	Evidence generation or validation does not depend on project insiders, and confidence in this evidence does not rely on trust in individuals	Evidence generated or checked automatically; external party can verify Very high confidence: evidence publicly verifiable or 3rd-party validated Automated reporting, metrics dashboards, external audits, or badges (e.g., OpenSSF Scorecard) External observers reach the same conclusion	Example: Evidence collected continuously; external observers can confirm compliance

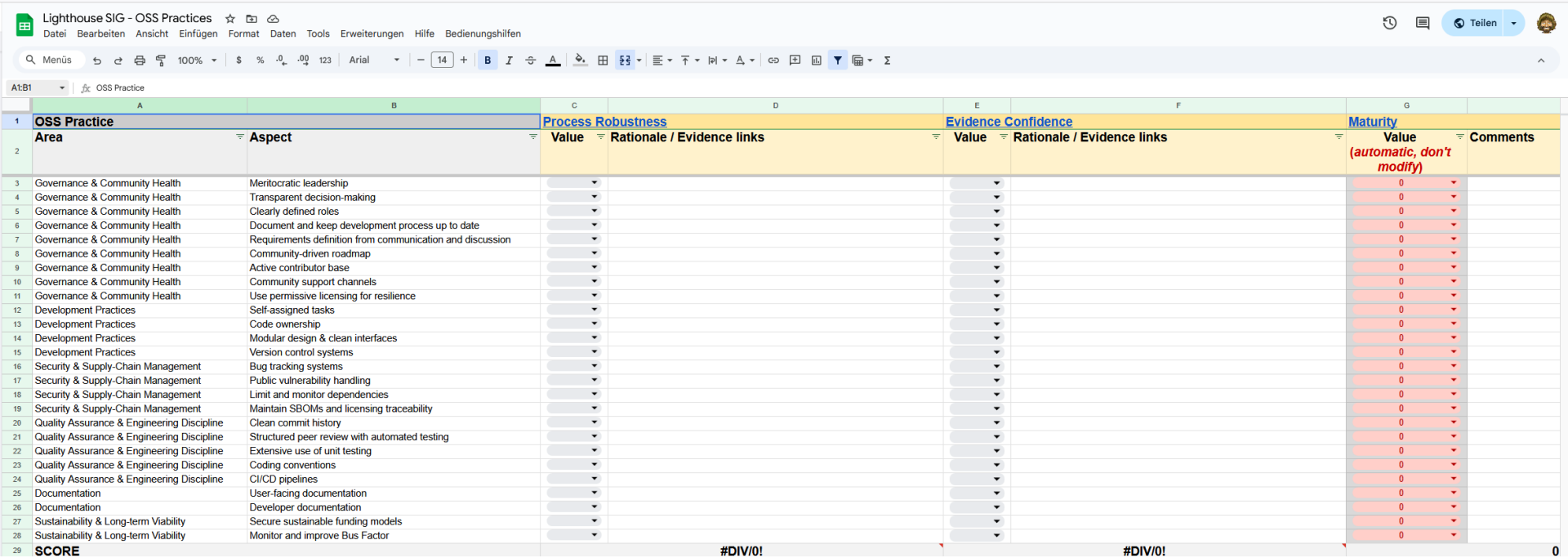
Check: Process robustness & Evidence Confidence

Process robustness	Evidence confidence
0 - no process	no evidence
1 - implicit	minimal evidence
2 - explicit	consistent evidence
3 - institutional	comprehensive evidence
4 – systematic & cont. improved	independent or automated verification

Every level must be verifiable through public artifacts.

(First draft is currently under refinement. Try to look what is existing elsewhere.)

Template


Lighthouse SIG - OSS Practices

Datei Bearbeiten Ansicht Einfügen Format Daten Tools Erweiterungen Hilfe Bedienungshilfen

Teilen

Menüs 100% $ % .0 .00 123 Arial 14

A1:B1 OSS Practice

	A	B	C	D	E	F	G	
1	**OSS Practice**		**Process Robustness**		**Evidence Confidence**		**Maturity**	
2	**Area**	**Aspect**	**Value**	**Rationale / Evidence links**	**Value**	**Rationale / Evidence links**	**Value** *(automatic, don't modify)*	**Comments**
3	Governance & Community Health	Meritocratic leadership					0	
4	Governance & Community Health	Transparent decision-making					0	
5	Governance & Community Health	Clearly defined roles					0	
6	Governance & Community Health	Document and keep development process up to date					0	
7	Governance & Community Health	Requirements definition from communication and discussion					0	
8	Governance & Community Health	Community-driven roadmap					0	
9	Governance & Community Health	Active contributor base					0	
10	Governance & Community Health	Community support channels					0	
11	Governance & Community Health	Use permissive licensing for resilience					0	
12	Development Practices	Self-assigned tasks					0	
13	Development Practices	Code ownership					0	
14	Development Practices	Modular design & clean interfaces					0	
15	Development Practices	Version control systems					0	
16	Security & Supply-Chain Management	Bug tracking systems					0	
17	Security & Supply-Chain Management	Public vulnerability handling					0	
18	Security & Supply-Chain Management	Limit and monitor dependencies					0	
19	Security & Supply-Chain Management	Maintain SBOMs and licensing traceability					0	
20	Quality Assurance & Engineering Discipline	Clean commit history					0	
21	Quality Assurance & Engineering Discipline	Structured peer review with automated testing					0	
22	Quality Assurance & Engineering Discipline	Extensive use of unit testing					0	
23	Quality Assurance & Engineering Discipline	Coding conventions					0	
24	Quality Assurance & Engineering Discipline	CI/CD pipelines					0	
25	Documentation	User-facing documentation					0	
26	Documentation	Developer documentation					0	
27	Sustainability & Long-term Viability	Secure sustainable funding models					0	
28	Sustainability & Long-term Viability	Monitor and improve Bus Factor					0	
29	**SCORE**			**#DIV/0!**		**#DIV/0!**		**0**

Filling... (currently looking into Xen, LLVM, Yocto...)

	A	B	C	D	E	F	G	H
1	**OSS Practice**		**Process Robustness**		**Evidence Confidence**		**Maturity**	
2	**Area**	**Aspect**	**Value**	**Rationale / Evidence links**	**Value**	**Rationale / Evidence links**	**Value (*automatic, don't modify*)**	**Comments**
3	Governance & Community Health	Meritocratic leadership	3	https://xenproject.org/about/governance/. Enforces by policy in gouvernance. Practice is institutionalized hence.	2	https://github.com/xen-project/xen/commit/075e7e35dbd9338d28f44f84b1fb573e521d51f1 https://xenbits.xenproject.org/governance/core-governance.html#elections Repeated activity in commits that nomination procces is followd as well on the vote via mailing list https://lists.xen.org/archives/html/xen-devel/2016-01/msg00503.html	2	Leadership roles are earned by merit and activity, as outlined in governance docs.
4	Governance & Community Health	Transparent decision-making	3	https://xenbits.xenproject.org/governance/core-governance.html#decision-making-conflict-resolution-role-nominations-and-elections	3	Repeated in docuemnts and commits	3	Voting and decision logs are public per governance documents. There are defined ways for technical, non-technical, sub projects and complete project affecting topics and voting procedures. (lazy consensus)
5	Governance & Community Health	Clearly defined roles	3	https://xenbits.xenproject.org/governance/core-governance.html#xen-project-wide-roles https://xenbits.xenproject.org/governance/core-governance.html#project-team-roles	3	Meassureable in Maintainers file	3	Roles such as community manager, advisory board, comitters, maintainers and project leads defined
6	Governance & Community Health	Document and keep development process up to date	3	https://xenbits.xenproject.org/governance/core-governance.html#change-history https://wiki.xenproject.org/wiki/Submitting_Xen_Project_Patches https://xenbits.xen.org/docs/latest/			3	See version history of government page on first link. Some parts of development are defined in wiki some in rst docs, parts of the wiki are outdated, eg.g bug handling process
	Governance & Community Health	Requirements definition from communication and discussion	3	https://xenproject.org/blog/xen-summit-2025-find-your-place-in-the-future-of-			3	Design sessions during xen-summits can be used for community

• • •

20								
21	Quality Assurance & Engineering Discipline	Structured peer review with automated testing	3	https://xenbits.xenproject.org/gitweb/?p=xen.git;a=blob;f=.gitlab-ci.yml;h=7974ac4e82bc278a4b3b7ea32684cde183d4d289;hb=HEAD https://xenbits.xenproject.org/governance/core-governance.html#project-team-roles			3	Yes, test via gitlab runners, review via committers and maintainers
22	Quality Assurance & Engineering Discipline	Extensive use of unit testing	2	https://gitlab.com/xen-project/xen/-/tree/staging/tools/tests			2	Existing but not extensive
23	Quality Assurance & Engineering Discipline	Coding conventions	4	https://xenbits.xen.org/docs/latest/misra/rules.html			4	Yes usage and test for MISRA via CI
24	Quality Assurance & Engineering Discipline	CI/CD pipelines	3	https://xenproject.org/contribute/ci/status/ https://xenproject.org/contribute/ci/ https://design-sessions.xenproject.org/uid/discussion/disc_fiY7dpMIeN2FASo6mMHp/view			3	All is runners on gitlab, but issues are known see design session wrt to speed and timeouts. Tests on x86, ARM32, ARM64, RISC-V, PowerPC for Dom0 boot, DomU launch, Dom0less, suspend/resume, PCI passthrough Static analysis with cppcheck, MISRA C, and more On QEMU emulated tests and real bare metal boards
25	Documentation	User-facing documentation	3	https://xenbits.xen.org/docs/latest/			3	Available
26	Documentation	Developer documentation	3	https://xenbits.xen.org/docs/latest/ https://design-sessions.xenproject.org/uid/discussion/disc_Jp2T9yq2I3sH00teD0CT/view			3	Wiki exists but sites are partially outdated and shall be deprecated. Docs for HV developers exists, but as written on the overview page needs sorting.
27	Sustainability & Long-term Viability	Secure sustainable funding models	4	https://xenproject.org/about/become-a-member/			4	Linux Foundation, robust advisory board support
28	Sustainability & Long-term Viability	Monitor and improve Bus Factor	3	https://xenproject.org/blog/a-recap-of-the-xen-project-developer-and-design-summit-community-health-development-trends-coding-changes-and-more/ https://insights.linuxfoundation.org/project/xen/contributors?timeRange=past365days&start=2024-10-20&end=2025-10-20&auth=success&widget=retention			3	Diverse maintainers and active efforts to reduce risk. Community Health is watched, and i.e. in 2018 improvements were done for better collaboration of x86 development. Retetion above 50% as per LFX insights, but 51% of contributions from 2 companies
29	**SCORE**		**3.24**		**3**		**3.076923077**	

ELISA
Enabling Linux in Safety Applications

Plans 2026

Aerospace · Automotive · Linux Features · OS Engineering Process · Safety Architecture · Systems · Tools · Lighthouse · Space Grade Linux

Photo by Aleksandr Popov on Unsplash

Enhance the framework & prepare for a standard

- Continue active tasks
  - Pending actions in spreadsheet like maturity level
  - Extended review of projects to get trends
- Add another pillar for evaluation: “existing records of best practices”
  - Compare frameworks & activities like OpenSSF Scorecard, LFX insights, Trustable, CHAOSS

  - Reach out to existing communities and “best practices” maintainer
- Prepare the input to know what is needed to create a standard
  - Reach out to JDF, OpenChain, domain experts
- Work with OSEP WG on a definition what “quality” actually means (& other terms)

ELISA
Enabling Linux in
Safety Applications

Join our efforts!

Aerospace · Automotive · Linux Features · OS Engineering Process · Safety Architecture · Systems · Tools · Lighthouse · Space Grade Linux

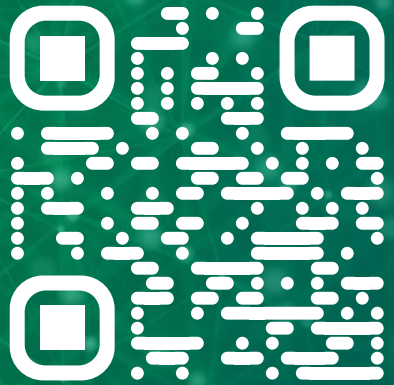
Get Involved

- Lighthouse SIG meets every other Friday 12:00 PM
  https://elisa.tech/community/meetings/
- *Mailing list*
  https://lists.elisa.tech/g/lighthouse
- *Discord*
  https://discord.gg/TnWZpDUu
- *Contributions in GitHub welcome!*
  https://github.com/elisa-tech/lighthouse-oss/
  https://github.com/elisa-tech/lighthouse-oss/wiki

Photo by Hannah Busing on Unsplash

ELISA
Enabling Linux in Safety Applications

Thank you!



https://elisa.tech

Aerospace · Automotive · Linux Features · OS Engineering Process · Safety Architecture · Systems · Tools · Lighthouse · Space Grade Linux

JOIN THE COMMUNITY

ELISA
Enabling Linux in Safety Applications

Our infrastructure and tools are open by default, so jump in and introduce yourself, ask questions and share ideas. Please consider this your invitation to participate.

Subscribe to Mailing Lists

Join Community Meetings

Contribute to Tools and Docs on GitHub

Participate in Working Groups

Attend Events