ELISA
Enabling Linux in Safety Applications

WORKSHOP

ELISA Workshop
Munich, Germany

November 18-20, 2025
Co-hosted with Red Hat

Licensing of Workshop Results

All work created during the workshop is licensed under Creative Commons Attribution 4.0 International (CC-BY-4.0) [https://creativecommons.org/licenses/by/4.0/] by default, or under another suitable open-source license, e.g., GPL-2.0 for kernel code contributions.

You are free to:

- Share — copy and redistribute the material in any medium or format
- Adapt — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Best Practices in Open Source and Standards – Evaluation of Example Projects

Simone Weiß

ELISA Enabling Linux in Safety Applications WORKSHOP

Introduction

Who am I?

1. Simone Weiß
2. Since…

   ~ 1 year more activ in ELISA

   ~ the creation +1 with the Lighthouse OSS WG

   ~ this month with Linutronix

Agenda

1. Recap: Goals WG Lighthouse OSS
2. Approach WG Lighthouse OSS
3. Template or determining the Status quo
4. Evaluation of Projects
5. Challenges and future work

What is the number one open-source development best practice that should be followed ?

Goals WG Lighthouse OSS

Background:

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

Photo by William Bout on Unsplash

Recap: Goals WG Lighthouse OSS

Goals:

- Evaluate and document established open-source development best practices
- Provide an assessment guide for users to rate the quality of open-source projects

Discussion Point: should we also provide tools and foster continuous improvement and community-driven quality assurance?

Approach WG Lighthouse OSS

We covered parts from here till now

Determine the Status
- Which OSS development practices exist?
- Which standards exist?
- Do they define practices that increase Quality

Definition of Practices
- Can we re-use pratices from standards or OSS development practices?
- Evaluate OSS project pilots to define practices
- Additional OSS projects to validate?

Creation of a framework for evaluation
- Are the defined practices suitable for an automated framework?
- Can we come to a common framework impl. across (some) OSS projects?

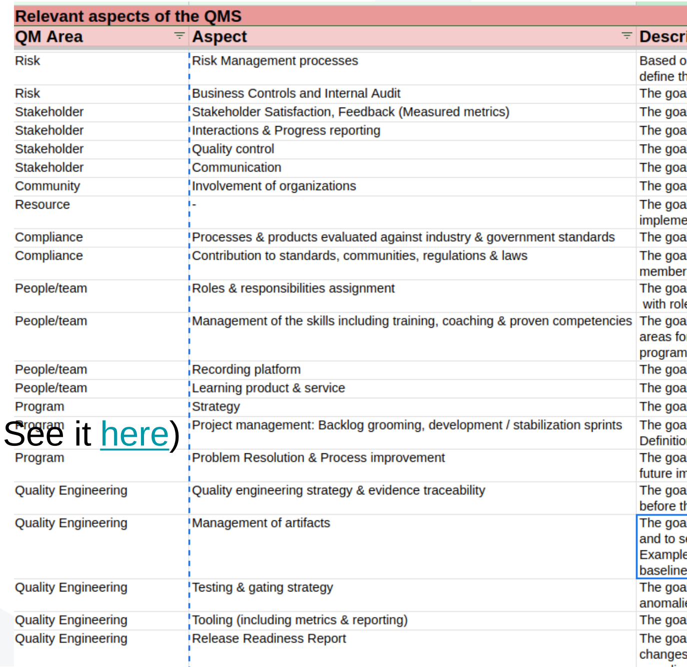
ELISA Enabling Linux in Safety Applications

WORKSHOP

Template or determining the Status Quo

WG considered QMS framework:

- 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	Descri
Risk	Risk Management processes	Based o define th
Risk	Business Controls and Internal Audit	The goa
Stakeholder	Stakeholder Satisfaction, Feedback (Measured metrics)	The goa
Stakeholder	Interactions & Progress reporting	The goa
Stakeholder	Quality control	The goa
Stakeholder	Communication	The goa
Community	Involvement of organizations	The goa
Resource	-	The goa impleme
Compliance	Processes & products evaluated against industry & government standards	The goa
Compliance	Contribution to standards, communities, regulations & laws	The goa member
People/team	Roles & responsibilities assignment	The goa with role
People/team	Management of the skills including training, coaching & proven competencies	The goa areas fo program
People/team	Recording platform	The goa
People/team	Learning product & service	The goa
Program	Strategy	The goa
Program	Project management: Backlog grooming, development / stabilization sprints	The goa Definitio
Program	Problem Resolution & Process improvement	The goa future im
Quality Engineering	Quality engineering strategy & evidence traceability	The goa before th
Quality Engineering	Management of artifacts	The goa and to s Example baseline
Quality Engineering	Testing & gating strategy	The goa anomalie
Quality Engineering	Tooling (including metrics & reporting)	The goa
Quality Engineering	Release Readiness Report	The goa changes

ELISA Enabling Linux in Safety Applications WORKSHOP

Template or determining the Status Quo

Additionally literature 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 softwaretooling in free/libre open source software environments (S. Tatschner)

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

Discussion Point: Any large gaps of practices we did not consider?

ELISA Enabling Linux in Safety Applications

WORKSHOP

First example: xen

ELISA Enabling Linux in Safety Applications WORKSHOP

Evaluation of xen as an example project

Governance & Community Health:

- Meritocratic leadership
- Transparent decision-making
- Clearly defined roles
- Document and keep development process up to date
- Requirements definition from communication and discussion
- Community-driven roadmap
- Active contributor base
- Community support channels
- Use permissive licensing for resilience

Evaluation of xen as an example project

Development Practices:

- Self-assigned tasks
- Code ownership
- Modular design & clean interfaces
- Version control systems

Security & Supply-Chain Management:

- Bug tracking systems
- Public vulnerability handling
- Limit and monitor dependencies
- Maintain SBOMs and licensing traceability

Evaluation of xen as an example project

Quality Assurance & Engineering Discipline:

- Clean commit history
- Structured peer review with automated testing
- Extensive use of unit testing
- Coding conventions
- CI/CD pipelines

Documentation:

- User-facing documentation
- Developer documentation

Sustainability & Long-term Viability:

- Secure sustainable funding models
- Monitor and improve Bus Factor

ELISA Enabling Linux in Safety Applications

WORKSHOP

Second example: yocto

ELISA Enabling Linux in Safety Applications

WORKSHOP

Evaluation of yocto as an example project

Governance & Community Health:

- Meritocratic leadership
- Transparent decision-making
- Clearly defined roles
- Document and keep development process up to date
- Requirements definition from communication and discussion
- Community-driven roadmap
- Active contributor base
- Community support channels
- Use permissive licensing for resilience

ELISA Enabling Linux in Safety Applications

WORKSHOP

Evaluation of yocto as an example project

Development Practices:

- Self-assigned tasks
- Code ownership
- Modular design & clean interfaces
- Version control systems

Security & Supply-Chain Management:

- Bug tracking systems
- Public vulnerability handling
- Limit and monitor dependencies
- Maintain SBOMs and licensing traceability

Evaluation of yocto as an example project

Quality Assurance & Engineering Discipline:

- Clean commit history
- Structured peer review with automated testing
- Extensive use of unit testing
- Coding conventions
- CI/CD pipelines

Documentation:

- User-facing documentation
- Developer documentation

Sustainability & Long-term Viability:

- Secure sustainable funding models
- Monitor and improve Bus Factor

Any patterns?

Repeating patterns between yocto and xen

- **Hard to evaluate:**
  - Requirements
  - Community roadmap
  - Limit and monitor dependencies
- **There might be very different views on**: Modular design & clean interfaces
- **Need to define what extensive is**: Extensive use of unit testing
- **Not easy to access (transparency):** Secure sustainable funding models
- **Always a challenge:** Monitor and improve Bus Factor, Missing public artefacts?

ELISA Enabling Linux in Safety Applications WORKSHOP

Differences between yocto and xen

- xen
  - contradictions in docs
    - Developer documentation
    - Bug tracking
  - Old information after restructering not clearly marked
- Yocto:
  - less formal on:
    - Transparent decision-making
    - Clearly defined roles

Further notes during evaluation

- General: what to evaluate when. E.g.: xen project vs xen hypervisor, sometimes fuzzy

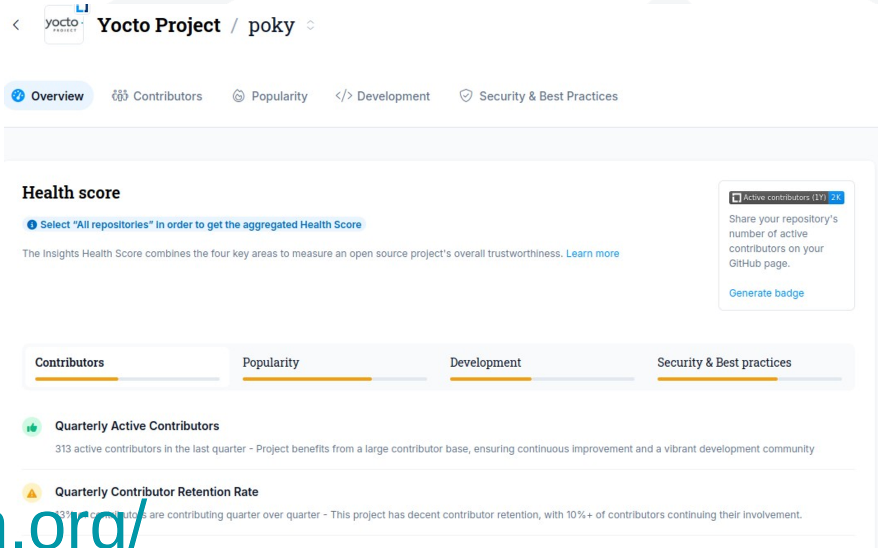
- High overlap with The Open Source Project Security (OSPS) Baseline from openssf: e.g.:
  - OSPS-LE-02.0* maps to “Use of permissive Licensing”
  - OSPS-DO-01.0* maps to “Document and keep development process up to date”
  - …
  - Should we perform a mapping to this and other metrics?

ELISA Enabling Linux in Safety Applications WORKSHOP

Further notes during evaluation

LFX Insights:

- Automatic evaluate:
  - Active contributor base
  - Monitor and improve Bus Factor
  - Code Review Activities


Yocto Project / poky

Overview · Contributors · Popularity · Development · Security & Best Practices

Health score

Select "All repositories" in order to get the aggregated Health Score

The Insights Health Score combines the four key areas to measure an open source project's overall trustworthiness. Learn more

Active contributors (1Y) 2K

Share your repository's number of active contributors on your GitHub page.

Generate badge

Contributors · Popularity · Development · Security & Best practices

Quarterly Active Contributors

313 active contributors in the last quarter - Project benefits from a large contributor base, ensuring continuous improvement and a vibrant development community

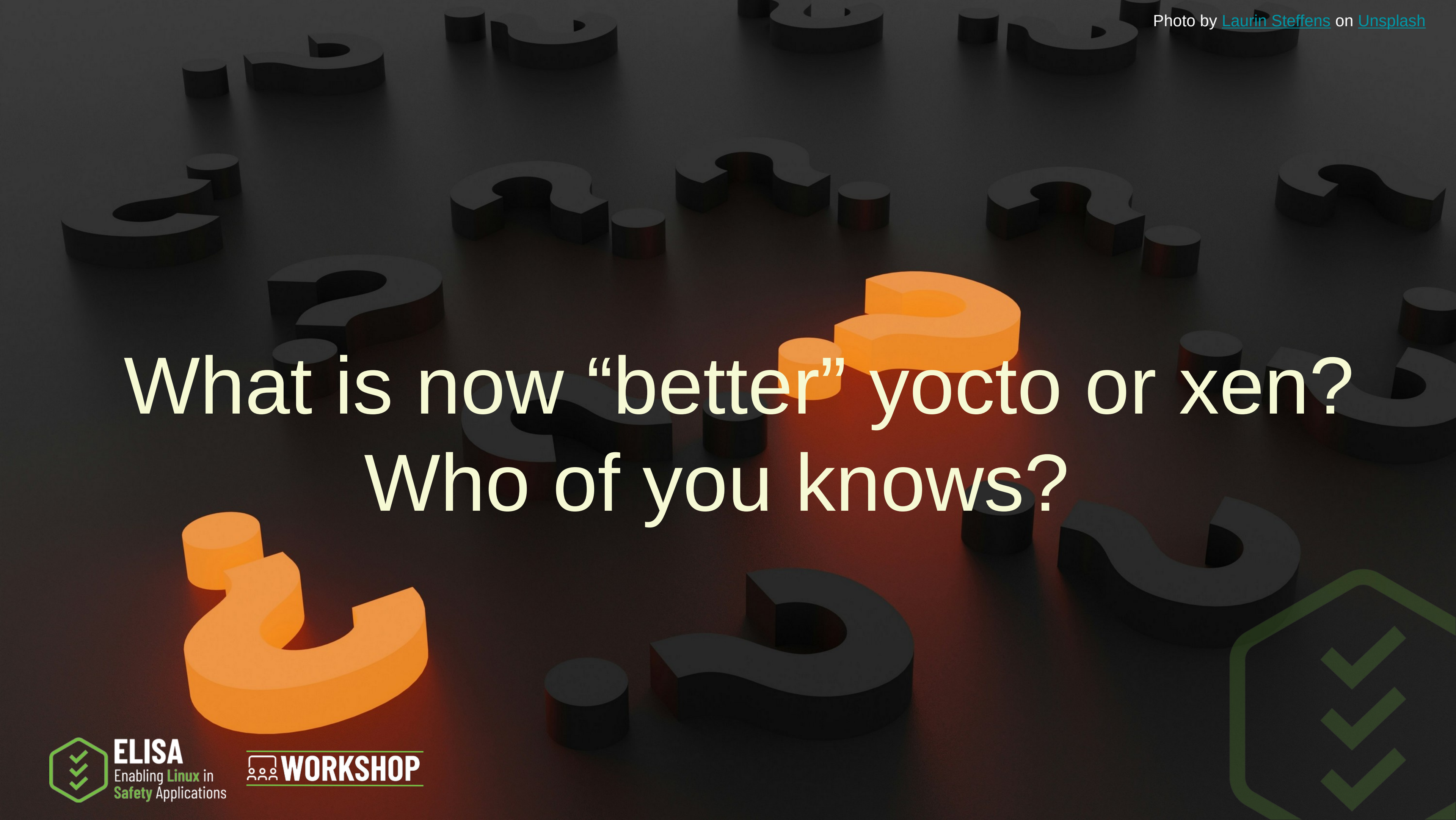
Quarterly Contributor Retention Rate

13% of contributors are contributing quarter over quarter - This project has decent contributor retention, with 10%+ of contributors continuing their involvement.

https://insights.linuxfoundation.org/

ELISA Enabling Linux in Safety Applications · WORKSHOP

Photo by Laurin Steffens on Unsplash



What is now “better” yocto or xen?
Who of you knows?

ELISA Enabling Linux in Safety Applications

WORKSHOP

Future work

There is now additional work for a maturity scale (proposed – Thanks Wendi!)

Process robustness x Evidence confidence

0 no process No evidence

1 - not systematically Minimal evidence

2 - process described Consistent evidence

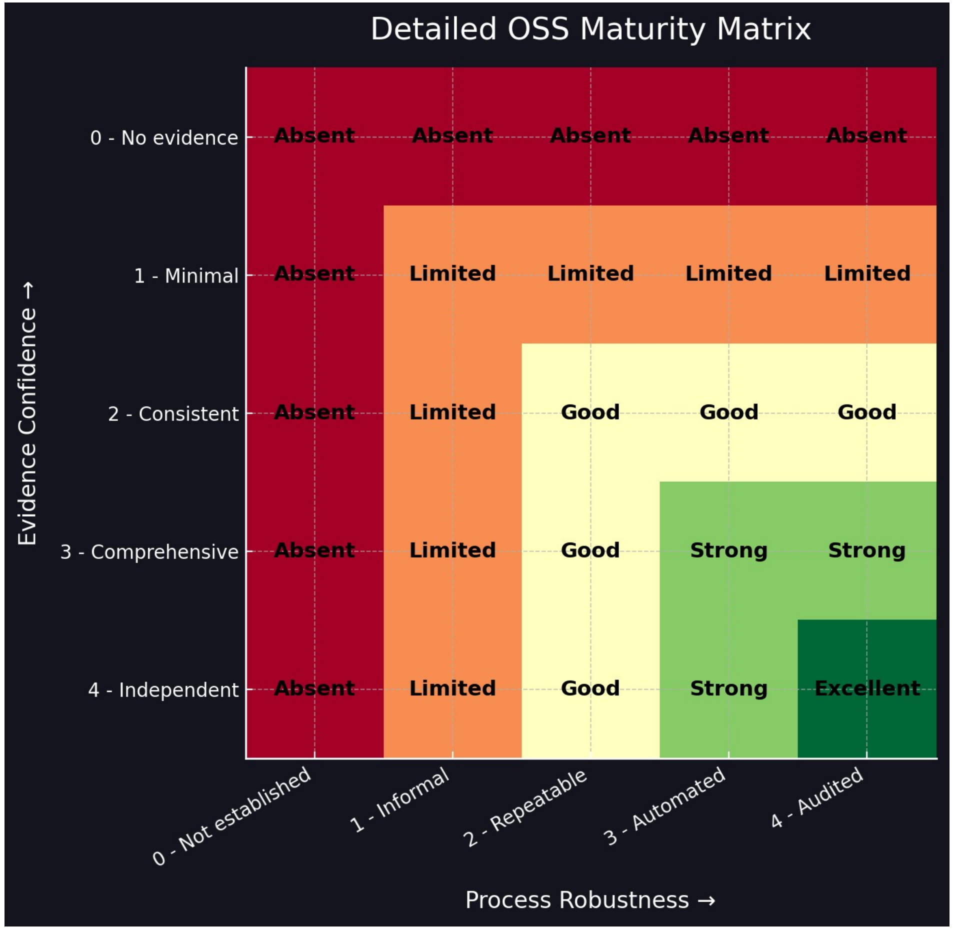
3 - process enforced Comprehensive evidence

4 - continuously improved and monitored Independent or automated verification

Every level must be verifiable through public artefacts

ELISA Enabling Linux in Safety Applications

WORKSHOP



Detailed OSS Maturity Matrix

Evidence Confidence → / Process Robustness →	0 - Not established	1 - Informal	2 - Repeatable	3 - Automated	4 - Audited
0 - No evidence	Absent	Absent	Absent	Absent	Absent
1 - Minimal	Absent	Limited	Limited	Limited	Limited
2 - Consistent	Absent	Limited	Good	Good	Good
3 - Comprehensive	Absent	Limited	Good	Strong	Strong
4 - Independent	Absent	Limited	Good	Strong	Excellent

Discussion Point:

- Is there a further scale needed that weighs the single practices to reach an overall score?
- How important are the practices when compared to each other: e.g. “Active contributor base” vs “Maintain SBOMs and licensing traceability”?

Approach – Future work

We covered parts from here till now

Determine the Status

- Which OSS development practices exist? **More**
- Which standards exist?
- Do they define practices that increase Quality?

Definition of Practices

- Can we re-use pratices from standards or OSS development practices?
- Evaluate OSS project pilots to define practices
- Additional OSS projects to validate? **Which**

Creation of a framework for evaluation

- Are the defined practices suitable for an automated framework?
- Can we come to a common framework impl. across (some) OSS projects?

ELISA Enabling Linux in Safety Applications WORKSHOP

Photo by Glenn Carstens-Peters on Unsplash

Thoughts?

ELISA
Enabling Linux in Safety Applications

WORKSHOP

Discussion Points

Are we on the right track?

Are we missing any existing efforts?

What more practices to consider in the template?

Thoughts on the maturity scale?

Remember the goal to evaluate and provide a guide: Should we not also foster continuous improvement and community-driven quality assurance?

Thank you

ELISA Enabling Linux in Safety Applications

WORKSHOP