



**RI.
SE**

JOHAN LINÅKER, RISE

Health Check-ups on Open Source Software Projects

Managing Risks while Promoting (Re)use

Open Source Software Health

- An Open Source Software project's capability to stay viable and maintained over time without interruption or weakening



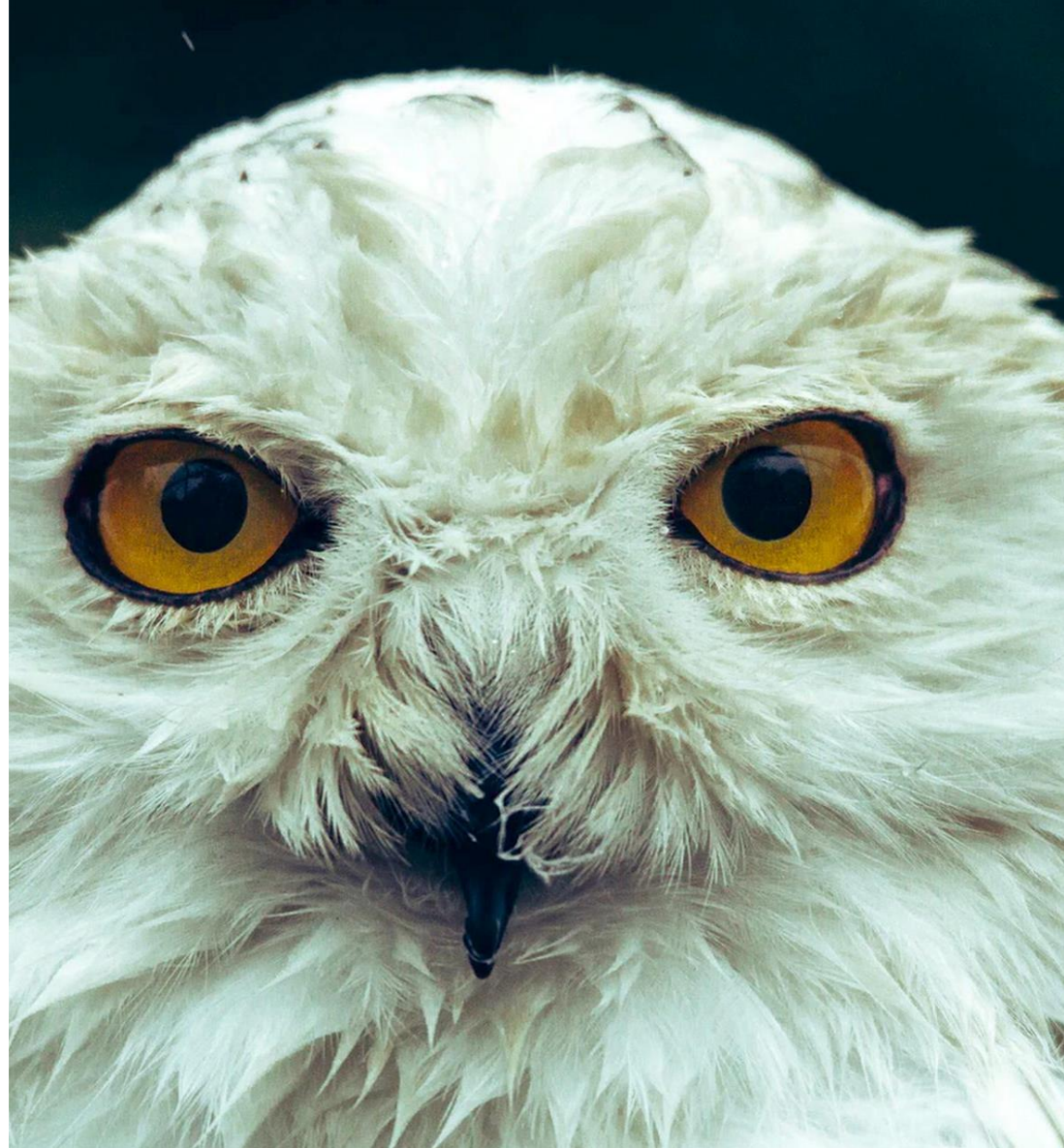
Open Source Software Health

- Productivity: There is an active development of the project
- Robustness: The development is open and spread out on several (independent) individuals
- Openness: Users of the project can influence and contribute to the development of the project



Linus' law

- "Given enough eyeballs, all bugs are shallow"
- Requires that enough eyeballs actually reaches the codebase
- Free-riding, for both good and bad



Brain-time as a Common Pool Resource

- “Brain-time” and maintenance effort is subtractable
- Maintainers are humans, not robots
 - Burnout, changed family or working conditions
- Companies must adapt to stay competitive
 - Refactorization, new products, changed business model

@johanlinaker

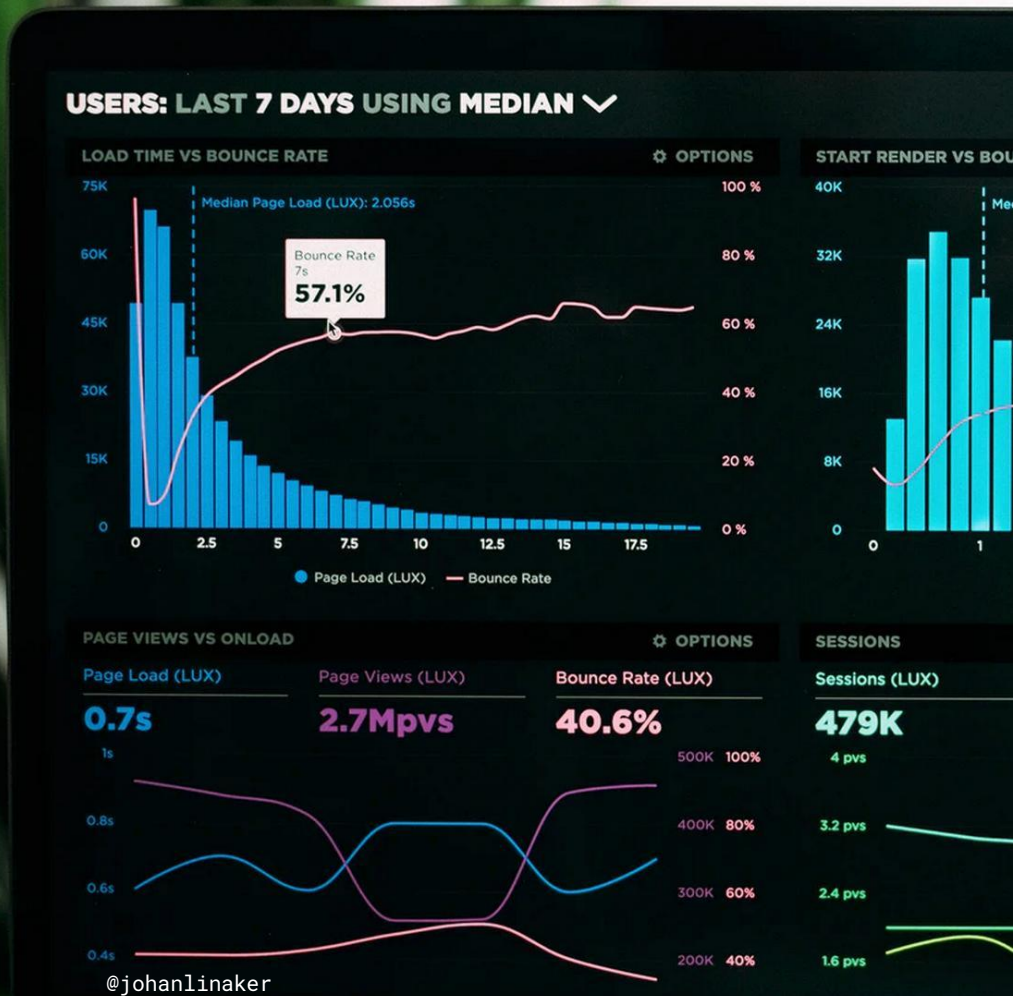




- An MD asks questions and uses tools at disposal to examine the patient, identify symptoms, arrive at a diagnosis, and prescribe a treatment.



- A developer asks questions and uses tools at disposal to examine the OSS project, identify symptoms, arrive at a sourcing decision, and potential actions for community engagement.



Health and Security Management for OSS (HASMOSS)

- 2021-23 Vinnova-funded R&D-project
- RISE, Scania, Debricked, Addalot
- Goals:
 - Enable health analysis at intake and acquisition of OSS, and ongoing consumption
 - Enable sourcing decisions and proactive health improving measures



What can we find in literature?

- 146 studies
- 107 characteristics (+associated metrics)
- Divided over 15 themes
- Supplementary material:
<https://doi.org/10.6084/m9.figshare.20137175>
- Paper:
<https://www.ri.se/sites/default/files/2022-09/opensym2022-6%20%281%29.pdf>



@johanlinaker

Photo by Jared Craig | <https://unsplash.com/photos/HH4WBGNyltc>





What does experts say?

- 17 interviews with industry and community experts
- 4 areas critical to classify projects, impacting what metrics to prioritize and how tough
- 21 areas of complementary metrics considering
 - Community productivity, and stability
 - Orchestration
 - Production process and outputs

@johanlinaker

Photo by Austin Distel | <https://unsplash.com/photos/smiling-man-reading-book-while-holding-mug-4r72LPPf41k>

RI
SE

Project Classifier

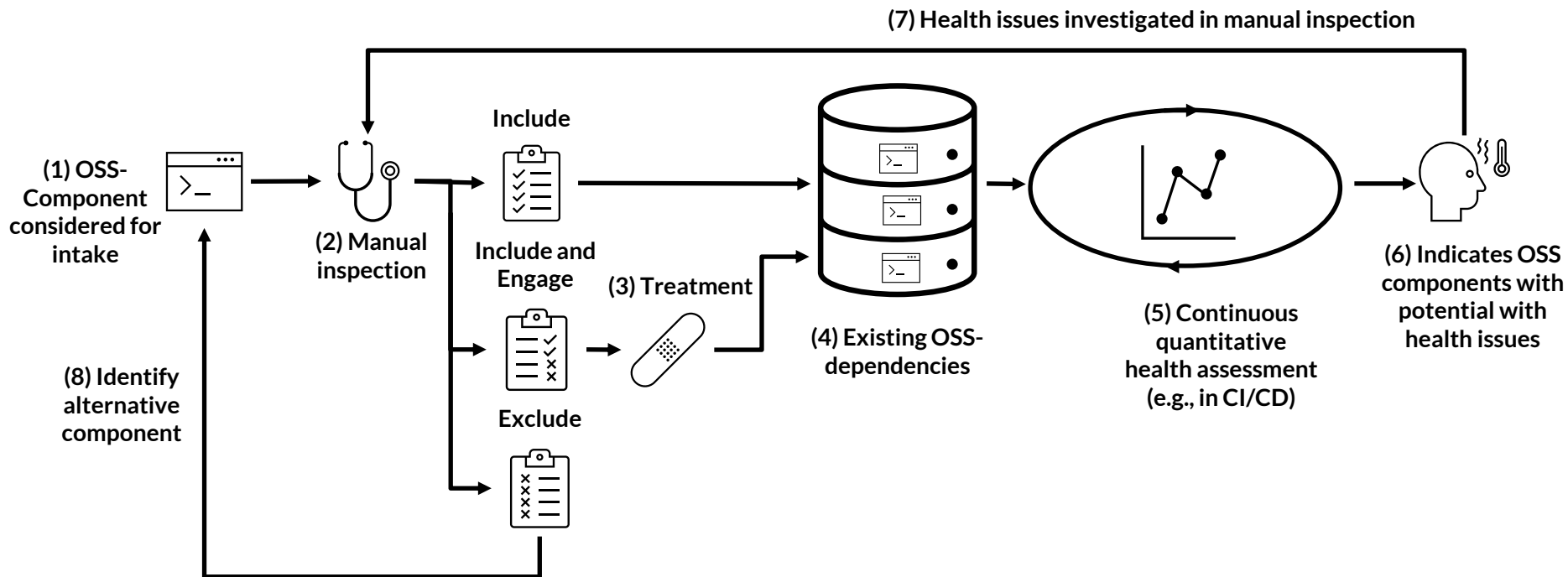
- Life-cycle stage
 - 1) inception, 2) growth, 3) stabilization, and 4) decline
- Project Complexity
 - scope, size, and technical complexity of the codebase
- Governance concentration
 - impact on the project's openness to input and external influence on decisions and transparency of discussions
- Strategic Importance
 - importance of the OSS project from a business and technical perspective



Going from theory to practice

- What:
 - Lower risk of OSS used and considered in the intake process
- How:
 - Set up an intake and screening process for new and existing OSS dependencies
 - Monitor health and make proactive decisions on sourcing options and community engagement
- Key requirements:
 - Decentralized, self-managed process
 - Enable but don't overburden developers
 - Enable follow-up and actionable insights

Semi-automating the health-check process



Human Infrastructure in support of a sustainable maintenance

- Maintainer resources
 - Managing social expectations and peer-pressure
 - Balancing of workload with capacity
 - Finding time through funding
 - Work-life balance and prioritization
- Community resources
 - Embracing the episodic contributors
 - Mitigating toxicity
 - Promoting inclusiveness
 - Managing impact of project characteristics
 - Low-cost contributor support
 - Marketing and outreach
 - Distributing knowledge

Resource funding

- Full-time employment dedicated to projects
- Partially-dedicated employment
- Entrepreneurship, a common but risky endeavor
- Sponsorship, a diverse and limited source of income



