JOHAN LINÅKER, RISE

Hälsa och säkerhet hos öppen programvara vid utveckling och anskaffning

Hälsa hos öppna programvaruprojekt

- Det öppna programvaruprojektets förmåga att hålla sig livskraftigt och underhållet över tid utan avbrott eller försämring

Photo by Jan Piatkowski | https://unsplash.com/photos/eopz9bkwROs

Hälsa hos öppna programvaruprojekt

- Produktivitet: Aktiv utveckling och underhåll av projektet
- Robusthet: Utvecklingen sker öppet och är utspritt över ett flertal (oberoende) individer
- Öppenhet: Projektets användare kan bidra och utöva inflytande över projektets utveckling och underhåll

Photo by Markus Frieauff | https://unsplash.com/photos/lJ0KiXl4uys

Linus lag

- ”Given enough eyeballs, all bugs are shallow”
- Kräver att tillräckligt manga ögon når koden (och gör något)
- Free-riding både positivt och negativt

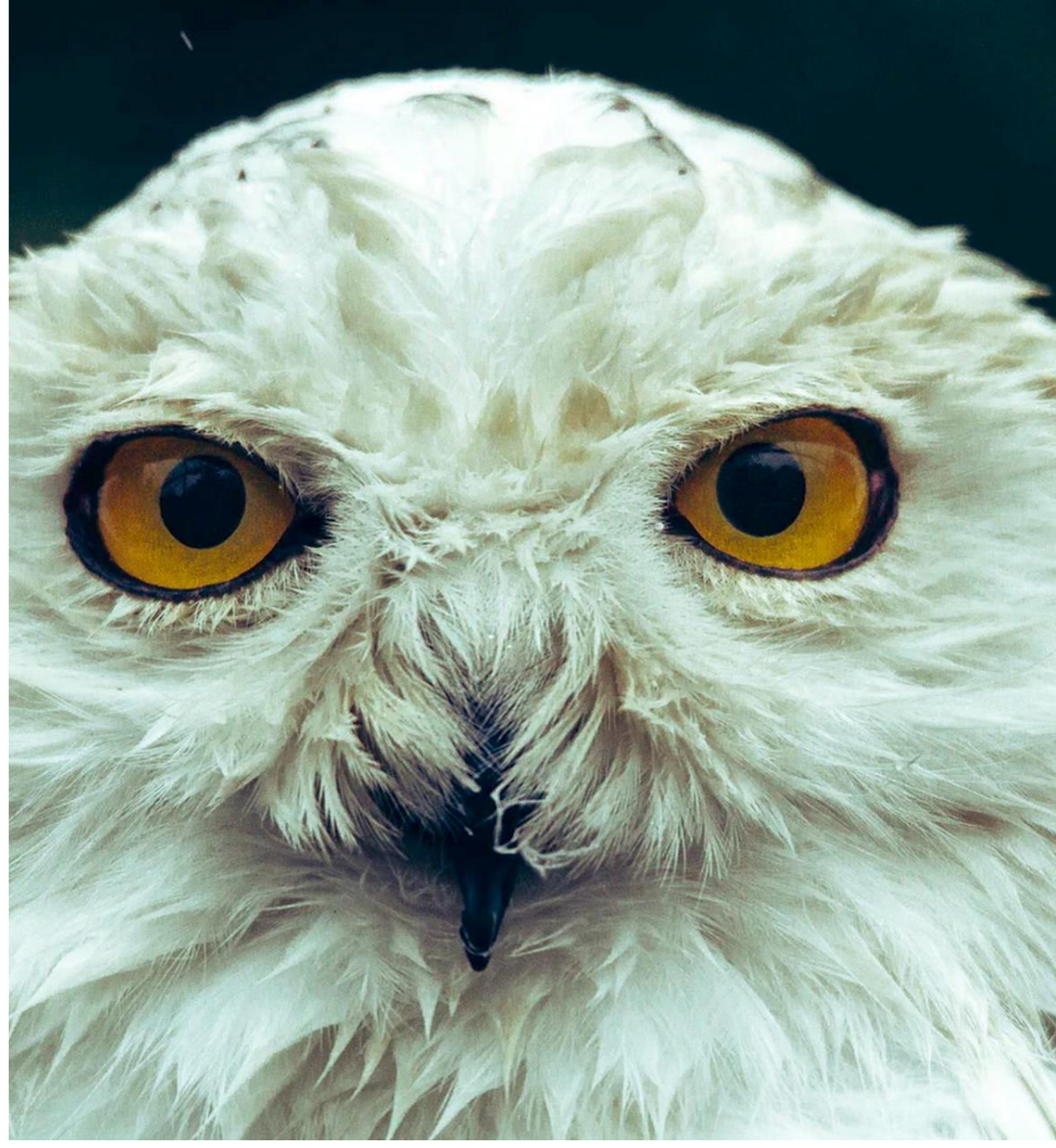


Photo by Frida Bredesen | https://unsplash.com/photos/c1fFv08N7qE

Utvecklares tid som allmäning

- Utvecklares tid och resurser är subtraherbar och kan ta slut
- Kodförvaltare är människor, inte robotar
  - Utbränning, förändrade familjeförhållanden, nytt jobb
- Företag måste anpassa sig för att förbli konkurrenskraftiga
  - Refaktorisering, nya produkter, ändrade affärsmodeller

Photo by NeONBRAND | https://unsplash.com/photos/KYxXMTpTzek

Sårbarheter ett växande problem

- Omkring 86% av granskade kodbaser under 2025 innehöll sårbarheter i öppen källkod, varav 81% hade allvarliga eller kritiska risker.
- Andelen högrisk-sårbarheter i öppen källkod ökade kraftigt från 48% av projekten 2022 till 74% år 2023, och fortsätter att öka kraftigt
- Livslängden på sårbarheter har ökat med 95% från 2017 till 2024
- 80% av beroenden förblir ouppdaterade i över ett år trots att säkrare versioner finns

- En läkare ställer frågor och använder tillgängliga instrument för att undersöka patienter, identifiera symptom och diagnos, samt bestämma en lämplig behandling.

- Utvecklare ställer frågor och använder tillgängliga verktyg för att undersöka öppna programvaruprojekt, identifiera symptom och diagnos, och bestämma lämplig intervention, som att bidra eller välja alternativ lösning.

@johanlinaker

RI.SE

Utveckling

- I utvecklingen återanvänds öppna programvarukomponenter dagligen
- Större svenskt företag tog in 60 000+ nya komponenter och versioner därav förg. år
- Kräver kontinuerlig granskning, både vid intag och över redan befintliga beroenden
- Kostnad och komplexitet måste vara låg

Anskaffning

- Vid anskaffning av större system, nya eller befintliga krävs mer omfattande analys
- Central del i början av anskaffningsprocessen vid utvärdering och jämförelse av öppna alternativ sinsemellan och med proprietära

RI.SE

Health and Security Management for OSS (HASMOSS)

- 2021-23 Vinnova-finansierat projekt
- RISE, Scania, Debricked, Addalot
- Mål:
  - Möjliggöra hälsoanalys vid intag och anskaffning av öppen programvara
  - Möjliggöra sourcing-beslut och proaktiva steg för att förbättra hälsan, och adressera risken den kan utgöra

RI.SE

Photo by Luke Chesser | https://unsplash.com/photos/JKUTrJ4vK00

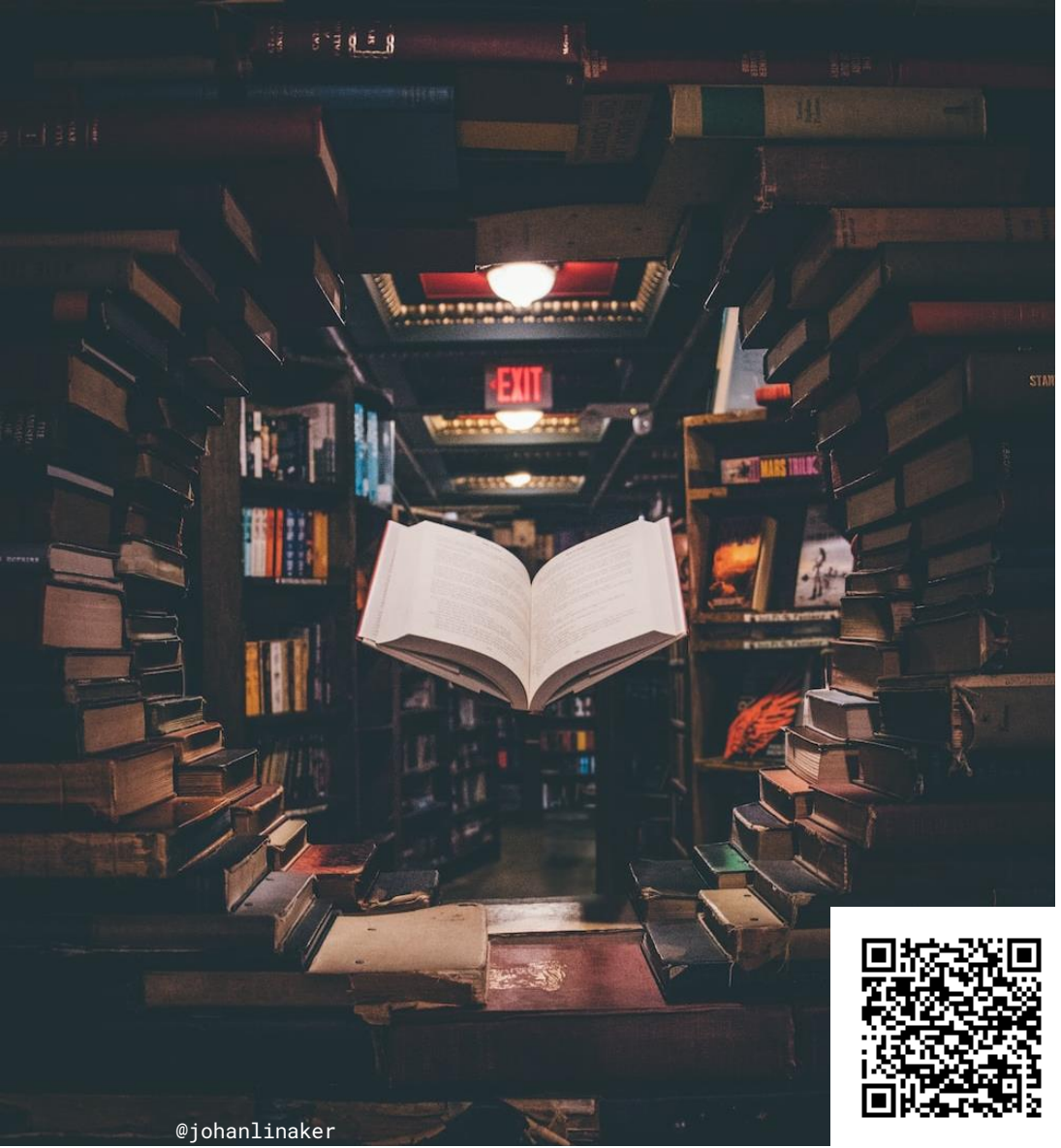


Photo by Jaredd Craig | https://unsplash.com/photos/HH4WBGNyltc

Vad säger literaturen?

- 146 studier
- 107 sätt att diagnostisera hälsan på
- Spänt över 15 teman
- Forskningsdata (open access): https://doi.org/10.6084/m9.figshare.20137175
- Artikel (open access): https://www.ri.se/sites/default/files/2022-09/opensym2022-6%20%281%29.pdf

RI.SE

Photo by Austin Distel | https://unsplash.com/photos/smiling-man-reading-book-while-holding-mug-4r72LPFh4Ik

Vad sager experterna?

- 17 intervjuer med exporter från industri och den öppna programvaruekosystemet
- 4 kritiska områden att ta hänsyn till vid bedömning och jämförelse av projekt, ex. avseende vilka indikatorer som bör användas och hur
- 21 områden med kompletterande indikatorer omfattande:
  - Projektens produktivitet och stabilitet
  - Projektens styrning och koordinering
  - Projektens utvecklingsprocess och leveranser

RI.SE

Se även befintliga resurser

- CHAOSS (Community Health Analytics for Open Source Software)
  - https://chaoss.community/
- OpenSSF (Open Source Security Foundation – Se scorecard och best practices badge
  - https://openssf.org/
- Franska states kriterier (baserade på CHAOSS)
  - https://github.com/codegouvfr/floss-criteria/tree/main/criteria/Communities

Dimensioner för jämförelse och analys och projekt

- Livscykelstadie
  - 1) uppstart, 2) tillväxt, 3) stabilisering, and 4) nedgång
- Projektens komplexitet
  - Scope, storlek och teknisk komplexitet av kodbasen
- Styrningens koncentration
  - Påverkan på projektets öppenhet för externt inflytande och bidrag, samt transparens därav
- Strategisk betydelse
  - Betydelsen av projektet från ett tekniskt och affärsmässigt perspektiv

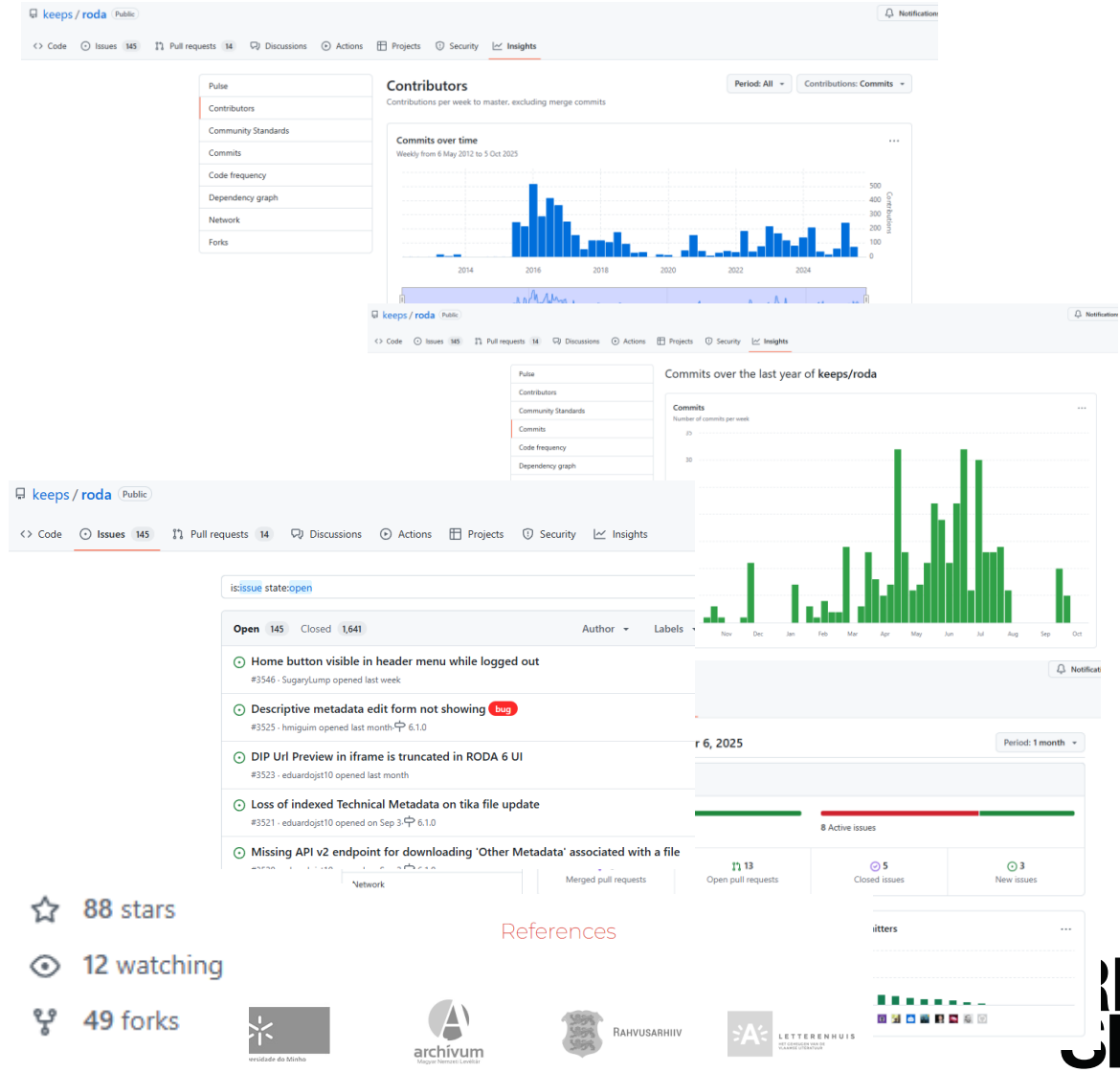
Photo by Annie Spratt | https://unsplash.com/photos/open-book-page-vpFiAD-WWEs

Anskaffnings-perspektivet

- Försök hos stor svensk myndighet
- Workshop-format med interna intressenter
- Målet var att utvärdera hälsan hos öppna e-arkiveringslösningar
- Enkät utvecklad genom iterationer baserade på CHAOSS-mått
- Möjliggör jämförelse mellan öppna och slutna alternativ vid upphandling
- Utvärderingen behöver vara noggrann och detaljerad
- Rapport (se sid 21):
  - https://gitlab.com/open-data-knowledge-sharing/wiki/-/wikis/uploads/filer/2021-05-28_Shared_eArkiv_Arbetsformedlingen_Akerlund Lundstedt.pdf

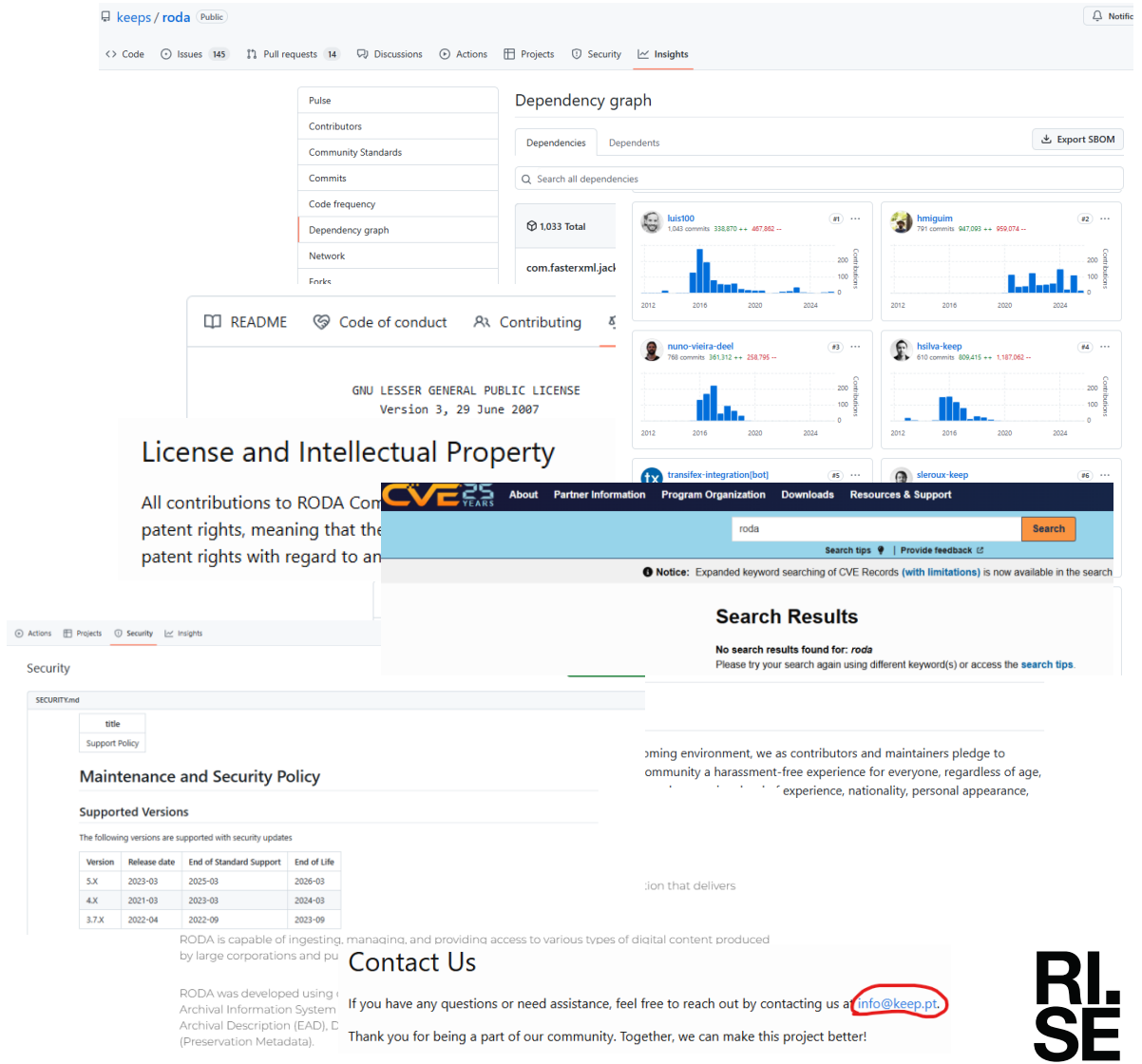
Analys av RODA

- Projektets produktivitet – Utvecklingsaktivitet
  - Bidragsaktiviteten till kodbasen över tid, t.ex. senaste 45, 90 och 365 dagar.
  - Aktivitet i utvecklingsrelaterade aktiviteter, såsom kodgranskningar, sammanslagning av pull requests och arbete med ärenden under samma tidsperioder.
- Projektets produktivitet– Responsivitet
  - Hur snabbt och kvalitativt svar ges, t.ex. på nya diskussioner, frågor, pull requests eller ärenden.
- Projektets stabilitet – Användning
  - Projektets användning signalerat genom olika popularitetsindikatorer, t.ex. stjärnor och följare på GitHub eller nedladdningar från paketförvaltare.



Analys av RODA

- Orchestration – Licenser
  - Närvaro och kvalitet på nödvändig juridisk information, t.ex. ingående licenser och information om exportkontroll
- Orchestration – Styrningsstruktur
  - Närvaro, tillgänglighet och aktuell dokumentation av styrningsstruktur; hur beslut tas, konflikter hanteras, av vem och hur representanter väljs.
- Produktionsprocesser – Säkerhetshantering
  - Förekomst av rapporterade sårbarheter, t.ex. senaste 1, 3 och 12 månader, och i vilken takt dessa har åtgärdats.
  - Processer för hur sårbarheter rapporteras, diskuteras, hanteras, offentliggörs och kommuniceras av OSS-projektet.
  - Implementering av bästa praxis enligt Open Source Security Foundation, såsom multifaktorsautentisering och automatisk sårbarhetsskanning.



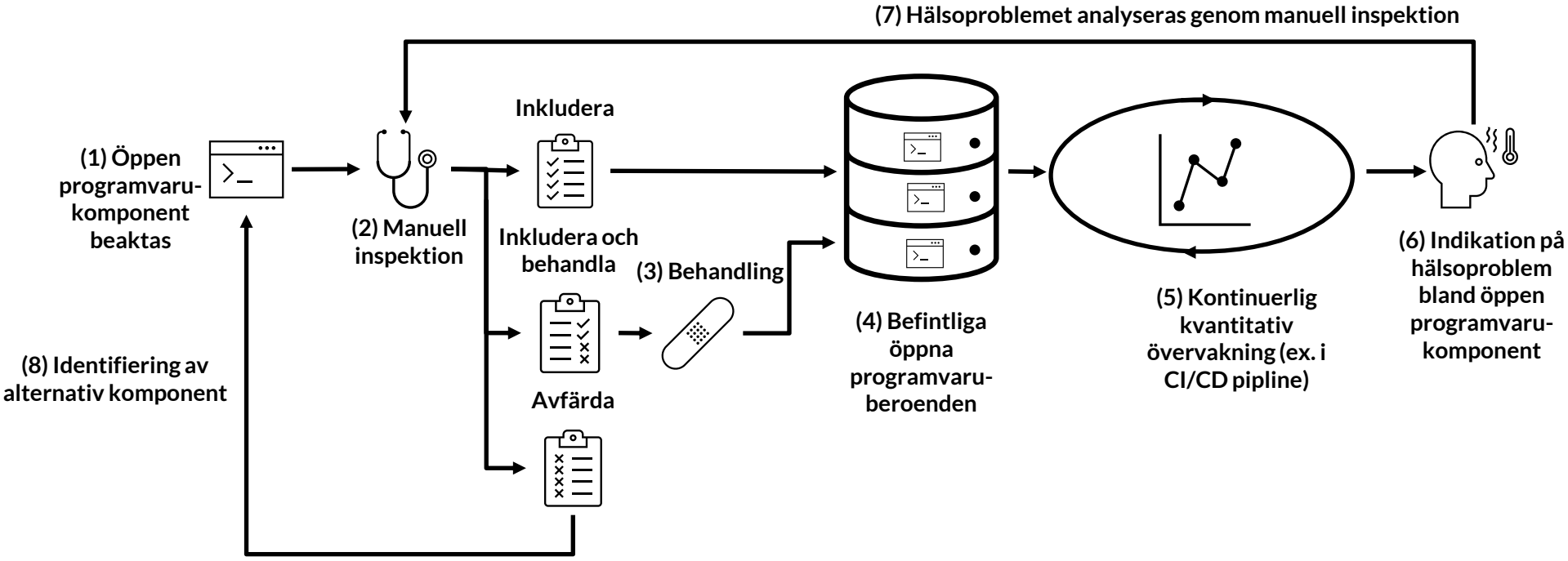
Utvecklingsperspektivet – från teori till praktik

- Vad:
  - Hantera hälsorisken som följer av användande av öppen programvara
- Hur:
  - Etablera en intags och screening-process för nya och befintliga öppna programvaruberoenden
  - Övervaka hälsan och göra proaktiva beslut kring sourcing och bidragsstrategier avseende öppna programvaruprojekt
- Nyckelkrav:
  - Decentraliserad och själv-administrerande process
  - Stärka utvecklare men begränsa bördan
  - Möjliggöra uppföljning och styrning
- Rapporterad i https://ospobook.todogroup.org/06-chapter/ >>

RI.SE

Mer läsning >>

Semi-automatiserad process för hälsobevakning

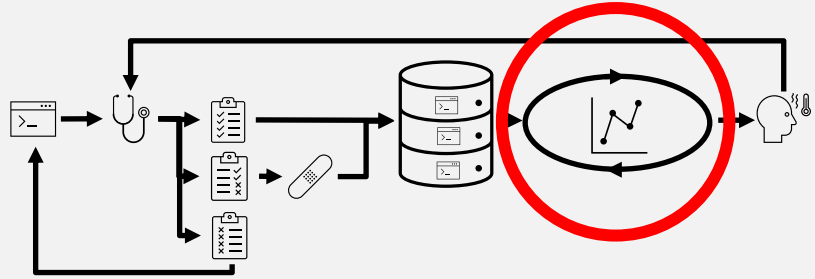


@johanlinaker

RI. SE

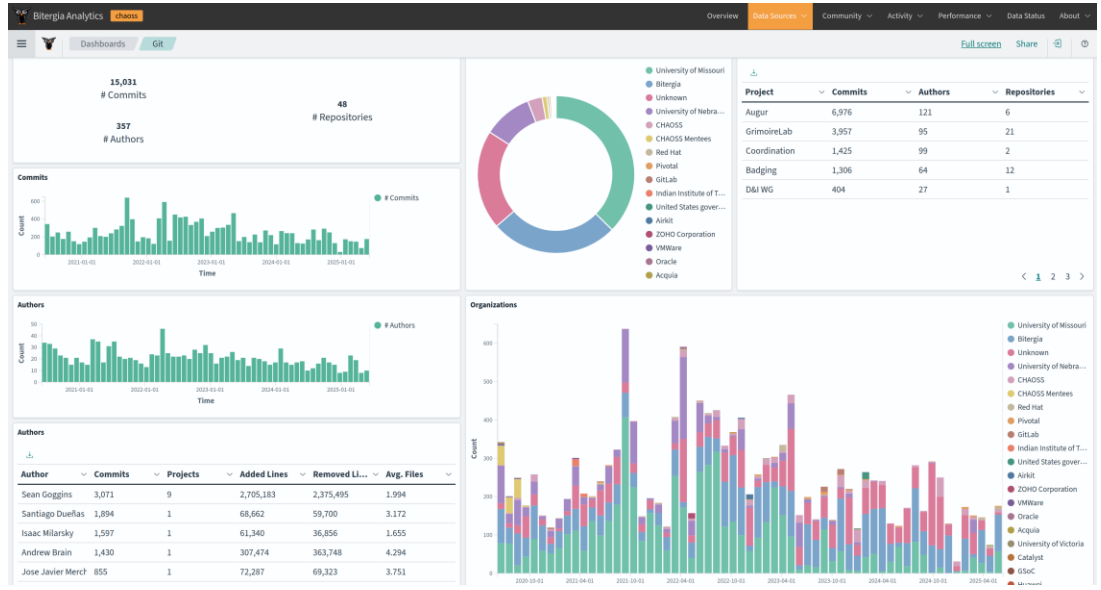
Kvantitativ screening

- Stora mängder beroenden är vanligt förekommande. Manuell översikt och granskning inte realistisk.
- Verktyg behövs, integrerat i CI/CD-pipelines eller som körs vid regelbundna tillfällen.
- Kör övergripande tester anpassade till typ av ekosystem och beroenden.
- Flagga projekt och rikta uppmärksamhet dit där indikatorer tillsammans pekar på en potentiell risk.
- Flaggade projekt följs upp av manuella granskningar av utvecklare eller analytiker.
- Specialanpassade verktyg och/eller färdiga lösningar finns. Se ex. GrimorieLab och Debricked OSS Intelligence.



Automation och översikt - GrimorieLab

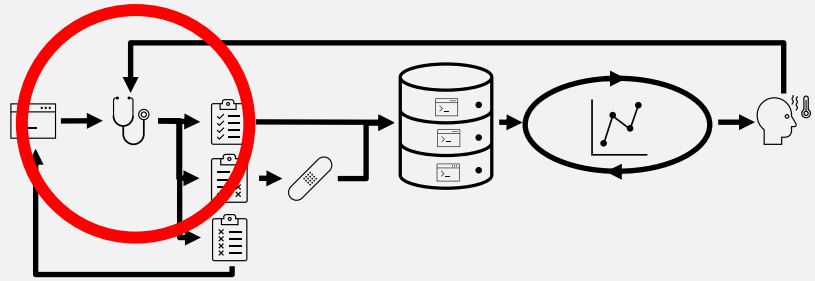
- Data samlas in från 30+ källor, inklusive historiska data, med löpande uppdateringar och hög datakvalitet.
- Rådata förädlas för djupare insikter, t.ex. analys av kärn-, regelbundna och tillfälliga bidragsgivare samt attraktion och retention.
- Intuitiva dashbordsverktyg möjliggör utforskning och delning av data samt skräddarsydda visualiseringar.
- API och gränssnitt stödjer hantering av organisationer



https://chaoss.biterg.io/app/dashboards

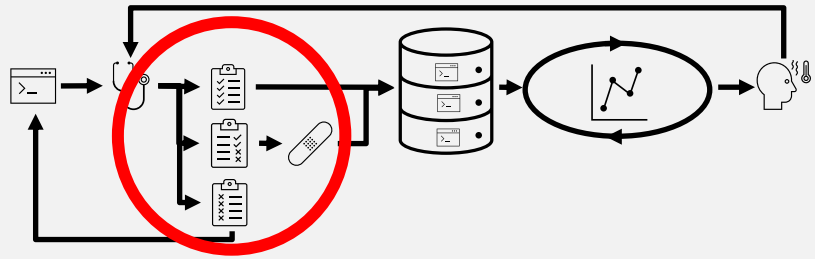
Manuella inspektioner

- Analys av enskilda projekt, antingen identifierade i screening, eller som underlag för sourcingbeslut
- Användning av standardiserad checklista med automatiserat verktygsstöd vid behov
  - Avvägning mellan noggrannhet och effektivitet
  - Intervjua och kartlägg huvudsakliga bekymmer från interna intressenter
  - Beakta vilka typer av projekt som används och behovet av anpassning
  - Kräver enkla svar (Ja/Nej) eller tydliga kategorier (1-5, 6-10…)
- Lättviktig dokumentationsprocess som bevarar och indexerar analyser för framtida uppföljning



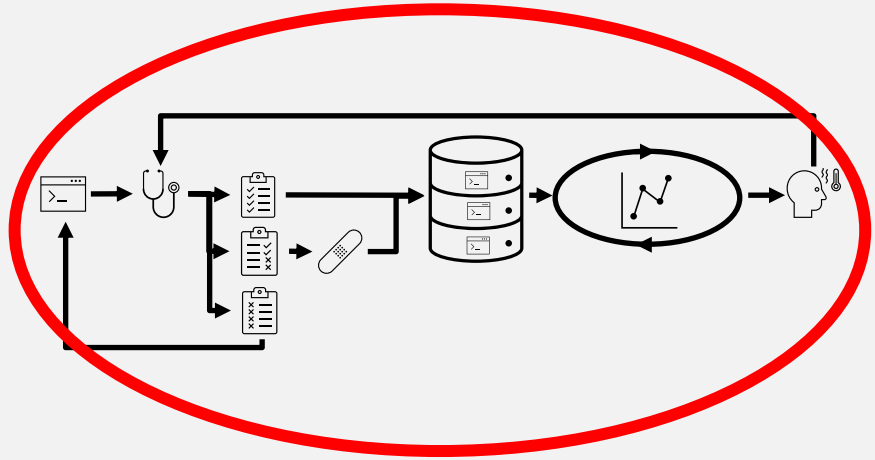
Vad ska man kolla efter?

- Behov att definiera målen med analysen och vilka frågor som ska besvaras
  - Huvudsakliga bekymmer och risker
  - Typer av projekt, inom vilka domäner etc.
- Litteratur och praxis har tillhandahållit en kunskapsbas som kan användas tillsammans med befintliga initiativ, t.ex. CHAOSS, OpenSSF
- Kräver arbete i förväg
- Utvärdering hos Scania
  - Fokusgrupp + användarobservationer
  - Sammanfattat i en checklista med 14 hälsattribut



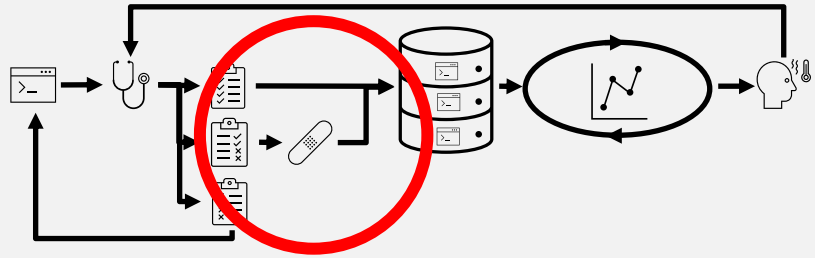
Behov av träning och uppföljning

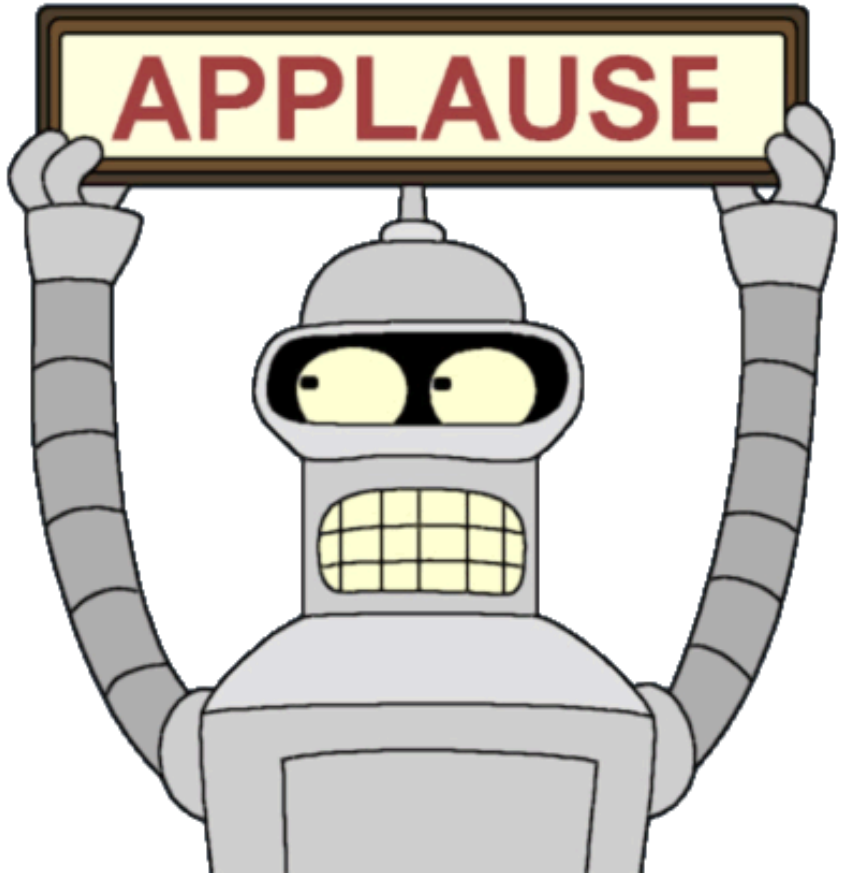
- Workshops för att introducera checklistor och analysprocess
- Integrera som standardiserad praxis i utveckling och Q&A
- Återkommande feedbacksession för presentation av analys av öppna programvaruprojekt
  - Uppmuntra diskussion, kunskapsdelning och kritiskt förhållningssätt
  - Kontrastera mellan olika typer av projekt, relevanta frågor att ställa samt tillämpning/tolkning av mätvärden



Behandling och riskmitigering

- Säkerställ och möjliggör de mänskliga resurser som behövs för ett hållbart underhåll
- Härstammar antingen från kodförvaltarna eller från communityt
- Kräver investeringar och stöd till projekten
  - Utveckling av ny funktionalitet och buggfixar
  - Kravställning, prioritering och planering
  - Testning och kvalitetssäkring
  - Dokumentation, marknadsföring och communitybyggande
  - Finansiering och support med infrastruktur




APPLAUSE