Community engagement:
tips for companies on open source

Matheus Tavares Bernardino
https://matheustavares.gitlab.io/

$ whoami

- Working with QEMU @ Qualcomm
- Git contributor for ~3 years
  - Parallel checkout and grep
  - git clone vulnerability
  - sparse-checkout
- MsC thesis on git
- Google Summer of Code 2019
- Kworkflow ex-maintainer
- Linux kernel: IIO subsystem

Why open source?

- Faster innovation
- Reduced maintenance costs
- Higher quality & security
- Recruitment & retention of skilled devs
- Influence the direction of industry standards

“Influence comes from participation”
- Ibrahim Haddad (Linux Foundation VP)

Contributing to external projects

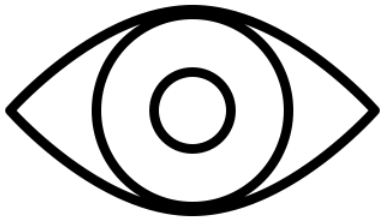
Open sourcing internal projects

External projects

1. Start by observing
2. Work together with the community
3. Give back through review
4. Give back through bug-reports

1. Start by observing

- You are a “stranger joining an ongoing conversation”
- Each project has its own culture, norms, and expectations
  - Contributing docs
  - Commit log
  - Mailing list, IRC, slack, etc.
  - *Unspoken rules*
- Be prepared to put aside your preferences

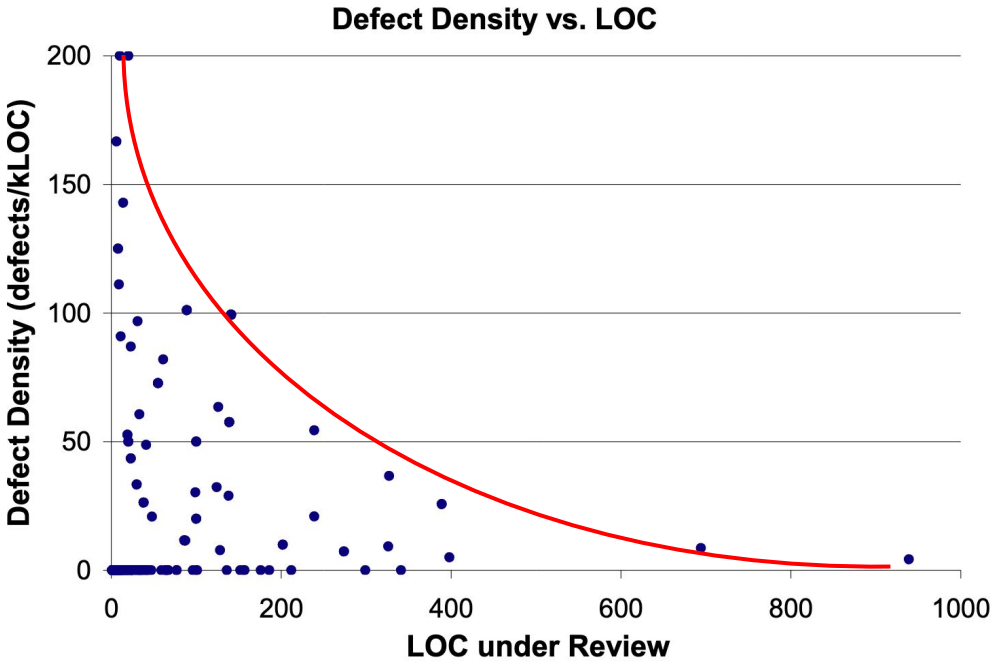


2. Work together with the community

- Involve upstream early and often
  - Don’t just dump your code
  - Avoids redesign costs later
  - consider “upstream first”
- Design with upstream in mind
  - Avoid temporary workarounds
  - Follow the code conventions and APIs
  - Provide documentation
  - Work on your commit messages (why is this useful for upstream?)
- Listen and respond to feedback. Be patient.

2. Work together with the community

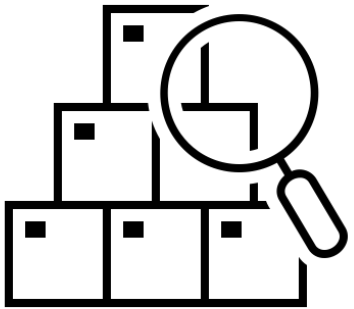
- Number of *defects* per 1000 LOC
- *"LOC under review should be under 200, not to exceed 400."*
- Patches should be small



SmartBear study of Cisco team (2006)

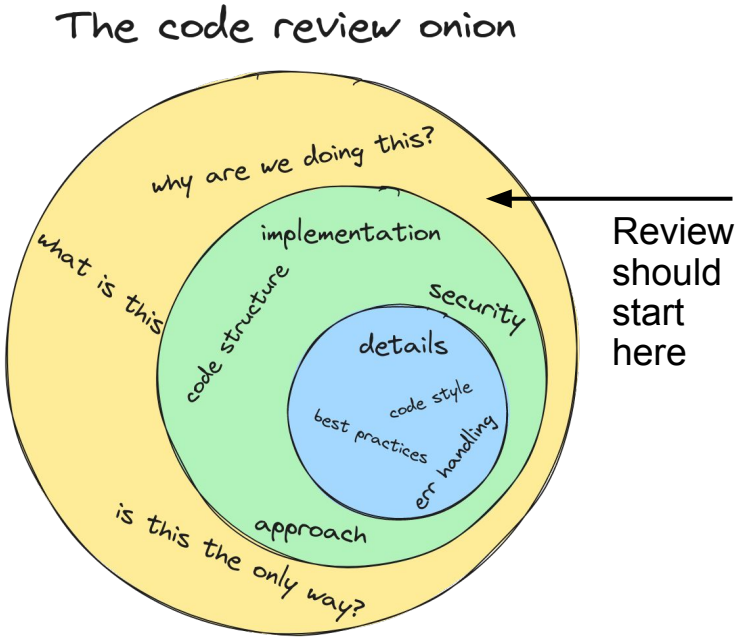
3. Give back through review

- **It goes both ways:**
  - Gain community trust
  - Learn and improve code
- Schedule time for reviews
- Respond in a timely manner
- Walk through the code “thinking out loud”



3. Give back through review

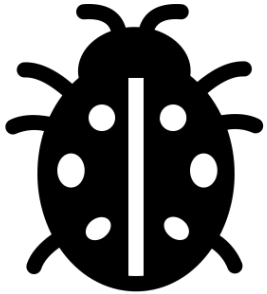
- **Don’t put shame** on the author
- **Explain the reasoning** behind your suggestion
- Compliment what’s good
- Group review comments to avoid extra rounds



https://trstringer.com/code-review-onion/

4. Give back through bug-reports

- Expected behavior vs actual result
- Describe the scenario
- **Minimal reproducible example**
- Bisect the commit history
- Known workarounds?



Also helpful: improve on reports from others

Internal projects

1. Foster a community
2. Improve processes

1. Foster a community

- Open documentation
- Avoid the need for vendor-specific tools
- Respond to issues/PRs
- GSoC/Outreachy mentoring
- Partner with universities and local groups
- Developer blog posts

2. Improve internal processes

- Fund elected less-recognized open source projects
  - *Employee pools?*
- Open source award program *(internal and external)*
- Collect metrics
- Fund for conference participations
- Internal mentoring for open source newbies
- Provide the tools & hardware
- Encourage personal OSS contributions

Thank you

Matheus Tavares Bernardino
https://matheustavares.gitlab.io/

Reading suggestions

- Google's Reviews guide
- Code Review Anti-pattern
- Linux Foundation's "Participating in Open Source Communities"
- Well-written commit messages: my slides and cbeams'.
- The Unspoken Rules of Collaborating on Open Source