Work Like Open Source

Adopting open source process constraints

Organization and Process

The hacker culture and its successes pose by example some fundamental questions about human motivation, the organization of work, the future of professionalism, and the shape of the firm.[1]

Eric Raymond

[1] The Cathedral and the Bazaar, 2001

Survivability of Process

- Process designed to open source constraints result in projects that run well, attract attention, and are self-perpetuating.
- The same project structured traditionally requires much more manual coordination and authoritative prodding.
- Most interaction is reduced to the basic feature set of email.

Open-Source Software

The process of systematically harnessing open development and decentralized peer review to lower costs and improve software quality [2]

Eric Raymond

[2] The Cathedral and the Bazaar, 2001

Development Constraints

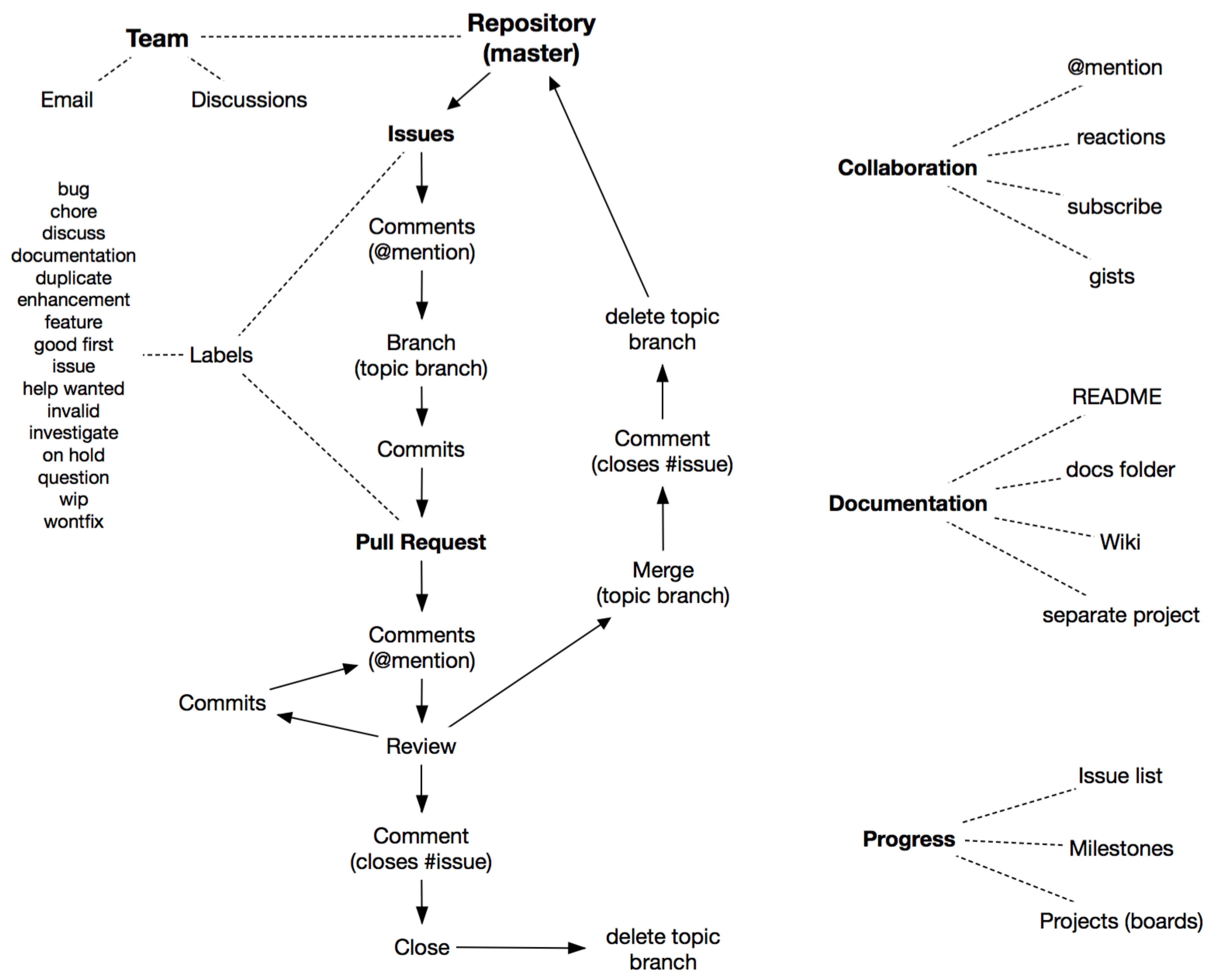
- **Asynchronous**: Almost no part of the process requires interrupting others.
- **Electronic**: Discussion and planning uses email, GitHub team discussions and Gists. Avoid meetings.
- **Available**: Work is visible, exposes process, has a URL (link). Collaborate using GitHub Issues, Pull Requests and Gists.
- **Lock Free**: Avoid process sync/lock points. Work is not blocked by approval. Use GitHub Flow.

Just Use GitHub

If your process isn't just "use GitHub", GitHub is still going to be the foundation of whatever process you use instead, 90% of the time.[3]

Giles Bowkett

[3] Flaws In Scrum And Agile


Team
Repository (master)
Email
Discussions
Issues
Comments (@mention)
Branch (topic branch)
Commits
Pull Request
Comments (@mention)
Commits
Review
Comment (closes #issue)
Close
delete topic branch
Merge (topic branch)
Comment (closes #issue)
delete topic branch
Labels
bug
chore
discuss
documentation
duplicate
enhancement
feature
good first issue
help wanted
invalid
investigate
on hold
question
wip
wontfix
Collaboration
@mention
reactions
subscribe
gists
Documentation
README
docs folder
Wiki
separate project
Progress
Issue list
Milestones
Projects (boards)

RIP The Office

- Traditional offices violate open source constraints.
- Offices are not required as center of planning, coordination, and communication for software development.
- Offices will not go away entirely, just not required for most aspects of the software product development lifecycle.
- Offices are required for defining broad vision, strategy, setting big goals, and coherent high level product design.

Show How, Don't Tell What

Show people how to plan, build, and ship product together

- Don't tell people what to do.
- Lead by example as loud as possible.
- Get people contributing.
- Make everyone a manager.
- Just do work.

Do the simple thing first

Just use GitHub

References

The Cathedral & the Bazaar
Your team should work like an open source project
Show How, Don't Tell What - A Management Style
GitHub Guides