Community Guide to Contributing to Mature Software Products

Michael Kromer
VP Technology & Architecture

$ whoami

- Mike, Geek.
- First computer with 6 (C64)
- Background in US military background (Scalability, FUBAR-safe)
- VP Technology & Architecture (Kopano)
- http://medozas.de/vita.html
- openSUSE Member und Committer
- Committed code to Kopano, the Linux kernel, Asterisk, LibreOffice, KVM, Virtualbox, Nomachine NX, X2Go, OpenSER, Chromium, USBoverIP, …….

There is no ***one*** community.

Every software community works a little or more different.

Examples

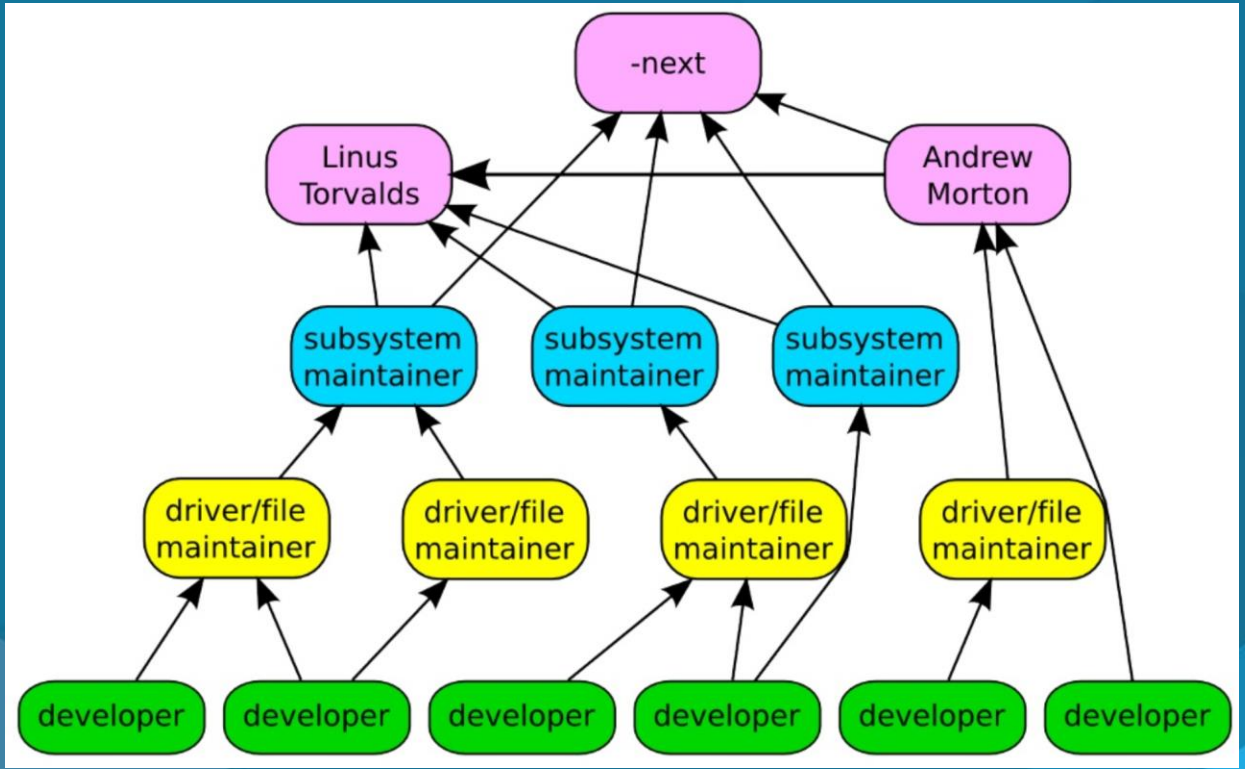
- Linux Kernel
- Iridium Browser
- Kopano
- openSUSE / SLE

Different organisational structures

- Linux Kernel → „demand-oriented“ with chain of trust
- Iridium Browser → „project-oriented“ with a specific goal
- openSUSE → open community with commercial support
- SLE → company-driven, customer & market oriented
- Kopano → „hybrid“ community with commercial backing

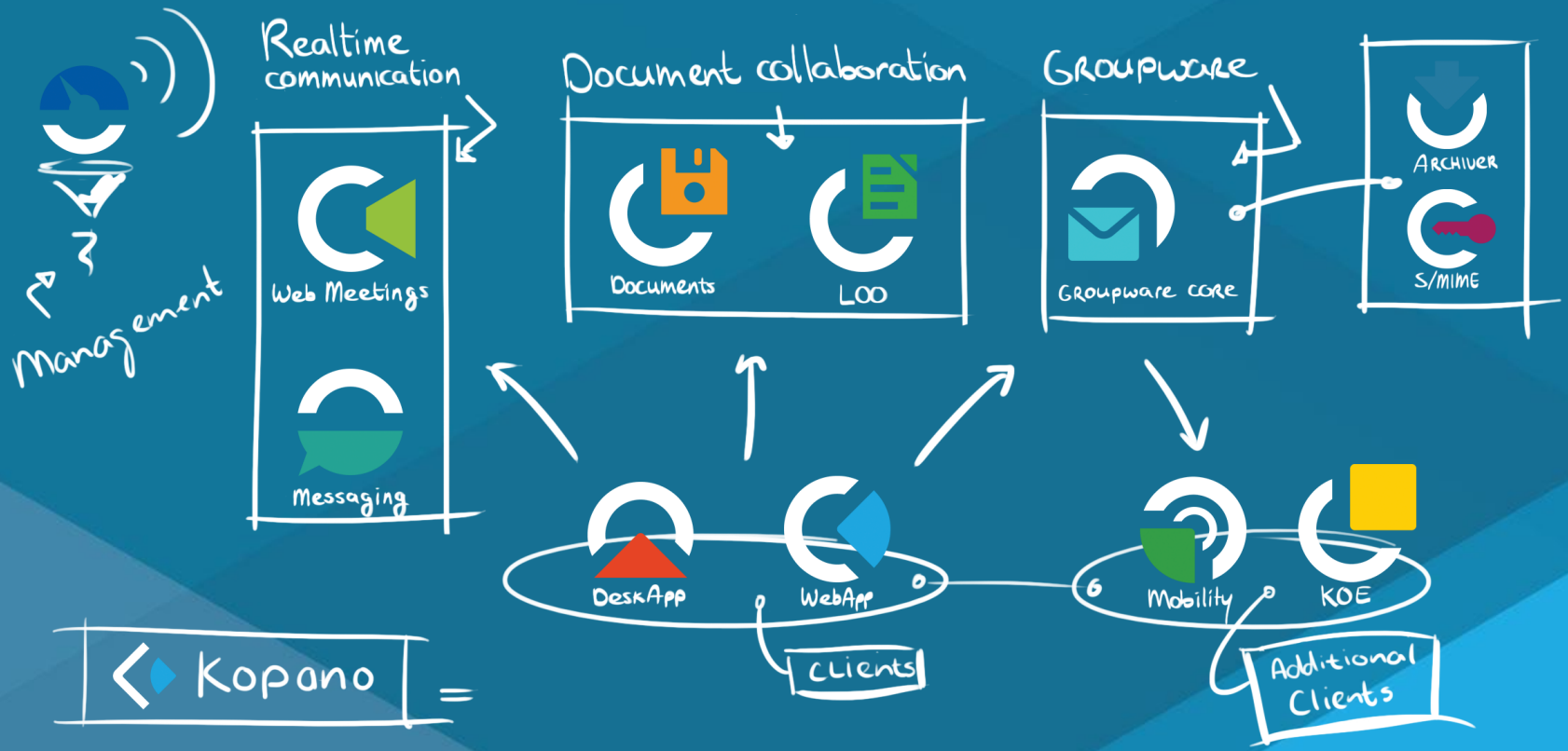
Example of code & project structure

Linux



Greg Kroah-Hartman gregkh@linuxfoudation.org

Kopano

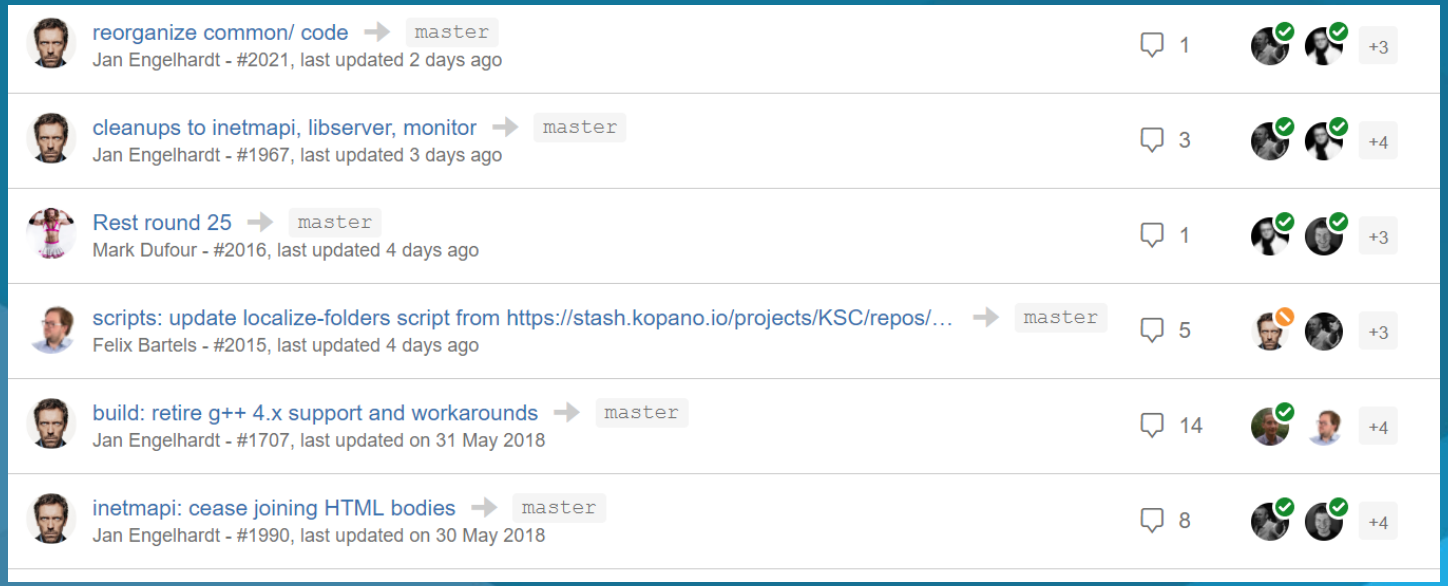

Management
Realtime communication
Web Meetings
Messaging
Document collaboration
Documents
LOO
Groupware
Groupware core
Archiver
S/MIME
DeskApp
WebApp
Clients
Mobility
KOE
Additional Clients
Kopano
=

Lifecycle overview of a mature software product

From code to quality code

Review

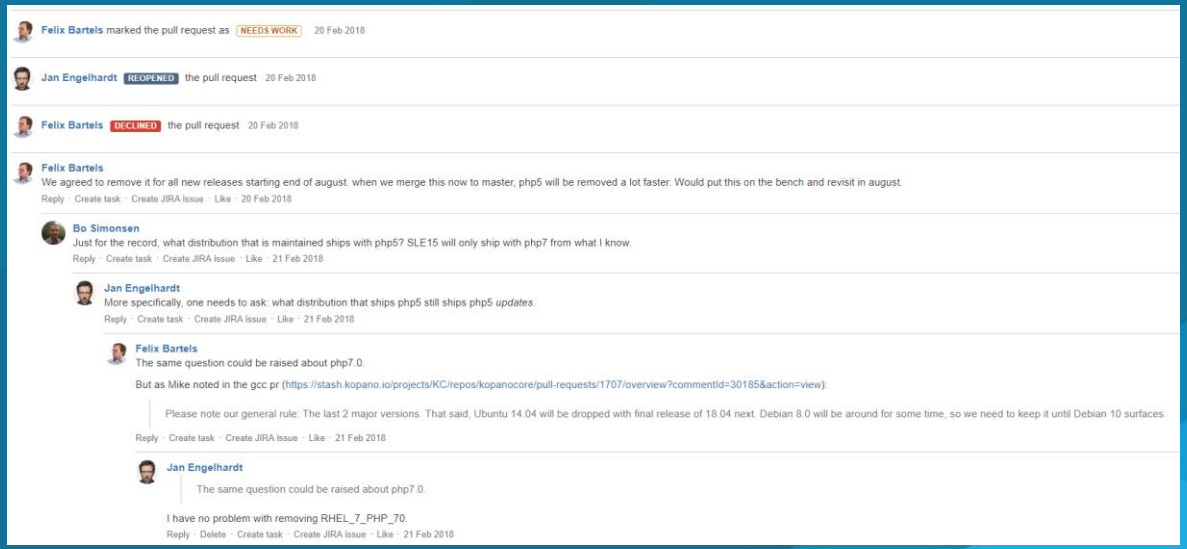
– Reviews by developers in that area (merit & experience based)



From code to quality code

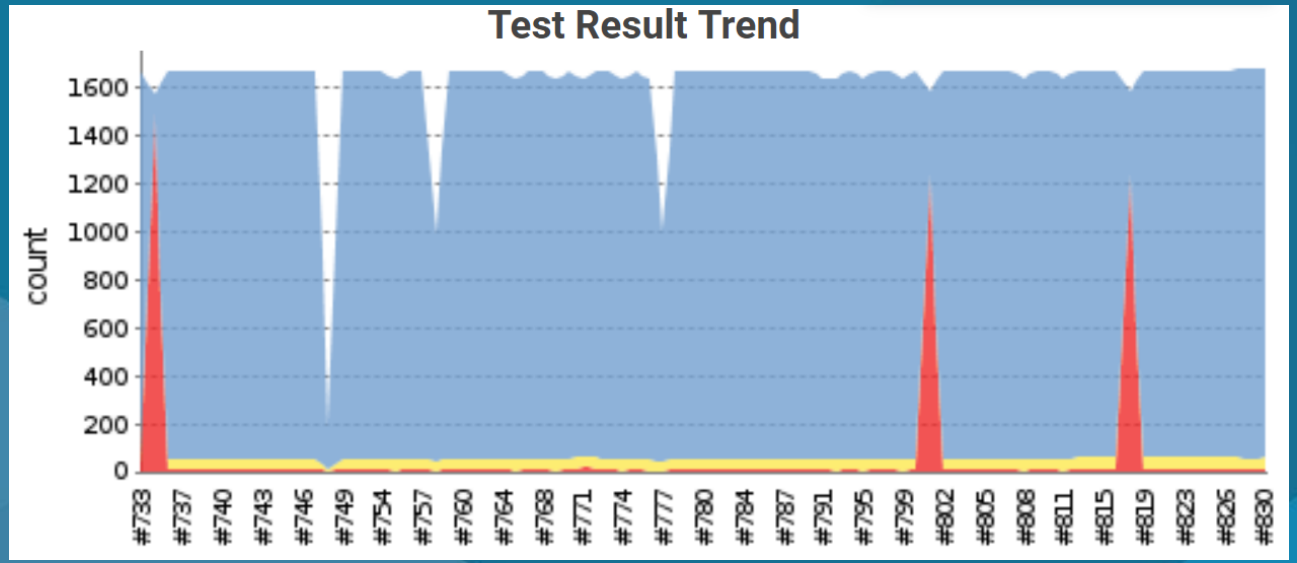
Review

- Code quality
- Potential side effects
- Style guidelines

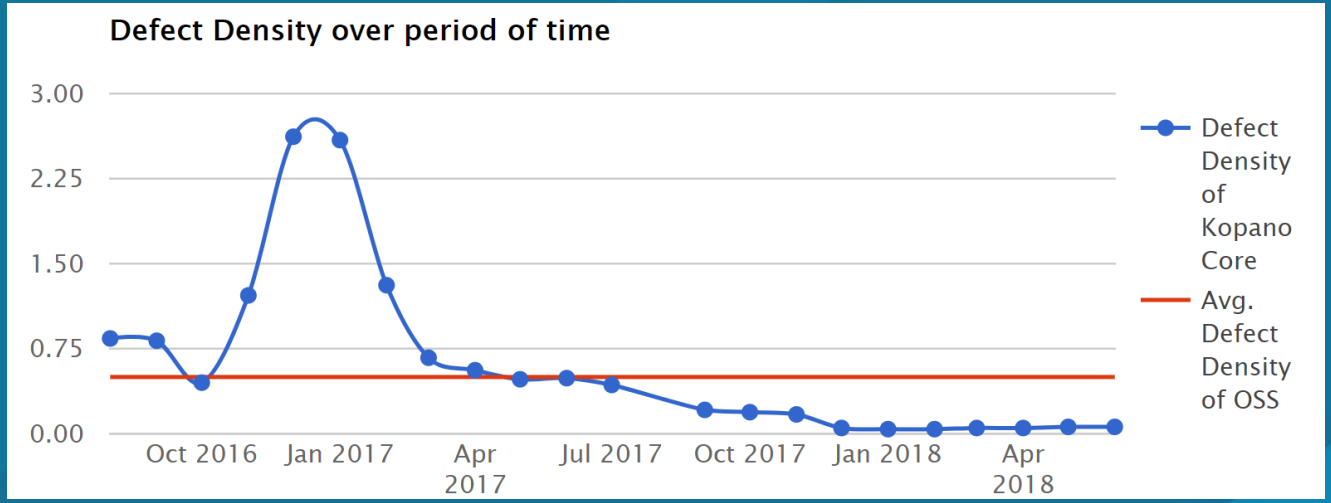


From code to quality code
CI

– Automated quality checks (unit tests, merge tests, sanity checks, linting checks, memory leak checks)



From code to quality code
CI



- Regressions?
- Instability?
- Degration of (scalability|performance|interoperability)?
- Upgrade-safe?

From code to quality code
QA

- Verification of all checks (Is Test-covered?)
- Manual check if „issue“ is solved
- Documentation changes

From code to quality code

Release

- Backwards incompatibilities? (API/ABI)
- Behavioral changes?
- Default settings?

☐	All commits merged to the release branch
☐	Commits announced at the forums https://forum.kopano.io/topic/901/z-push-2-4-x-beta-pre-final-updates/
☐	Create a tag on the master (final release) or release branch (beta release)
☐	Check if the builds are available at http://repo.z-hub.io/z-push:/final/ or http://repo.z-hu If not, check Release procedures#Buildingpackages page how to build packages
☐	Update the ZP-1335 - Maintain 2.4 release branch IN PROGRESS
☐	Create release notes for the forums
☐	Announce the release at https://forum.kopano.io/category/26/general-usage-of-z-push
☐	For final release do blog post at http://z-push.org/category/news/
☐	For final release update information at http://z-push.org/download/
☐	Update releases page at https://jira.z-hub.io/projects/ZP?selectedItem=com.atlassian.
☐	Update Version Lifecycle page
☐	Update Roadmap page
☐	For final release email Toos and Bob, cc Felix and Z-Push devs with the link to the rele
☐	Announce release at https://mattermost.kopano.com/core/channels/kopano-releases
☐	For final release Send an email to the Z-Push mailing list (Z-Push@lists.inai.de)

Questions?