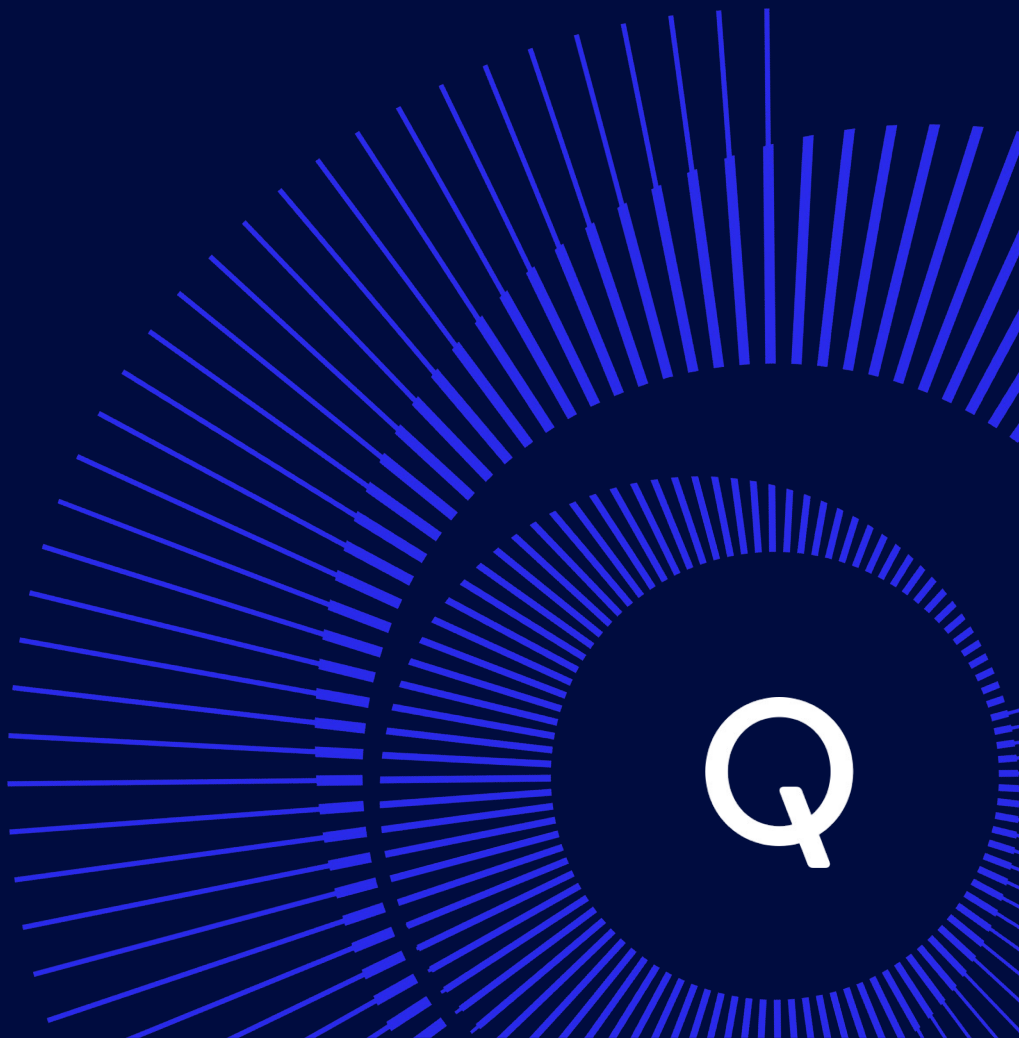
Qualcomm

From Closed To Collaborative: Lessons From Qualcomm's Open Development Experience

Rashmi Chitrakar

Qualcomm Technologies Inc



From Closed to Collaborative

In-Depth Perspectives & Lessons from Qualcomm's Open Development Experience

Open Development Evolution	The journey to Qualcomm Linux 2.0
New Ways of Thinking	OSDO – Open Source Development Office, Identities and Branding, Policies & Mandates, Support Maintainers, Build a Community & Track Contributions
Open Development Workflow & Integrations	Investment in existing systems in an Open Workflow
Roll Out Strategy	Incremental expansion from "boot to shell" to broader enablement
Lessons & Insights	From Upstreaming & the overall transition effort

Open Development Evolution

Open Development Evolution

The Old BSP Model

- Proprietary, Android-based BSPs
- Customized per platform/customer/ project
- No open collaboration or ecosystem
- Misaligned with Yocto/upstream practices
- Code dumps and fragmented efforts

meta-qcom

- Community maintained, clean layer following Yocto best practices, developed in the open
- Limited HW support, focused on open source based implementations (ex: freedreno)
- Not officially supported by Qualcomm; Reference BSP for Linaro

Qualcomm Linux 1.0

- Closer to traditional Embedded Linux
- Yocto/OE layers more aligned with upstream expectations
- Huge improvements over old “LE” targets
- meta-qcom-hwe as the Yocto hardware enablement layer for QLI supported targets
- Dependency on meta-qcom
- Closed development

Qualcomm Linux 2.0 Open Development Goals

Upstream based software distributions

- Mainline serves as the baseline for all other derivatives & is developed in the open
- Limited forked projects, downstream patches, or other ”custom” enablement
- Pristine upstream focused software distributions

Public open development model

- Enable the community & Qualcomm engineers to collaborate on software projects
- Drive for transparency, trust & visibility into the software we deliver into the open

Continuously validated baselines

Enable completely automated hardware testing to keep the baseline from introducing regressions by leveraging CI/CD systems integrated with Github

Downstream enablement on Upstream baseline

- Enable CI/CD loops for value-add technology against the latest mainline
- Work w/ tech teams to resolve issues & create a stable integration

New Ways of Thinking

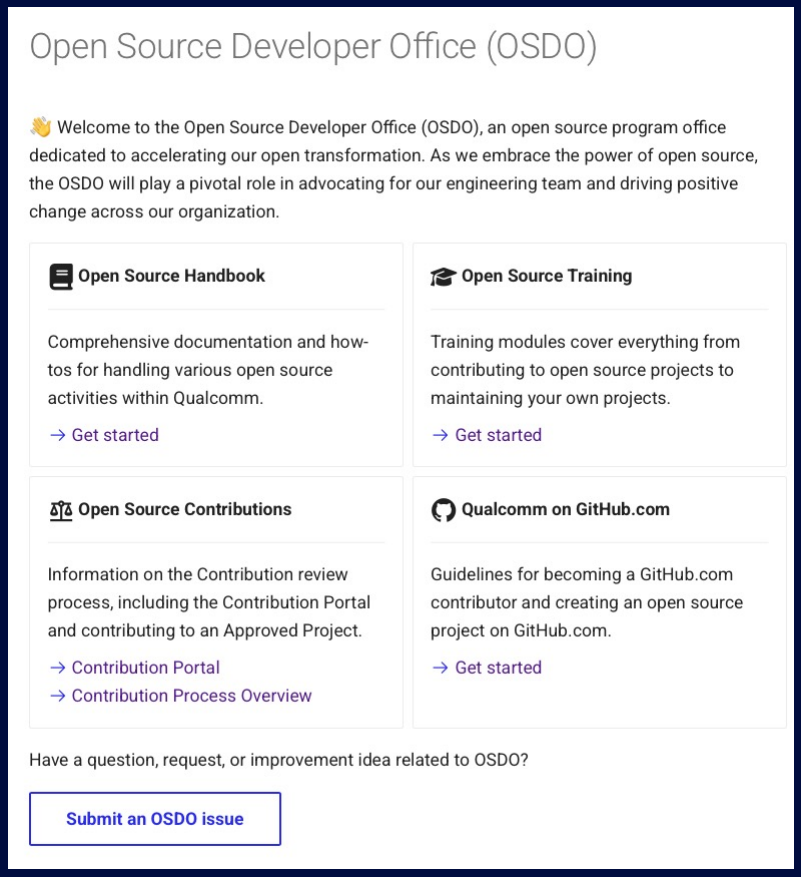
OSDO – Open Source Developer Office

Qualcomm's OSPO dedicated to accelerating our open transformation

Run as an internal open source project, inviting participation across our organization & fostering partnerships among different organizations & roles.

Mission

- Accelerate changes needed by our open source developers.
- Streamline processes and remove internal roadblocks.
- Provide a clear and centralized voice for engineering across the company.



New Ways of Thinking

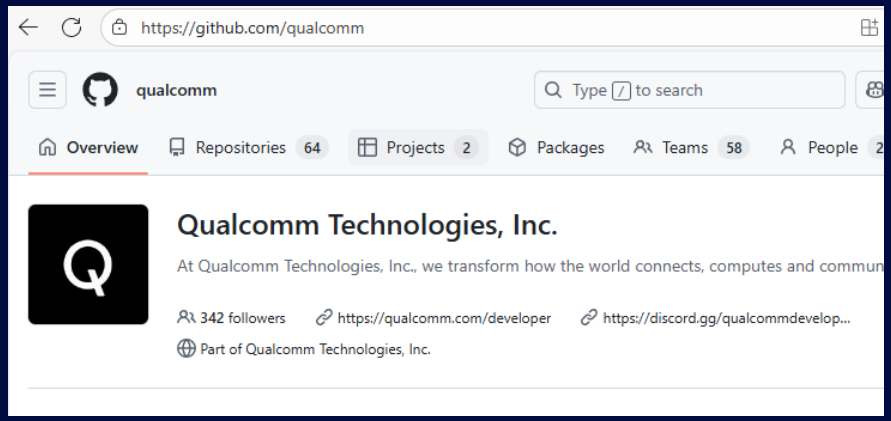
Rethink our approach across several domains

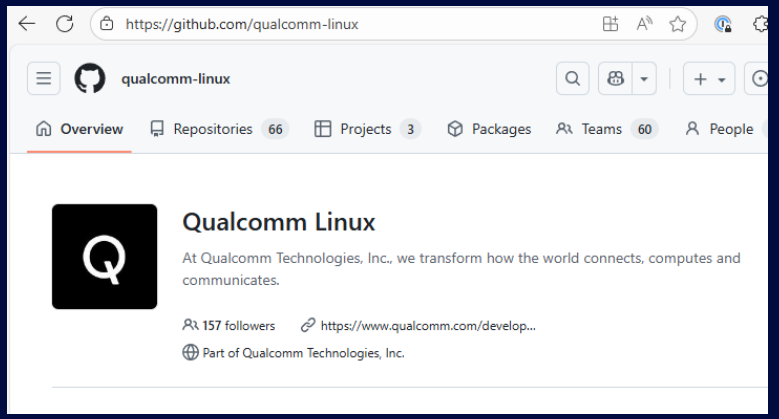
Identities & Branding	Show up as Qualcomm
Policies & Training	Upstream-first approaches
Support Maintainers	Support & retain talent
Build a Community & Track Contributions	Insights & Analytics

Identities and Branding

Showing up in the community as Qualcomm

- Move to Qualcomm Technologies Inc. identity and branding for our Open Source presence
- New e-mail service, new domain – oss.qualcomm.com
  - Challenging and disruptive to some aspects of our development and the communities in which we participate, but necessary.
- New GitHub Organizations – github.com/qualcomm, github.com/qualcomm-linux





Policies & Mandates

New Policies

Mainline/Upstream first

Strict enforcement; exceptions require approval & must be ported to mainline unless not applicable

Semantic versioning

Transition to enable external understanding of releases

Security

New processes & tooling suited to open development & community vulnerability reporting

Training

Contributor & maintainer training covering best practices, templates & compliance checks

Key Mandates

Downstream overlays built against both mainline (0.0) and LTS (QLI 1.0/2.0) branches

Kernel tracks linux-yocto-dev master; Yocto tracks to active development upstream version

Upstream drivers must enable boot and basic functionality for all peripherals on supported SoCs

L1+ automated testing; no commercial support on mainline

Support Maintainers

Drive towards a more maintainer friendly development approach, support tooling and processes for developers.

OSDO unified cross-company maintainer and OSS developer input into focused Working Groups and drove cross-functional execution to resolution.

Efficient Processes

Progress: 35% Complete

Focuses on removing the "red tape" associated with legal and compliance workflows.

Key Themes

Compliance Automation, Firmware workflows

Developer Experience

Progress: 64% Complete

Focuses on resolving immediate "quality of life" issues for developers.

Key Themes

Infrastructure, Tools

Community

Progress: 19% Complete

Focuses on Qualcomm's external reputation and culture.

Key Themes

Governance

Strategy

Progress: 39% Complete

Focuses on long-term health, security, and demonstrating the value of open-source work to leadership.

Key Themes

Metrics, Security

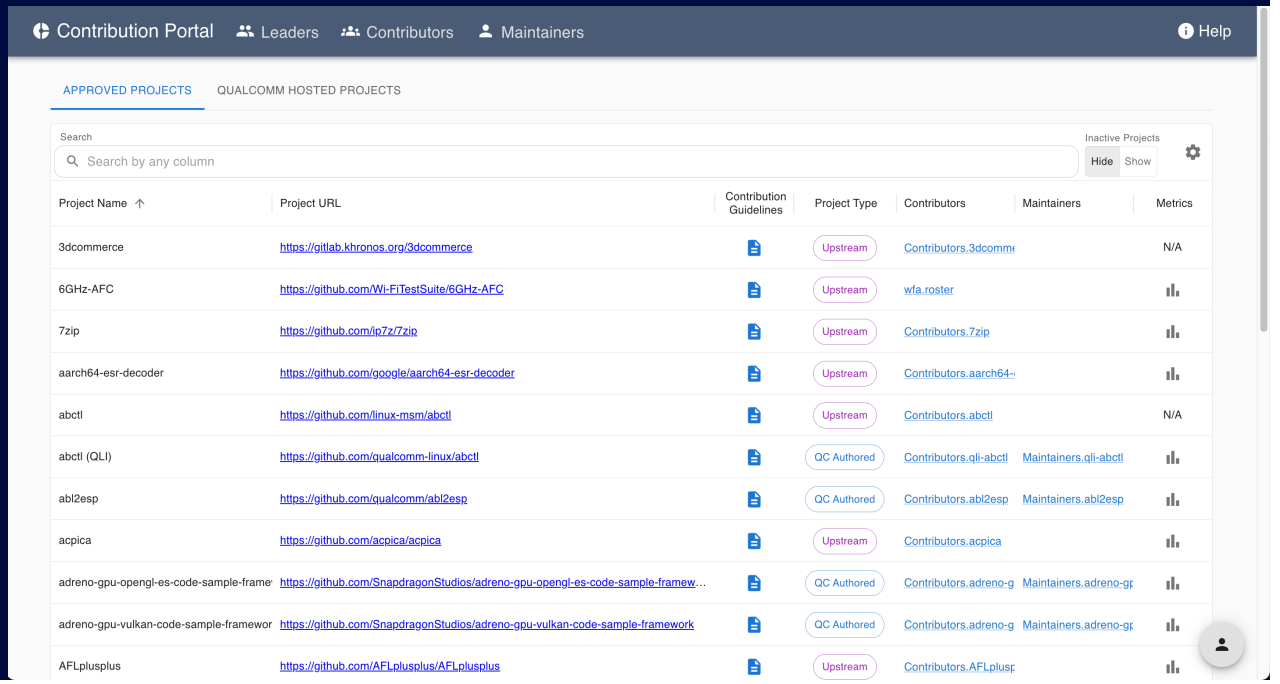
Build a Community & Track Contributions

Contribution Portal

System that tracks Open Source Projects cleared for contributions

Key highlights

- 600+ Approvals & 1200+ Projects
- Includes internal leads for each project
- View contribution guidance
- Join the contributor list
- Launchpad for Contribution Metrics

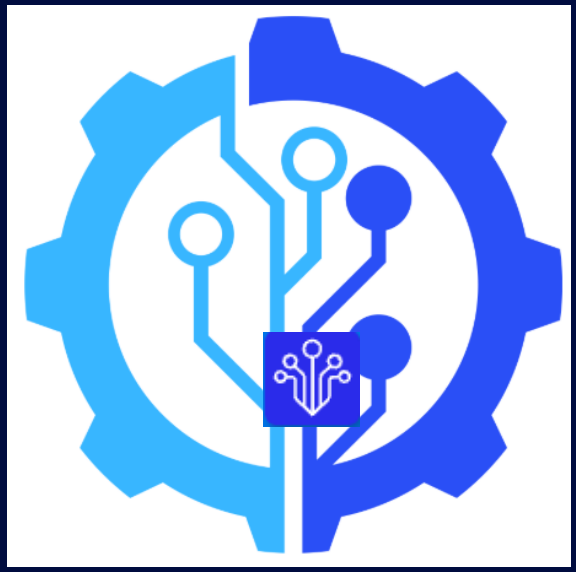


Development Workflow & Integrations

Connecting the Two Worlds

Product Delivery

- Validate against all commercialization lines
- Comprehensive device tests (external & internal)
- Change Request (bug & feature) traceability
- Qualcomm commercial deliverables must adhere to compliance requirements



Open Development

- Tech teams transitioning to develop in the open
- Encourage external contributions
- Modern CI/CD pipelines native to GitHub
- Fast transparent feedback
- Adherence to one-mainline model

Open Development Workflow & Integrations

GitHub Workflow

- GitHub Actions & Applications to facilitate workflows
- AWS Infrastructure provided by IT
- Key repositories hosted in "Qualcomm Linux" organization

Integrations

- Integrated with internal unified CI.
- Testing w/ LAVA and Internal test tools
- Integration with internal change tracking

Qualcomm Repository Template

Starter project with required files + OSS best practices

- Includes required and sample files e.g. README.md, LICENSE.txt, CONTRIBUTING.md
- Issue and PR templates
- Workflows
  - Qualcomm Preflight Checks
  - Stale Issues: warn and then close inactive Issues/PR
- Dependabot config

Qualcomm Preflight Checks Workflow

Runs a series of preflight checks on Pull Requests

1. **Repolinter** for open source best practices
   - To be replaced
2. **Semgrep** static analysis tool and security scanner
   - Investigating alternative tools for different ecosystems
3. **Dependency Review Action** detects vulnerable dependencies
4. **copyright-license-checker-action** checks copyright and licenses
5. **commit-emails-check-action** validates commit email addresses
6. **Commit-msg-check** Ensures commit messages follow standards (optional)
7. **ARMOR Compatibility Checkers** - Ensures source code follows API and ABI backward compatibility (optional)

Roll Out Strategy

Develop in the Open Roll Out Strategy

MVP Pilot for Yocto "Phase 1"

- Enable pilot technology teams and Yocto Linux on GitHub.com
- qcom-next kernel tree
- Publicly available CI including build, checks & test cases
- Basic testing integration

Develop in the Open

Expand & Enable "Phase 2"

- Expand footprint of technology teams & downstream value-add enablement on the upstream baseline
- Enable additional Linux distribution models to support Ubuntu & Debian
- Integration with internal QC Unified CI and CR (Change Request) Tracking

Increase community developer engagement

Maturing engagement

- Expand capability to ensure compatibility for a broad range of open source distributions on QCOM hardware
- Incrementally build a more sophisticated CI and testing approach to scale community engagement and enable commercial product requirements

Continuous Upstream Validation

Lessons & Insights

Lessons & Insights

Strategy	Outcome
Intersect with Community Efforts	Accelerated progress
Engage tech champions & mentors	Unblock bottlenecks early, reduce rework
Dedicate tech team bandwidth	Faster mainline enablement
Decouple upstream from marketing name	Unblock base DT upstreaming
Upstream FW early	Eliminate FW upstreaming delays
Templatize SOC upstream execution	Faster, consistent upstream delivery

From Closed to Collaborative: The Results Speak

3x contributor growth. 188 unique contributors. 12+ teams.

Qualcomm Linux 2.0 Journey

- Created **94** new OSS repositories across relevant organizations
- **3,243** merged/4,814 total PRs in the qualcomm-linux organization
- **673** issues filed — *54% already closed*
- **188** unique contributors

Huge cross functional effort

- First phase had **126** people with meaningful contributions from **12+** different organizations:
  - Program Management, Tools, Build & CI, Configuration Management, Test, Technology Teams, Security, IT and OSDO
- And then we expanded further!

Strategic Wins

- Incremental enablement was a good strategy
  - Immediate parity with existing closed development approach was not feasible
  - MVP of basic functionality & adding different technologies over time worked well
- Aligning with existing community effort (meta-qcom)
  - Strategically merged initiatives kickstarted momentum.

The Qualcomm Linux journey demonstrates that incremental, community-aligned open development scales

Thank You!