

arm

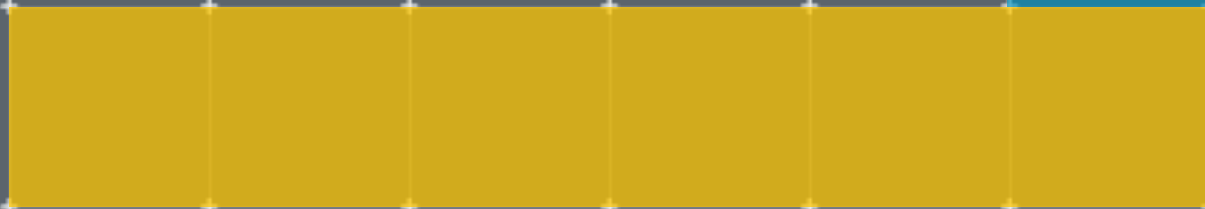
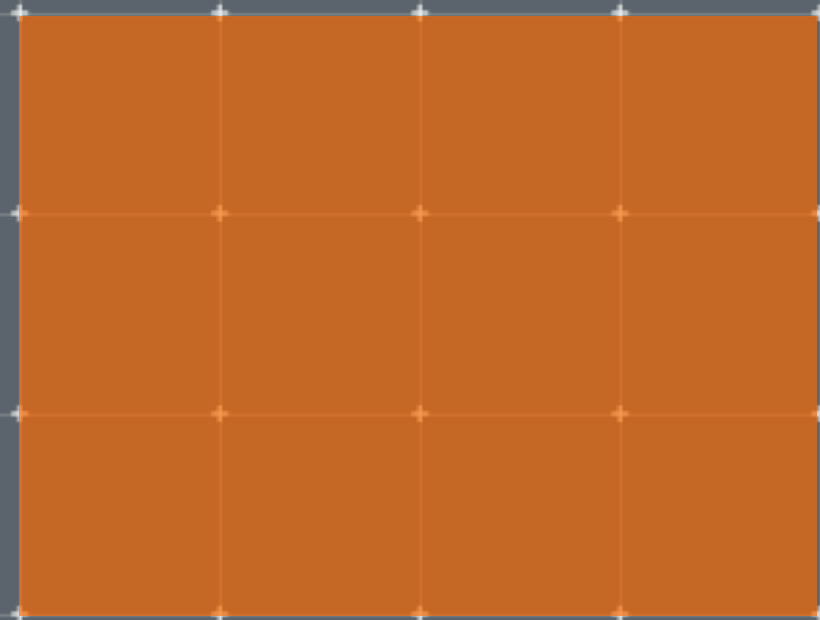


# Maintenance Process

Proposal

March-April 2020

# Initial draft proposal



# Goal of this Proposal

- Define a clear maintenance process for trustedfirmware.org projects
  - Initially for TF-A and TF-M (might extend to OP-TEE, ...)
- Define key developers roles within the project
- Enlarge maintainership to developers outside of Arm
- Document the code review process
  - Set expectations for contributors and reviewers
  - Define who does what
- Define a life cycle for platform support
- Start a discussion within the community

# Roles, Rights and Responsibilities

## Contributor

- Default role

## Code owner

- Responsible for specific module(s)
- Self-nominated or elected by their peers

## Maintainer

- High level view, ensure changes adhere to project's architecture
- Have merges rights
- Elected by their peers (**meritocracy basis**)
- Cumulative with code owner role (controversial point)

# Life Cycle of a Patch

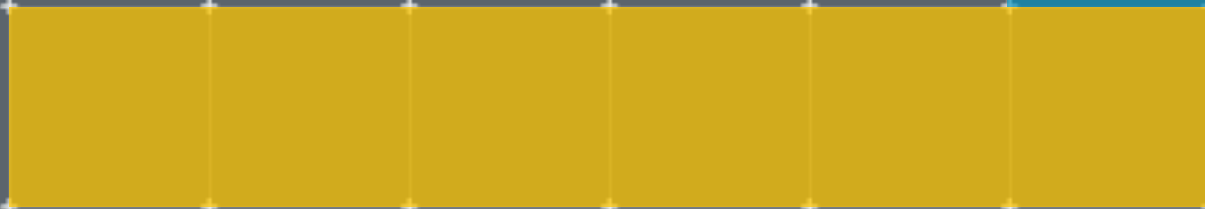
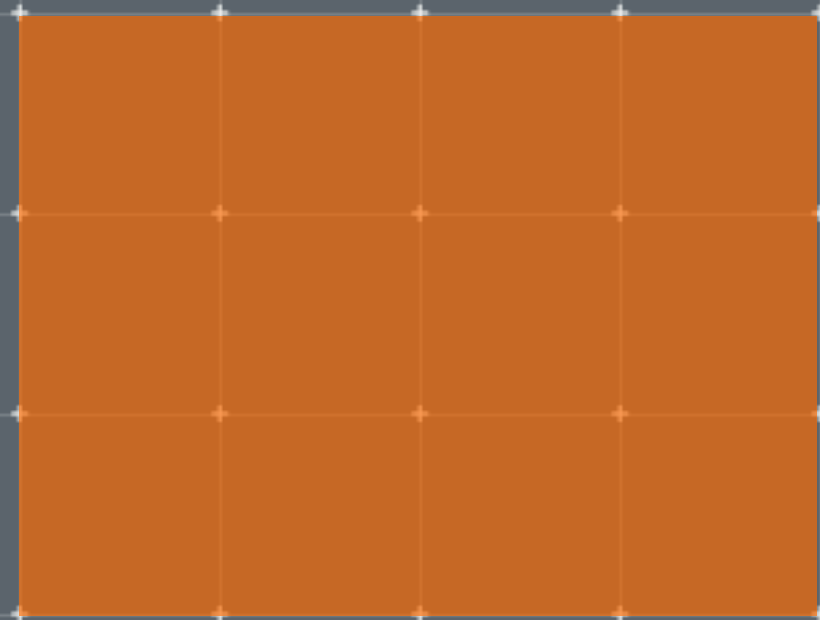
1. Patch is submitted for review. Changes  $n$  modules.
2. Patch author adds reviewers:
  - 1 code owner per module
  - 1 maintainer
3. Code owners review and approve the patch.
4. Maintainer reviews and approve the patch.
5. Maintainer merges the patch in the tree.

Note: 3 and 4 may happen in parallel while review progresses but maintainer has final word.

# Life Cycle for Platform Support

- Platform support involves maintaining:
  - Code (e.g. API breakages)
  - Tests
  - Documentation
  - OpenCI (soon)
- Define several levels of support
  - Fully supported
  - Limited support
  - Orphan
  - Out of date
  - Deprecated
- Users know what is supposed to work on a platform
- Define a process to deal with deprecated platforms

# Community Feedback



# Early Feedback on the Projects' Mailing Lists

- How to avoid review bottlenecks?
  - Allow self-review.
  - **Define minimum wait time for feedback before merge.**
  - Mandate at least 2 reviewers (distinct code owner and maintainer).
  - Post-merge review (requested or voluntary).
- Platform life cycle needs to be refined (too many states?)
- Split out "*Module has an active owner*" from "*What's working/broken*"
  - Could be orphan, but still working for some time
  - Could be actively maintained but temporarily broken (e.g. API migration)
- Add versioned project's list of features. Each platform fills it in.
- Standard labels for platforms/versions/features in ticketing system



# Some things are better defined at project scope

- Add a complementary project-specific document
- Patch review timeline ("All patches must be reviewed in a *timely* manner")
  - How long to wait for initial feedback?
  - How long to wait for comments to be addressed?
  - Silence == approval?
- Process for transitioning platform states (and how to give notice)
- How to measure code owners & maintainers (in)activity?
- Code review guidelines
- Testing requirements
- Branching strategies (separate maintainers for different branches?)

# References

- Proposal

<https://developer.trustedfirmware.org/w/collaboration/project-maintenance-proposal/>

- RFC on TF-A mailing list (replicated to TF-M, OP-TEE and TSC)

<https://lists.trustedfirmware.org/pipermail/tf-a/2020-March/000337.html>

Thank You

Danke

Merci

谢谢

ありがとう

Gracias

Kiitos

감사합니다

धन्यवाद

תודה

arm

# arm

The Arm trademarks featured in this presentation are registered trademarks or trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. All rights reserved. All other marks featured may be trademarks of their respective owners.

[www.arm.com/company/policies/trademarks](http://www.arm.com/company/policies/trademarks)