RChain Improvement Process (RIP Process)

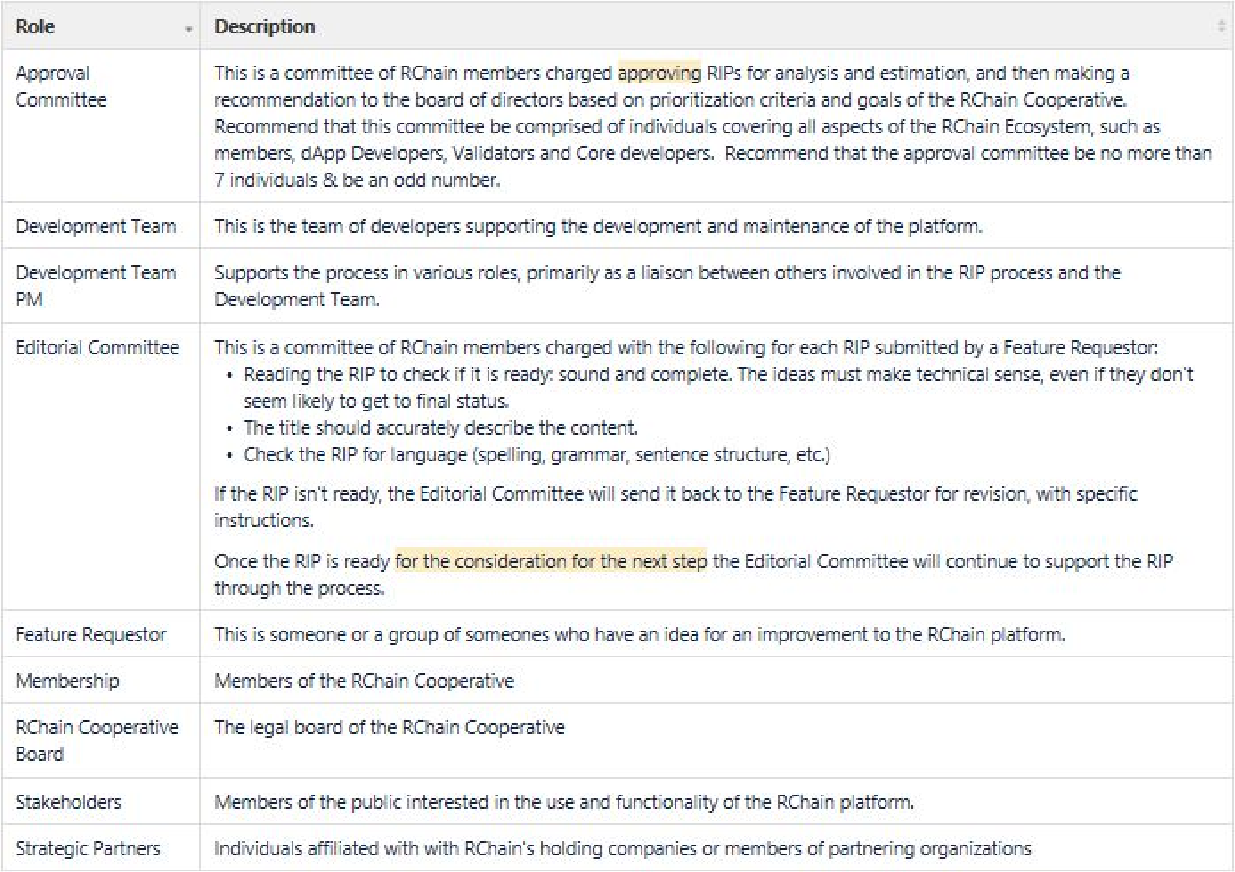
This document seeks to propose a process by which updates to the RChain blockchain are defined, approved, prioritized and released.

Tooling:

The RChain feature backlog is captured in Jira at: http://rchain.atlassian.net at: https://rchain.atlassian.net/secure/RapidBoard.jspa?projectKey=RIP&rapidView=19&view=planning.nodetail

The RChain Improvement Proposals are captured in Confluence at: RChain Improvement Proposals

Actors:



Role	Description
Approval Committee	This is a committee of RChain members charged approving RIPs for analysis and estimation, and then making a recommendation to the board of directors based on prioritization criteria and goals of the RChain Cooperative. Recommend that this committee be comprised of individuals covering all aspects of the RChain Ecosystem, such as members, dApp Developers, Validators and Core developers. Recommend that the approval committee be no more than 7 individuals & be an odd number.
Development Team	This is the team of developers supporting the development and maintenance of the platform.
Development Team PM	Supports the process in various roles, primarily as a liaison between others involved in the RIP process and the Development Team.
Editorial Committee	This is a committee of RChain members charged with the following for each RIP submitted by a Feature Requestor: • Reading the RIP to check if it is ready: sound and complete. The ideas must make technical sense, even if they don't seem likely to get to final status. • The title should accurately describe the content. • Check the RIP for language (spelling, grammar, sentence structure, etc.) If the RIP isn't ready, the Editorial Committee will send it back to the Feature Requestor for revision, with specific instructions. Once the RIP is ready for the consideration for the next step the Editorial Committee will continue to support the RIP through the process.
Feature Requestor	This is someone or a group of someones who have an idea for an improvement to the RChain platform.
Membership	Members of the RChain Cooperative
RChain Cooperative Board	The legal board of the RChain Cooperative
Stakeholders	Members of the public interested in the use and functionality of the RChain platform.
Strategic Partners	Individuals affiliated with with RChain's holding companies or members of partnering organizations

Process:

Specify the Feature

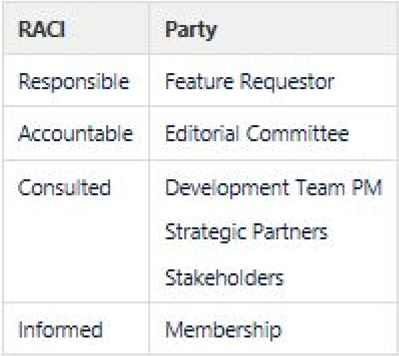
A feature request should be proposed in written format. For small features, a Jira Story issue is suitable.

If a feature cannot be implemented within a single component (Interpreter, Consensus, Node), then an Epic is required. An Epic should contain a link to a document that describes the feature with enough specificity that an Engineering plan can be drafted and estimation can take place.

Required Information:

- Category for Prioritization
- Score for Category from 1 → 10 (1 being lowest, 10 being greatest)
- Metrics to measure success (specific goal that justifies the score)
- Detailed description of the feature & benefits
    - Does the feature open up new markets, if so how?
    - Are users blocked from doing something without the feature? Or is this a UX change?
    - Are there contractual obligations related to the feature? (This alone won't bump priority. Contracts should never be signed against un-implemented features)

RACI



RACI	Party
Responsible	Feature Requestor
Accountable	Editorial Committee
Consulted	Development Team PM Strategic Partners Stakeholders
Informed	Membership

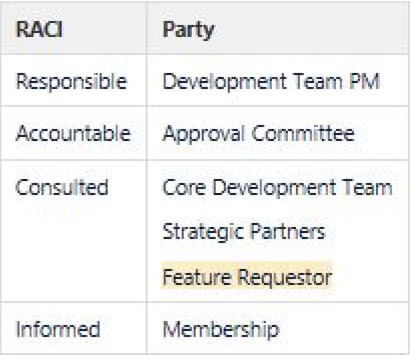
Approval for Analysis

Proposed features should be vetted for completeness appropriateness and value before being analyzed. Analysis is a costly process, and should not be applied to all proposed features without some kind of prior vetting.

Criteria:

- Is the request complete?
    - Category present & Scored?
    - Feature is described with enough detail for analysis
    - Metrics to measure success are described
    - Acceptance criteria are described sufficiently.

RACI



RACI	Party
Responsible	Development Team PM
Accountable	Approval Committee
Consulted	Core Development Team Strategic Partners Feature Requestor
Informed	Membership

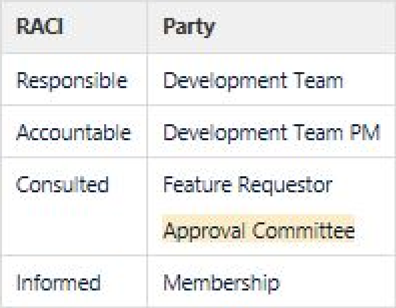
Analysis & Estimation

Once a feature has been specified and approved for analysis, the engineering team performs an analysis and proposes an implementation. This analysis will include the following:

- A review of the specification / description of the feature.
- High level design.
- Estimate of effort in story points.
- Tickets for the work along with the estimate are linked into the wiki page as individual ticket links (for the purpose of history tracking of scope changes)

This part of the process will likely involve a good deal of back and forth between the requestor and the development team, until agreement on the details of the feature is reached. Additionally, the feature may be re-scoped or re-sized as a result of these discussions.

RACI



RACI	Party
Responsible	Development Team
Accountable	Development Team PM
Consulted	Feature Requestor Approval Committee
Informed	Membership

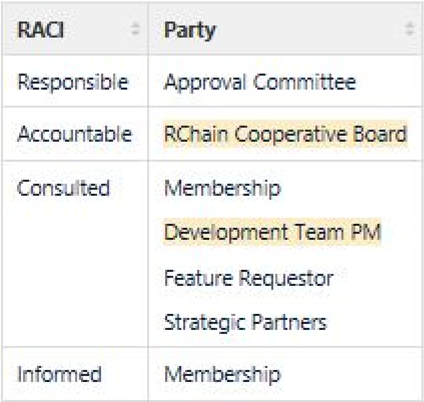
Approval for Implementation

Once estimated, the feature should be approved for implementation. If the feature presents as a hard fork, then approval from the validator pool is needed, to prevent a fork of the platform when the update is applied.

- There is a way for validators to signal their approval for the feature?
- There is a way for the CoOperative to signal their approval for the feature.

- This mechanism should be auditable and transparent.
- There needs to be some measure of a number of votes / approvals to signal consent.
- The Approval Committee should set the threshold for the number of votes to signal consent.
- Recommend that voting take place inside an issue, where persons can 'vote' on the approval for implementation. The votes will be tallied & an image captured on the implementation wiki page.

RACI

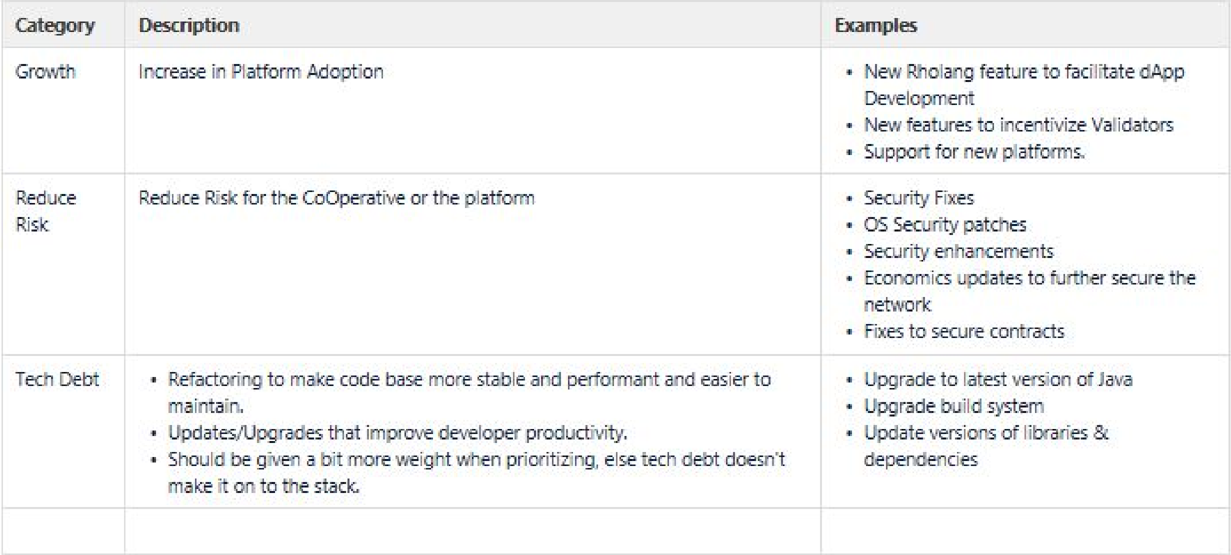


RACI	Party
Responsible	Approval Committee
Accountable	RChain Cooperative Board
Consulted	Membership Development Team PM Feature Requestor Strategic Partners
Informed	Membership

Prioritization

Once the above items have taken place, the feature will be slated into a release vehicle and the tickets will be assigned to a sprint. The prioritization of the feature will depend on its priority relative to other items on the backlog.   Prioritization of work is a challenge. Recommendation is that all RIPs be categorized into 1 of these categories, and given a relative weight in terms of value.

Strategic Categories



Category	Description	Examples
Growth	Increase in Platform Adoption	• New Rholang feature to facilitate dApp Development • New features to incentivize Validators • Support for new platforms.
Reduce Risk	Reduce Risk for the CoOperative or the platform	• Security Fixes • OS Security patches • Security enhancements • Economics updates to further secure the network • Fixes to secure contracts
Tech Debt	• Refactoring to make code base more stable and performant and easier to maintain. • Updates/Upgrades that improve developer productivity. • Should be given a bit more weight when prioritizing, else tech debt doesn't make it on to the stack.	• Upgrade to latest version of Java • Upgrade build system • Update versions of libraries & dependencies

Prioritization Process

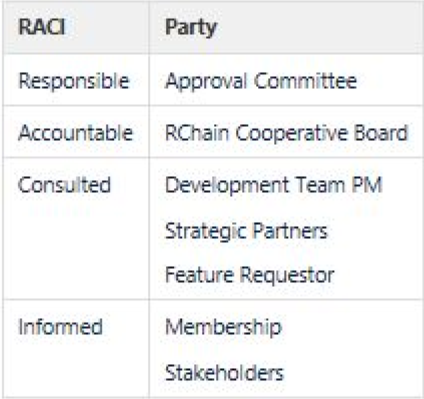
The Board of Directors should define the goals / theme for the platform. The goals for the platform should be set annually. Doing this more frequently isn't productive. Examples of such goals are:

- *"By the end of 2019, there will be 100 dApps & 100,000 transactions / week on the platform"*
- *"By the end of 2019, there will be 1000 developers proficient in Rholang"*
- *"By the end of 2019, the CoOperative membership will have grown to 100,000"*
- *"By the end of 2019, all governance tools for the CoOperative will operate on the RChain platform"*

Recommend that no more than 2 goals be set (1 for platform, 1 for CoOperative) for any given period. Any more than that, creates confusion. Recommend that the goal be annual. Any shorter than that, results in the same effect as having too many goals.

Using this theme, a prioritization committee can use the https://articles.uie.com/kj_technique/#close technique to prioritize the RIPs and communicate the priorities to the Development Team PM.

RACI



RACI	Party
Responsible	Approval Committee
Accountable	RChain Cooperative Board
Consulted	Development Team PM Strategic Partners Feature Requestor
Informed	Membership Stakeholders

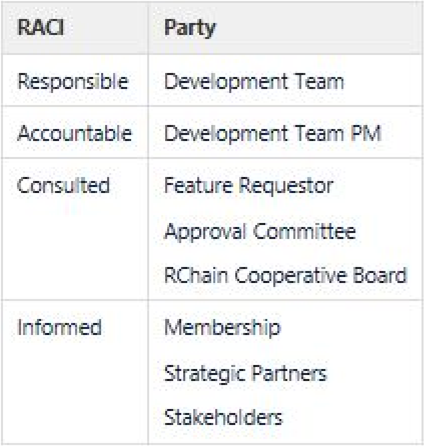
Implementation

The feature is implemented & Tested by Development

Acceptance testing & Release readiness

The feature is deployed to the public test net for acceptance testing. Stakeholders have the opportunity to examine the patch on their own systems.

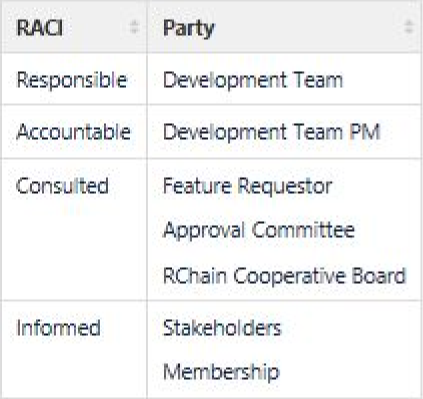
The acceptance testing process must also include release readiness criteria.



RACI	Party
Responsible	Development Team
Accountable	Development Team PM
Consulted	Feature Requestor Approval Committee RChain Cooperative Board
Informed	Membership Strategic Partners Stakeholders

Release

The feature (release package) is deployed to the public network.



RACI	Party
Responsible	Development Team
Accountable	Development Team PM
Consulted	Feature Requestor Approval Committee RChain Cooperative Board
Informed	Stakeholders Membership