

Codice Incubator Proposal Template

Version 1.0.0-SNAPSHOT

This template is a derivation of Apache's: <http://incubator.apache.org/guides/proposal.html>

Every proposal is different. There may be sections which don't seem to be useful. It's fine to leave them out and to add new ones that the proposal requires. Best practice evolves. Innovation is acceptable. The format is less important than the content.

Abstract

A short descriptive summary of the project. A short paragraph, ideally one sentence in length. The abstract should be suitable for reuse in the board resolution used to create the official project upon graduation.

Proposal

A lengthier description of the proposal. Should be reasonably declarative. More discursive material should be included in the rationale (or other later sections).

Background

Provides context for those unfamiliar with the problem space and history. Be sure to address any contractual or organizational culture issues that may need to be addressed as part of an incubation process.

Explain terms whose meanings may be misunderstood (for example, where there is not a single widely adopted definition).

Rationale

Explains why this project needs to exist and why should it be adopted by Codice. This is the right place for discursive material.

Initial Goals

A complex proposal (involving multiple existing code bases, for example) may cause concerns about its practicality. A good way to address these concerns is to create a plan that demonstrates the proposal is feasible and has been carefully thought through.

Current State

This section (and the contained topics) describes the candidate's current status and development practices. This should be an honest assessment balancing these against Codice's principles and development ideals.

For some proposals, this is a chance to demonstrate understanding of the issues that will need to be addressed before graduation. For others, this is a chance to highlight the close match with Codice that already exists. Proposals without an initial code base should just simply state that.

Meritocracy

Codice is a [meritocracy](#).

Once a developer has submitted enough good patches then it should be natural that they are elected to committer. It should be natural that active committers are elected to the project management committee (PMC).

This process of renewal is vital to the long term health of Codice projects. This is the right place to demonstrate that this process is understood by the proposers.

Community

Codice is primarily interested in communities. Candidates should start with a community and have the potential to grow and renew this community by attracting new users and developers. Candidates with a single developer or all development activity within a single corporate entity are at a much greater risk with regard to establishing community participation.

Explain how the proposal fits this vision and will go about improving community participation.

Core Developers

Codice is composed of individuals.

It is useful to provide a brief introduction to the developers on the initial committers list. This is best done here (and not in that section). This section may be used to discuss the diversity of the core development team.

Alignment

Describe why Codice is a good match for the proposal. An opportunity to highlight links with Codice projects and development philosophy.

Known Risks

An exercise in self-awareness. Risks don't mean that a project is unacceptable. If they are recognized and noted then they can be resolved during incubation. At a minimum be sure to address the following risks.

Orphaned products

A public commitment to future development. Recruiting a diverse development community and strong user base takes time. Codice needs to be confident that the proposers are committed.

Inexperience with Open Source

If the proposal is based on an existing open source project with a history of open development, then highlight this here. If the list of initial committers contains developers with strong open

source backgrounds then highlight this here.

Inexperience with open source is one reason why closed projects choose to apply for incubation. Codice has developed over the years a store of experience in this area. Successfully opening up a closed project means an investment of energy by all involved. It requires a willingness to learn and to give back to the community. If the proposal is based around a closed project and comes with very little understand of the open source space, then acknowledge this and demonstrate a willingness to learn.

Homogenous Developers

Healthy projects need a mix of developers. Open development requires a commitment to encouraging a diverse mixture. This includes the art of working as part of a geographically scattered group in a distributed environment.

Starting with a homogenous community does not prevent a project from entering incubation. But for those projects, a commitment to creating a diverse mix of developers is useful. Those projects who already have a mix should take this chance to highlight that they do.

Documentation

References to further reading material.

Initial Source

Describes the origin of the proposed code base. If the initial code arrives from more than one source, this is the right place to outline the different histories.

If there is no initial source, note that here. Also be sure to explain and provide any technical documentation that may exist, or be currently under development.

Source and Intellectual Property Submission Plan

Complex proposals (typically involving multiple code bases) may find it useful to draw up an initial plan for the submission of the code here. Demonstrate that the proposal is practical.

External Dependencies

External dependencies for the initial source is important. Only some external dependencies are allowed by Codice policy. These restrictions are (to some extent) initially relaxed for projects under incubation.

If the initial source has dependencies which would prevent graduation then this is the right place to indicate how these issues will be resolved.

ITAR/Export Constraints

If the proposal involves cryptographic code either directly or indirectly, Codice needs to know so that the relevant paperwork can be obtained.

Required Resources

Resources that infrastructure will be asked to supply for this project.

Google Groups / Mailing lists

The minimum required lists are project-private (for confidential PMC discussions) and project-dev lists.

If this project is new to open source, then starting with these minimum lists is the best approach. The initial focus needs to be on recruiting new developers. Early adopters are potential developers. As momentum is gained, the community may decide to create commit and user lists as they become necessary.

Existing open source projects moving to Codice will probably want to adopt the same mailing list set up here as they have already. However, there is no necessity that all mailing lists be created during bootstrapping. New mailing lists can be added by approval from the board.

It is conventional to use an all lowercase, dash-separated (-) prefix based on the project name. By default, commits for {podling} will be emailed to {podling}-commits. It is therefore recommended that this naming convention is adopted.

Example (project):

- * project-private (with moderated subscriptions)
- * project-dev
- * project-commits
- * project-user

The foundation is researching other alternatives for private, PMC communications as well, such as tools like HipChat, Campfire, and Flowdock.

Source Code Management

Where is the code currently, and what is the future plan. The expectation is that all Codice projects will use Bitbucket.org to host the “master” repository.

Agile Planning and Issue Tracking

Agile planning and issue tracking are managed with JIRA. Each project can have multiple sub-projects with the tool, to best manage the development, delivery and maintenance.

Wiki and Code Review

Each project will be expected to use the wiki for collaboration on live documents, including usage and source documentation. Projects will have the opportunity to use an integrated code-review tool, but at the discretion of the mentor and PMC.

Other Resources

Describe here any other special infrastructure requirements necessary for the proposal. Note that the infrastructure team usually requires a compelling argument before new services are allowed on core hardware. Most proposals should not require this section.

Most standard resources not covered above (such as continuous integration) should be added after bootstrapping.

Initial Committers

List of committers (stating name and an email address) used to bootstrap the community. Mark each which has submitted a contributor license agreement (CLA). Existing committers should use their Codice.org email address (since they require only appropriate karma).

It is a good idea to submit CLAs at the same time as the proposal. Nothing is lost by having a CLA on file at Codice but processing may take some time.

Sponsors

Champion

The Champion is a person already associated with Codice who leads the proposal process. It is common - but not necessary - for the Champion to also be proposed as a mentor.

A Champion should be found before the proposal is formally submitted.

Mentors

They don't have to be Codice-related.