# Structure of the Technical Community

The Technical Community consists of multiple projects and a Technical Steering Committee that spans across all projects.

# Project Management

## Project Roles

### Contributor

A Contributor is someone who contributes to a project. Contributions could take the form of code, code reviews, or other artifacts. Contributors work with a project’s Committer and the project’s sub-community. A Contributor may be promoted to a Committer by the project’s Committers after demonstrating a history of meritocratic contribution to that project.

### Committer

For each project there is a set of Contributors approved for the right to commit code to the source code management system (the “Committers”) for that project.

* Committer rights are per project; being a Committer on one project does not give an individual committer rights on any other project.
* The Committers will be the decision makers on all matters for a project including design, code, patches, and releases for a project.
* Committers are the best available individuals, but usually work full-time on components in active development.

#### Adding Committer

* Initial Committers for a project will be specified at project creation. In order to preserve meritocracy in selection of Committers while ensuring diversity of Committers, each initial project is encouraged to taking on at least three Committers from different companies (subject to meritocracy).
* Committer rights for a project are earned via contribution and community trust. Committers for a project select and vote for new Committers for that project, subject to TSC approval.
* New Committers for a project should have a demonstrable established history of meritocratic contributions.

#### Removing Committer

A Committer may voluntarily resign from a project by making a public request to the PTL to resign (via xgvela-tsc email lists).

A Committer for a project who is disruptive, or has been inactive on that project for an extended period (e.g., six or more months) may have his or her Committer status revoked by the project’s Project Technical Leader or by 2/3 super-majority vote of the project’s committers.

The Project Technical Leader is responsible for informing the Technical Steering Committee (TSC) of any committers who are removed or resign via the xgvela-tsc email list.

Former committers removed for reasons other than being disruptive may be listed as ‘Emeritus Committers’. That title expresses gratitude for their service, but conveys none of the privileges of being a Committer.

#### Adding Committer to moribund projects

In the event that a project has no active committers (e.g., due to resignations, etc.), the TSC may appoint an interim Committer from a project’s active Contributors. This term shall last until the next release date, after which time the Committer must stand for election from amongst other Committers on the project to maintain his or her status. In this special case, approval requires a majority of committers who respond within two weeks. If no one responds by the deadline, then the committer status is approved. This provision allows a project to continue development following an unexpected change in personnel.

The method by which the TSC appoints an interim committee is first by request to the XGVela-TSC email list indicating the request to appoint an interim Committer for a project. After the reception of such an email, the normal TSC decision process applies.

### Project Technical Leader

A project is required to elect a Project Technical Leader (“PTL”). The PTL acts as the de facto spokesperson for the project.

#### PTL Candidates

Candidates for the project’s Project Technical Leader will be derived from the Committers of the Project.

Candidates must self nominate.

#### PTL Voters

Only Committers for a project are eligible to vote for a project’s Project Technical Lead.

#### PTL Election Mechanics

An election for Project Technical Leader occurs when any of the following are true:

* The project is initially created
* The Project Technical Leader resigns his or her post
* The majority of committers on a project vote to call a new election
* One year has passed since the last Project Technical Leader election for that project

## Project Decision Making Process

Technical and release decisions for a project should be made by consensus of that project’s Committers. If consensus cannot be reached, decisions are taken by majority vote of a project’s Committers. Committers may, by majority vote, delegate (or revoke delegation) of any portion of such decisions to an alternate open, documented, and traceable decision-making process.

## Project Lifecycle

### XGVela project lifecycle Overview

The activities of the XGVela community are articulated around projects and releases. The scope of each project is aligned with the XGVela architecture and the scope of each release is defined with the objective to fulfill a particular use case(s).

A project is a long-term endeavor setup to deliver features across multiple releases, which have a shorter lifespan.

The project lifecycle provides the freedom for each team to conduct its project according to their needs, culture and work habits. Thus, the project lifecycle is not prescriptive on how each project operates.

An XGVela release can be composed of 1 to N projects. As such the number of contributing projects to a particular release may vary overtime.

A release is initiated to deliver a set of project deliverables.

The project lifecycle process does not impose a duration for the project nor for the release. There is an independent Release Plan document for each release to specify release timelines.

### Project Type

As PaaS is a group of independent functionalities which are used case by case, XGVela projects are usually not tightly-connected with each other. Usually XGVela projects can be separated into two types:

* Code type: major delivery of a code-type project is code and document. It has one or a group of related functions (considering donators’ opinion)
* Doc type: major delivery of a doc-type project is only document. Mostly requirement doc, architecture doc for the entire platform or a group of platform users.

### Project Lifecycle States

The lifecycle of a project is depicted on the following diagram:

|  |  |
| --- | --- |
| Project State | Description |
| Proposal | Project doesn’t exist yet or not belong to XGVela. Project has not real resources from XGVela, but is proposed and is expected to be created due to business needs. |
| Incubation | Project has been established as a XGVela project. It has resources and can attend XGVela releases. |
| Mature | Project is fully functioning and stable, has achieved at least one successful release, and has been used by end users or support to be integrated by many commercial products. |
| Achieved | Project can reach Archived state for multiple reasons. Either project has successfully been completed and its artifacts provide business values, or project has been cancelled for unforeseen reasons (no value anymore, technical, etc.). |

To move from one state to the next state, the Project Team has to propose to TSCs about their moving-up goals, project data, and request for formal review discussion. TSC will vote on project state change.

|  |  |  |
| --- | --- | --- |
| From State | To State | Review Description |
| Proposal | Incubation | Incubation Review |
| Incubation | Mature | Maturity Review |
| Incubation/Mature | Archived | Termination Review |

## Project Review

### Review Progress

For all reviews, the following steps are required:

* The project review request and review materials are posted to XGVela-project-lifecycle Confluence (under creation).
* The project review is emailed to xgvela-tsc mailing list.
* Present state-changing proposal on TSC meeting and get vote by TSC members (usually more than half of attending TSC voted means approval).

### Review Contents

#### Incubation Review

The goal of Incubation Review is to officially launch the project as XGVela project and to support its needs until project Termination Review.

Once a project has passed the Incubation Review, the project is in Incubation State and may span over multiple releases.

To apply entering Incubation state, the following contents are required:

* Name of the project: (appropriate, no trademark issues etc.)
* Repository name: (lower-case without any special characters)
* Project contact: (name, company, email)
* Initial committer list: (name, company, email)
* Contributor list: (name, company, email)
* Meets XGVela TSC policy
* Project goal and purpose:
* Project functionality and major usage scenarios: (for code-type project, explain functionality of code, under what scenario the project should provide service; for doc-type project, explain brief document direction and using scenarios)
* Project relationship with General PaaS/ Adaptation Layer/ Telco PaaS: (used which General PaaS functions, this project functionality should be located in adaptation layer or telco paas)
* Relationship with other projects
* Project scope:
* Project release plan: (plan for at least two releases)
* Tools required:

#### Maturity Review

The goal of Maturity Review is to ensure:

* Artifacts for Incubation State are completed and accepted
* Project is mature to be used in open source/lab/commercial products/use cases
* Project will still have right to use XGVela resources

Once a project has passed the Maturity Review, the project is in Mature State and may span over multiple releases.

To apply entering Mature State, the following contents are required

* History release: (how many release attend and major deliveries in each release)
* Architecture get along with XGVela architecture:
* Project is active and contributes to XGVela: The project demonstrates a stable or increasing number of contributions across recent releases. Contributions are commits which got merged to a repository of an XGVela project or a related upstream project. Commits can for example be patches to update the requirements document of a project, code addition to an XGVela or upstream project repository, new test cases and so forth.
* Project usage: The project can be successfully deployed, configured, and used by end users. Or it can be successfully integrated by commercial products.

#### Termination Review

The goal of the Termination Review is to ensure that:

* Artifacts for Mature State are complete and accepted
* Mature project artifacts are acceptable
* Project Team has the confidence that its artifacts can be used outside the XGVela community
* Metrics for Termination review are available

## Release Process

A Project’s Committers make all decisions about Releases of that Project. However, to be able to measurable, the project must demonstrate a history of following the Release Process. The purpose of the Release Process is to insure openness and maximum opportunity for participation. The idea is to have a simple, clear, public declaration of what a project intends to do and when, and what was actually done in a release cycle. Towards that end, a project following the ‘Release Process’ should have a Release Plan published at the beginning of its release cycle by its committers, and a Release Review just prior to the project release.

Both Release Plan and Release Review documents are intended to be relatively short, simple, and posted publicly on the wiki to assist project in coordinating amount themselves and the general world in gaining visibility.

Release Plan and Release Review should contain roughly the following sections:

* Introduction
* Release Deliverables
* Release Milestones
* Expected Dependencies on Other Projects
* Compatibility with Previous Release
* Themes and Priorities
* Features delivered
* Non-Code Aspects (user docs, examples, tutorials, articles)
* Architectural Issues (if any)
* Security Issues (if any)
* Quality Assurance (test coverage, etc)
* End-of-life (API/Features EOLed in Release)
* Summary of Outstanding Bugs
* Summary of Standards Compliance
* Delta between planed schedule and actual schedule
* Other