# Proposal of Governance Model

A governance model is needed in an Open Source project to let the community know the order of how the things work internally in the project, the roles and responsibilities.

Repositories

The product will have two remote repositories, one *public* and other *private*, with different roles and responsibilities from the people that is involved in each repository.

History of a Commits

Every commit has to have some specific behavior, fist all tasks related to a change (features, bug fix, etc.) have to be implemented in a local development branch. We will follow the ideology of “*Branch early, branch often*” also we will encourage to describe every commit to give the Q&A and the Project Lead an idea of what the commit was for when they review the code.

The contributors with commit privileges to the private repository, only need to follow the mentioned rule. But the Contributors that want to push some code to the public repository need to follow an extra set of rules that comply with the pull request behavior.

The pull request behavior follows the following conduct:

1. Fork the public repository
2. Make changes to the code (following the previous rules using branches and commits over branches).
3. Then the user can create a Pull Request from a personal branch to the public repository.

Roles

Contributor

The contributors are those individuals that are make contributions to the project source code. There are 3 kinds of contributors depending in where they commit and the trust level they have. The kinds are the following:

* *External Contributor*: this are individual developers who are contributing to Android on their own behalf.
* *Internal Contributor*: this are individuals that used to be *External Contributors* but via achievements and history and after a screening process has been promoted to *Internal Contributor*.
* Internal Developer: this is a developer that have an economic remuneration to write code for the project. It has been screened, and authorized by an organization.

Here is a table that shows some characteristics of the kind of contributors:

|  |  |  |  |
| --- | --- | --- | --- |
|  | External Contributor | Internal Contributor | Internal Developer |
| Access to public repository |  |  |  |
| Access to private repository |  |  |  |
| Contribute on their own behalf |  |  |  |
| Has an economic remuneration |  |  |  |

All the contributors use basically the same technologies and are subject to the same requirements on code style and code testing requirements. Also they have to write their own test for the code they have done.

User / Tester

This are users that download and test the product in their own infrastructure. This role is important because this are people that eventually repot bugs and new feature requests and help shape the overall functionality and vision of the product.

Approver

This is the role assigned to a member of the project that has demonstrated their design skills and have made an important technical contribution to the project. They can review the code and test the pull requests and decide to include or exclude a request. Also they test constantly the commits made to the private repository also to include or exclude commits.

QA/Tester

Design and create extra testing scenarios than the one provided by the contributors. And can advice the Approvers about some possible scenarios where the code can break. They test everything that get into the Public and Private repository they can have their own project a side of the product where they can create more sophisticated testing scenarios.

Project Lead

When the product reach enough maturity it can be divided in sub-projects for every project there will be a Project Lead. This one are senior contributors who oversee the engineering for individual projects. Typically, will be Internal Developers and are responsible for the following:

* Lead all technical aspects of the project, including the project roadmap, development, release cycles, versioning, and quality assurance (QA).
* Ensure the project is tested by QA in time for scheduled for platform releases.
* Designate Verifiers and Approvers for submitted patches.
* Be fair and unbiased while reviewing Pull Requests. Accept or reject patches based on technical merit and alignment with the overall strategy.
* Review changes in a timely manner and make best efforts to communicate when changes are not accepted.
* Optionally maintain a web site for the project for information and documents specific to the project.
* Act as a facilitator in resolving technical conflicts.
* Be a public face for the project and the go-to person for questions related to the project.
* Apply the overall strategy to the processes.

NOTE: this document might change until come to the available