

Savanna development process

This document is a compilation from OpenStack and Launchpad wiki pages.

Launchpad groups

There are several different groups with different responsibilities and permissions:

- **savanna-maintainers** - team that is responsible for configuring launchpad project
- **savanna-drivers** - guys who create/release milestones, approve/assign blueprints, schedule/clean bugs. This is the team for core contributors (currently these are the people who wrote Savanna)
- **savanna-dev** - team for all Savanna contributors - guys who directly works on Savanna project
- **savanna-all** - team for everyone interested in Savanna project

Milestones and releases

- **Series:** parallel lines of development within the project, working towards a major release. See [Savanna timeline](#).
- **Series goals:** bugs or blueprints that will be fixed in a particular series.
- **Milestones:** points in the development of a series, such as future minor releases or release candidates.
- **Releases:** once a milestone results in a release of software, you can turn it into a release in Launchpad.

For details see:

- [OpenStack release cycles](#)
- [OpenStack branch model](#)
- [Page on Launchpad wiki](#)

Blueprints

A blueprint is a simple specification. It describes an idea - such as a feature or process - and tracks enough metadata to show its implementation status and who's involved.

In OS community Launchpad blueprints are used to track the implementation of significant features. Keeping their status current is critical to the success of the release and the project as

a whole. It avoids unnecessary reporting, pings and discussions, and keeps everyone on the same page.

Blueprint lifecycle

Creation

If you intend to work on a given feature, you should create a blueprint for that. Describe the feature summarily in the blueprint itself, and link to another document (using the specification link) if you have more. Note that you may track the peer-review of your blueprint using the Drafter and Definition fields. Once it is ready for PTL review, you should set;

- Approver: <Someone from **savanna-drivers** team>
- Assignee:
- Series goal: Proposed for
- Milestone: When you think your work will be proposed for merging

Inclusion in the release roadmap (PTLs)

The PTL reviews the blueprint, then may include it as a release goal:

- Definition: Approved
- Series goal: Accepted for
- Priority: (see above)

During development (assignee)

The "Implementation" field should reflect progress in your work:

- Implementation: (see above)
- Please update the implementation status regularly to avoid being pinged about it :)

When merged (assignee)

When the work is fully merged, finalize the spec by setting:

- Implementation: Implemented

For details see:

- [Page on OpenStack wiki](#)
- [Page on Launchpad wiki](#)

Bugs

Bug attributes

- **Importance:** Critical, High, Low, Undefined

- **Assigned To:** The person currently working to fix this bug. Must be set by *In progress* stage.
- **Milestone:** The milestone we need to fix the bug for, or the milestone/version it was fixed in.

Bugs lifecycle

Reporting

Anyone with Launchpad account can report a bug

Confirming & prioritizing

1. If you are lacking information to properly reproduce or assess the importance of the bug you should you should ask the original reporter for more information and set the bug status to *Incomplete*
2. Once you have reproduced the issue (or are 100% confident that this is indeed a valid bug), you should set bug status to *Confirmed*
3. Core developer (or a member of the project bug supervision team) should also prioritize the bug

Bugfixing

Developer working on the bug should:

- Set status to *In progress*
- Set assignee to himself

When the fix is ready, he will propose the change and get it reviewed.

Note that Gerrit will automatically set the status and assignee when a change is proposed that mentions the bug number.

After the change is accepted

Once the change is reviewed, accepted, and has landed in master, it will automatically move to:

- *Status:* Fix Committed
- When the fix makes it into a milestone or release branch, it will automatically move to:
- *Milestone:* Milestone the bug was fixed in
- *Status:* Fix Released

For details about bugs lifecycle see:

- [OpenStack wiki page](#)
- [Launchpad wiki page](#)

Gerrit Workflow

Setup

Skipped. See link to OpenStack wiki page below.

Review

- Once the code is committed, it appears at <https://review.openstack.org>
- Automatic testing will be done and the results will be displayed once they are complete. Reviewers will come and review and leave comments in the comment box or in the code itself.
- Any Openstack developer may propose or comment on a change (including voting +1/0/-1 on it). Openstack projects have a policy requiring two positive reviews from core reviewers. A vote of +2 is allowed from core reviewers, and should be used to indicate that they are a core reviewer and are leaving a vote that should be counted as such.
- When a review has two +2 reviews and one of the core team believes it is ready to be merged, he or she should leave a +1 vote in the "Approved" category. You may do so by clicking the "Review" button again, with or without changing your code review vote and optionally leaving a comment. When a +1 Approved review is received, Jenkins will run tests on the change, and if they pass, it will be merged.

For details see:

- [OpenStack Gerrit workflow](#)
- [Detailed Gerrit workflow on OpenStack wiki](#)

Other useful links:

[Contributing to OpenStack](#)