# Stabilizing a release

由 Myrle Krantz创建, 最后修改于一月 07, 2016

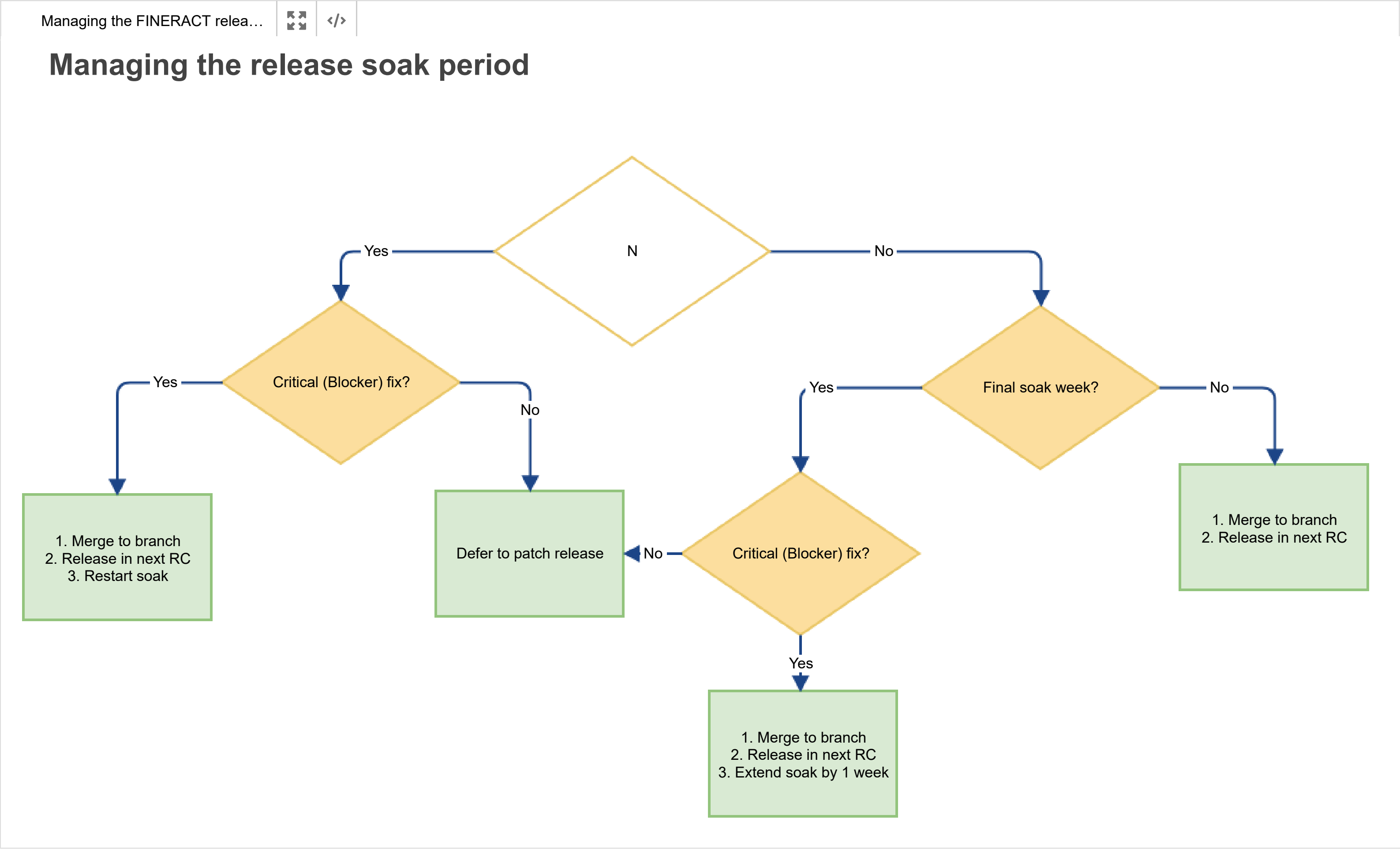
## Stabilization

Minor and major number releases go through a stabilization period before release, and remain in maintenance (bugfix) mode after release. To start the release process, we create an "A.B.x" branch based on the latest master.

The stabilization period for a new A.B.0 release normally lasts four weeks, and allows us to make conservative bugfixes and discover showstopper issues. The stabilization period begins with a release candidate with the version A.B.0-rc.1. Further release candidates may be made as blocking bugs are fixed.

At the beginning of the final week of the stabilization period, a new release candidate should be made if there are any showstopper changes pending since the last one. The final week of the stabilization period is reserved for critical bugfixes; fixes for minor bugs should be deferred to the A.B.1 release.

A picture speaks a thousand words:



Under some circumstances, the stabilization period will be extended.

- If a potentially destabilizing change must be made in order to fix a bug, the entire four-week stabilization period is restarted. A potentially destabilizing change is one which could affect many parts of Fineract in unpredictable ways, or which involves adding a substantial amount of new code. Any incompatible API change (only allowable in the first place if the new release is an A.0.0 release) should be considered a potentially destabilizing change.
- If a critical bugfix is made during the final week of the stabilization period, the final week is restarted. The final A.B.0 release is always identical to the release candidate made one week before (with the exceptions discussed below).

If there are disagreements over whether a change is potentially destabilizing or over whether a bug is critical, they may be settled with a committer vote.

After the A.B.0 release is out, patch releases (A.B.1, A.B.2, etc.) follow when bug-fixes warrant them. Patch releases do not require a four week soak, because only conservative changes go into the line.  As long as changes are backwards compatible however, we prefer to put them into new minor releases rather than continue releasing patches for an old minor release.

Certain kinds of commits can go into A.B.0 without restarting the soak period, or into a later release without affecting the testing schedule or release date:

- Documentation fixes.
- Changes that are a normal part of release bookkeeping.

Core code changes, of course, require voting, and restart the soak or test period, since otherwise the change could be under-tested.

## Voting

A change to the A.B.x line must first be proposed to the dev mailing list via a JIRA ticket.  The rest of the voting process takes place in JIRA.  The mail might look something like this: "I'm proposing that we include the solution to ticket FINERACT-20045 in release 1.22.0.  I'm calling on all committers and interested contributors to join the discussion there." Please don't use the mailing list for the actual discussion.

Before proposing adding a change, you should check that it doesn't produce merge conflicts in the master branch. If conflicts occur, please create a new temporary branch in your fork with your changes merged and the conflicts resolved. The branch should be named A.B.x-YYYY, where YYYY is the number of the FINERACT JIRA ticket. Add a note in the JIRA ticket about the existence of the temporary branch and which fork it can be found in. If the change involves further work, that work can be performed on the branch.

In order to make it possible to find all tickets in a given release, please make sure the fix version on the JIRA ticket is set. While a ticket is in voting, add the label "submitted_for_vote" to the ticket.  Once the vote is complete, remove the label.  If the ticket is voted out, change the fix version appropriately.

A change needs three +1 votes from committers and no vetoes to go into A.B.x.

A vote should be added to the bottom of the ticket description with the following information:

- an at-most-one-line justification of why it should be in A.B.x
- optionally some notes (brief summaries to help a reader get oriented – actual discussions take place in the ticket comments),
- optionally some concerns (again, brief summaries) and finally

- the votes and vetoes.

If you cast a veto (i.e. -1), please summarize the reason in the concerns field, and start the discussion by commenting on the JIRA ticket.

Voting +1 on a change means that you have reviewed the change using the community guidelines for code reviews. The +1 voters, the proposer, the programmer, and the committer that pulls a change into a release are all equally responsible for any resulting bugs.

Non-committers are welcome to vote via the JIRA ticket comments, thus taking on shared responsibility for a change.  Their vote is non-binding.

If you add revisions to a ticket, you can create a new block and skip any sections that are redundant.  Use the same schema if you want to vote a change into a later release.  You might want to do this if you believe it requires further review or testing.

Here's an example:

```
Proposed for version 1.2.1 by @mage
Justification:
  Mindreading is what every user always wanted.
Notes:
  There was consensus on the desirability of this feature.
Concerns:
  Vetoed by @myrle due to privacy concerns.
Votes:
  +1: @mage, @gstein
  +1: (non-binding): @stylesen
 -1: @myrle

Proposed for version 1.3.0 by @myrle
Justification:
  We should be able to isolate it to the prefrontal cortex with another month of development effort.
Votes:
  +1: @myrle
```

Votes are only communicated by comments on the JIRA ticket, and not by any other means. If you believe the information you or someone else is adding to the discussion is interesting for the broader community, please send a mail to dev that the issue is being discussed on ticket FINERACT-XYZ.

If you have reviewed a change and like it but have some reservations, then ask questions and communicate concerns using the JIRA ticket comments. If you wish to track down people who are following the change and might be willing to spend more time on it, the watcher list on its JIRA ticket is a good place to start.

*(Thanks to the Subversion team for the content and ideas we 'borrowed' from the Subversion release guideline)*

无标签

## 3 评论

Sander van der Heijden 发表：

Is a 4 week soak period going to be realistic at this stage with last years quarterly release calendar or would a shorter period also be appropriate looking at the current release process?

Myrle Krantz 发表：

I don't have a strong opinion on this.  In my professional experience at other projects a 3 to 4 week soak/alpha phase/stabilization phase has been a good idea even with quarterly releases.  But I'm new to open source and to FINERACT, so I'm willing to try out something different if you want to.  What do you suggest?

Markus Geiss 发表：

One thing to remember is that we don't have a sort of freeze on the current code base. While preparing a release the development will go on at the develop branch. If we go with a continues release plan ,like today, the soak phase will be shorter then 4 weeks. If we turn back to quarterly releases the phase needs to be longer given that more changes were introduced.