When to branch

As of 2023: In recent years, experience has shown that it takes about 40 days after branching to comfortably produce a Tiki release. It would be nice to have it be shorter with a nice clean 30 days, but in practice, this puts pressure on the community and release managers. It's not that there is that much work in number of hours (because trunk/master/main is in relatively good shape before branching, and the release process is almost all automated). However, we have multiple phases (alpha, beta, release candidates) and multiple sites to update (most of the *.tiki.org sites), which we do in phases, from less critical to most critical. So, we need to proceed to the upgrades, and allow enough time for feedback to come in, and to deal with it. If the schedule is too right, we don't have leeway to deal with unexpected issues. Thus, it's better to acknowledge that a regular release process should take about 40 days. The release managers can schedule the various steps at a good pace (as if it was 30-35 days), and leave the 5-10 days buffer at the end. So either it is used to deal with unexpected issues, or to give more time for the community to fix bugs, and improve the documentation. These are general guidelines, and the release manager can shorten or extend as per When to release - think popcorn

During this 40-day period, any risky work should be done in a merge request or a feature branch. We want innovation and experimentation to happen at all times, and be visible to the community.

The content below is kept for posterity.

Historically, we branch several weeks/months before an official release. However, it would also be possible to agree on a freeze and branch much later. Please add your arguments and preferences below.

Benefits of branching early (ex.: 4-6 weeks before)

- Devs can still work on unfinished or new features
  - after all, not everyone may want to just fix, or would have to make himself familiar with unknown code first
- (Psychological) Signal that freeze is in effect (only disadvantage in case if we do the way as in past merging to trunk is that people need to SVN switch, see below **Merge or not to merge**)
- no accidental breaking code by devs not knowing of feature freeze (living in a bubble?)
- clear and better controllable border between trunk and feature-freezed code
  - try on trunk. working? backport to branch
    - new and unexperienced devs won't have to fear of doing something really bad when working on trunk

Accept	Undecided	Reject
2	0	1
• Kissaki • luci		• marclaporte

Benefits of branching as late as possible

- Less work of merging from branches/4 to trunk (in case we need to merge - see below)
  - and possibly the other way around
- (Forcibly) Route manpower to only fixing and improving present and presently working (freezed) features.
- When release is soon coming, new features can be done in an experimental branch (and merged into trunk later)

Accept	Undecided	Reject
2	2	1
• jonnybradley • marclaporte	• luci • Kissaki	• sylvieg

Data

To help with decision making, it would be nice to have some historical perspective

Version	Branch date	.0 release date (SecDB creation)	Freeze duration (days)	Date last merge
3.x	2009-02-28 r16998	2009-05-19 r18916	80	2009-05-19 r18920 (18880 to 18916)
4.x	2009-11-03 r22815	2009-11-16 r23323	13	2009-12-18 r23957 (23892 to 2390□
5.x	2010-03-09 r26025	2010-06-07 r27534	61	2010-08-07 r28381 (28273 to 2837□
6.x	2010-09-21 r29485	2010-11-09 r30607	49	2010-12-14 r31414 (31378 to 31409)
7.x	2011-03-24 r33628	2011-06-09 r34878	77	2011-08-16 r36217 (36203 to 36216)
8.x	2011-10-03 r37900	2011-11-10 r38776[1]	38	?

[1] 8.1 data. 8.0 was released a week earlier, but was broken and never announced.

Marc's thoughts

Some things to think about:

- How many hours will be needed (can differ from version to version)
  - Tiki24 was a small, incremental release so it takes fewer hours
  - Tiki25 is a huge release with major non-optional changes like Bootstrap 4 to 5
- How many weeks to absorb changes
  - Ex.: coordinate with many people about upgrading various sites
- Continuous access to Pre-dogfood servers during the release process help keep trunk stable and thus, help reduce work during the release period.

Conclusion: We should fix all known regression bugs before branching in order to avoid backports, and thus proceed to serious testing on Pre-dogfood servers. We should try to have the shortest realistic branching period, and this depends on the state of trunk. Even if trunk is in super good shape, it's unrealistic to have it faster than 3 weeks (for a small release) because we have many actions to coordinate with many people (like upgrading various sites, getting feedback, etc.). For a big release, we should try to keep branching to a maximum of 6 weeks.

Related

- Semi-automatic merging period
- when to release
- Translation branching strategy