Contributions of each Team to the release process

The Tiki community is organized in many Teams. At release time, here are the responsibilities for each.

Table of contents

Technical

Release coordinator

- Overall management & planning, schedules (branching and releasing date).
- Also known as "Release manager"
- See Release Team

Developers

1. Make Tiki good enough (fix bugs, etc.) so that it can be released by the Packaging Team
2. Provide Documentation Team with basic information about features.
3. Update all Developer documentation to take into account new version number and features. Ex.: Download, Commit, Hello World, etc....
4. Review all Experimental branches and delete if appropriate (could be done at any time, but a 6-month reminder is good).

Packaging

1. Run Release scripts for packaging all releases (Alpha/Beta/RCs/Official)
2. Make sure everything is OK with Packages
3. Submit to and keep up to date each of the 1-click installers and Distro
4. Maintain Installation guides

Security

1. Review all previously reported issues on dev & sent to security list.
    1. Ask bug reporters how they would like to be acknowledged.
2. Contact all people that have helped in the past.
3. Proceed to security audit as per our release procedures.
    - run doc/devtools/securitycheck.php and check each "potentially unsafe" file.
    - Check for presence of all .htaccess files
    - Add files to robots.txt (printed pages, etc.)
4. Update security.tiki.org with sections for new version
5. Run Security DB

Performance

- Monitor performance for each release before .0 is released
    - Google PageSpeed
    - Yahoo! YSlow
    - etc.

Continuous Integration

If (when!) the Continuous Integration Team will have everything set up, there should be nothing special to do with respect to the release.

Infrastructure

- Progressively update each Domain to the new version, and the Dogfood Team's plan.

- The goal is that all major sites are upgraded before it's released.

UX & Themes

1. Make use of the test all themes profile and you will then have a drop down to easily successively test all themes.
   - Install the Tiki version to be released and apply successively the featured profiles.
   - Use the Pre-Dogfood Server to make sure themes work well with real world data.
   - Test with major browsers, including on desktop/laptop and mobile devices (tablets and phones).
   - Click around and fix what you can (layout, bad colors, missing padding, etc.).
     - Make sure to try to fix main problem so it fixes in all themes (ex.: layout CSS).
     - Keep in mind that bundled serve as a base for most of the custom themes that are made. Thus, they need to be as good as possible!
2. Make sure basic features are intuitive and appealing for end users and content creators. For site admins, see Configuration Profiles Team.
   - Pretend you are a end-user and install Tiki and address any unclear user interface.

Non-technical

Documentation

1. Update Features list & ratings
2. Make doc:Tiki11 nice
3. Make sure basic docs are updated (Requirements, Download, Backup, Install, Upgrade and Initial Config, etc.)
4. Run Preferences report
   - check all links from the app to the sites -> Add & Remove to doc:Keywords & dev:Keywords see Restore Help Pages
   - Put the report on doc.tiki.org
5. Make sure all new features have at least a stub.
6. Make an .odt and .pdf version of the current documentation for stable releases.

Communications

1. Publish news on tiki.org, SourceForge, Freshmeat, etc.
2. Send tiki.org newsletters to subscribers:
   1. Tiki Releases
   2. Tiki News
3. Update info on all Listings
4. Inform Try Tiki sites. (OpenSourceCMS.com, etc)
5. Promote on all relevant Social networking sites. (Facebook, etc.)
6. Promote on all Wiki and CMS-related resources (ex.: CMScritic.com)

Branding

1. Update Fact Sheet
2. Update Promo Sheet

Community Building

1. Identify new volunteers
2. Monitor support forums and help identify any blockers & regressions.

Configuration Profiles

1. Determine which profiles are "Featured" and make them excellent
2. Make sure admin panels are intuitive and appealing for site admins. For end users and content creators, see UX and Themes Team.
    - Go through the steps of a new end-user installing Tiki, and note and correct any unclear user interface.

Wishlist Triage

1. Test/triage all reported patches
    - Contact them to invite them to commit directly and be active in the community
2. Test/triage all reported bug reports
    - Test the fixes and close the bugs
3. Review How to Submit a new item on the Wishlist to make sure it's still relevant and update to take advantage of new features.
4. Carry out manual test plans: Instructions for Tiki Testers

Dogfood

1. Determine an upgrade calendar for all relevant domains, from less risky to most risky. Order is Branding, Themes, Documentation, Community (tiki.org), Development, Profiles. It's the ultimate DogFood and the goal is that all major sites are upgraded before it's released.
2. In collaboration with the Infrastructure Team, proceed to these upgrades and ensure everything is still working fine.
3. After the upgrade, implement new features on *.tiki.org to help test and make the community more open efficient and welcoming
    - In the weeks and months before an upgrade, this can be tested daily thanks to the Pre-Dogfood Server
    - Update Dogfood with any new features used. This serves as a case study.
4. Review Tiki Welcome, as all new registrants are sent to it after validating their email link.

i18n

1. Remove any out of sync English strings
2. Get in contact with all past translators and coordinate updates
3. Upstream/merge/resolve potential conflicts of translations in stable branch to trunk
4. Help translators prioritize most important information to translate. Ex.: Promo Sheet.

Legal

- Check the licenses of all newly included or updated code.

Video

Tentatively:

- Confirm that video resources management pages/trackers (coming soon™) are updated and current-version compatible.
- Handle any problems with listed videos becoming out of date
    - Revise video if possible, especially for most-needed videos.
    - Mark as out of date/archived if not.

Analytics

1. Provide stats to the Communications Team
    - How many commits, committers, etc.
    - Monitor % of completion stats, which is mostly automated at http://i18n.tiki.org/Status (now offline, but it will return)
    - any other useful stats
2. Update List of all code contributors

Consulting

None identified at the moment

Finance

None identified at the moment

Fundraising

None identified at the moment

Admins

- At each release, Admins do whatever needs doing, and whatever falls between the cracks.

Questions

If I participate in a release, how many hours do I need to invest?

There is no simple answer and "it depends".

1. The number of hours is "as many as we can get". The more people we are, the more we can split the load. And the easier it'll be for all.
2. There is a law of diminishing returns. Before putting hundreds of hours on X, we should try to improve Y which becomes our weakest link.
3. On the other hand, volunteers can't interchangeably do any task. It's according to their interest and type of skills.
4. And some teams are easier than others because of more involvement in the past. Ex.: The release scripts are so much easier than they once were.

This being said, globally, for most teams, if we get 15-25 hours. It makes a significant difference.

Assuming we have 20 people in all. And they each invest 15 hours. That 300 hours will make this release be very smooth.

And as we play our cards right and document everything (we are a Wiki community right?), the next release will be that much easier.

We are *not* in a micro-managed environment. Regardless of what team you are in, it is important to be flexible and attentive to the needs of other contributors which depend on them. For example, improvements in the release script (which might require some Developers' help), Documentation would need a script that checks all links to doc.tiki.org -> http://dev.tiki.org/How+to+release#script_to_check_all_links

alias

- Release Team Roles
- Release Roles