

Revisiting Tiki's translation process (2015)

What?

Product localization is the process of offering a tool, product, software in a different language than it was firstly conceived. When we talk about this process, there is an important subject that always comes to mind. We as a community that to be able to understand that translation and localization are distinct things.

1. Translation

This applies to fairly literal, "word for word." This is often out of necessity. If you want to make sure that a person in Japan understands how to use a product, it is important that the source and target-language text match up precisely.

1. Localization

This is a more involved process whereby the target-language content is adapted to more effectively convey a similar meaning or connotation in the target culture. The key point here is that your target-language version will often not be a literal translation. As an example, if you want to convey the phrase "Like father, like son" in Chinese, it would read as something like "Tigers do not breed dogs."

Who?

Ideally, open source projects who get to this maturity level, should have peers responsible for recruiting, training, maintaining infrastructure and troubleshooting. Volunteers for these responsibilities are:

- gezza
- msaad
- Torsten

When?

As soon as possible. L10N is a topic we need to catch up with. We have to stimulate people interested in contributing with localization by providing better ways for them to work with us.

Why?

It is not so easy to translate Tiki as it should/could be. A translator is not necessarily a developer, for example an administrative person at a company who speaks multiple language can be assigned to do translation, but we can not expect a user will have to download source files, get commit access, learn how to commit in SVN, etc. The contribution process has to be more organic.

We have interactive translation, but it is only for what is visible, for example you can not translate error messages that appear only if something goes wrong.

Also, we need to manage different Tiki releases, strings might come and go between releases.

How?

We have quite a few contestants. The options are to go with a proprietary service provider like Transifex (free for open source projects), host our own instance of Pootle, or improve our existing infrastructure.

- **Transifex**

- <https://www.transifex.com>

- Pros

- It provides integration for back and forth deployment
 - They already have an user base of contributors. Reaching out to new people is an easier step.
 - API for integration
 - Glossary
 - Translation Memory

- Teams
- Support for multiple products inside a project (we could have 12x, 14x, 15x separated)
- One less subsite to maintain (i18n.tiki.org)
- Infrastructure maintenance is outsourced, no resources required
- Lower entrance barrier for translators (no need for source files, commit access, PHP, SVN knowledge)
- Cons
 - Integrating might take some time.
 - We need to have at least one translation leader per language we want to deploy (This person will review strings). **** In the case of a one-person/one-language translation effort, that person would also be the reviewer. The downside is an extra step to mark the translated strings as reviewed; the upside (to be optimistic) is having a review stage built in to encourage a second check.
 - One more tool (|msaad| = I don't think it's bad to have yet one more tool if it actually gets the job done).
 - 翻译指令 (translation instructions) 是一个可选的字段，用于提供上下文信息。在 Transifex 页面中，每个源字符串都有一个“翻译指令”字段，该字段会在字符串被翻译时出现在所有语言中。There is a "translation instructions" field in the Transifex page, per source string, for context information that will then appear for all languages when that string is being translated.
 - You can open Tiki in a browser to see the frontend and search for the string in transifex. Not the same as clicking on a string but not so complicated either.
 - transifex imports commented out translation lines, but then treats them as the source string so the exported translations are also commented out
- see also: [Translation workflow using transifex](#)
- **Pootle**

<http://pootle.translatehouse.org> and a demo: <http://pootle.locamotion.org/>

 - Pros
 - Open source!
 - Multiple file formats supported
 - Teams
 - Translation Memory (suggests how this word was translated before)
 - Supports multiple products inside a project
 - One less subsite to maintain (i18n.tiki.org)
 - Lower entrance barrier for translators (no need for source files, commit access, PHP, SVN knowledge)
 - Cons
 - May be slower for querying when dealing with lots of entries (JS dependent).
 - Integrating might take some time.
 - UI going through changes for worse
 - Doesn't have core statistics about contributors when compared to Transifex
 - Someone needs to learn to deploy and maintain it, requires extra resources
- **Crowdin**

<https://crowdin.com>

asked and got opensource license, created tiki project: <https://crowdin.com/project/tiki>

already got 8 translator volunteers checked in for the project

a translator contacted me via email about helping out with translation. Turned out he uses both transifex and crowdin, asked him to compare these two:

"The Crowdin is better for translate but Transifex improved a little lately. If i needed to decide now I'll chose Crowdin. "

- Pros

- Free Open Source and Academic Licenses
<https://crowdin.com/pricing>
- SSO using tiki.org or any Tiki installation account?
<https://support.crowdin.com/for-project-managers/integration/hybrid-single-sign-on-integration/>
- in-context translation tool:
<https://support.crowdin.com/for-project-managers/project-management/in-context-localization/>

◦ Cons

- crowdin does not import commented out translation lines

• **Tiki** build in interactive translation on community server

<http://i18n.tiki.org>

Have an active i18n team which uses the existing tools as the last commit was done on [2013-09-23](#) (22 months ago)

◦ Pros

- Dog Fooding
- It's already there
- In-context translation
 - Permits higher quality translation because of context
 - Makes it easier to prioritize translations of strings that are first / most seen by users
- With some work, any Tiki instance could become a source of translations
- There is a script to commit translations from i18n.tiki.org to svn with user "tikii18nbot" Just needs to be run regularly or added to a cron job. Adding to [release process](#) would be smart as well.
 - <https://www.openhub.net/p/tikiwiki/contributors/169651656406> and <http://sourceforge.net/u/tikii18nbot/profile/>
 - Lower entrance barrier for translators (no need for source files, commit access, PHP, SVN knowledge)
- [Possible to fix English strings without invalidating all the translations](#), where as in Transifex, a manual intervention is necessary for each translation
 - You can use the english translation too to correct what is displayed in english

◦ Cons

- The results obtained with it are not outstanding.
 - The question is : is it a problem with the tool, or a problem of coordination / activity / communication of the [i18n Team](#). There are several people who volunteered to help, but didn't get any responses.
 - Currently it seems like the team concept doesn't generate enough activity, there are a lot of teams with little real activity. There were discussions about this situation at the recent [TikiFestCEST](#) and roundtable meetings, how to adjust this to current resources and participation intentions (eg: to have in the tiki.org user tracker a selection list "Interested in topics" or similar), no outcome yet.
 - Be it teams, groups, topics, maintainers or any way you come up with to cluster activity, the fundamental issue is that things need to be done. People need to take on responsibilities and be reliable. And when there is no identified leadership in a topic or if that leadership doesn't respond or is otherwise unreliable, things stall, and new potential contributors never become active contributors. Translation is a fantastic 1st place to onboard new contributors and success here will have very nice side effects. Tiki's most prolific code committer (Sylvieg) started off "just translating".
- We barely have control on what is done and what's none
 - Can you elaborate?
- We don't need to be good at everything. We have to focus on improving Tiki.

- We only have a few locales 100% complete.
 - True, but can the tool really take the blame for this?
- Documentation on this is sparse. Newcomers have a hard time getting set-up.
- One more subsite to maintain (i18n.tiki.org)
- Lack of tools like a translation memory
- 翻訳の管理が難しい。翻訳の品質を上げるには、翻訳者のスキルアップが必要。翻訳の自動化は、まだ実現していない。翻訳の管理は、まだ手作業で行っている。 - It is already there.
- Lack of management for various Tiki releases
 - Yes, this is a problem indeed: how do Transifex and Pootle solve this?
 - Answer to Marc Laporte byTorsten: I have been working with Transifex fora few hours now. I Registered a free account, applied to be added to the Tiki Translations Team there (granted surely) and started to co-manage the translations. Due to the fact we are all new to Transifex a few languages have been setup wrong in the first place. I could figure that out and instantly fix it. Gezza, Luci and a few others have started; I have continued today with fixing plus adding new languages. Right now we have 17 translations running, more to come. The moist difference is, that the whole team can co-manage the translations and everybody with commit access can upstream to our codebase until we might figure out some automative script solution. i18n.tiki.org has to be managed server side and we surely wont dare to add a high number of people to server accessing sysadmin team each one having to handle scripts with SSH. Most of us might be capabl, but anyway that would still not allow a really overseeable workflow. With Transifex everything is as much obvious and accessible, that some basic communication in IRC and Dev-List should be absolutely enough to coordinate.
 - Transifex solves this with team oriented management functionalities - many team members can co-manage the account (and translations) without the need of server access (other than on i18n.t.o where we need server access to handle translation files and commit scripts
 - @Torsten Fabricius: The translation robot script can be put on a cron job and the commits would be done automatically.
- Existing tools + transifex/pootle - do we have to choose one or can they coexist?
 - Pros
 - people can continue to do as they did so far
 - ...
 - Cons
 - too many ways to translate (is it really a con?)
 - ...
- lang folder size
 - it has been noted, that Tiki installation is quite large, /lang folder takes about 75mb
 - Tiki could be shipped with languages that are eg: at least 50% translated (can be more-less)
 - have an "Upload translation" button to add new languages to a Tiki install. When uploading, you can select file type (transifex/crwding/Tiki source) and the file will be processedd accordingly
 - use transifex/crowdin to store translations
- Other tools
 - OneSky
 - free for crowdsourced projects as they phares
(<http://www.oneskyapp.com/pricing/translation-management-system/>)
 - does not seem to handle php arrays
(<https://developer.oneskyapp.com/tutorial/php-localization-guide>)
 - webtranslateit
 - free for open-source (<https://webtranslateit.com/en/projects/public/about>)
 - supports php arrays (https://webtranslateit.com/en/docs/file_formats/php_array)

- Phrase
 - <https://help.phrase.com/help/php-array>
 - <https://phrase.com/docs/guides/migrating/>
- TranslateWiki
 - used mainly for MediaWiki localization
 - process requires giving commit access to translatewiki staff (https://translatewiki.net/wiki/Translating:New_project)
- POEditor
 - does not handle .php files (<https://poeditor.com/features/>)
- Other ideas?

How much?

Aside from human resources, Transifex or Pootle won't cost us anything. However, the benefits of this improvement are intrinsic. We'll ship in different languages, we'll increase our user base, we'll be more appealing to contributors, we'll be better prepared for moving forward.