Git

1. Introduction

What is Git?

Git is a version control system (vcs) originally developped for the Linux Kernel development. Git as a vcs is used to develop and to deploy software, such as other vcs'es like CVS, Subversion (svn), Perforce, Bazaar, and so on, whereas Git has a unique decentralized approach.

In 2005, the relationship between the community that developed the Linux kernel and the commercial company that developed BitKeeper broke down, and the tool's free-of-charge status was revoked. This prompted the Linux development community (and in particular Linus Torvalds, the creator of Linux) to develop their own tool based on some of the lessons they learned while using BitKeeper. Some of the goals of the new system were as follows:
Speed, Simple design, Strong support for non-linear development (thousands of parallel branches), Fully distributed, Able to handle large projects like the Linux kernel efficiently (speed and data size)

The **Pro Git book** was written by Scott Chacon and Ben Straub and published by Apress, is available online and as print version.
All content is licensed under the Creative Commons Attribution Non Commercial Share Alike 3.0 license.
Print versions of the book are available on Amazon.com and the online verion on the git community website here: https://git-scm.com/book/en/v2.

Git is being used to implement an additional contribution process via a Distributed version control system for the Tiki community. The reasons, pros and cons are documented on Distributed revision control page.

Despite the benefits of using Git, there are many concerns about how each contributor will behave, work and what kind of problems they will find on their daily activities. So, this document and derived pages are an attempt to guess what knowledge a Tiki developer should have for contributing to Tiki.

2. Related pages

Git: You are reading this. Root page of the Git structure. An attempt at consolidating/linking all the git information the information. Also a lot of introductory content.

- Using Git with Tiki: Basic instructions on checking out tiki with Git. Status: Recently modified but a lot of obsolete information. Purpose vs Git isn't well documented (I think it was supposed to be a quick start type of page, but unclear).
- Git Usage: A more git task oriented reference, compared to the workflow oriented Git Workflow. Status: Up to date.
  - Git concepts for SVN users: Tutorial/cheat sheet for svn users new to git. Definitely useful.
  - Git Tips: show some tips on how to do various developer tasks with Git
  - Git cherry-pick: Detailed info on a specific git task.
  - Merging back changes from a fork: Status: Some information is now incorrect
- Git clone Tiki: Status: Brand new page (june 2019). I don't understand why a new page was created vs the larger and more complete Git Usage most of the content is redundant, but in some cases the approach slightly different.
- Worflows
  - Git Workflow: The new version control development workflow we will follow from now on. Status: In heavy development.
  - Git and SVN combined workflow: Details the workflow behind the current Git and SVN

integration. Status: Still up to date
        - How to rescue a commit lost from SVN on 2018-04-10: Documentation of a specific svn synchronization problem. Status: Will become obsolete once the git svn workflow is decommissioned
- Obsolete
    - Git SVN HowTo Documentation of using git-svn command, not to be confused with the Git and SVN combined workflow. Status: obsolete.
    - Distributed revision control Page documenting the rationale for the migration. Status: Probably obsolete.
    - Choosing the Git Workflow Migration plan page. Status: probably obsolete
    - Best practices for a local Git repository and interacting with the Tiki community: Draft of a workflow. Status: Probably obsolete

Pages in the Git structure we may want to document above (or maybe put the info above in the page description):

3. Status as of 2018-06-12

There is a combined workflow using SVN and Git to make a more smooth transition and less impact on daily activities of Tiki developers. This workflow can be found on Git and SVN combined workflow page and it details how is the integration of SVN and Git now.

~~Also, the current phase of migration strategy involves writing enough documentation and close write access to SVN repository, making it read-only. Probably soon, we will have a~~ Round Table Meeting ~~to check if it is time to close SVN.~~ Only once the following is done can we talk about closing write access to the SVN repo.

1. The Git workflow is compelling enough that the majority of developers want to move to it
2. All those who would like some support / training have received it.
3. The Git-related documentation is ready (we need something equivalent as what we have for SVN)
4. The relevant automated scripts (such as the release scripts) have all been ported to use Git
5. There is a strategy for usernames (how to request access, etc.)

See Using Git with Tiki and Git Tips to start developing with Git.

4. Roadmap

1. Setup workflow to work with Git and SVN
2. Write documentation
3. Talk on Round Table Meeting about Git
4. Setup help channel for developers struggling with Git
5. Close SVN write access
6. Publish the new Tiki repository

5. Discussions

Distributed revision control has discussions which led to Git being the best option.