Backport changes using Git

Some fixes submitted on master branch are also desired on other Tiki major versions. Each Tiki major version is also a branch, so backport means get a change present on master branch and submit it on other branch. Example, get a commit from master and submit on `19.x`.

This page is the Git equivalent of: Merge a commit between branches

The best tool to achieve this procedure is the `git cherry-pick`.

Example

Folder structure

The next example assumes an existing folder named `tiki19` with versioned by Git and another folder `tikimaster`, also versioned by Git. The `tiki19` is tracking branch **19.x** while `tikimaster` folder is tracking branch **master**.

```
| .
| ..
| tikimaster
| tiki19
```

1. Getting commit and message

Type `git log`, search for commit to be backported, copy it's hash and assign it to a variable name `TARGET_COMMIT`.

```
TARGET_COMMIT=1f0d6c95b936aeface9a4d7bf4d2a5b18782d1e7
TARGET_MESSAGE=$(git log $TARGET_COMMIT -n1 --format='%B')
```

2. Get SVN revision

It is needed to look for the SVN revision number to satisfy the commit message standards required by Tiki community. An useful command to list the last 5 entries from svn is:

```
svn log svn://svn.code.sf.net/p/tikiwiki/code/trunk -l 5
```

Write down the revision number.

```
TARGET_REV=r69552
```

3. Update repository

```
cd tiki19
git pull
```

4. Cherry-pick commit

```
git cherry-pick $TARGET_COMMIT
```

5. Rephrase the message

Follow Tiki standards to commit messages https://dev.tiki.org/Commit-Tags.

```
git commit --amend -m "[bp/${TARGET_REV}]${TARGET_MESSAGE}"
```

6. Push it to 19.x

```
git push
```