Preparing new release

1. Sync locale strings across language files (add test to check for differences? or execute system tests in different locales?).
2. Make sure all tests pass on each merged issue branch, on all supported Redmine versions:
    1. Start in Redmine's directory:

    ```
    cd /var/lib/redmine
    ```

    2. (only before 1st run) prepare database (load default data separately, https://www.redmine.org/boards/2/topics/48044):

    ```
    RAILS_ENV=test bundle exec rake db:drop db:create db:migrate
    RAILS_ENV=test bundle exec rake redmine:plugins:migrate NAME=issue_recurring
    RAILS_ENV=test bundle exec rake redmine:load_default_data
    ```

    3. run tests (migration tests run separately as they are not run within transaction and can leave db in unspecified state when failing):

    ```
    RAILS_ENV=test bundle exec rake redmine:plugins:test NAME=issue_recurring
    RAILS_ENV=test bundle exec rake redmine:plugins:test:migration NAME=issue_recurring
    ```

    4. (optionally) rerun failing tests separately:

    ```
    RAILS_ENV=test bundle exec ruby plugins/issue_recurring/test/integration/issue_recurrences
    _test.rb --name test_create_recurrence --verbose
    ```

3. [Merge](#) all outstanding branches into master.
4. Rerun above tests on master branch if there were:
    1. multiple branches merged,
    2. conflicts during merge.
5. Update [source:CHANGELOG.md](#) and [source:README.md](#): **Features** and compatibility list in **Installation** paragraphs.
6. Bump plugin version number in [source:init.rb](#).
7. Commit and push changes.
8. Create and push git tag with new version number:

```
$ git tag -a 1.1
$ git push origin 1.1
```

9. Close corresponding issues (if not closed by appropriate commit messages).
10. Update plugin information on https://redmine.org/plugins/issue-recurring
11. Add release news on IT https://it.michalczyk.pro/projects/issue-recurring/news
12. Add new version on IT https://it.michalczyk.pro/projects/issue-recurring/settings/versions