

Development Workflow Using GIT

Vicente Benjumea

(DRAFT)

Abstract

This document describes the workflow for development using GIT. It is largely based on the `gitworkflows` manual page¹ [1]. So many rules presented in this guide, as well as their rationale, can be found in such document.

1 Version Numbering

Version numbering of development uses three numbers: *Version Major*, *Version Minor* and *Revision Number*, with the following meanings:

- *Version Major*: each number in the sequence defines a new release that incorporates major changes in source code that affect the *public interface*, API and functionality. It usually incorporates new added features, and the public interface may not be compatible with a previous release.
- *Version Minor*: each number in the sequence defines a new release that incorporates changes in source code that do not affect the *public* behavior of the interface. The incorporated changes usually improve the way things are done, but do not add new features. The public interface should be fully compatible with previous release.
- *Revision Number*: each number in the sequence defines a new release that incorporates changes in source code that fixes some bugs present in previous releases and do not affect the *public* behavior of the interface.

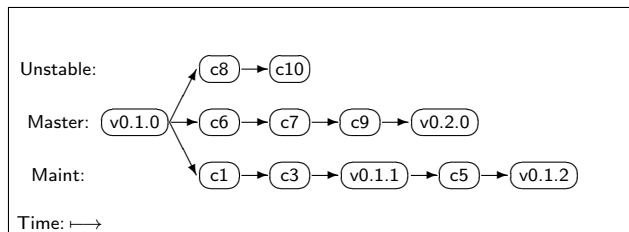
2 Development Branches

There are three main development *branches* (*maint*, *master* and *unstable*), as well as some maintenance branches for previous releases, and a *topic* branch for each bug-fix, minor changes and new added features.

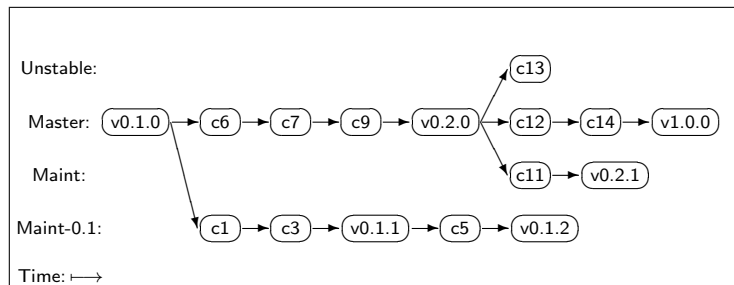
- The *maint* branch tracks the commits for bug-fixes for the last released version, and hold all released revisions for the previous released version. For example, if the last released version is 0.1.0, then this branch tracks all later bug-fix commits and revisions, such as 0.1.1, 0.1.2, 0.1.3, and so on.
- The *master* branch tracks the commits for minor and major changes that should go into the next released version (X.Y.0).

¹<http://schacon.github.com/git/gitworkflows.html>

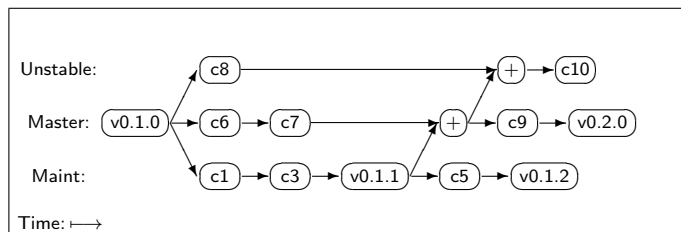
- The *unstable* branch is used to commit minor and major changes for testing their stability. After successful testing, such changes can be incorporated into the *master* branch.



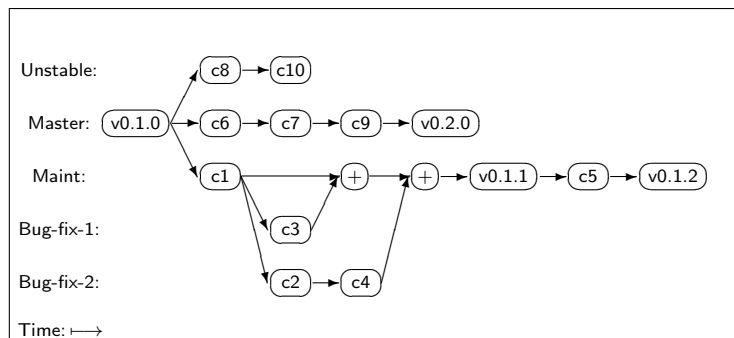
- As new versions are released, older maintenance branches are renamed after the version number they maintain, e.g.: *maint-0.1*



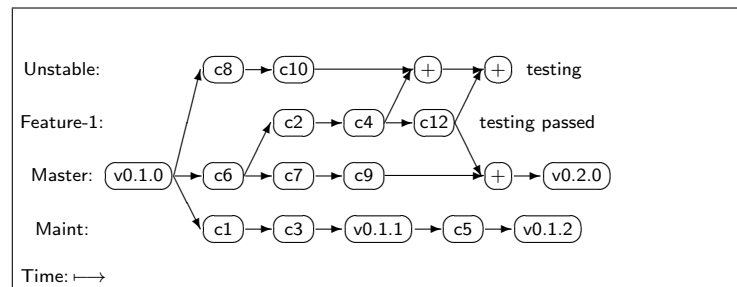
- From time to time (at least for every revision/maintenance release), merge from *maint* into *master*, and from *master* into *unstable*. Never merge in the opposite way.



- Each topic (bug-fix, feature, etc) will be tracked in an individual *topic* branch, which will be forked off from the branch (*maint* or *master*) the topic will be eventually incorporated into (merged).



Relevance changes will be incorporated from the topic branch into the *unstable* branch (for stability testing) before being incorporated from the topic branch into the destination branch. Note that changes and bugfixes should be applied into the suitable *topic* branch, and not into the *unstable* branch. Never merge from the *unstable* branch into the other ones.



- As a general rule, changes should be split into small logical steps, and commit each of them separately. They should be consistent, working independently of any later commits and pass the test suite.
- All these branches (*maint*, *master*, *unstable*, and *topic* branches) compose the development history of the project, and therefore, they all should be uploaded into the *public repository*.

References

- [1] Git Workflows Manual Page. <http://schacon.github.com/git/gitworkflows.html>

Copyright (c) 2012 Vicente Benjumea, University of Malaga, Spain

Redistribution and use in source (LaTeX) and 'compiled' forms (SGML, HTML, PDF, PostScript, RTF and so forth) with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code (LaTeX) must retain the above copyright notice, this list of conditions and the following disclaimer as the first lines of this file unmodified.
2. Redistributions in compiled form (transformed to other DTDs, converted to PDF, PostScript, RTF and other formats) must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS DOCUMENTATION IS PROVIDED BY THE COPYRIGHT HOLDERS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS DOCUMENTATION, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.