

24/01/2012

# Git interactive rebase

Tobias Schröder

## Interactive rebase

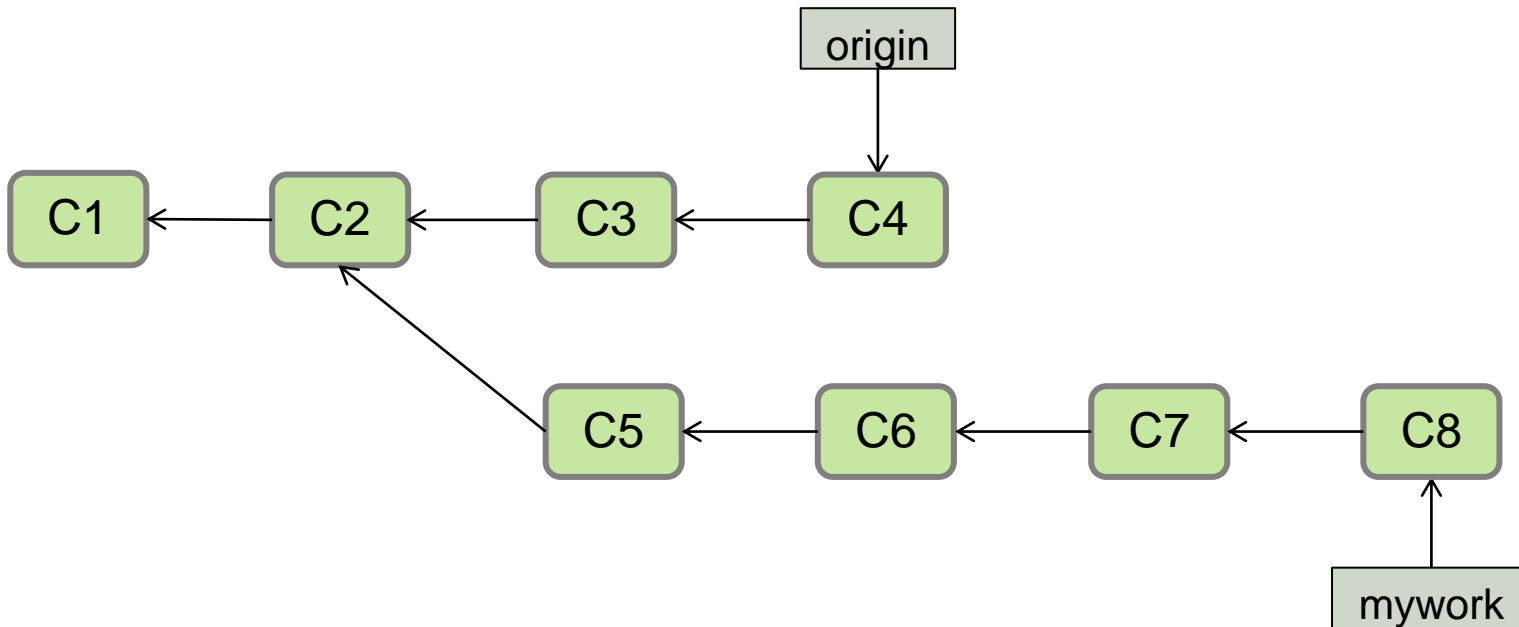
- Rewrite local history interactively
- Remove, rearrange, reword, edit and squash
  
- `git rebase -i last_good_commit`
  
- `git rebase -i HEAD~number_of_commits`

## When to rebase?

- **Local branches.**
- Pushed branches? Rather not.
- Master branch? Rather not.
- Tags? never.

# An example

- `git rebase -i mywork~4`



## An example

```
git-rebase-todo (c:\Users\schro...sentation\.git\rebase-merge) - VIM
pick 4c33866 not a useful message
pick 89c6b21 message
pick c18e348 some documentation
pick 6da3966 minor fixes

# Rebase 7a4305e..6da3966 onto 7a4305e
#
# Commands:
# p, pick = use commit
# r, reword = use commit, but edit the commit message
# e, edit = use commit, but stop for amending
# s, squash = use commit, but meld into previous commit
# f, fixup = like "squash", but discard this commit's log message
# x, exec = run command (the rest of the line) using shell
#
# If you remove a line here THAT COMMIT WILL BE LOST.
# However, if you remove everything, the rebase will be aborted.
#
<presentation\.git\rebase-merge\git-rebase-todo [unix] (16:00 24/01/2012)1,1 All
```

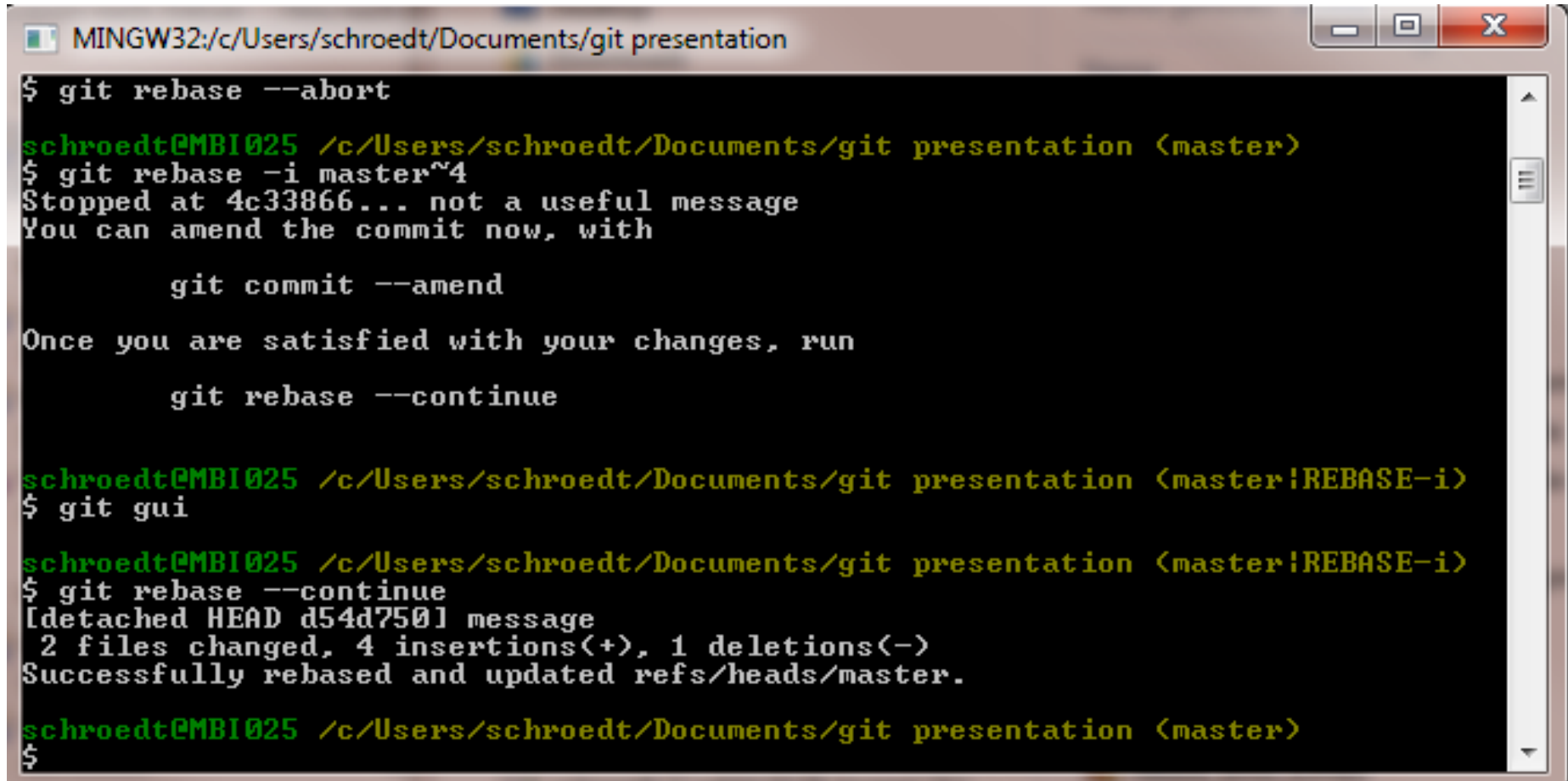
(action) (partial-sha) (short commit message)

# An example

```
git-rebase-todo + (c:\Users\schr...sentation)\.git\rebase-merge) - VIM
edit 4c33866 not a useful message
pick 89c6b21 message
squash c18e348 some documentation
pick 6da3966 minor fixes

# Rebase 7a4305e..6da3966 onto 7a4305e
#
# Commands:
# p, pick = use commit
# r, reword = use commit, but edit the commit message
# e, edit = use commit, but stop for amending
# s, squash = use commit, but meld into previous commit
# f, fixup = like "squash", but discard this commit's log message
# x, exec = run command (the rest of the line) using shell
#
# If you remove a line here THAT COMMIT WILL BE LOST.
# However, if you remove everything, the rebase will be aborted.
#
~
~
~
~
~
~
<sentation\.git\rebase-merge\git-rebase-todo[+] [unix] <16:00 24/01/2012>3,8 All
-- INSERT --
```

## An example



```
MINGW32:/c/Users/schroedt/Documents/git presentation
$ git rebase --abort

schroedt@MBI025 /c/Users/schroedt/Documents/git presentation (master)
$ git rebase -i master~4
Stopped at 4c33866... not a useful message
You can amend the commit now, with

    git commit --amend

Once you are satisfied with your changes, run

    git rebase --continue

schroedt@MBI025 /c/Users/schroedt/Documents/git presentation (master!REBASE-i)
$ git gui

schroedt@MBI025 /c/Users/schroedt/Documents/git presentation (master!REBASE-i)
$ git rebase --continue
[detached HEAD d54d750] message
 2 files changed, 4 insertions(+), 1 deletions(-)
Successfully rebased and updated refs/heads/master.

schroedt@MBI025 /c/Users/schroedt/Documents/git presentation (master)
$
```

Note:

`git rebase --abort`

Stops rebasing and returns to the state before starting rebase

# An example

The screenshot shows the gitk application window titled "gitk: git presentation". The window has a menu bar with "Datei", "Bearbeiten", "Ansicht", and "Hilfe".

The main area is divided into two columns. The left column shows a commit history with a vertical axis of colored dots (yellow for master, blue for other commits) and commit messages. The right column shows the corresponding commit details, including the author "Tobias Schroeder <t.schroeder@dkfz.de>" and the commit time.

The commit history shows the following messages from top to bottom:

- minor fixes
- message
- now it is a useful message
- changes to first file
- some minor changes
- changes on file another\_textfile
- some initial text to file another\_textfile
- added another text file
- added text file

The commit details for the selected commit (SHA1 ID: 4e88859645e75c019ed8de1e2c36becc607a988f) are:

Author	Date
Tobias Schroeder <t.schroeder@dkfz.de>	2012-01-24 09:39:37
Tobias Schroeder <t.schroeder@dkfz.de>	2012-01-24 10:03:49
Tobias Schroeder <t.schroeder@dkfz.de>	2012-01-24 16:04:09
Tobias Schroeder <t.schroeder@dkfz.de>	2012-01-24 09:24:19
Tobias Schroeder <t.schroeder@dkfz.de>	2012-01-24 09:31:09
Tobias Schroeder <t.schroeder@dkfz.de>	2012-01-24 09:21:06
Tobias Schroeder <t.schroeder@dkfz.de>	2012-01-24 09:19:59
Tobias Schroeder <t.schroeder@dkfz.de>	2012-01-24 09:18:41
Tobias Schroeder <t.schroeder@dkfz.de>	2012-01-24 09:17:47

Below the commit list, the SHA1 ID is shown as "4e88859645e75c019ed8de1e2c36becc607a988f". Navigation buttons for "nächste" and "vorige" are present. The "Suche" (Search) section includes a search box and buttons for "Suchen", "nächste", and "vorige". The "Version nach" (Sort by) dropdown is set to "Beschreibung" (Description). The "Exakt" (Exact) dropdown is set to "Alle Felder" (All fields). The "Vergleich" (Compare) section has radio buttons for "Vergleich" (selected), "Alte Version" (Old version), and "Neue Version" (New version). The "Patch" and "Baum" (Tree) view options are also visible.

The diff view shows the changes for the selected commit. The "Beschreibung" (Description) field is highlighted in blue. The diff content shows the commit message "now it is a useful message" and the file "another\_textfile.txt".



