

9/17/13

Git rerere

Andreas Fetzner
Division for Medical And Biological Informatics

Git rerere

- Git rerere = **re**use **re**corded **re**solution
 - Git will remember a resolved hunk conflict
 - Next time it will resolve the conflict automatically!

Szenario

- You have a rather old branch which you want to merge into Master
- Let's assume you didn't create an integration branch
- Most likely you will get a lot of conflicts

```
CONFLICT (content): Merge conflict in Modules/Segmentation/Interactions/mitkFastMarchingTool.h
Auto-merging Modules/Segmentation/Interactions/mitkFastMarchingTool.cpp
CONFLICT (content): Merge conflict in Modules/Segmentation/Interactions/mitkFastMarchingTool.cpp
Auto-merging Modules/Segmentation/Interactions/mitkEraseRegionTool.h
CONFLICT (content): Merge conflict in Modules/Segmentation/Interactions/mitkEraseRegionTool.h
Auto-merging Modules/Segmentation/Interactions/mitkEraseRegionTool.cpp
CONFLICT (content): Merge conflict in Modules/Segmentation/Interactions/mitkEraseRegionTool.cpp
Auto-merging Modules/Segmentation/Interactions/mitkErasePaintbrushTool.h
CONFLICT (content): Merge conflict in Modules/Segmentation/Interactions/mitkErasePaintbrushTool.h
Auto-merging Modules/Segmentation/Interactions/mitkErasePaintbrushTool.cpp
CONFLICT (content): Merge conflict in Modules/Segmentation/Interactions/mitkErasePaintbrushTool.cpp
Auto-merging Modules/Segmentation/Interactions/mitkDrawPaintbrushTool.h
CONFLICT (content): Merge conflict in Modules/Segmentation/Interactions/mitkDrawPaintbrushTool.h
Auto-merging Modules/Segmentation/Interactions/mitkDrawPaintbrushTool.cpp
Auto-merging Modules/Segmentation/Interactions/mitkCorrectorTool2D.h
CONFLICT (content): Merge conflict in Modules/Segmentation/Interactions/mitkCorrectorTool2D.h
Auto-merging Modules/Segmentation/Interactions/mitkCorrectorTool2D.cpp
Auto-merging Modules/Segmentation/Interactions/mitkAddContourTool.h
CONFLICT (content): Merge conflict in Modules/Segmentation/Interactions/mitkAddContourTool.h
Auto-merging Modules/Segmentation/Interactions/mitkAddContourTool.cpp
```

Szenario

- Resolving the conflicts will take some time...



Szenario

- After you are done you try to push the Master
- But:

```
Merge the remote changes before pushing again. See the 'Note about fast-forwards' section of 'git push --help' for details.
```

Szenario

- In the meantime a colleague pushed changes to the master...



Szenario

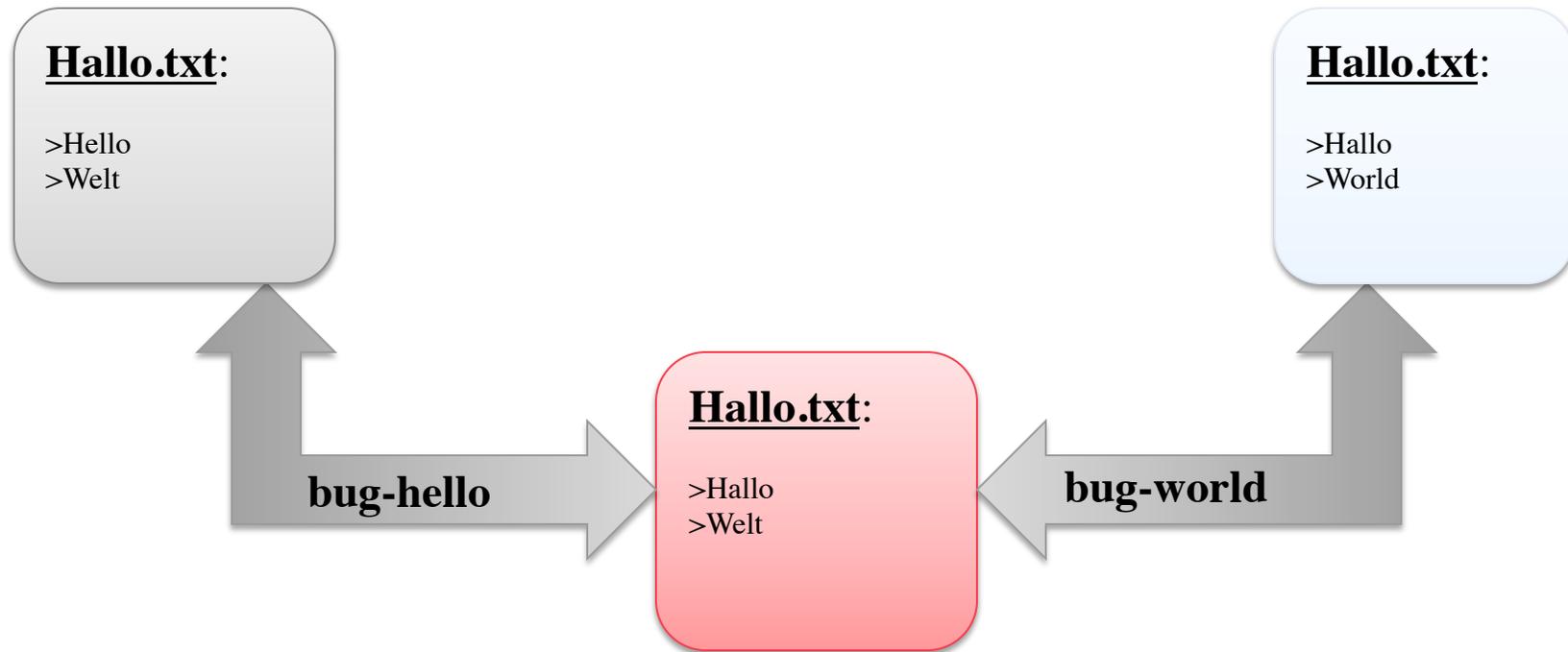
- What now?
 - Reset changes
 - Pull current master
 - Merge again
 - And...

```
CONFLICT (content): Merge conflict in Modules/Segmentation/Interactions/mitkFastMarchingTool.h
Auto-merging Modules/Segmentation/Interactions/mitkFastMarchingTool.cpp
CONFLICT (content): Merge conflict in Modules/Segmentation/Interactions/mitkFastMarchingTool.cpp
Auto-merging Modules/Segmentation/Interactions/mitkEraseRegionTool.h
CONFLICT (content): Merge conflict in Modules/Segmentation/Interactions/mitkEraseRegionTool.h
Auto-merging Modules/Segmentation/Interactions/mitkEraseRegionTool.cpp
CONFLICT (content): Merge conflict in Modules/Segmentation/Interactions/mitkEraseRegionTool.cpp
Auto-merging Modules/Segmentation/Interactions/mitkErasePaintbrushTool.h
CONFLICT (content): Merge conflict in Modules/Segmentation/Interactions/mitkErasePaintbrushTool.h
Auto-merging Modules/Segmentation/Interactions/mitkErasePaintbrushTool.cpp
CONFLICT (content): Merge conflict in Modules/Segmentation/Interactions/mitkErasePaintbrushTool.cpp
Auto-merging Modules/Segmentation/Interactions/mitkDrawPaintbrushTool.h
CONFLICT (content): Merge conflict in Modules/Segmentation/Interactions/mitkDrawPaintbrushTool.h
Auto-merging Modules/Segmentation/Interactions/mitkDrawPaintbrushTool.cpp
Auto-merging Modules/Segmentation/Interactions/mitkCorrectorTool2D.h
CONFLICT (content): Merge conflict in Modules/Segmentation/Interactions/mitkCorrectorTool2D.h
Auto-merging Modules/Segmentation/Interactions/mitkCorrectorTool2D.cpp
Auto-merging Modules/Segmentation/Interactions/mitkAddContourTool.h
CONFLICT (content): Merge conflict in Modules/Segmentation/Interactions/mitkAddContourTool.h
Auto-merging Modules/Segmentation/Interactions/mitkAddContourTool.cpp
```

How to enable it?

- Hidden feature
- `git config --global rerere.enabled true`
- That's it! From now on your resolutions will be recorded!

Example



Example

- Merge **bug-hello** into **bug-world** to see if everything works
 - You will get conflicts...and notice a new git message:



```
Auto-merging hallo.txt
CONFLICT (content): Merge conflict in hallo.txt
Recorded preimage for 'hallo.txt'
Automatic merge failed; fix conflicts and then commit the result.
```

- Rerere created a new entry for this merge conflict!
- Resolve and commit them...again notice a new git message:

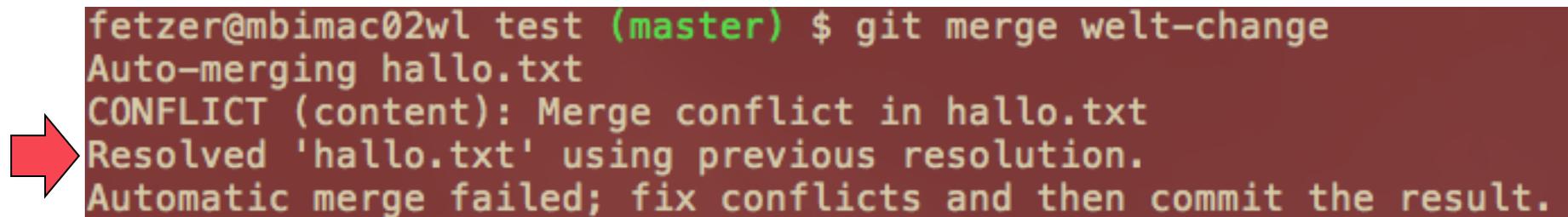


```
fetzer@mbimac02wl test (test-rerere2) $ git commit
Recorded resolution for 'hallo.txt'.
```

- Rerere recorded your resolution!

Example

- Merge your branches into the Master:



```
fetzer@mbimac02wl test (master) $ git merge welt-change
Auto-merging hallo.txt
CONFLICT (content): Merge conflict in hallo.txt
Resolved 'hallo.txt' using previous resolution.
Automatic merge failed; fix conflicts and then commit the result.
```

- The conflict has automatically been resolved by **git rerere!**

Git rerere

- Very usefull
- Hidden git feature
- Totally simple to use
- Nearly no drawbacks

Git rerere useful options

- `git rerere status`
 - Shows current created records = conflicted files
- `git rerere diff`
 - Shows the state of the resolution (what is done, what still has to be done)
- `git checkout --conflict=merge hallo.txt`
 - Restore conflicted state
- `git rerere`
 - Re-resolve conflicts (after previous command)

Another usefull applikation

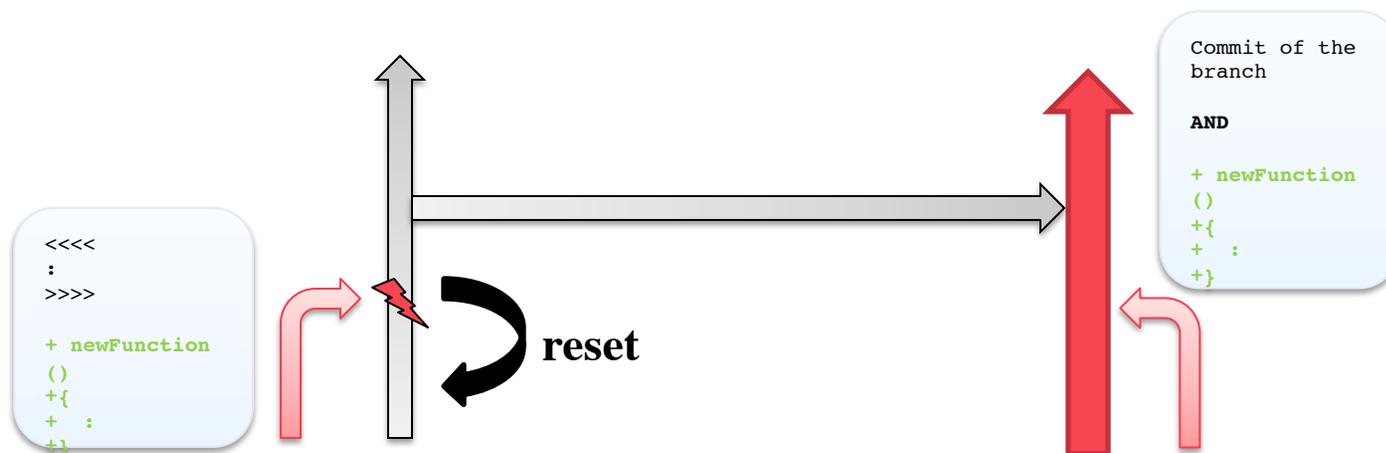
- You have a long lived topic branch
- Want to merge cleanly at the end (who not?)
- No intermediate commits (because WIP)
- Solution:
 - Merge occasionally, resolve conflicts, undo merge
 - Resolutions will be recorded
 - Final merge will be much easier thanks to **rerere**

Summary

- Rerere is a very useful but hidden git functionality
- Useful to avoid resolution of conflict twice or more
- Easy to use because it is working in the background
- **But...**(there is always a but 😊)

But...

- Incorrect resolutions will also be remembered (`forget`)
- Rerere remembers how you chose to resolve the conflicted regions
- Rerere also remembers how you touched up outside the conflicted regions to adjust to semantic changes
- Rerere can reuse previous resolution even though you were merging two branches with different contents than the one you resolved earlier.



References

- „Fun with rerere“, <http://gitster.livejournal.com/41795.html>
- <http://git-scm.com/blog/2010/03/08/rerere.html>