

Automating GIT for development on large *distributed* teams

Yuri D'Elia
2023-11-10

Branching in git

- Work on the same code without interference
- Test separate changes independently
- Reconcile conflicting changes *"easily"*
- **Allows history rewriting!**

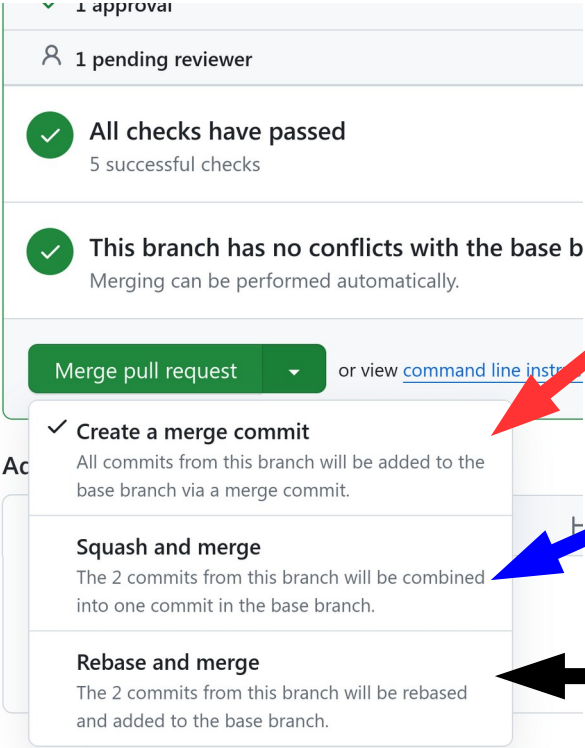
Keeping repository hygiene

• added new function belt test under calibration menu

- Merge pull request #2195 from leptun/MK3-fix-LFN
- Fix short filenames scrolling in sdprint menu and status screen
- Merge pull request #2188 from mkbel/cut_last_try
- Use MMU cutter only as last resort. Cut filament only before last load retry.
- Merge pull request #2155 from leptun/MK3_SD_scroll_workaround
- SD Scroll workaround
- Merge pull request #2057 from leptun/MK3_fix_M350
- Update Marlin_main.cpp
- M350 for all axis - MK3
- Merge pull request #1899 from wavexx/fix_lcd_pause_function
- Fix the LCD pause function
- Merge pull request #1768 from wavexx/unknown_z_improvements
- Fix include guard to silence duplicate declaration warnings
- Improved handling of unknown Z position
- Merge pull request #1752 from wavexx/introduce_editorconfig
- Introduce an editorconfig file
- Merge pull request #1739 from draeath/MK3-fix-sdcard-sorting
- Merge branch 'MK3' into MK3-fix-sdcard-sorting
- use modification times for sdcard sort
- Merge pull request #1704 from leptun/MK3_Enable_fan_waiting_cooldown
- Merge branch 'MK3' into MK3_Enable_fan_waiting_cooldown
- Merge branch 'MK3' into MK3_Enable_fan_waiting_cooldown
- Enable print fan while waiting for cooldown
- Merge branch 'MK3' into MK3_Dutch
- cv3.8.1 Merge pull request #2293 from mkbel/increment_version
- Version 3.8.1 final.
- Merge pull request #2292 from mkbel/de_lang_fix
- Shorten "Automatisches Laden Filament nur bei einge schaltetem Filament- sensor verfuegbar..." to fit on single screen.
- Updated 'lang_en_nl.txt' with newest translations
- Merge branch 'MK3' into MK3_Dutch
- Merge pull request #2291 from DRracer/lang-381-fixes-4
- Skipped translated sentence, because total translation size was > 65535B.
- Better EN for new messages
- Merge pull request #2279 from leptun/MK3_Octoprint_fixes
- Enable beeping after fan error is fixed
- Merge branch 'MK3' into MK3_Octoprint_fixes
- Merge pull request #2277 from DRracer/lang-381-fixes-3
- tiny lang fixes, part 1
- Merge pull request #2263 from leptun/MK3-Fix_PSU_DELTA_SILENT
- PSU_DELTA fix force_high_power_mode()
- Merge pull request #2256 from mkbel/unify_silent_mode
- Initialize EEPROM_SILENT in one place both for MK3 and MK25 printers and make it more clear.
- Reenable command processing when issue is fixed if printer not active
- Some more missing translation
- Updated po files
- Update Dutch translation to v3.8.1-RC1
- Merge branch 'MK3' into MK3_Dutch
- <v3.8.1-RC1> Merge pull request #2253 from mkbel/increase_version
- Increase version.
- Merge pull request #2252 from mkbel/increase_version
- Increase version and commit number.
- Merge pull request #2251 from DRracer/lang-381-fixes-2
- tiny lang fixes for 3.8.1
- Merge pull request #2250 from mkbel/fix_MK25_wizzard
- Move check for clear EEPROM just after factory_reset() to ensure nothing is written to EEPROM before check.
- Updated Dutch translation to latest merge some messages have been deleted
- Merge branch 'MK3' into MK3_Dutch
- Merge pull request #2217 from mkbel/MM_lay1_cal
- Merge branch 'MK3' into MM_lay1_cal
- Merge pull request #2246 from mkbel/reset_live_Z
- Use content supplied strings for lcd_first_layer_calibration_reset(), as we have translations prepared for release. It will be reverted later to show c
- Continue to lcd_v2_calibration() after selection is made in lcd_first_layer_calibration_reset(). Do not ask for selection if sheet is not calibrated, it is
- Do all text formatting in first layer calibration reset dialog in one line.
- Print sheet name and Z offset in first layer calibration reset dialog.
- Ask to reset Z offset before first layer calibration.
- Merge pull request #2245 from DRracer/knob-long-press
- fix the knob long-press causing hang in modal dialogs
- Fixes after first feature tests
- Allow knob-long-press only in listed menus
- Increase MENU_DEPTH_MAX to 7 to allow sheet calibration with MMU
- Merge branch 'MK3' into MM_lay1_cal

[main] e54497d80266ff8186cc08979cc53fee1a47b8e7 - commit 4405 of 7819

Which strategy?



The image shows a screenshot of a GitHub pull request interface. At the top, there are status indicators: '1 approval', '1 pending reviewer', 'All checks have passed' (5 successful checks), and 'This branch has no conflicts with the base b' (Merging can be performed automatically). Below these is a green button labeled 'Merge pull request' with a dropdown arrow, followed by the text 'or view [command line instructions](#)'. Three merge strategy options are listed below:

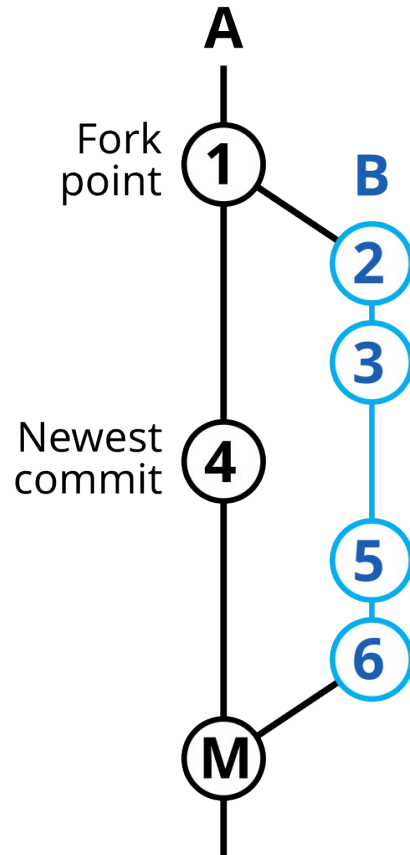
- Create a merge commit**: All commits from this branch will be added to the base branch via a merge commit.
- Squash and merge**: The 2 commits from this branch will be combined into one commit in the base branch.
- Rebase and merge**: The 2 commits from this branch will be rebased and added to the base branch.

Three arrows point from text labels to these options:

- A red arrow points from the text **"Noisy history"** to the 'Create a merge commit' option.
- A blue arrow points from the text **"Destroy history"** to the 'Squash and merge' option.
- A black arrow points from the text **"Fake history"** to the 'Rebase and merge' option.

Strategies compared

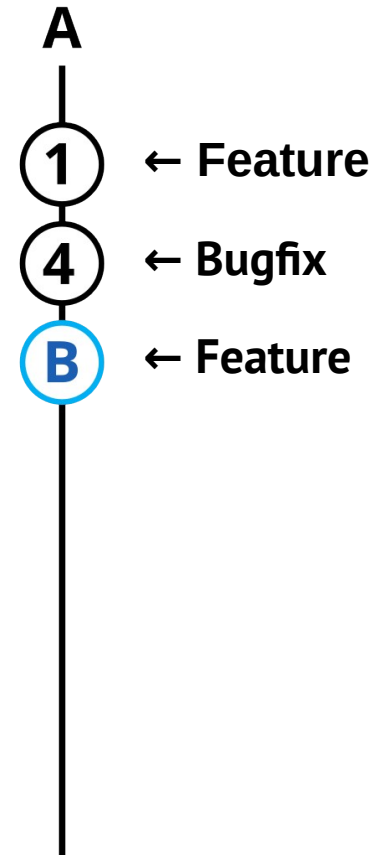
MERGE



REBASE



SQUASH



Workflow-based development

- Each branching methodology has pros and cons
- A branching policy for a project is called a "workflow":
 - Eg: "Always split logical changes":
 - Each **feature** into it's own branch
 - Each **bugfix** into it's own branch
- Tools to handle popular workflows:
git-flow / GitHub flow / GitLab flow / OneFlow ...

Workflow-based development

- Workflows help when working in large projects
- ... but increase reliance on branching
- Branching already increases with **team size**
- Branch dependency now common!

Workflow dependency example

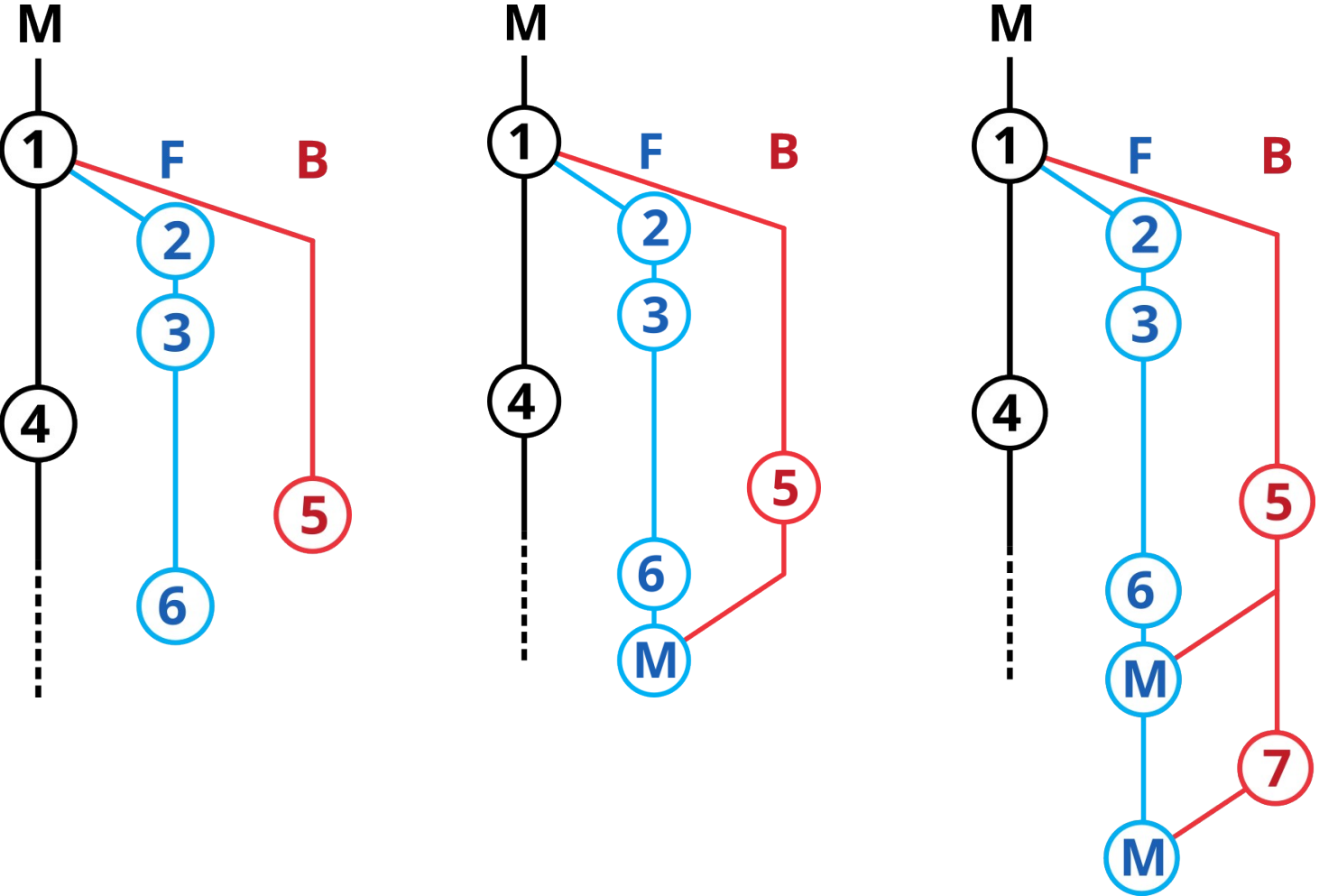
- Start working on a **feature**
- Bug discovered preventing **feature** from working:
 - re-branch from mainline for **bugfix**
 - submit bug for approval (stall **bugfix**)
- Return to feature:
 - **feature** now depends on **bugfix**
 - merge/rebase or **stall** feature?

What if **bugfix** requires changes as **feature** is updated?

Manual branch management

- Updating the **bugfix** branch:
 - switch to **bugfix** branch
 - modify, commit as required...
 - push
- Updating the **feature** branch:
 - switch to **feature** branch
 - merge **bugfix**
 - continue development as usual...

Bugfix & Feature evolution




Stalled branch reasons

- Testing:
 - Lengthy CI
 - Physical verification undergoing
- Policy:
 - "Feature can't be merged in current release"
- Review:
 - Timezone delays
 - Relevant reviewers busy
 - Unknown/different remote team priorities
 - No upstream interest

example: languishing PRs









prusa3d / Prusa-Firmware

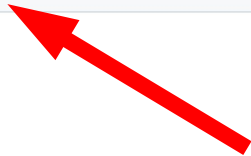
Type  to search

<> Code Issues 243 **Pull requests 70** Actions Projects Wiki Security Insights

Filters Labels 92 Milestones 3 [New pull request](#)

Clear current search query, filters, and sorts

<input type="checkbox"/>	 3 Open ✓ 193 Closed	Author ▾	Label ▾	Projects ▾	Milestones ▾	Reviews ▾	Assignee ▾	Sort ▾
<input type="checkbox"/>	 Improve filament sensor PCB upgrade logic #3155 opened on May 21, 2021 by wavexx • Review required  FW 3.14.0							
<input type="checkbox"/>	 Fix filament change behavior when issued in paused state #2390 opened on Dec 28, 2019 by wavexx • Review required  FW 3.14.0					1		18
<input type="checkbox"/>	 Perform a small extrusion prior to unload #2318 opened on Nov 9, 2019 by wavexx • Review required							20



Branch management

- Development on stalled branched **can** continue with *more burden*
- Branch dependencies can be expressed and resolved with merge/rebase operations
- ... why not introducing automation?

git-assembler

- At its core:
 - Declare how branches should be combined with either **rebase** or **merge** with rules
- What it does:
 - Performs the operation **only when necessary**
 - Performs dependent operations **in topological order**
 - **Shows the branch dependency state!**

git-assembler example

- Keep **feature** updated by merging **bugfix** when new commits appear using one rule:

merge feature bugfix

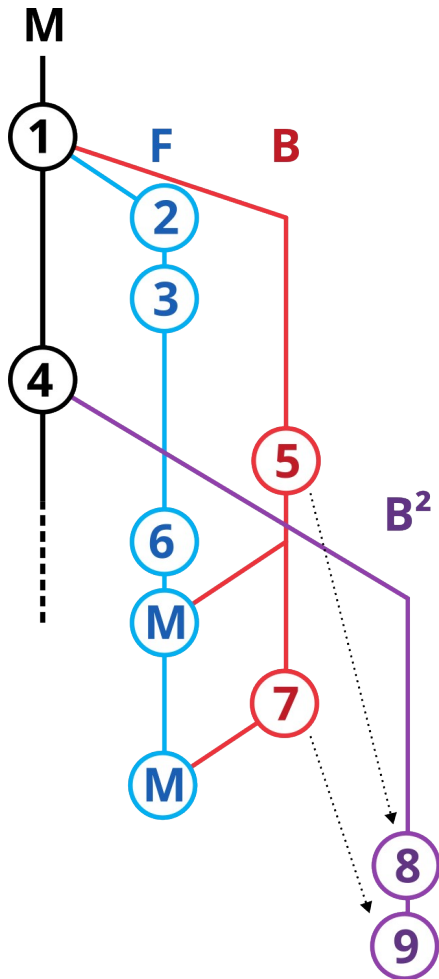
- How do run it:

\$ git as -a

- How it helps:

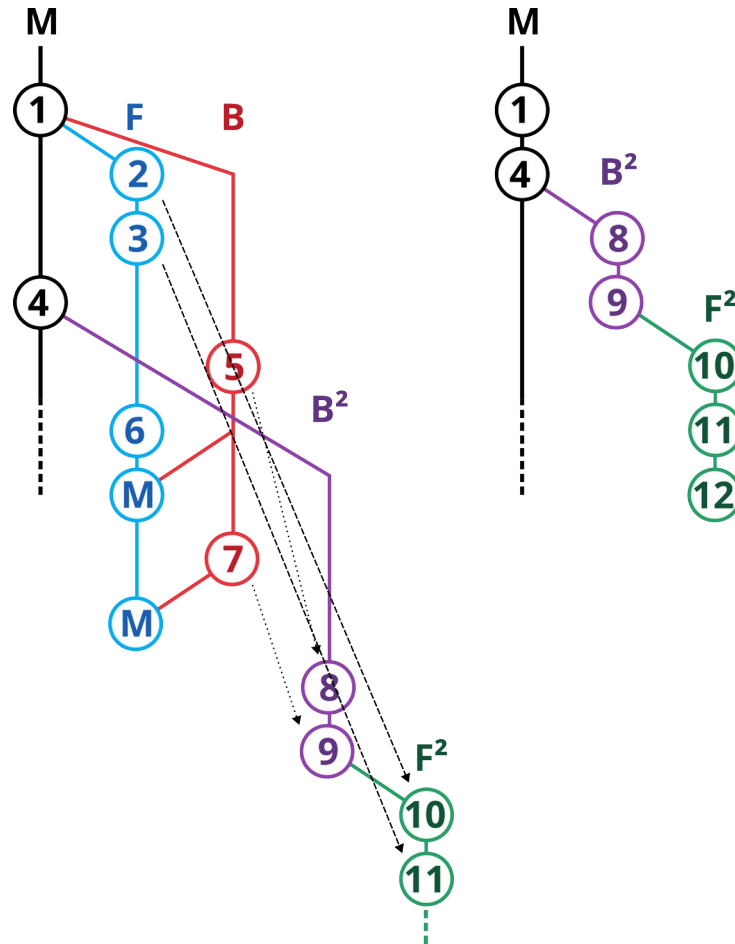
- switches to feature branch *if needed*
- merge bugfix *if needed*
- switches to the original branch *if needed*

Bugfix & Feature evolution²

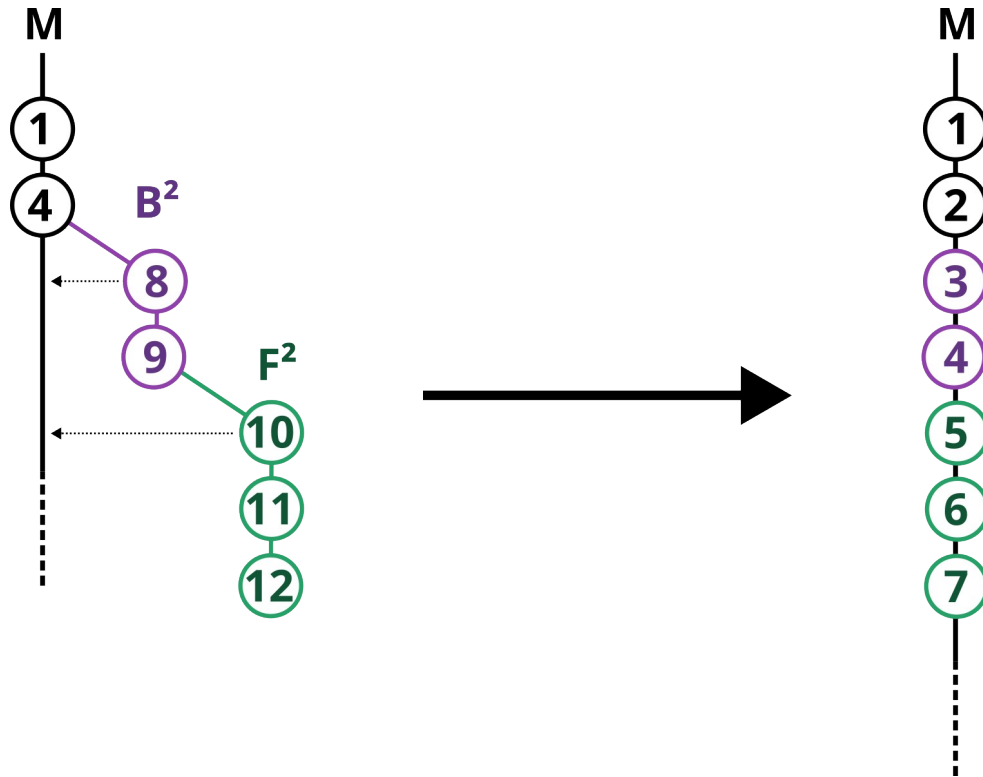


- Reviewer calls for **rebase** of **B**
- merge **can't** be used on feature anymore

Bugfix & Feature evolution²



Bugfix & Feature evolution²



git-assembler example²

- Keep **bugfix** (and **feature**) rebased on master:

```
rebase bugfix master
```

```
rebase feature bugfix
```

- How to see the branch topology:

```
$ git as
```

```
feature <- bugfix <- master
```

- How it helps:

- rebases only *when required*
- rebases in the *correct order*

Usage in practice

```
base MK3 upstream/MK3
merge MK3 tune_during_pause #2390
merge MK3 enhanced_unload #2318
merge MK3 stop_raise_higher #3455
merge MK3 unify_fullscreen_msg_handling #3090
merge MK3 filament_load_consistency #3456
merge MK3 remove_runout_support #3457
#merge MK3 lcd_temperature_jumping #3494
#merge MK3 extrude_min_temp #3495

merge MK3 panayiotis/MK3_ButtonLongPress_Fix #3014
merge MK3 panayiotis/MK3_NoBedHeating_onLoadUnload #2335

merge MK3 leptun/M73_fixes

rebase pinda_newcal MK3

stage master MK3
merge master pinda_newcal
```

```
$ git as
>master <= [MK3]
  pinda_newcal <- [MK3]
MK3 .. upstream/MK3
  tune_during_pause
  enhanced_unload
  stop_raise_higher
  unify_fullscreen_msg_handling
  filament_load_consistency
  remove_runout_support
  panayiotis/MK3_ButtonLongPress_Fix
  panayiotis/MK3_NoBedHeating_onLoadUnload
  leptun/M73_fixes
$ █
```

↑
declarations

resulting
structure ↑

Branch operations

- Merge
- Rebase
- **Stage**
 - **Always** re-create a new branch from a known starting point
- **Base**
 - **Optionally** re-create a new branch from a known starting point

(git) tools for patch management

- **st-git**: Linear stack of patches (similar to quilt)
<https://github.com/stacked-git/stgit>
- **topgit**: Patch history and dependency
<https://github.com/mackyle/topgit>
- **git-assembler**: Branch/tree dependency
<https://gitlab.com/wavexx/git-assembler>
- **git-replay**: Branch dependency
<https://github.com/olets/git-replay>

Why git-assembler?

- Assumes *nothing* from upstream, contributors or collaborators:
 - Any **remote** workflow is acceptable
 - Any **local** workflow can be replicated
- **Can** be used cooperatively (with a versioned configuration)
- Stateless:
 - Repository can be **manipulated by hand**
 - **Rules can be changed at any time**
- Supports worktrees:
 - Branches are updated in their respective worktree!