

Git Rebase Workflow

Demo Day

David Shi

September 2024



Table of Contents

- 1 What's a Github?
- 2 Merge Workflow
- 3 Rebase Workflow
- 4 Pre-Demo Warmup



Table of Contents

1 What's a Github?

2 Merge Workflow

3 Rebase Workflow

4 Pre-Demo Warmup



What's a Github?



What's a version control?

version control [vər'zHən kən'trōl] *noun*

records changes to a file or set of files and allows you to recall the specific version of the file you want later



What's a version control?

version control [vər'zHən kən'trōl] *noun*

records changes to a file or set of files and allows you to recall the specific version of the file you want later

Pros

- multiple people can work on the same files
- can restore to backup if you made a boo boo
- see exactly what was changed, when and by who



What's a version control?

version control [vər'zHən kən'trōl] *noun*

records changes to a file or set of files and allows you to recall the specific version of the file you want later

Pros

- multiple people can work on the same files
- can restore to backup if you made a boo boo
- see exactly what was changed, when and by who

Cons

- Nothing :^)



What's a version control?

version control [vər'zHən kən'trōl] *noun*

records changes to a file or set of files and allows you to recall the specific version of the file you want later

Pros

- multiple people can work on the same files
- can restore to backup if you made a boo boo
- see exactly what was changed, when and by who

Cons

- Nothing :^)
- a little bit of a learning curve :^)



Table of Contents

1 What's a Github?

2 Merge Workflow

3 Rebase Workflow

4 Pre-Demo Warmup



Typical Merge Flow

Write some code

Commit to branch

Merge in to master

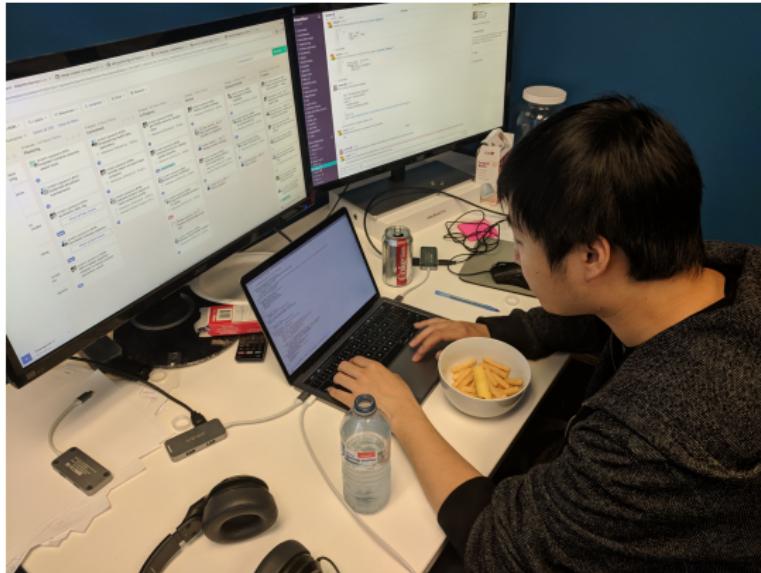
Start working on a new
feature

Check in your changes to
a Git repository

Finish feature and bring
changes in to master



Typical Merge Flow - Solo Developer



- commit directly to master
- no conflicts
- no wait time for merging



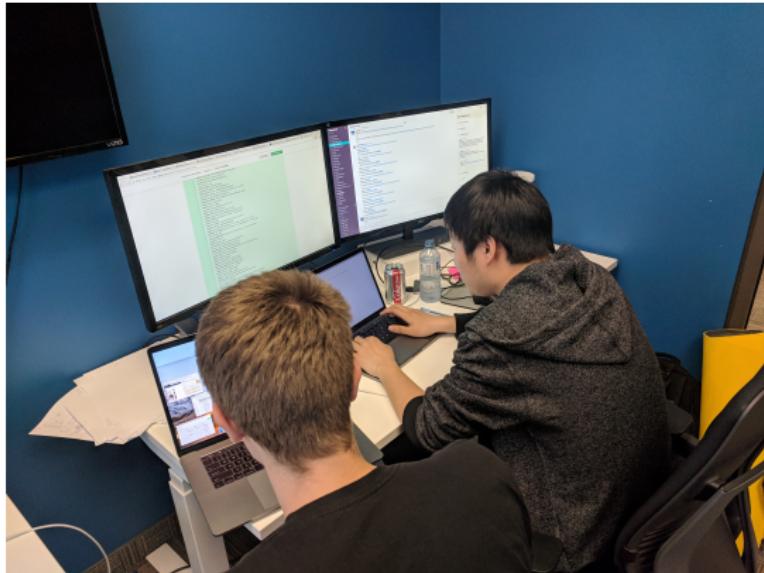
Typical Merge Flow - Solo Developer Git History

- Merge pull request #2 from AdeptMind/URL
url filtering based on regex/xpath
- add canadagoose
- fix config spider
- add new spiders
- fix bug
- avoid invalid keys
- merging
- fix
- extracting configurations
- prepare for another round of extraction
- do local color normalization
- update persistence
- update color extractor
- working on color
- add facts resolver
- fix color extractor
- command-line fix
- got macys colors
- fix filter url
- fix routes management; apply url filter universally
- labeler fix
- primitive labeler
- wip
- move to ubuntu
- adding random review text to test set
- adding search expansion
- remove redundant url management
- updated response checking
- adopt the document model
- add primary url index; move broken page out of tests
- split amazon spider; add extraction tests
- extraction optimization
- further optimization
- performance improvements
- no url expansion for now
- upgrading amazon search crawler with separate storage and search group management
- price extractor update
- update extraction storage
- fix extraction
- update
- mre restrictive link chasing
- continues on to property extraction
- decoding search results
- add noahsnaturalfoods; cosmetic fixes
- macys api decoded; speed fix
- deal with page inconsistency
- fix
- db as dependency injection
- much more sites; generalized evaluator
- first commit with tests

Siwei Yang <yangsiwei880813@gmail.com>
Huangxiao Li <tony@adeptmind.ai>
Siwei Yang <yangsiwei880813@gmail.com>
Siwei Yang <s8yangi@uwaterloo.ca>
Siwei Yang <s8yangi@uwaterloo.ca>
Siwei Yang <s8yangi@uwaterloo.ca>
Siwei Yang <s8yangi@uwaterloo.ca>
Siwei Yang <yangsiwei880813@gmail.com>
Siwei Yang <yangsiwei880813@gmail.com>



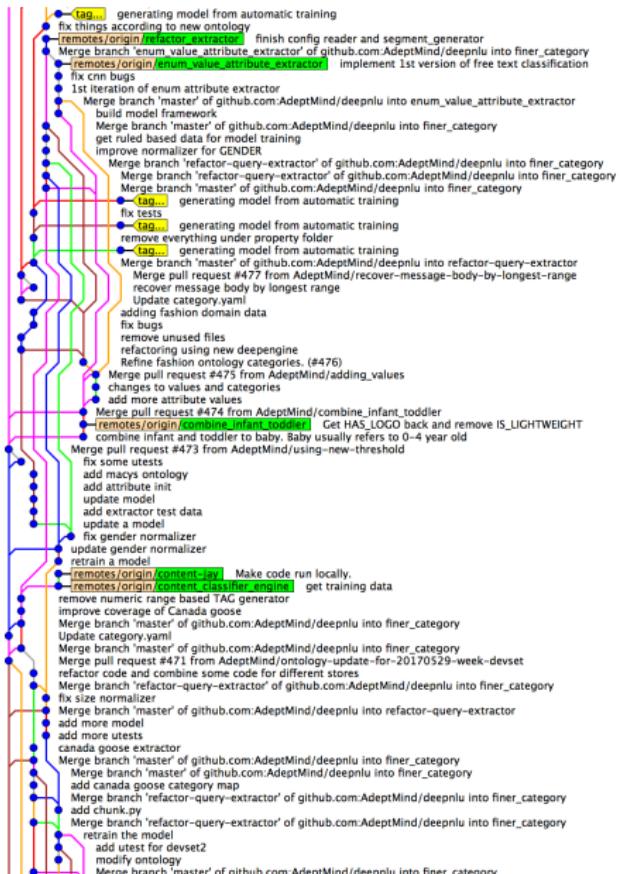
Typical Merge Flow - 2-4 Developers



- start to use branching strategy
- start to see conflicts
- need to wait between merges
- git history gets more complex, but manageable

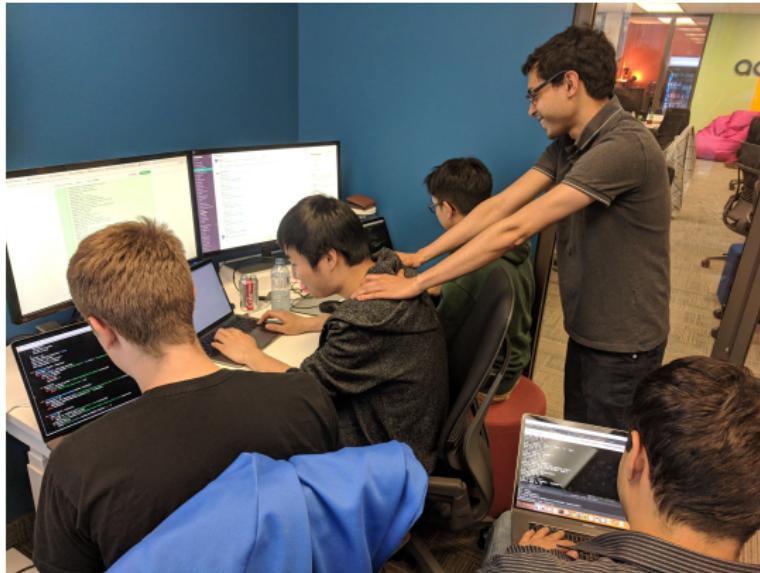


Typical Merge Flow - 2-4 Developers Git History



Siwei Yang <yangsiwei180813@gmail.com>
Jing He <peaceful.he@gmail.com>
warrior97 <warrior97@gmail.com>
warrior97 <warrior97@gmail.com>
james <warrior97@gmail.com>
james <warrior97@gmail.com>
warrior97 <warrior97@gmail.com>
Siwei Yang <yangsiwei180813@gmail.com>
Jing He <peaceful.he@gmail.com>
Siwei Yang <yangsiwei180813@gmail.com>
Siwei Yang <yangsiwei180813@gmail.com>

Typical Merge Flow - 5+ Developers



- lots of branches
- start to see conflicts
- one developer dedicated to merging

Typical Merge Flow - 5+ Developers Git History

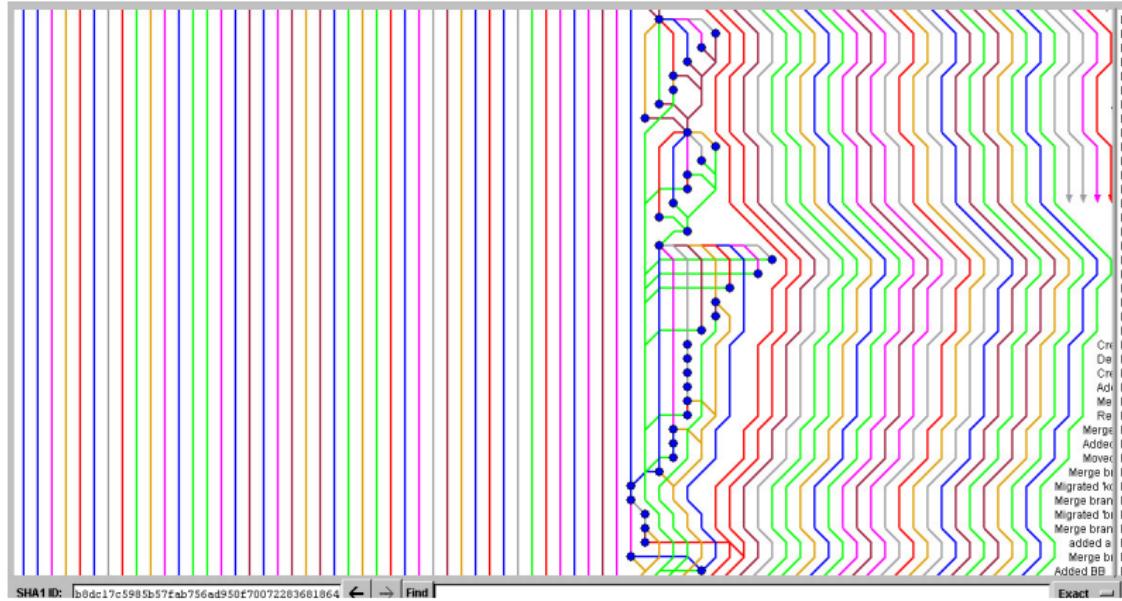




Table of Contents

1 What's a Github?

2 Merge Workflow

3 Rebase Workflow

4 Pre-Demo Warmup



What do?

Solution

- `git pull --rebase origin develop`
- `git rebase -i origin/develop`



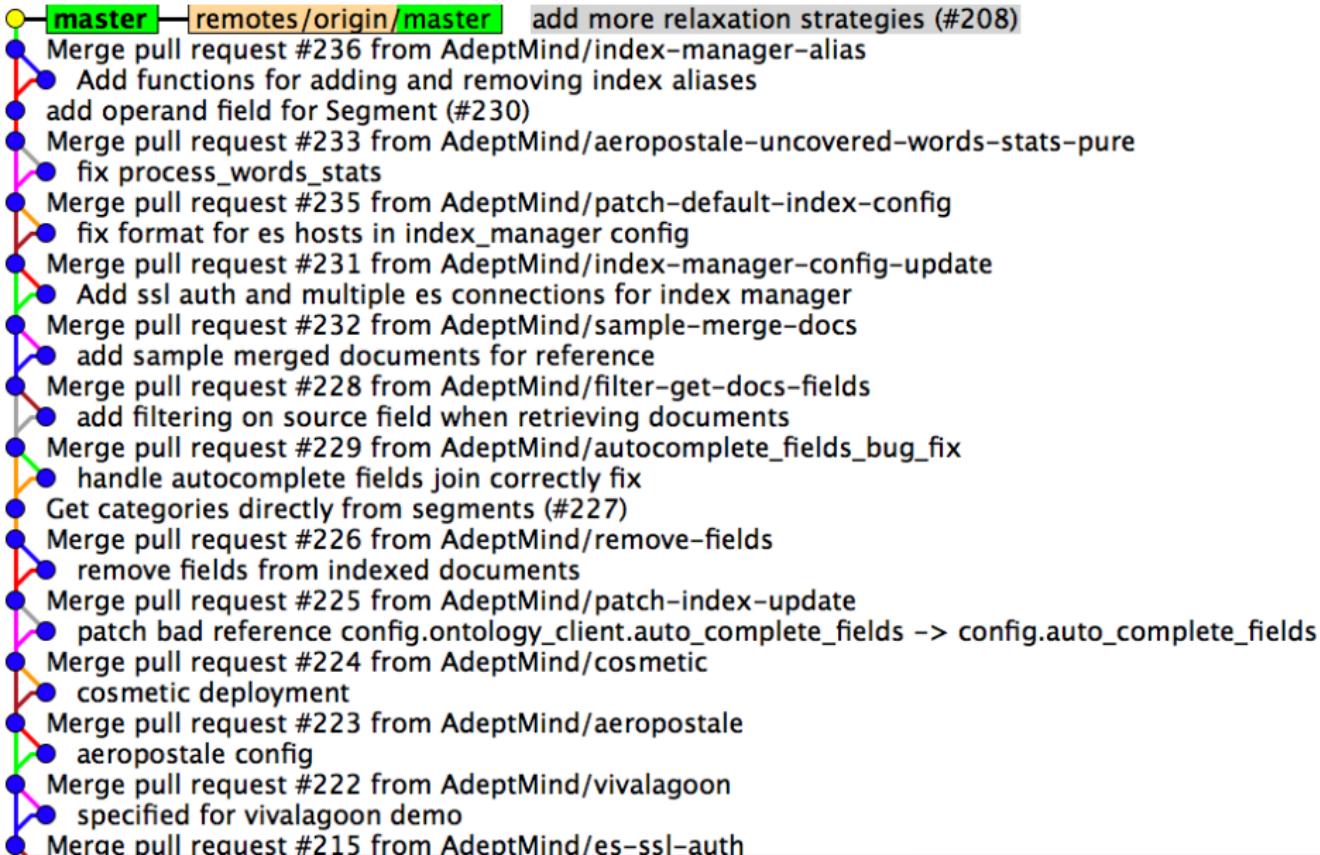
What do?

Solution

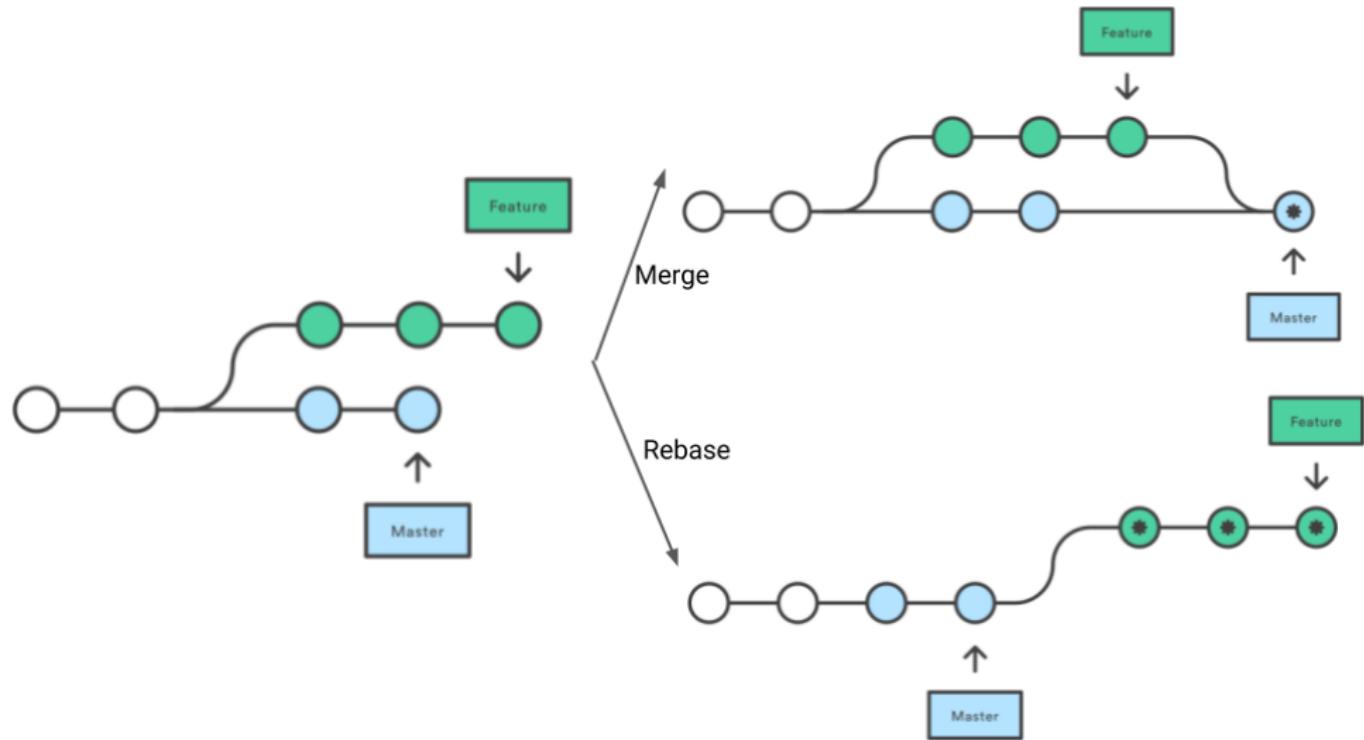
- `git pull --rebase origin develop`
 - `git rebase -i origin/develop`
-
- Creates a linear history
 - Easy to handle product release, backporting and release
 - Conflicts are solved by one developer, and everyone else gets resolutions for free (mostly)
 - Proven to scale, as proven by all large open source projects



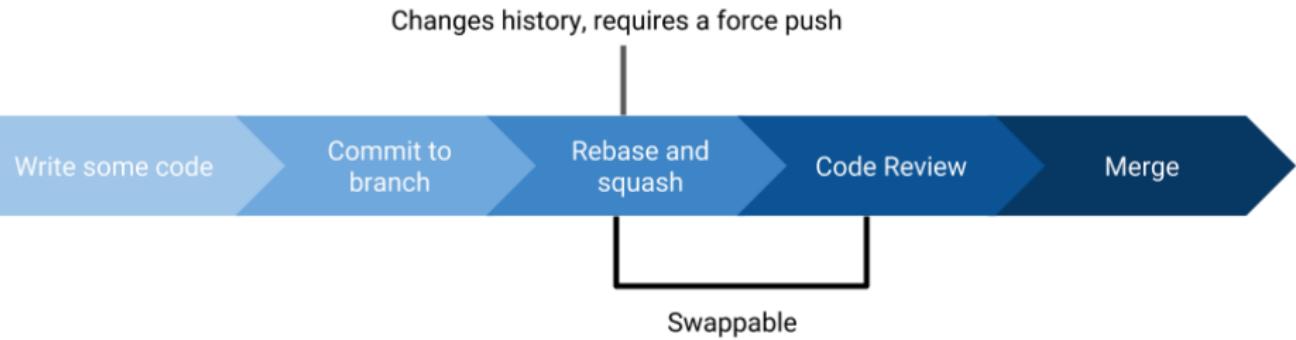
Rebase Flow - Taste the Teamwork



What's the Difference, Actually?



Rebase Flow



Only You Can Prevent Forest Fires



Table of Contents

1 What's a Github?

2 Merge Workflow

3 Rebase Workflow

4 Pre-Demo Warmup



Recap Git Fundamentals

what am I doing?

- git branch
- git status
- git diff
- gitk --all

making changes

- git checkout
- git commit
- git push

rewriting history

- git pull --rebase
- git rebase -i

TEAM?!

Thank you
(demo time?)

