

# Git in real life

Alessandro Sciarra

Z02 – Software Development Center

XX December 2022

# Outline of the talk

- 1 Rewriting history
- 2 Git rebase
- 3 Git tag
- 4 Semantic versioning
- 5 Gitflow

Rewriting history

# About rewriting history

Yes, you can change the history of a repository!



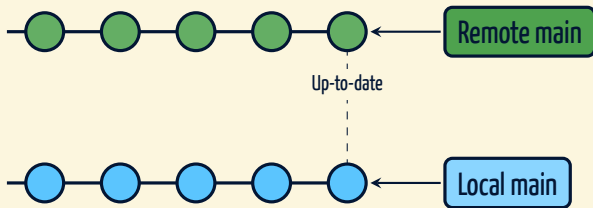
- **Very bad if there are different copies of the history** { E.g. branches, remote(s) }
- **If you rewrite shared history,**
  - it is generally hard to make the same change elsewhere and
  - merging (and hence pull/push) can lead to duplicated commits in history
- **Fine as long as**
  - yours is the only repository containing the rewritten history or
  - you work on a git project alone and you know what to do then

# About rewriting history

Yes, you can change the history of a repository!



- **Very bad if there are different copies of the history** { E.g. branches, remote(s) }
- If you rewrite shared history,
  - it is generally hard to make the same change elsewhere and
  - merging (and hence pull/push) can lead to duplicated commits in history

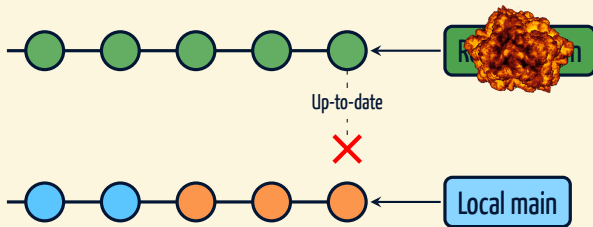


# About rewriting history

Yes, you can change the history of a repository!



- **Very bad if there are different copies of the history** { E.g. branches, remote(s) }
- If you rewrite shared history,
  - it is generally hard to make the same change elsewhere and
  - merging (and hence pull/push) can lead to duplicated commits in history



As long as you rewrite history contained elsewhere, you are in troubles!

# Changing the last commit (message)

## 1 How can I fix a typo in the **last** commit message?

This procedure will change history!

Said differently, feel free to do it if you **did not** share history!

```
$ git commit -m "Added engine implementation"
[airplane f8b0c25] Added engine implementation
24 files changed, 1340 insertions(+), 476 deletions(-)
# Ops! I used a verb in past form, ✎not conforming!
```

# Changing the last commit (message)

## 1 How can I fix a typo in the last commit message?

This procedure will change history!

Said differently, feel free to do it if you **did not** share history!

```
$ git commit -m "Added engine implementation"
[airplane f8b0c25] Added engine implementation
24 files changed, 1340 insertions(+), 476 deletions(-)
# Ops! I used a verb in past form, ✎not conforming!
```

# With clean staging area:

```
$ git commit --amend -m "Add engine implementation"
[airplane 33f53f4] Add engine implementation
Date: Fri Nov 25 08:52:22 2018 +0100
24 files changed, 1340 insertions(+), 476 deletions(-)
```



# Changing the last commit (content)

2 I forgot some changes in the **last** commit! And now?

This procedure will change history!

Said differently, feel free to do it if you **did not** share history!

```
$ git commit -m "Review take-off system"
[airplane e1df32a] Review take-off system
1 file changed, 230 insertions(+), 61 deletions(-)
# Ops! I forgot a file!
```

# Changing the last commit (content)

2 I forgot some changes in the **last** commit! And now?

This procedure will change history!

Said differently, feel free to do it if you **did not** share history!

```
$ git commit -m "Review take-off system"
[airplane e1df32a] Review take-off system
1 file changed, 230 insertions(+), 61 deletions(-)
# Ops! I forgot a file!
```

```
$ git add wheels_electronic.h
$ git commit --amend --no-edit
[airplane c13e34f] Add engine implementation
Date: Fri Nov 25 08:45:18 2022 +0100
2 files changed, 345 insertions(+), 88 deletions(-)
```

# Changing the last commit (content)

2 I forgot some changes in the **last** commit! And now?

This procedure will change history!

Said differently, feel free to do it if you **did not** share history!

```
$ git commit -m "Review take-off system"
[airplane e1df32a] Review take-off system
1 file changed, 230 insertions(+), 61 deletions(-)
# Ops! I forgot a file!
```

```
$ git add wheels_electronic.h
$ git commit --amend -C HEAD # use original commit timestamp
[airplane c13e34f] Add engine implementation
Date: Fri Nov 25 08:33:01 2022 +0100
2 files changed, 345 insertions(+), 88 deletions(-)
```

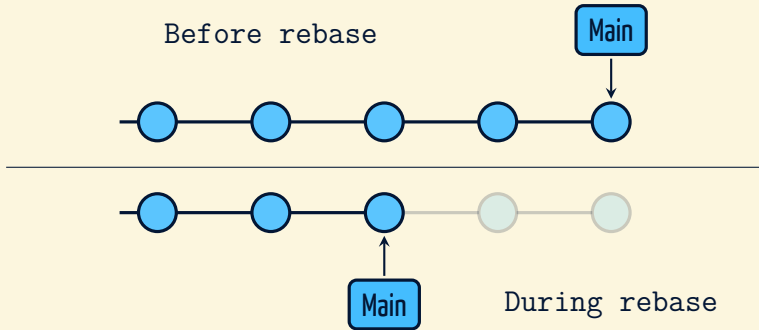
# Git rebase

# Which is the abstract idea of a rebase?



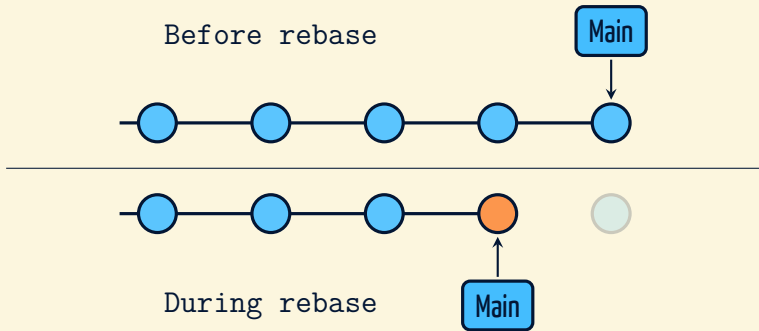
# Which is the abstract idea of a rebase?

- 1 Go back in history till a given point



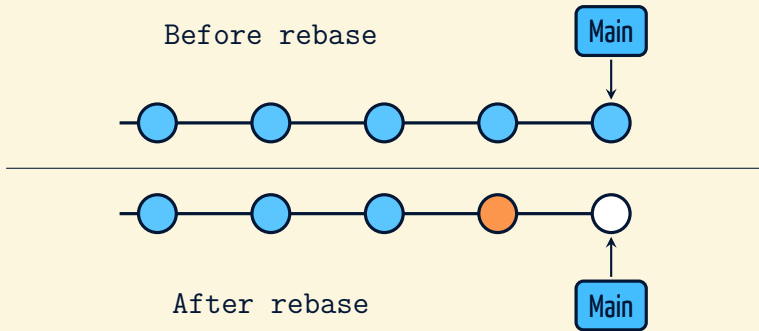
# Which is the abstract idea of a rebase?

- 1 Go back in history till a given point
- 2 Replay commits in a given order
  - ↳ possibly taking action on applied commit as specified



# Which is the abstract idea of a rebase?

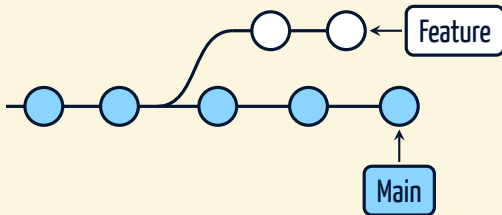
- 1 Go back in history till a given point
- 2 Replay commits in a given order
  - ↳ possibly taking action on applied commit as specified



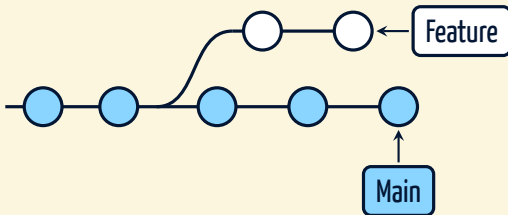


# Rebasing instead of (three-way) merging

Three way merge from Feature:

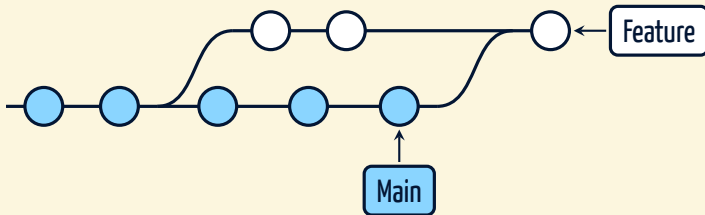


Rebase from Feature:

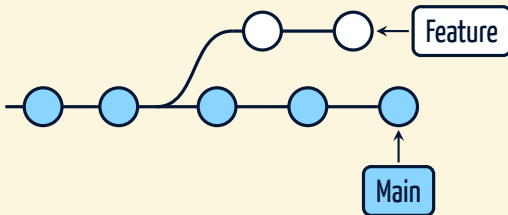


# Rebasing instead of (three-way) merging

Three way merge from Feature:

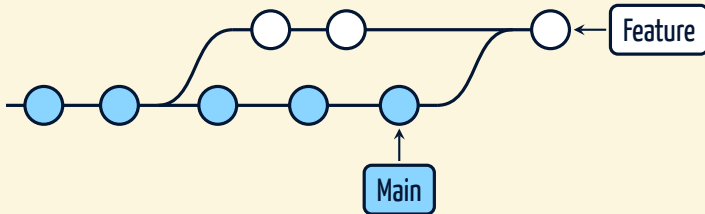


Rebase from Feature:

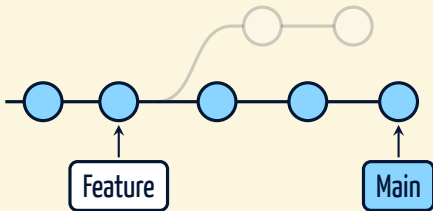


# Rebasing instead of (three-way) merging

Three way merge from Feature:

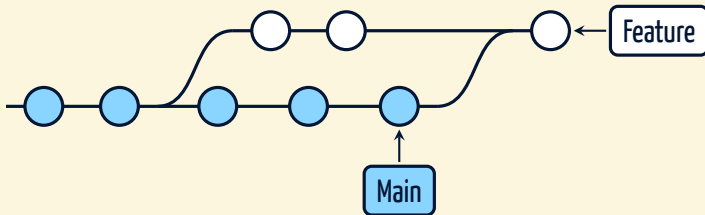


Rebase from Feature:

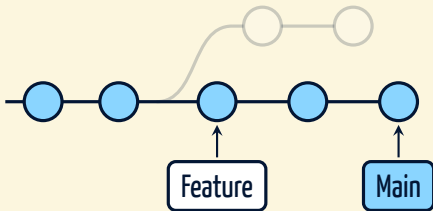


# Rebasing instead of (three-way) merging

Three way merge from Feature:

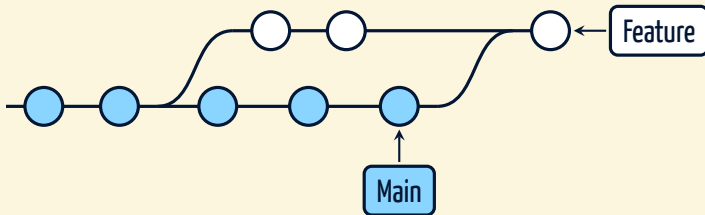


Rebase from Feature:

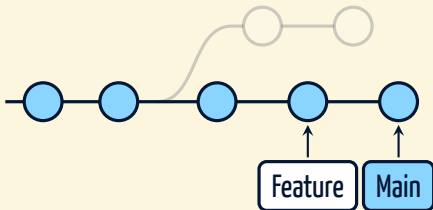


# Rebasing instead of (three-way) merging

Three way merge from Feature:

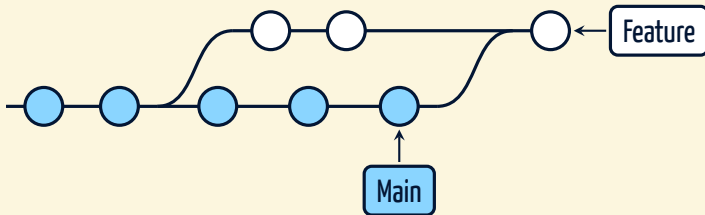


Rebase from Feature:

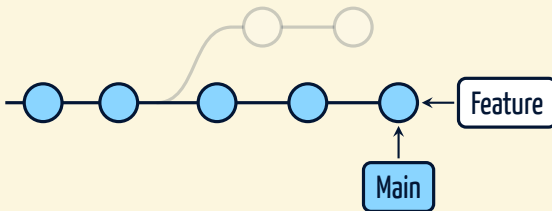


# Rebasing instead of (three-way) merging

Three way merge from Feature:

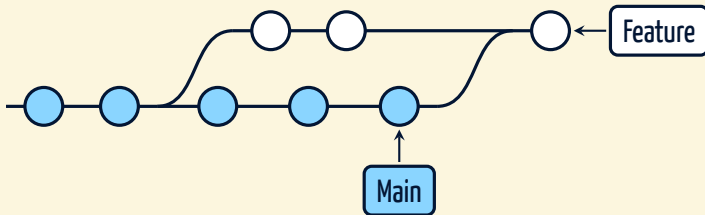


Rebase from Feature:

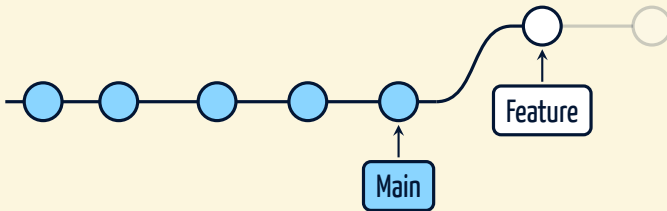


# Rebasing instead of (three-way) merging

Three way merge from Feature:

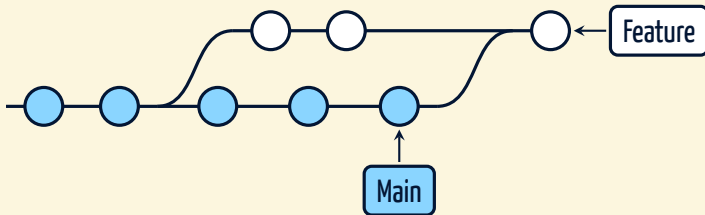


Rebase from Feature:

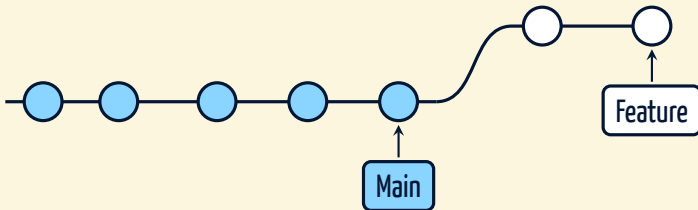


# Rebasing instead of (three-way) merging

Three way merge from Feature:



Rebase from Feature:





Git tag

# Semantic versioning

Gitflow