Agenda

- Regular git workflow
- Common mistakes and solutions
- Handling community contributions
Regular git workflow

$ git checkout bb-10.2-bug

$ # Code, build, test...

$ git add .

$ git commit

$ git push

Make sure your branch is up to date before starting work!
Regular git workflow

$ git checkout bb-10.2-bug

$ # Code, build, test...

$ git add .

$ git commit

$ git push

Keep changes small, do one thing!
Test!
Regular git workflow

$ git checkout bb-10.2-bug

$ # Code, build, test...

$ git add .

$ git commit

$ git push

Make sure you only add your changes.
Regular git workflow

$ git checkout bb-10.2-bug

$ # Code, build, test...

$ git add .

$ git commit

$ git push
$ git checkout bb-10.2-bug

$ # Code, build, test...

$ git add .

$ git commit

$ git push

Push to staging branch.
Regular git workflow

$ git rebase 10.2

$ git checkout 10.2

$ git merge bb-10.2-bug

$ git push

Make sure main branch is up to date locally.
Regular git workflow

$ git rebase 10.2
$ git checkout 10.2
$ git merge bb-10.2-bug
$ git push

This merge will be a fast-forward due to previous rebase.
Common mistakes and solutions

- Forgetting to rebase

- After git merge feature-branch
Common mistakes and solutions

- **Undo merge commit:**
  
  ```
  $ git checkout master; git reset --hard HEAD~1
  ```

- **Rebase feature branch**
  
  ```
  $ git checkout feature-branch
  
  $ git rebase master
  ```

- **Merge again**
  
  ```
  $ git checkout master
  
  $ git merge feature-branch
  ```
Common mistakes and solutions

- Bugfix should be applied to different branch.

    $ git checkout correct-branch

    $ git cherry-pick <commit-hash>

    # Resolve merge if there are any conflicts
Common mistakes and solutions

- Wrong commit message, wrong commit order, too many commits, etc.

- Solution: git rebase -i
Common mistakes and solutions

- Wrong commit message, wrong commit order, too many commits, etc.

- Solution: `git rebase -i`
Common mistakes and solutions

- How to write a good commit message?
- For regular bug fixes use template:
  MDEV-XXXX: mdev-title
  <blank row>
  Verbose commit message.
- Keep title to ~72 chars if possible
- Commit message should focus on WHY something was done.
- Title should hint at WHAT.
- Use proper capitalized sentences.
Handling community contributions

- We get a lot of community contributions

- After code review, ask author to improve commit.

- If author does not respond & it makes sense, fix it yourself.

- Take care to give credit!

  git commit --author “Contributor-Name <contributor-email>”
  --signoff

- If commit can be merged through Github, credit is automatically given.