Introduction to Git and Github

Joachim Mammele

What is Git?

- Most commonly used version control system
- Tracks changes you or anybody else makes to files
- Provides a record of what has been done
- Allows to revert to previous versions
- Makes collaboration easier when several people work simultaniously on the same code base

Repositories

- Container for a project that is tracked by git
- Local repository
  - Isolated repository stored on your own computer
  - Allows you to work on the local code version
- Remote repository
  - Stored outside of your own computer
  - Usually on a remote server (github, gitlab, bitbucket)
  - Easies up collaboration with others, eg. Code review

Basic usage

- Initialize a repository
  - Command: git init
  - Creates a hidden .git folder in which all relevant information is stored
- Check for changes in repository
  - Command: git status

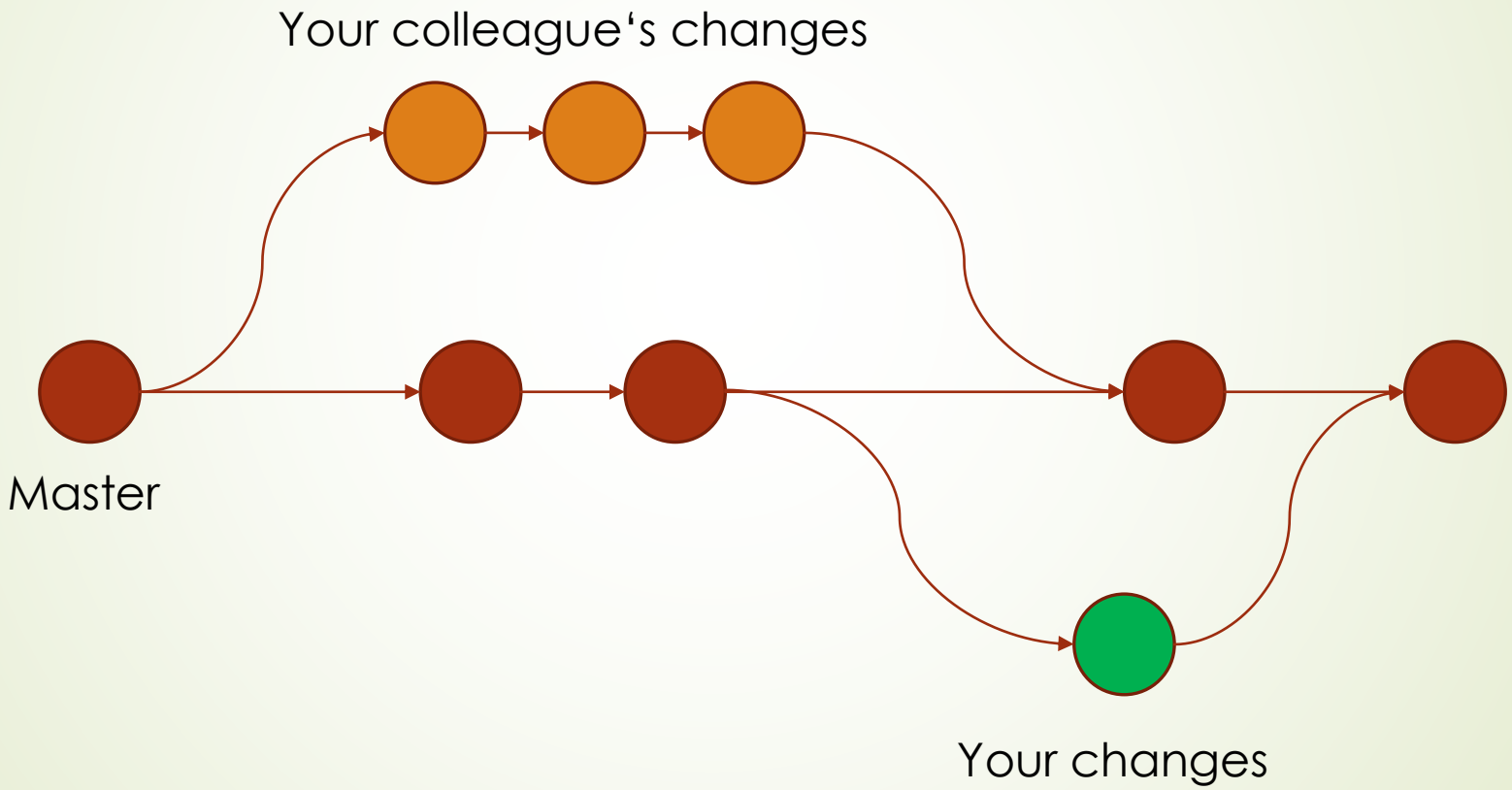
Basic usage

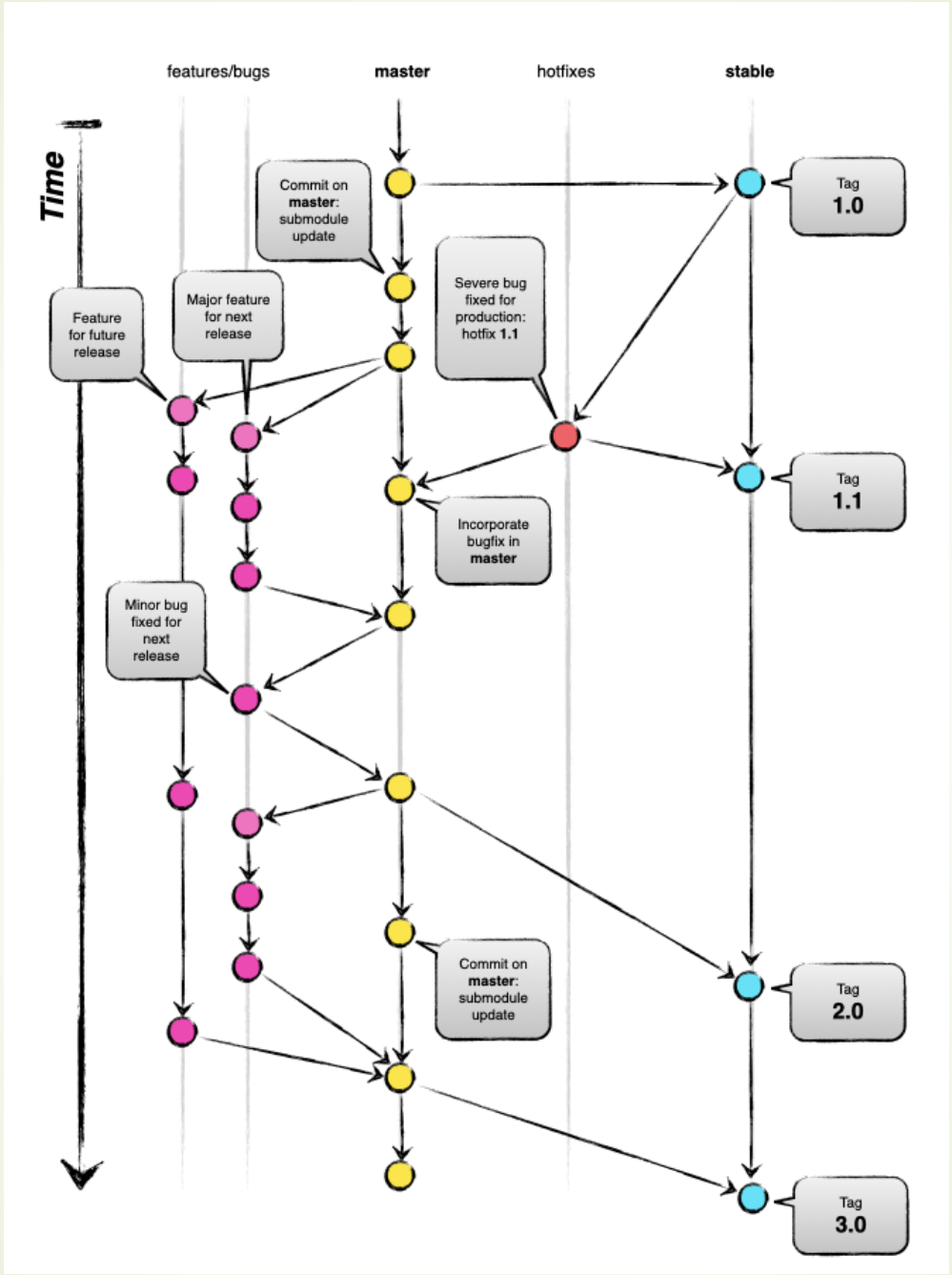
- Stage files
  - Makes git aware to monitor new files
  - Command: git add filename
- Commit files
  - Creates a snapshot of current code changes
  - Command: git commit –m „my changes“
- Show history of changes
  - Command: git log

Branches

- Allow to group changes
- Branches are created when working on
  - New features
  - Bugfixes
  - Doing quick experiments
- Help to keep the main code base clean
- Default branch is usually called „main“ or „master“

Branches - Usage





Source: https://gist.github.com/digitaljhelms/4287848

Further commands

- Merge
  - Process of bringing changes from a branch back to main
- Pull
  - Process of bringing changes from remote repository into the local repository
- Push
  - Process of bringing changes from local repository into the remote repository
- Pull requests
  - Allow to discuss/ code review changes before merging them into the main branch
  - Can be combined with automatic code checks (linting) and testing
  - Changes can only be merged if code is
    - Approved by maintainer
    - (Automatically) tested
    - Complies with coding rules

Branches - Usage

- Create a new branch and directly switch to it
  - Command: git checkout –b branch-name
- Switch to an already existing branch
  - Command: git checkout branch-name

Fork

- Copy of the whole repository
- Managed by you and not the original maintainer
- Allow to make changes without affecting the original repository
- You can fetch updates from or submit changes to the original repository with pull requests.
- Problem: with the time forks get outdated if not synched regularily

Taking it one step further

- Continous Integration
  - Developers merge their changes back to the main branch as often as possible
  - Changes are validated by creating a build from them and running automatic tests
  - Helps to avoid problems when branches with big changes need to be merged a few days before release day
- Continous Delivery
  - Extension of the previous one
  - Deploys all changes automatically to a testing and/or production environment after successful building and testing

- Screenshot of gitlab – removed for online version

Github Pages

- Automatically generate a website from your source code
- Slogan: Just edit, push – and your changes are live
- Example:
  - Website for the Latin-American ICCM
  - URL: https://cicm-al.org
  - Repository: https://github.com/cicm-al/cicm-al.github.io

Questions?

Hands-On Part

- Let‘s suppose we are a development team and need to work together on the same source code base
- We want to generate a calculator that sums up all numbers contained in different files
- Task:
  - Add your own file with a number on a branch
  - Create a pull request on github
  - Verify that build succeeds

```
▶ Run python calculator.py
Reading content of file test_file.txt
53
33
3

Reading content of file textfile1.txt
479
44

Reading content of file mysupernumber.txt
997



************************************************************
Summary of all numbers in all files is: 1609
************************************************************
```

- Check instructions on
- https://github.com/security-companion/iccm-eu-git-workshop