Version Control – Agile Workflow with Git/GitHub

15 - 17 November 2022 | Guido Trensch (JSC, Simulation & Data Lab Neuroscience)

Content

Content

Introduction

- Version control is one aspect of configuration management (CM) which is concerned with:
  - System building and Release Management
    - *Preparing software for releases and keeping track of system versions.*
  - Change management
    - *Keeping track of requests for changes, working out the costs and impact.*

[Ian Sommerville, *"Software Engineering"*]

Introduction

Why do we need version control?

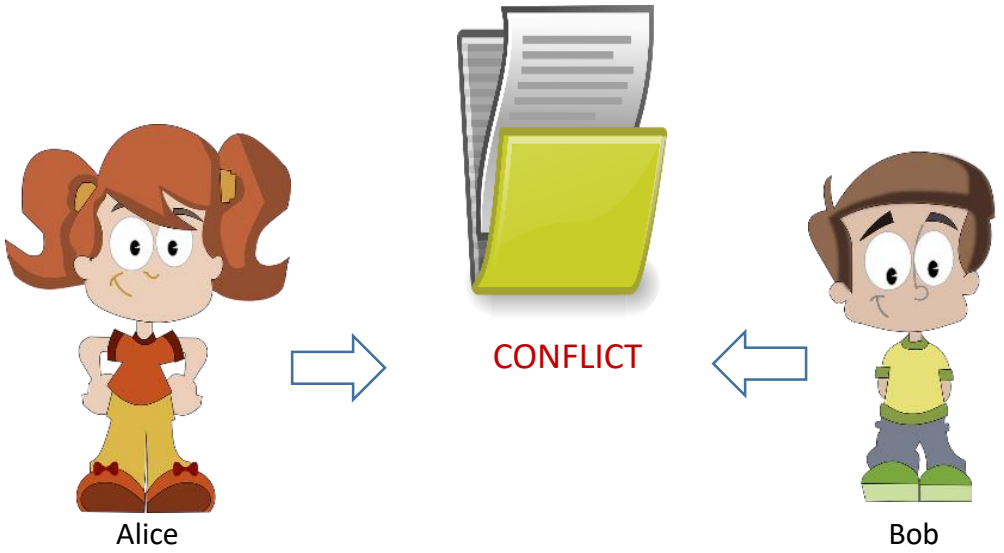
- Keep track of different versions of software components.
- Identify, store, organize and control revisions and access to it.
- Track contributions.
- **Enable independent development in multi-developer projects.**

Introduction

Ensure that changes made by different developers do not interfere with each other

and

provide strategies to solve conflicts in independent development!



Content

Version Control (VCS)

There are two types of version control systems:

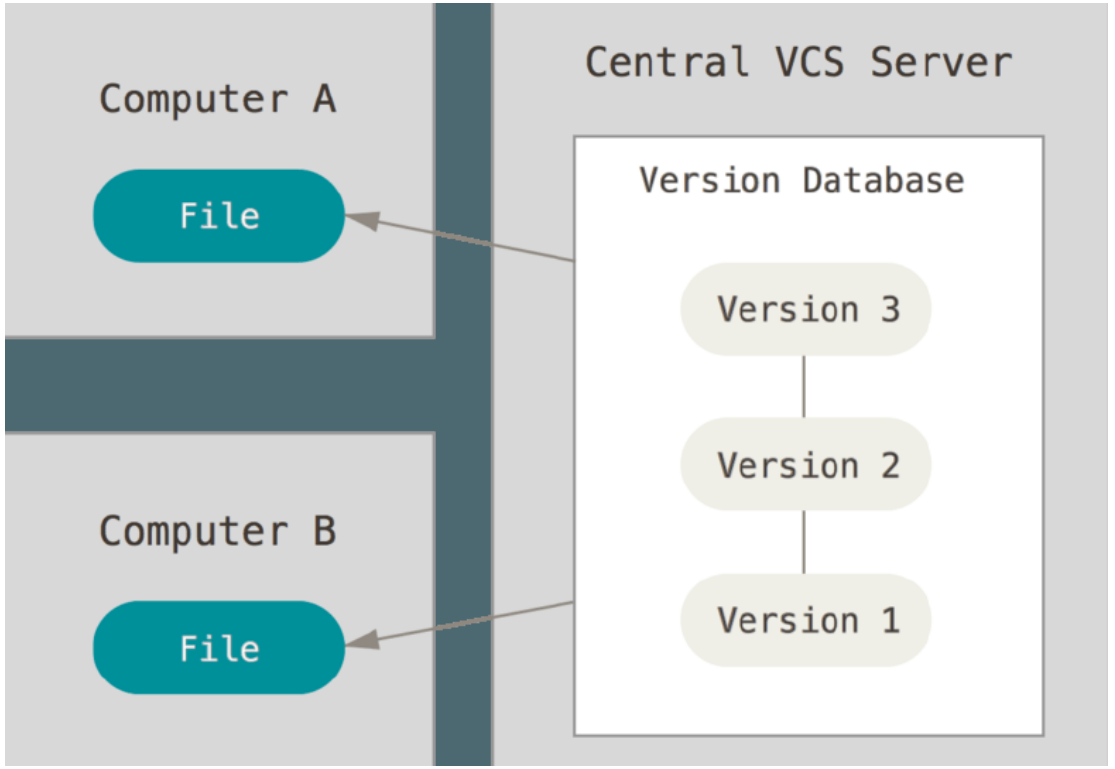
- Centralized systems
- Distributed systems

Version Control (VCS)

Centralized systems

- Maintain a single main (master) repository

- *Revision Control System (RCS, 1982)*
- *Concurrent Versioning System (CVS, 1986)*
- *Subversion (SVN, 2000)*

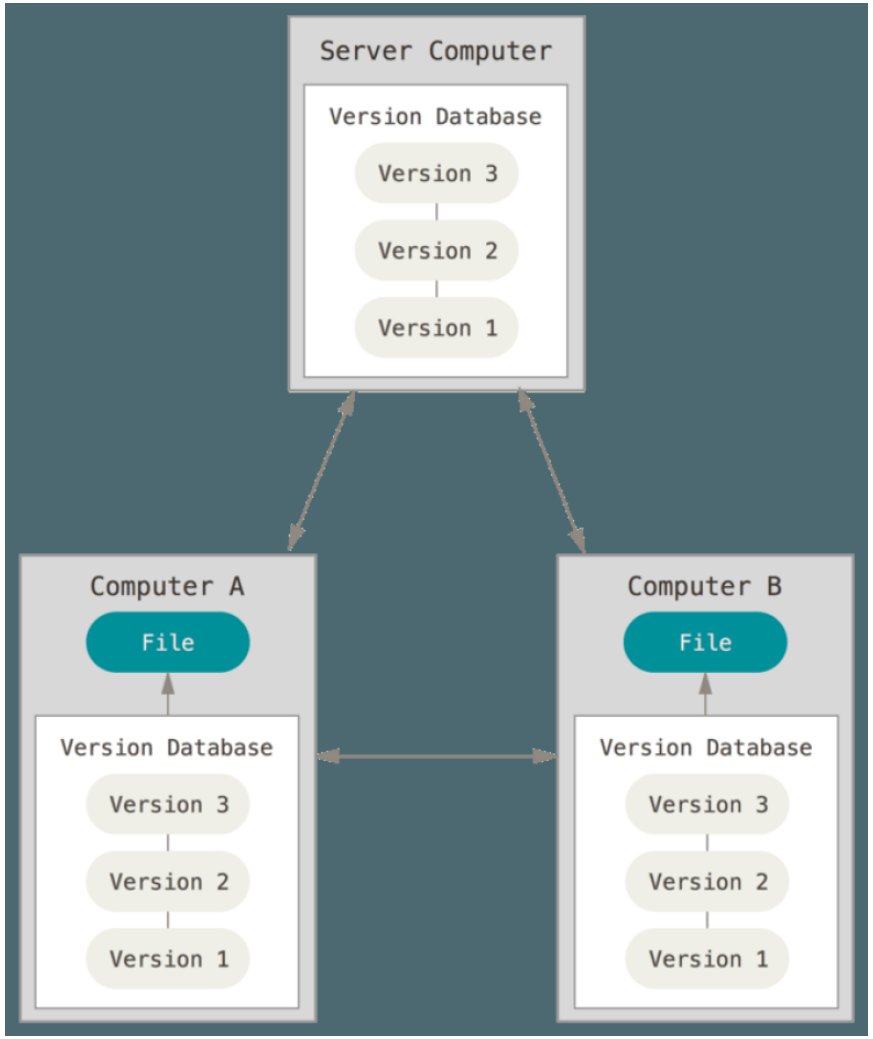


[Scott Chacon and Ben Straub, “Pro Git”]

Version Control (VCS)

Distributed systems

- Multiple versions of the component repository exist at the same time.
- *Git (by Linus Torwalds, 2005)*



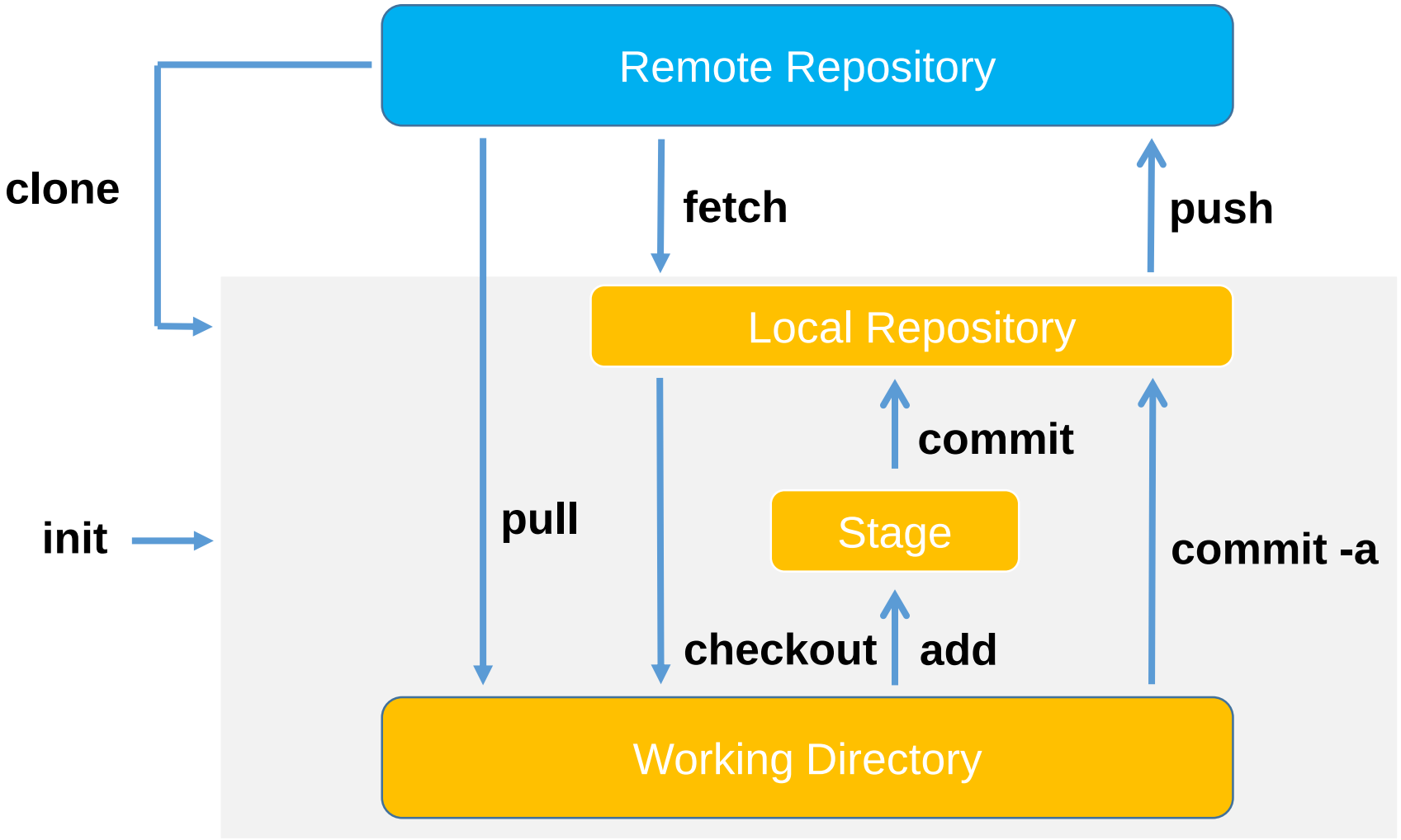
[Scott Chacon and Ben Straub, “Pro Git”]

“**Distributed version control** is essential for open-source development where several people may be working simultaneously on the same system without any central coordination.”

[Ian Sommerville, *“Software Engineering”*]

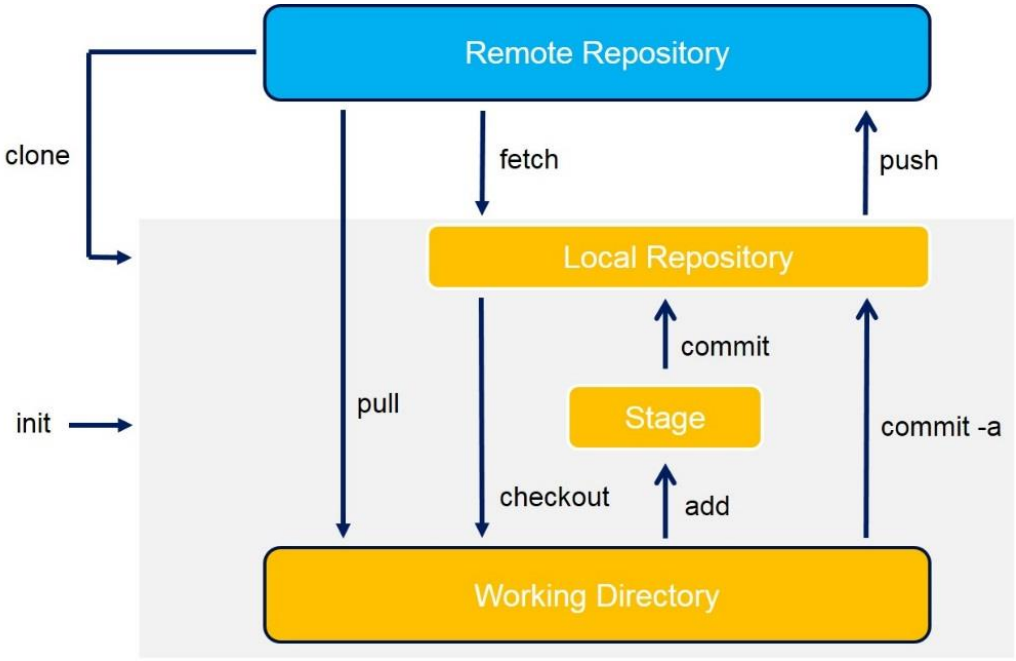
Content

Understanding Git



Understanding Git

Obtain a repository



- **git init**

  Create an empty Git repository or reinitialize an existing one.

- **git clone <repository>**

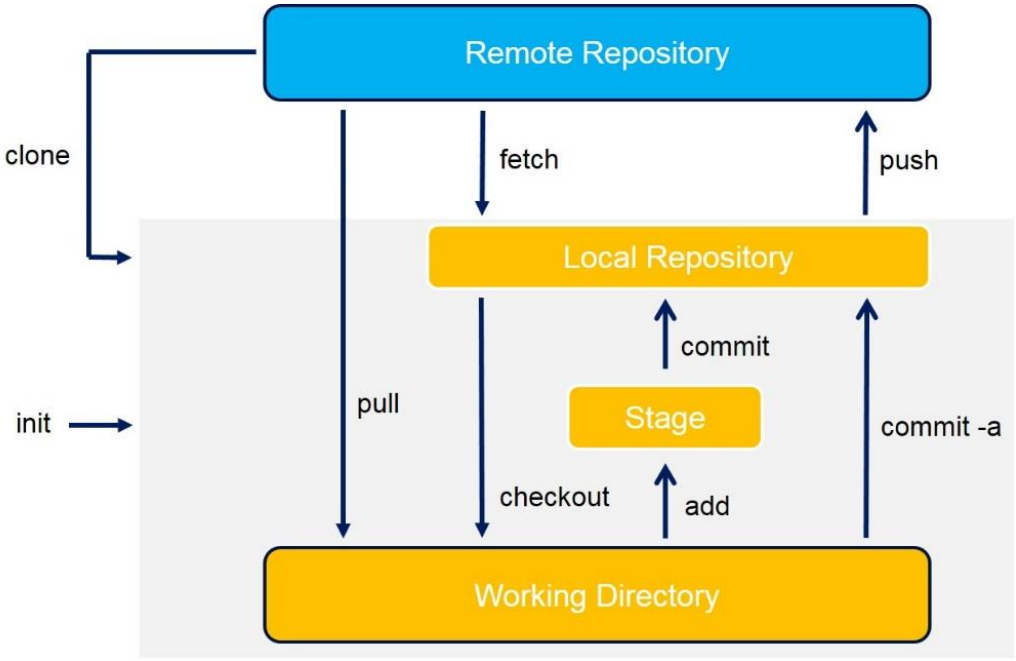
  Clone a repository into a new directory.

Example (SSH URL):

git clone git@github.com/MyGitHubUserName/SoftwareDevInScience.git

Understanding Git

Get changes from a remote repository



- **git fetch**

  Download objects and refs from another repository.

- **git pull <repository>**

  Fetch from and integrate with another repository or a local branch.

  (shorthand for ***git fetch*** followed by ***git merge FETCH_HEAD***)

Understanding Git

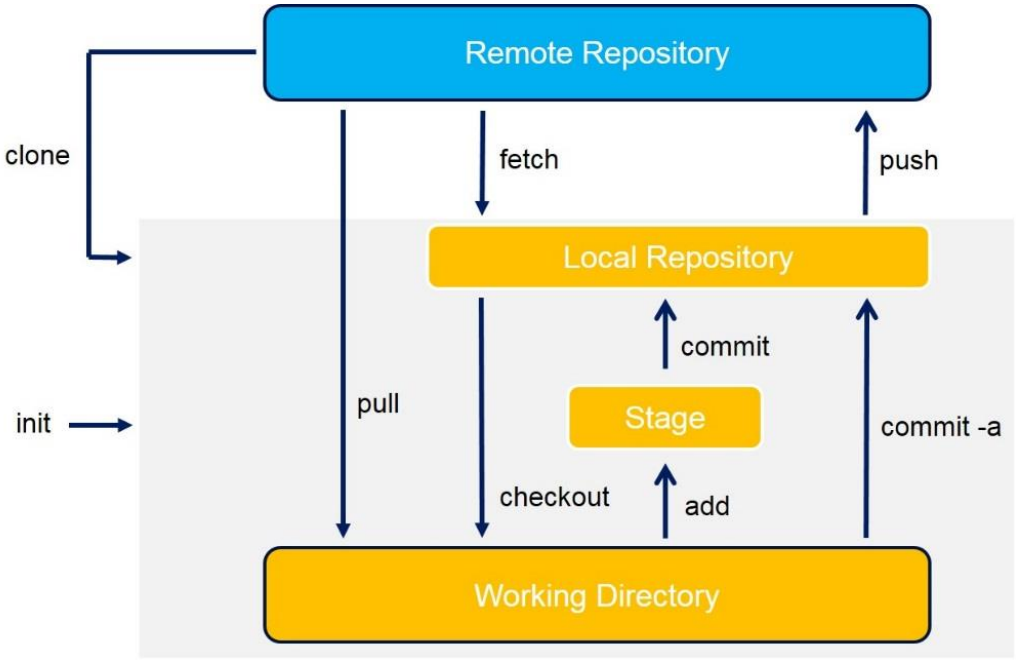
Apply changes to a remote repository

- **git add <file_name(s)>**

  Add file contents to the index (stage).
- **git commit -m <message>**

  Record changes to the repository.
- **git push <repository>**

  Update remote refs along with associated objects.



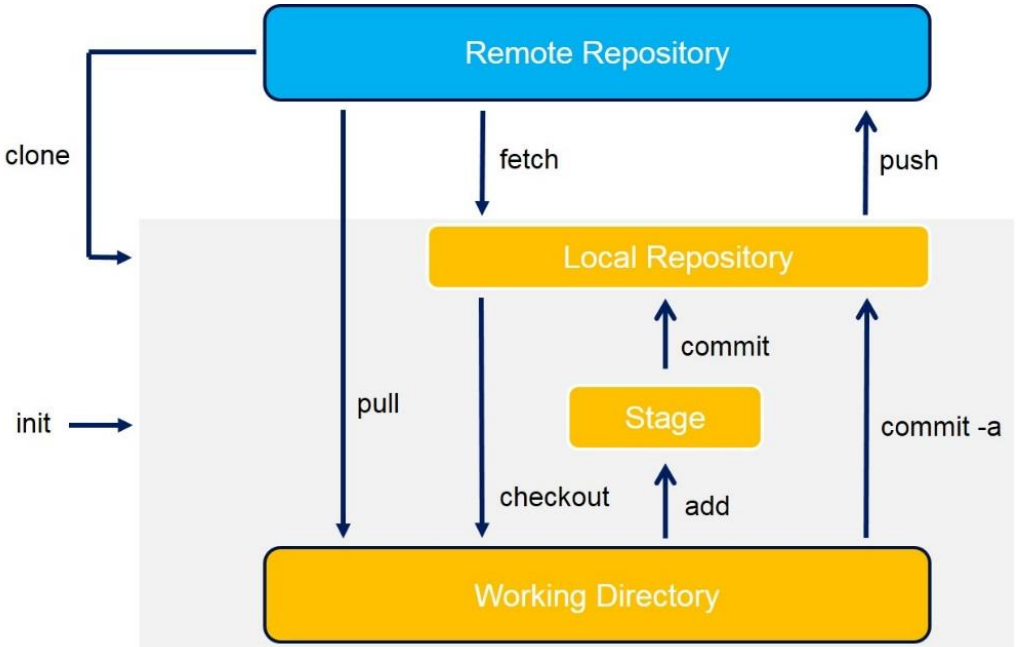
Understanding Git

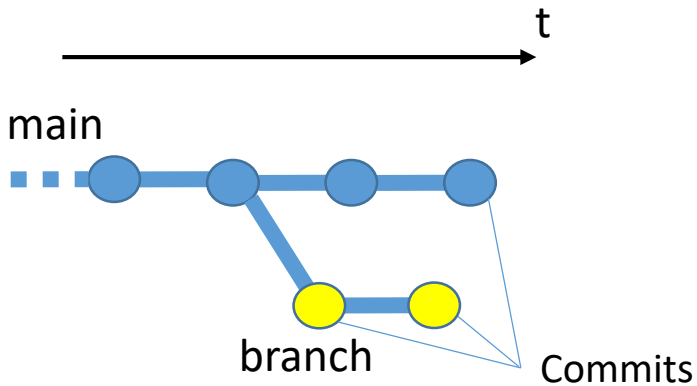
Branching

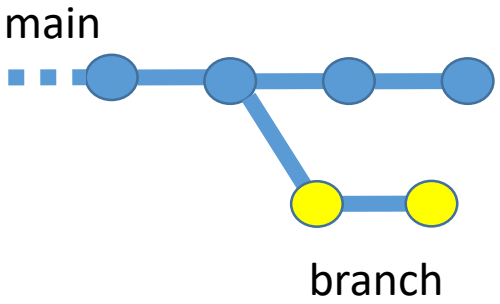
- **git checkout –b <new_branch>**
- **git branch <new_branch>**

  Both commands create a new branch.
- **git branch –d <old_branch>**

  Deletes a branch.







Avoid working on your main-branch !

(Except you know what you do.)

Understanding Git

Other useful Git commands

- **git status**

  Show the working tree status.

- **git reset**

  Reset current HEAD to the specified state.

- **git diff**

  Show changes between commits, commit and working tree, etc.

- **git merge**

  Join two or more development histories together.

- **git remote**

  Manage set of tracked repositories.

Content

GitHub (Agile Workflow)

What is GitHub?

- Web-based Git repository hosting service
- Platform to share open-source projects

www.github.com

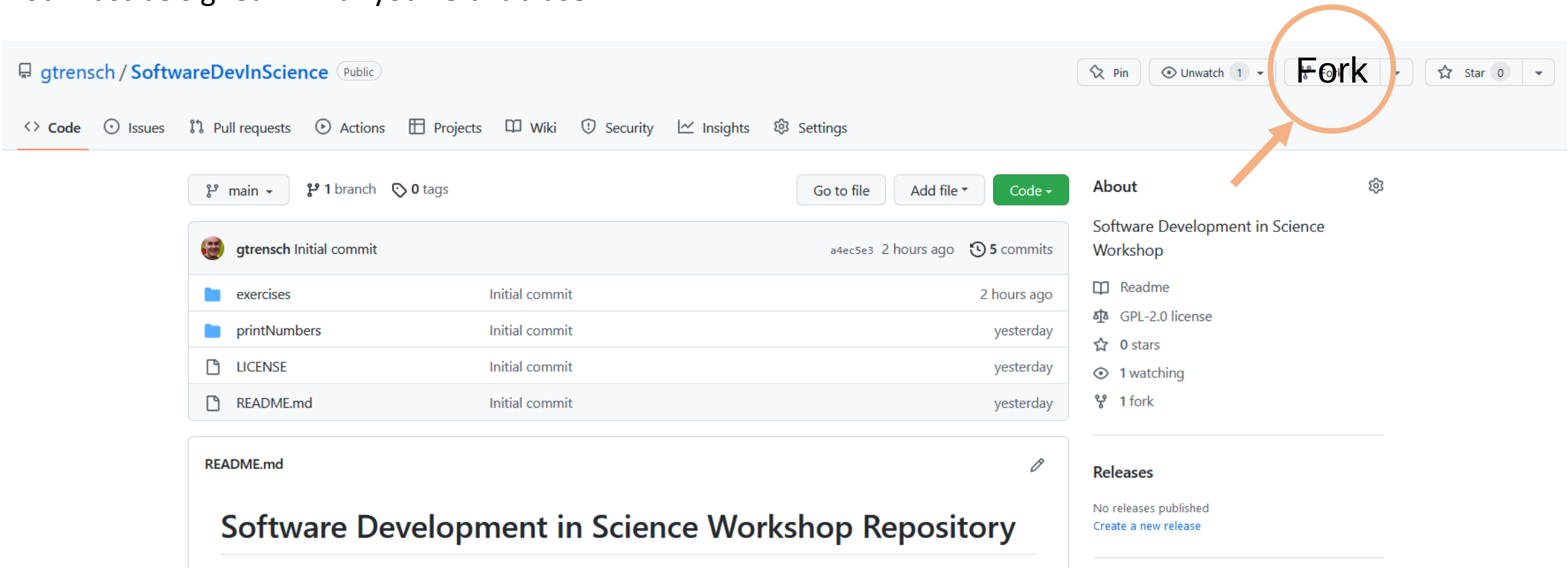
- As of January 2020, GitHub reports having **over 40 million users** and more than 190 million repositories!
- Supports agile practices:
  - Code review workflow
  - Continuous Integration and Delivery (CI/CD)
    - GitHub Actions
  - Lightweight project management

GitHub (Agile Workflow)

- Issue tracker
- Wiki functionality
- GitHub ***Fork***
  - Enables you to copy a repository from a user‘s account.
  - You can make changes under your own account and share your work by issuing a so-called ***Pull Request***.

GitHub (Agile Workflow)

You must be signed in with your GitHub user.

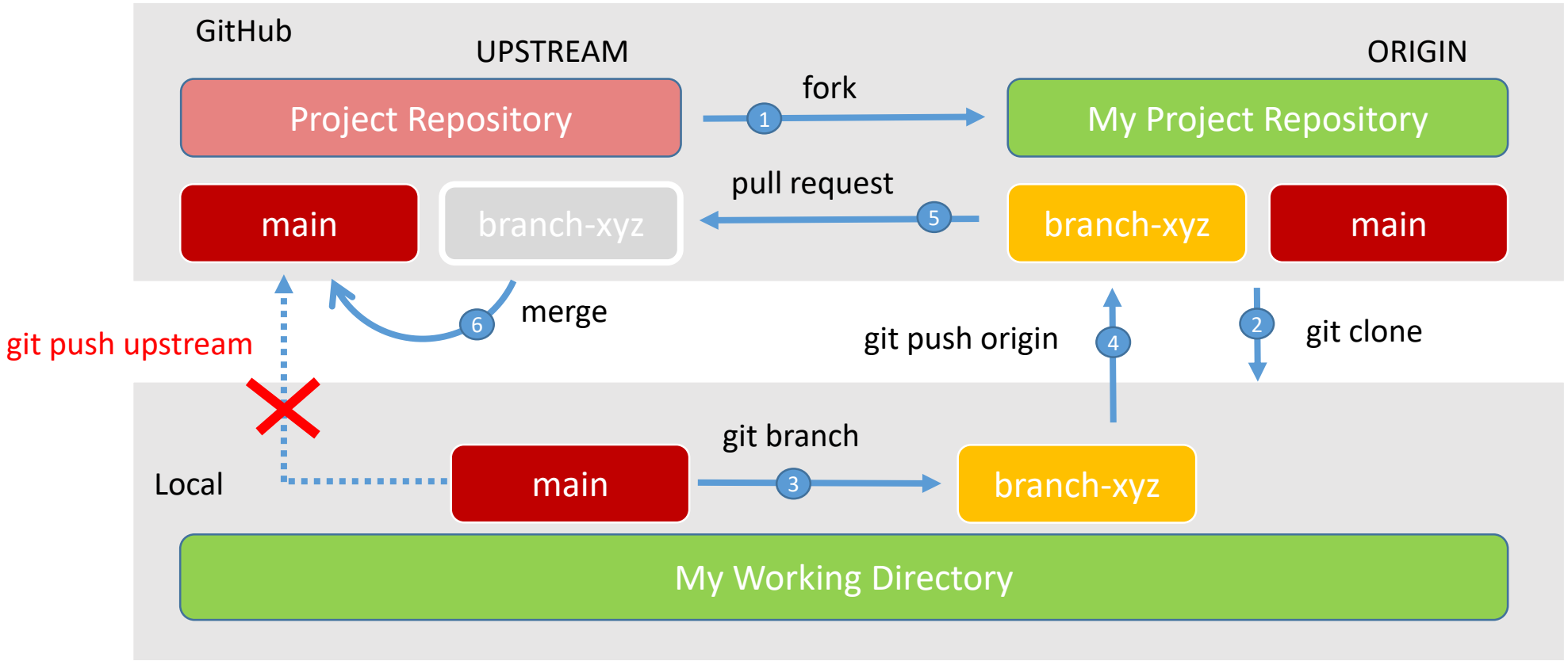


Four common GitHub workflows proven to be useful …

Not rules, rather guidelines and best practices for developers.

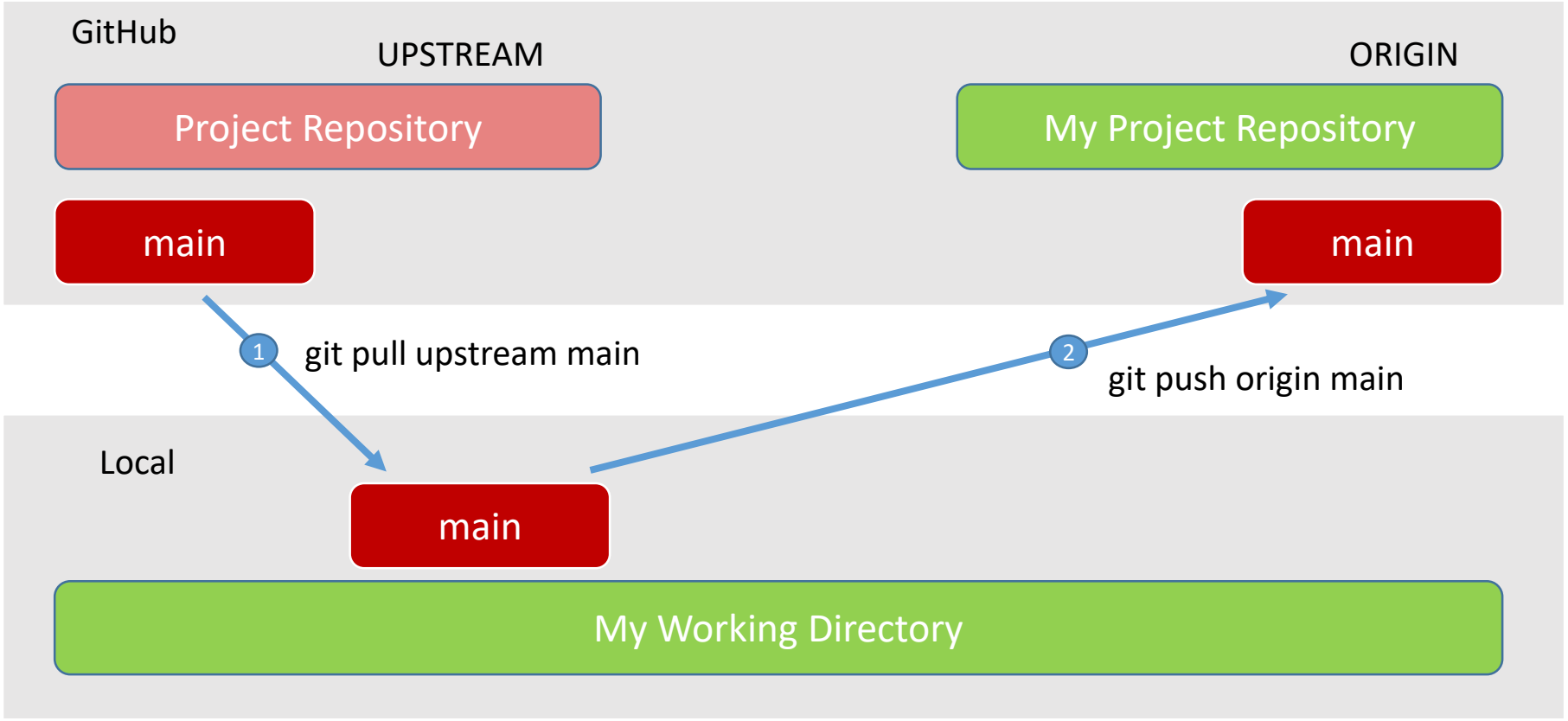
GitHub (Agile Workflow)

Contribute to a project



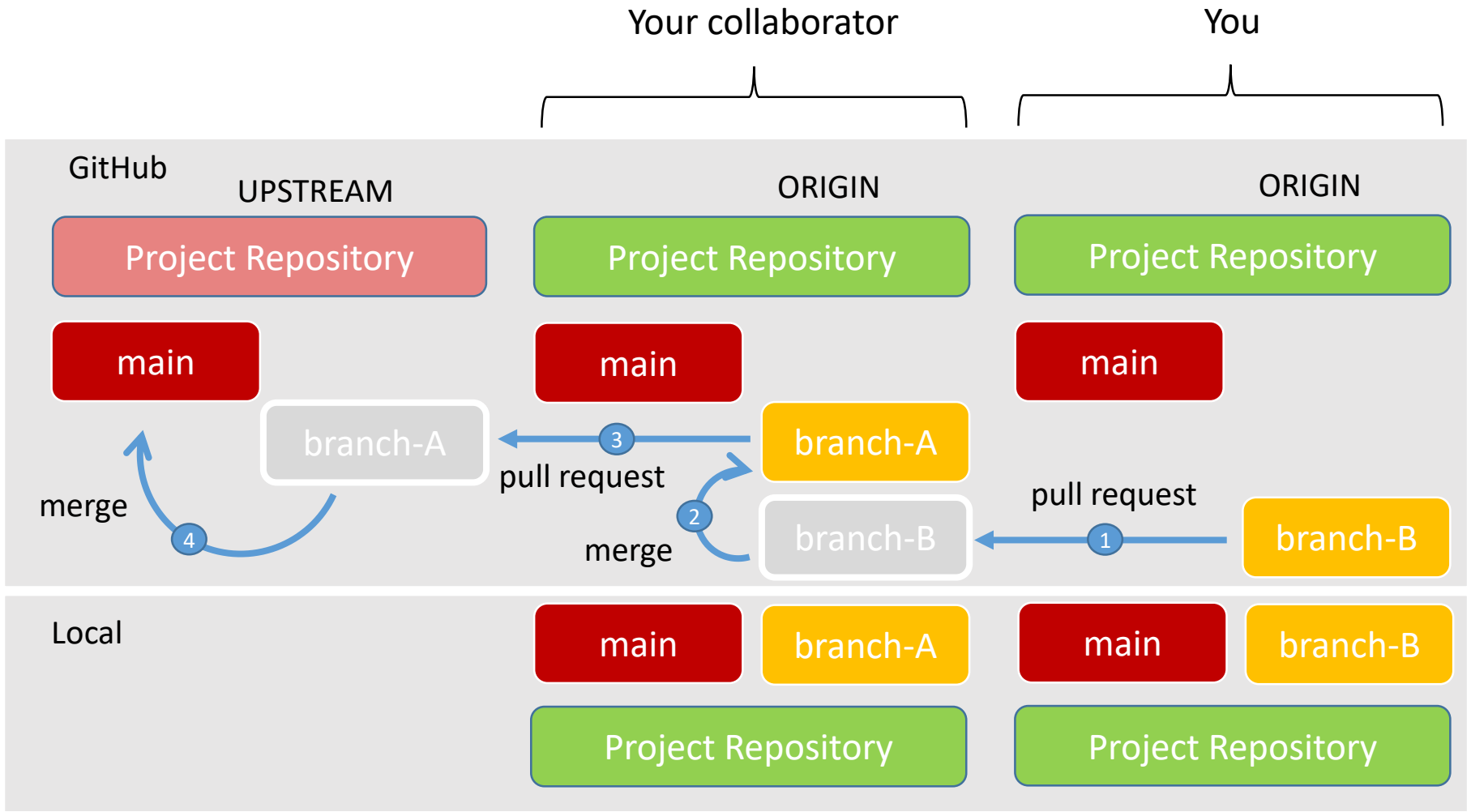
GitHub (Agile Workflow)

Keeping your local repository in sync with GitHub



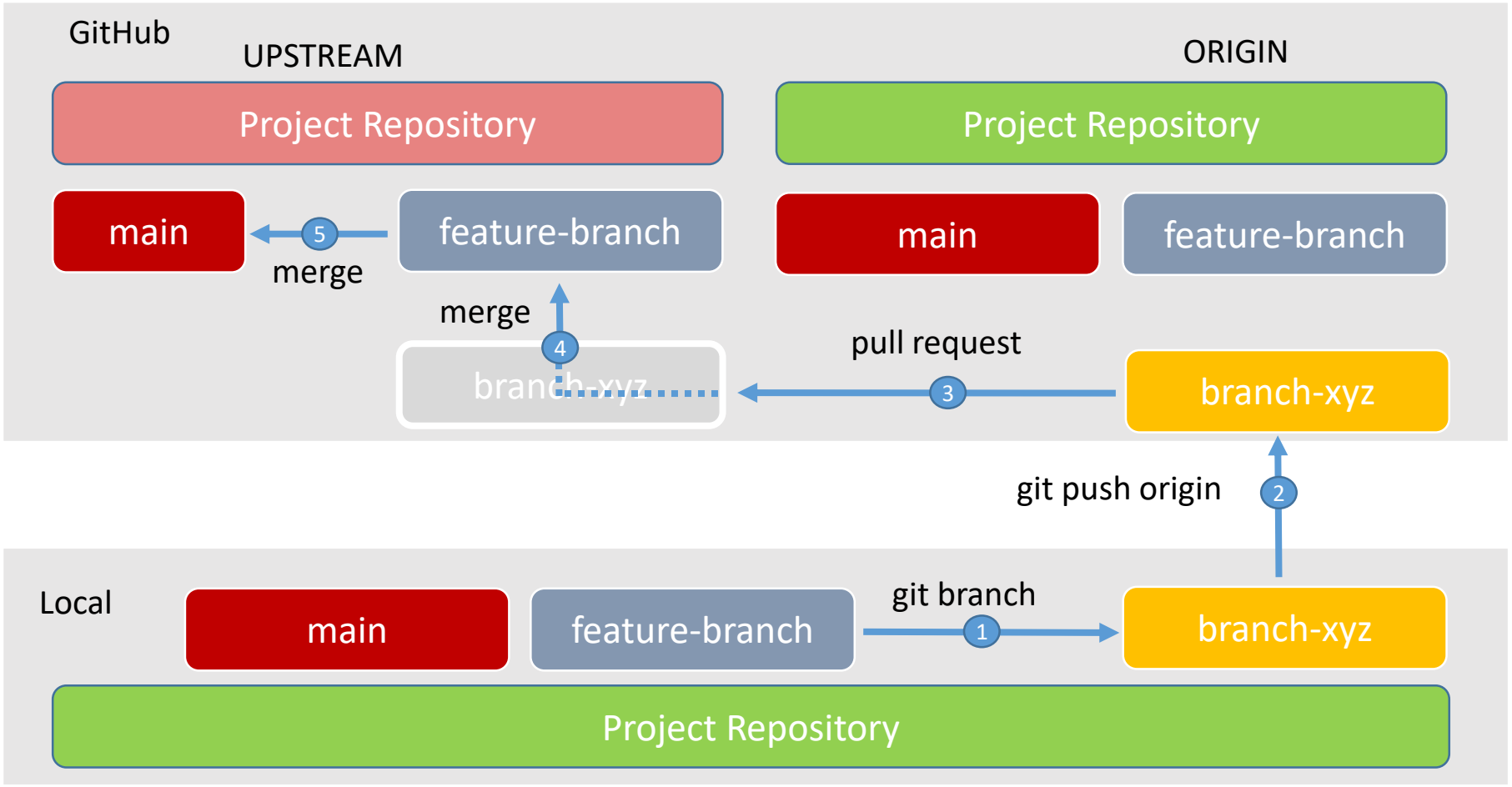
GitHub (Agile Workflow)

Collaborate



GitHub (Agile Workflow)

Feature Branch



Content

References

- Everything you need to know about Git.

  https://git-scm.com/book/en/v2

- Git Reference

  https://git-scm.com/docs

- GitLab

  https://about.gitlab.com/