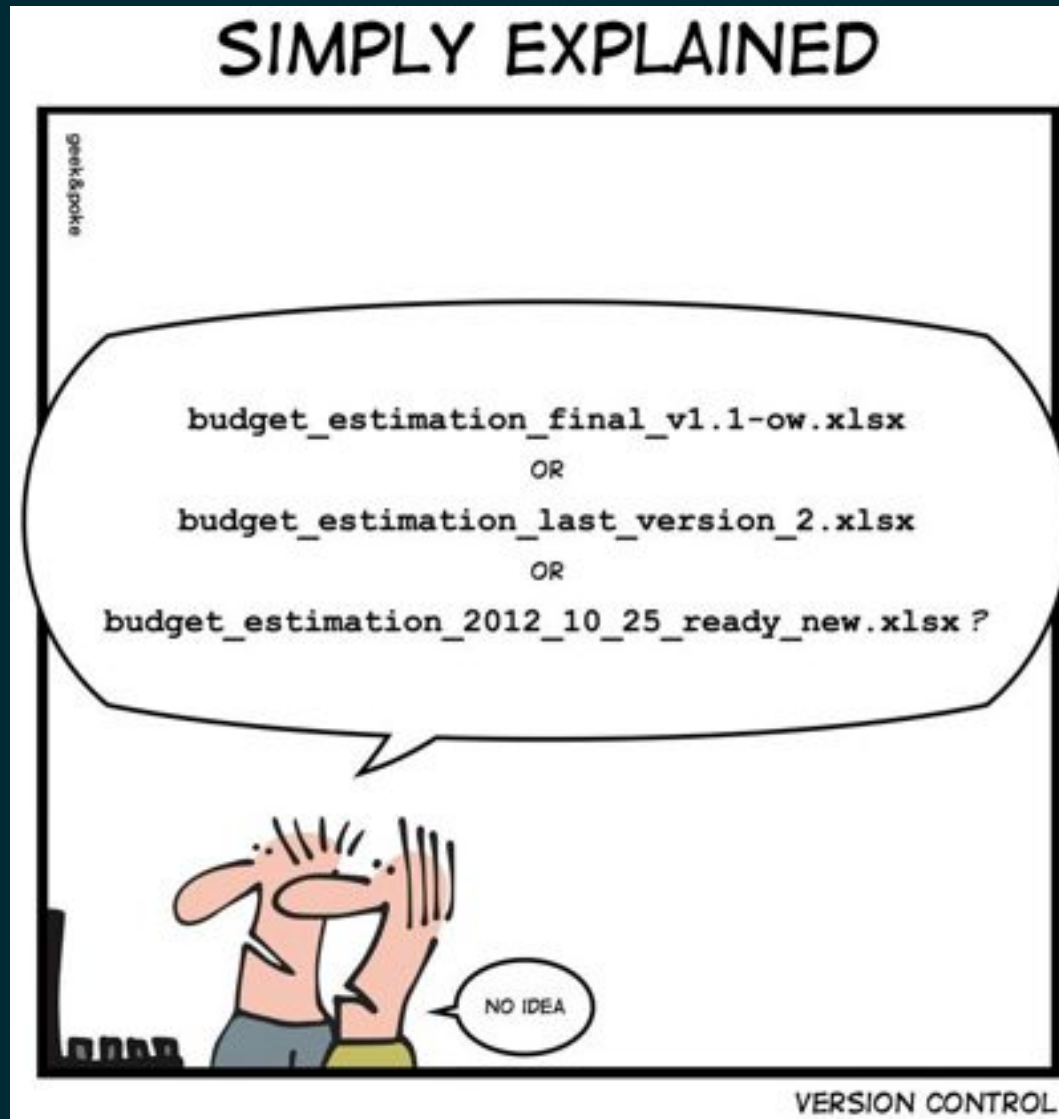










VERSION CONTROL WITH GIT

WHY USE VERSION CONTROL?



WHY USE VERSION CONTROL?

Name	Date modified
 Originals	4/24/2019 2:06 PM
 20190410Backup.xlsx	4/15/2019 12:25 PM
 20190410indVSmex.xlsx	4/24/2019 1:10 PM
 Mex_vs_Ind_Mod1.xlsx	4/24/2019 2:23 PM
 Mex_vs_Ind_Mod2.xlsx	4/24/2019 2:29 PM
 Mex_vs_Ind_Mod3.xlsx	4/24/2019 2:51 PM
 Mex_vs_Ind_Mod4.xlsx	4/25/2019 9:30 AM
 Mex_vs_Ind_Mod5.xlsx	4/25/2019 10:57 AM

e:	Mex_vs_Ind_Mod1.xlsx
e:	Excel Workbook (*.xlsx)

WHY USE VERSION CONTROL?



- Keep track of changes
- Revert to previous versions if needed
- Transparency
- Easier collaboration
- Increased code quality
- Allow experimentation

GIT

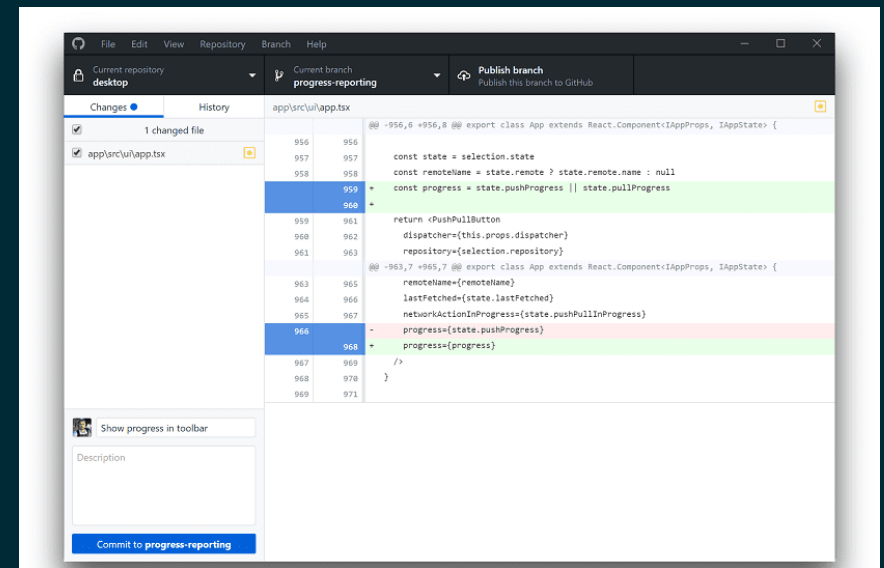
Git is a powerful tool for managing code changes and collaborating with others on a project.

Use Git from the *command-line*, or a *graphical user interface*.

```
> git add foo.py
```

```
> git commit -m "Fixed"
```

```
> git push
```

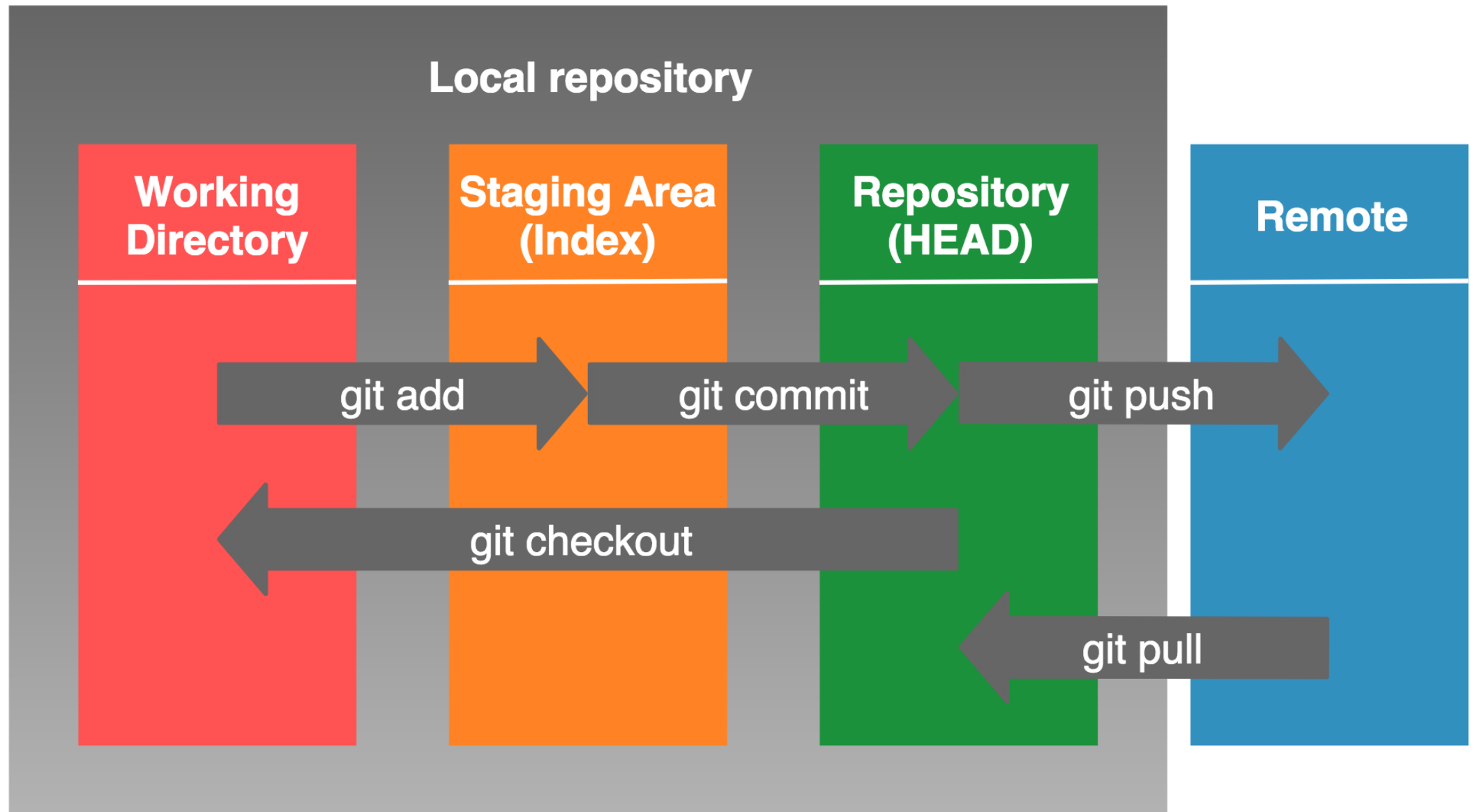


BASIC GIT COMMANDS

- `git add`: adds a file to the staging area
- `git commit`: creates a new commit with the changes in the staging area
- `git stash`: temporarily save changes that are not ready to be committed

WORKING WITH REMOTE REPOSITORIES

- `git clone`: creates a copy of the codebase on your local machine.
- `git push`: pushes changes back to the remote repository.
- `git pull`: pulls changes from the remote repository.



GITHUB

- Git repository hosting service
- Collaborate with others on codebase
- **Pull requests** for code review and merging changes
- **Issue tracking** and project management tools
- **GitHub Pages** for hosting websites

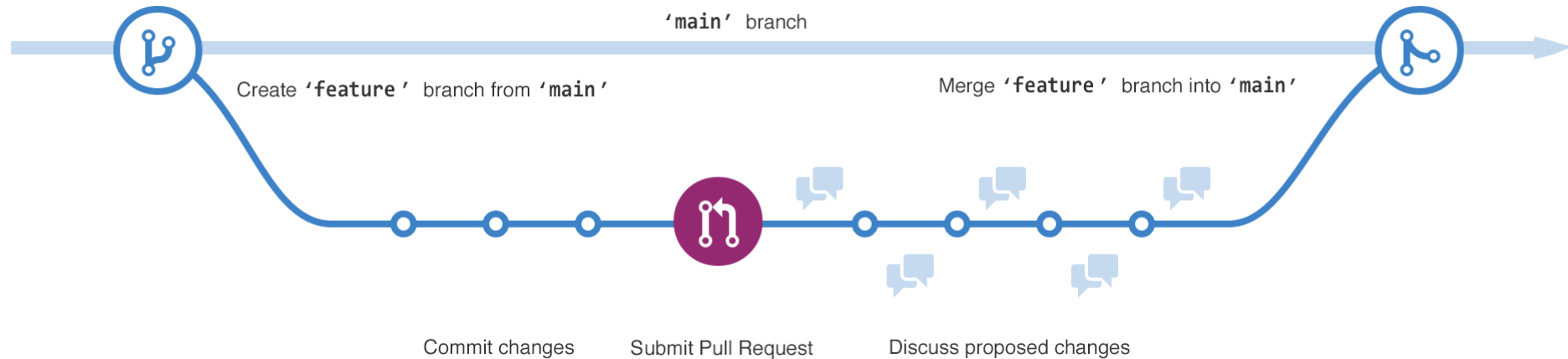
DEMO 1

1. Create a new repository on GitHub
2. Clone the repository to your local machine
3. Add a file
4. Commit the file
5. Push the file to GitHub

BRANCHING AND MERGING

- A branch is a separate version of your code that you can work on independently from the main branch.
- `git merge`: merges changes back into the main branch (we will do this from GitHub)

GITHUB FLOW



1. Create a branch
2. Make changes
3. Create a pull request
4. Review
5. Merge

DEMO 2

1. Create a new branch
2. Make changes
3. Commit the file
4. Create a pull request
5. Get feedback and merge changes

GITHUB BEST PRACTICES

- Commit often
- Use descriptive commit messages
- Keep pull requests small and focused
- Use “issues” to track work
- Review code regularly

GETTINGS STARTED

1. Create a [GitHub](#) account
2. Install [GitHub Desktop](#) (includes Git)
3. Become part of DHI-organization on GitHub (ask THB)
4. Create a new repository on GitHub

RESOURCES

- [GitHub: quickstart](#)
- [RealPython: git and github intro](#)

WORD LIST

Clone

making a local copy of a remote repository on your computer.

Remote

a reference to a Git repository that is hosted on a remote server, typically on a service like GitHub.

Commit

a record of changes made to a repository, including the changes themselves and a message describing what was changed.

Stage

selecting changes that you want to include in the next commit.

Push

sending changes from your local repository to a remote repository.

Pull

retrieving changes from a remote repository and merging them into your local repository.

Branch

a separate line of development that can be used to work on new features or bug fixes without affecting the main codebase.

Pull request

a way to propose changes to a repository by asking the repository owner to “pull” in the changes from a branch or fork.

SUMMARY

- Version control is a tool for managing changes to code
- Git is a version control system (*software*)
- GitHub is a *platform* for hosting and collaborating on Git repositories
- GitHub Desktop is a GUI for Git (and GitHub)
- Pull requests are a way to propose changes to a repository

