Github for Documentation

For E16/17 Unified Project
2020-10-21/2021-06-30

If you have any comments/questions : https://gihan.me/contact nuwanjaliyagoda@eng.pdn.ac.lk

Content

1. VCS
2. Git
3. Github
4. Markdown
5. Github Markdown
6. Github Pages
7. E16 Unified Project Documentation
8. A simple Git workflow for groups
9. Git Command Line (If time permits)

1. Version Control Systems

Need:

Need for backing up code at different places : backups, snapshots, versions.

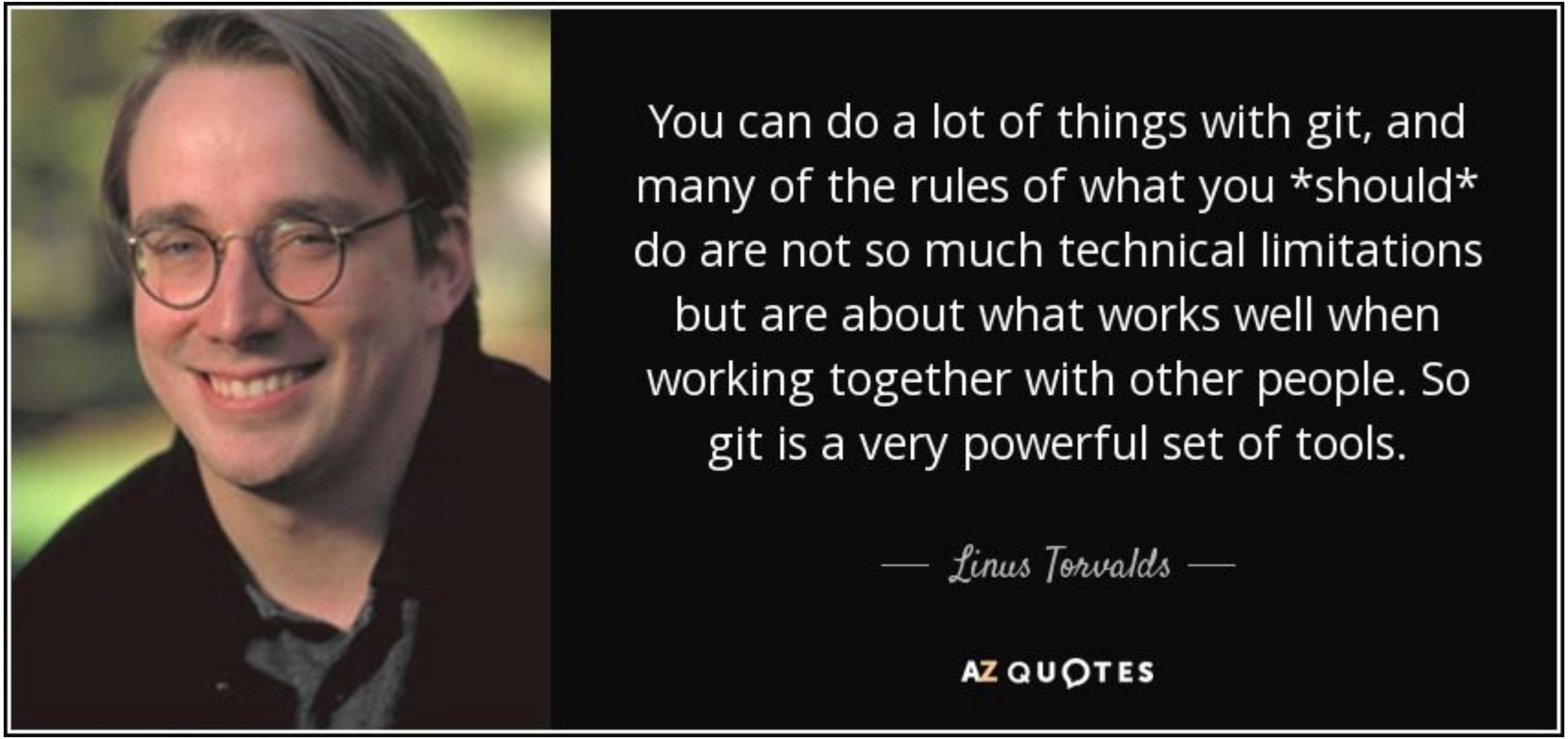
Version Control Systems :

Maintains snapshots of code.

“Branches”. (out of scope for this presentation)

Can backup on different locations.

Git



You can do a lot of things with git, and many of the rules of what you *should* do are not so much technical limitations but are about what works well when working together with other people. So git is a very powerful set of tools.

— Linus Torvalds —

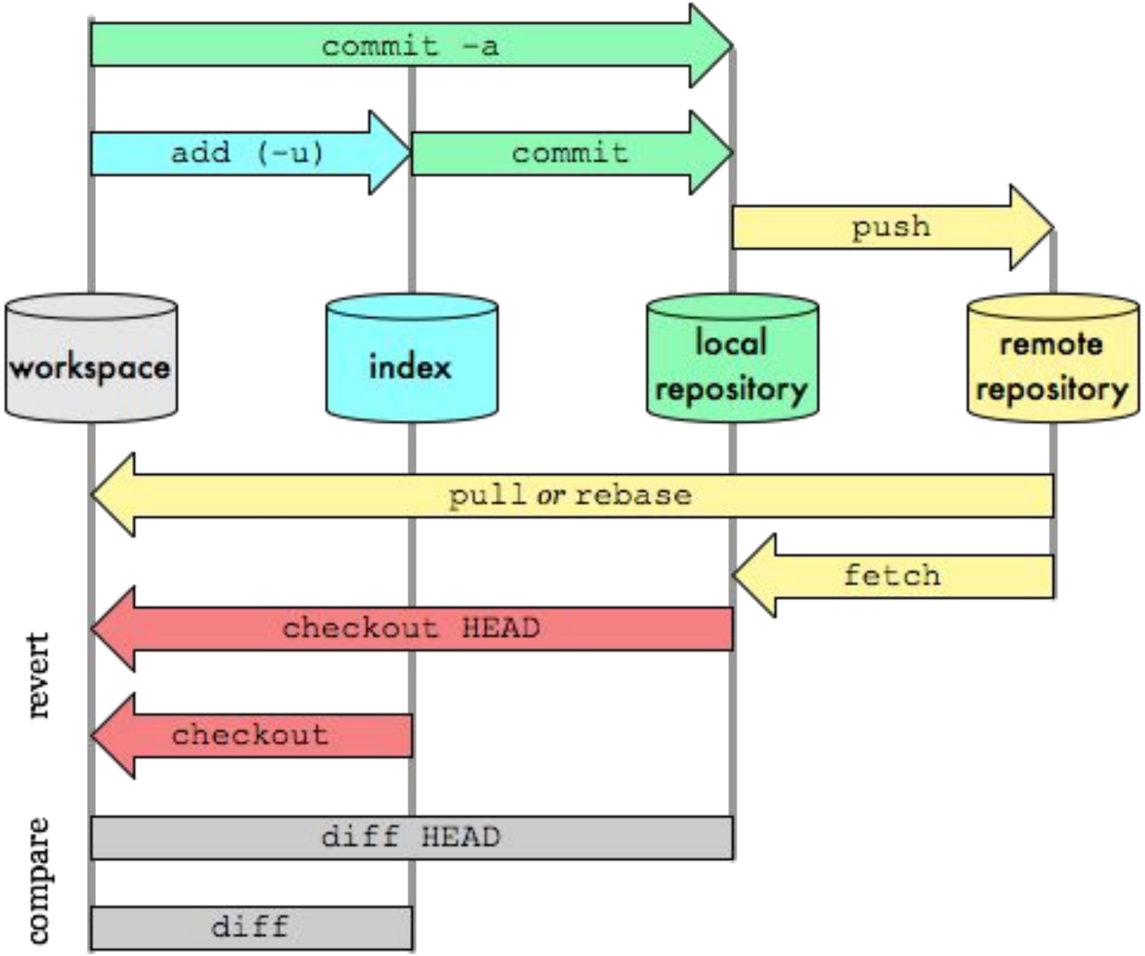
AZ QUOTES

Git

- [Opinion:]
    - Very well designed.
    - Too much features. Steep learning curve.
- Decentralized
- Fast (written in C++)
- Patches : we can send the changes we made to another person by a smaller file (compared to copying the full repository).
    - But this feature is seldom used thanks to Pull Requests
- [Opinion:]
    - First start with a minimum subset of git features. Learn other things on the go.

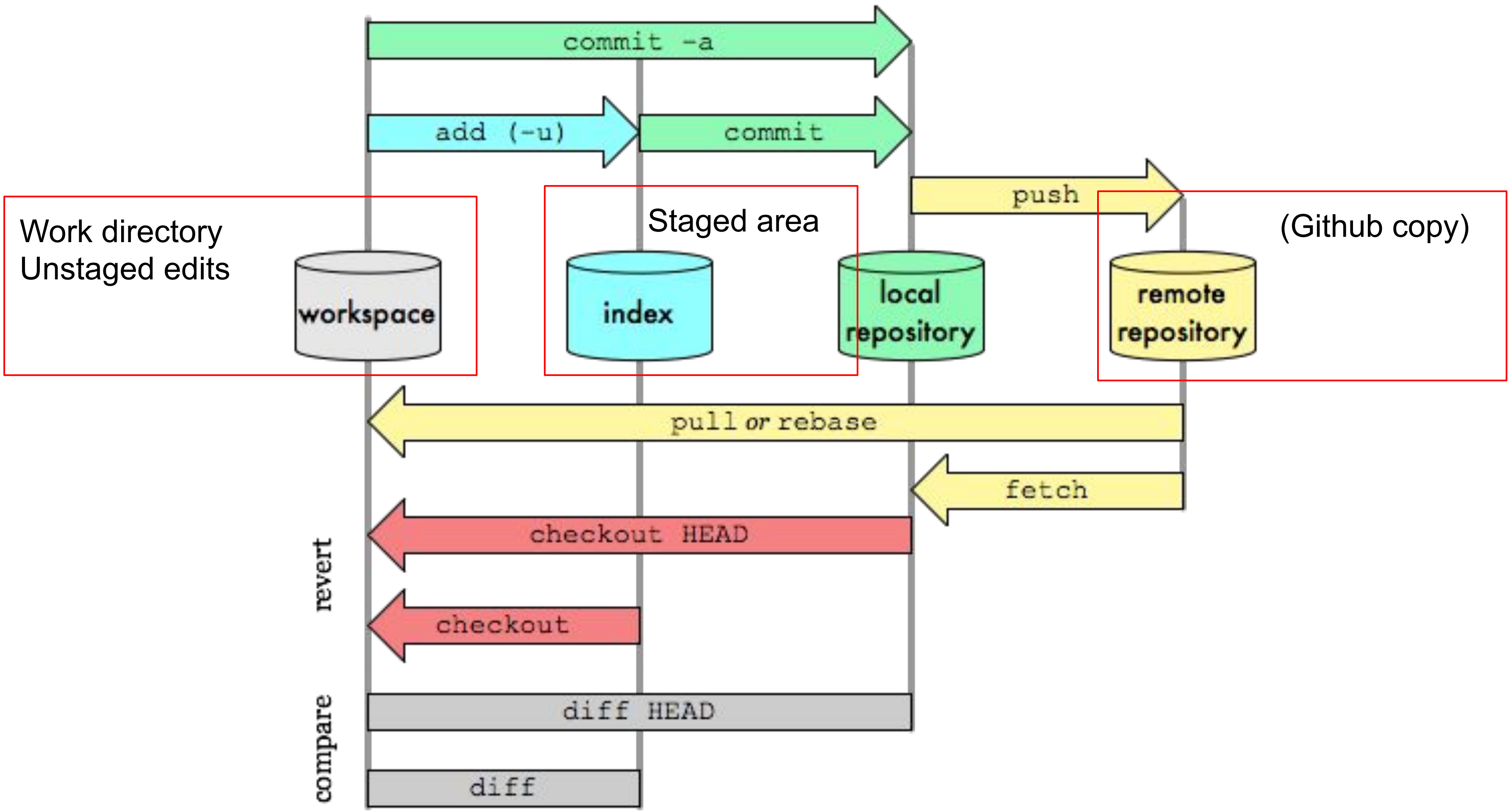
Git Data Transport Commands

http://osteele.com



Git Data Transport Commands

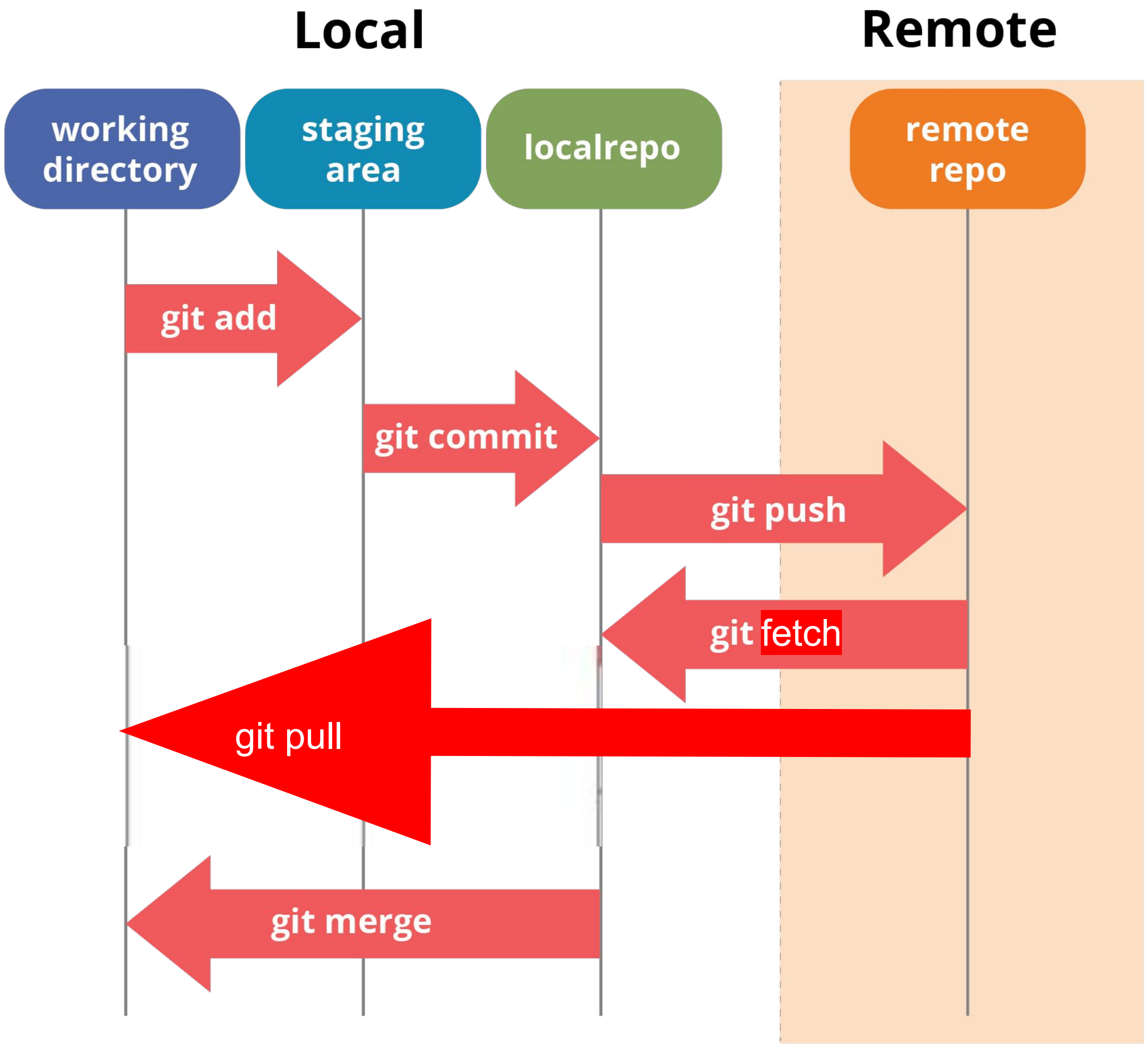
http://osteele.com



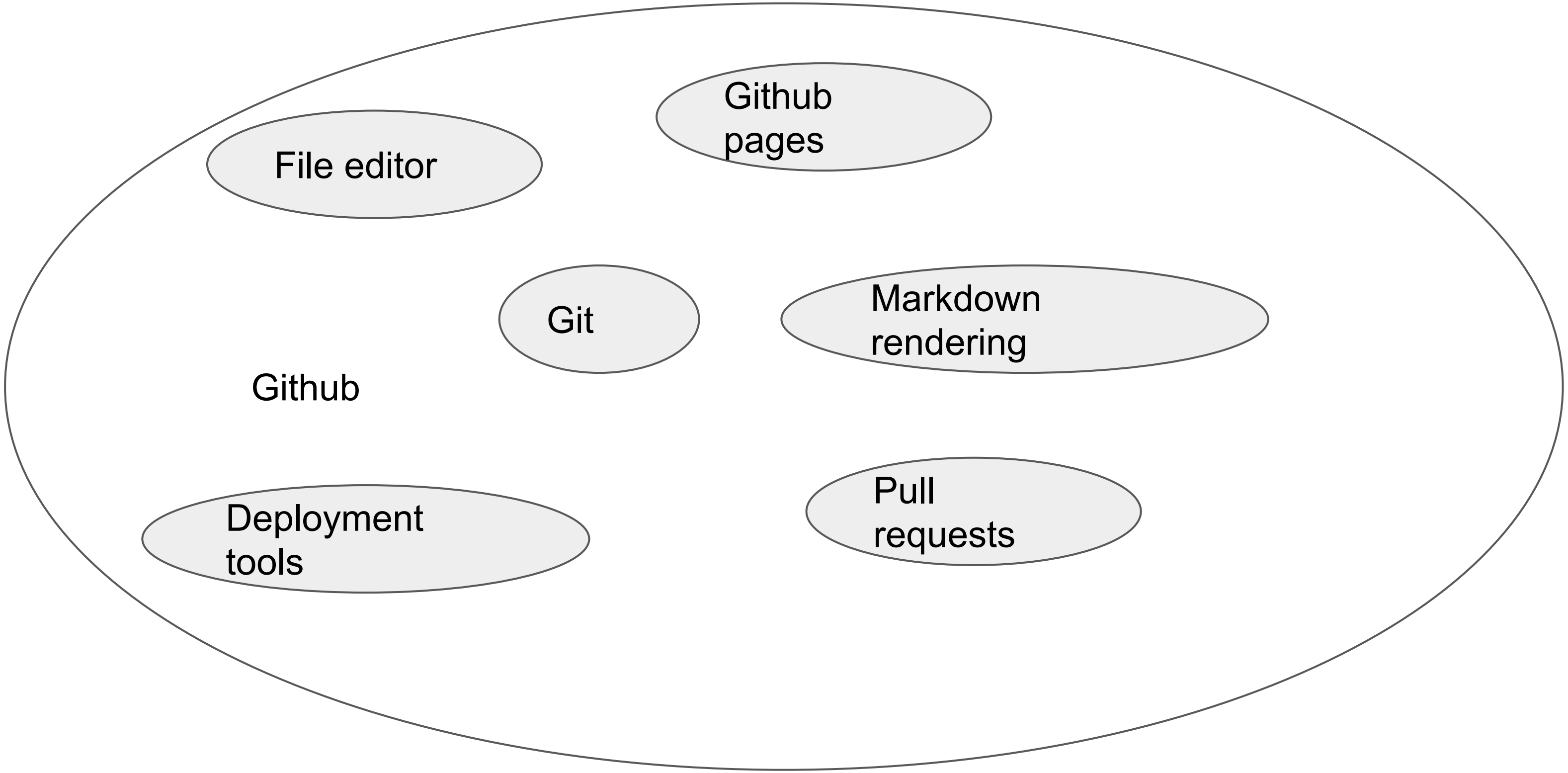
Work directory
Unstaged edits

Staged area

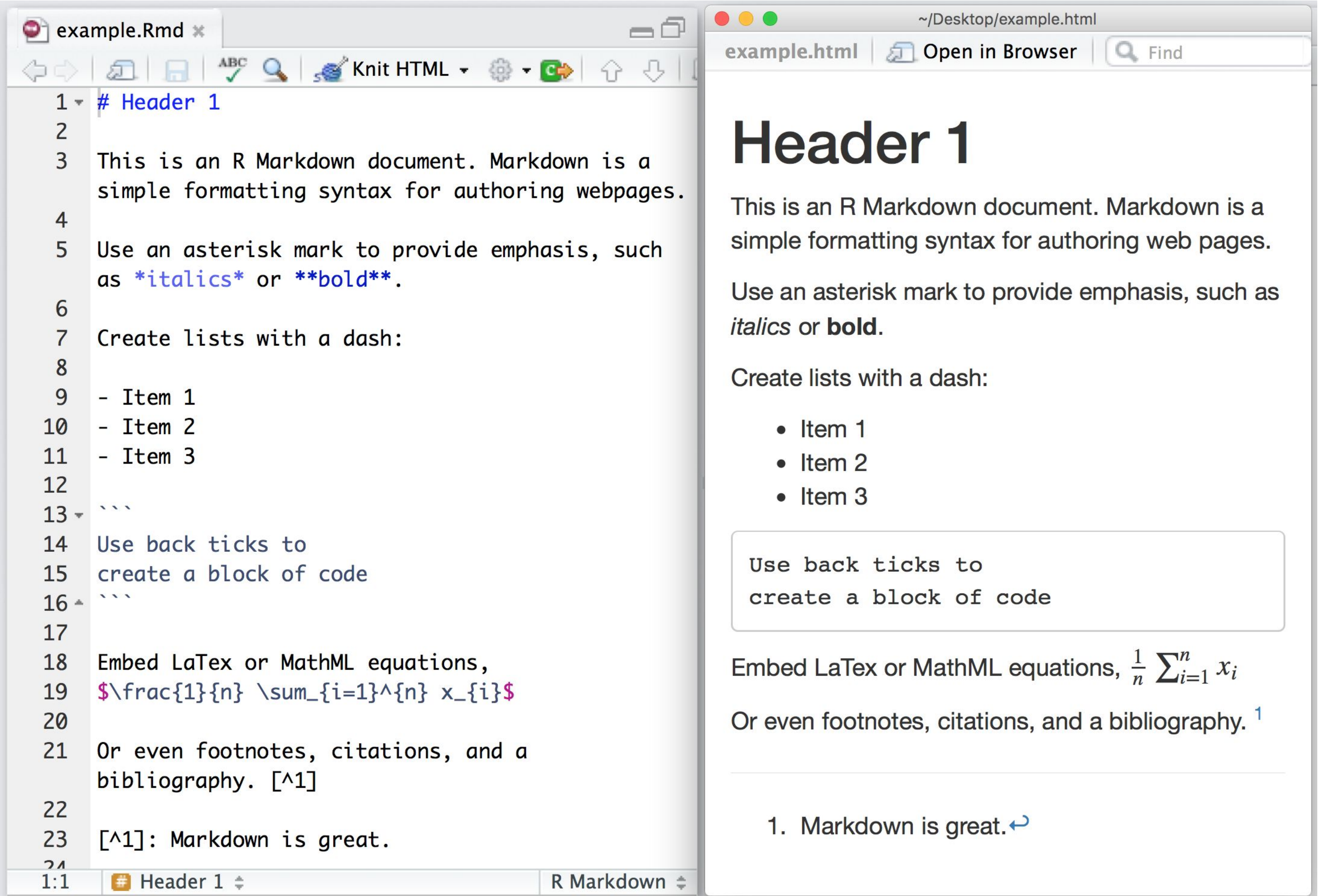
(Github copy)


Local
Remote
working directory
staging area
localrepo
remote repo
git add
git commit
git push
git fetch
git pull
git merge

Github


Github
pages
File editor
Git
Markdown
rendering
Github
Deployment
tools
Pull
requests

Markdown : A document preparation language



```
# Header 1

This is an R Markdown document. Markdown is a
simple formatting syntax for authoring webpages.

Use an asterisk mark to provide emphasis, such
as *italics* or **bold**.

Create lists with a dash:

- Item 1
- Item 2
- Item 3

```
Use back ticks to
create a block of code
```


Embed LaTex or MathML equations,
$\frac{1}{n} \sum_{i=1}^{n} x_{i}$

Or even footnotes, citations, and a
bibliography. [^1]

[^1]: Markdown is great.
```

Header 1

This is an R Markdown document. Markdown is a simple formatting syntax for authoring web pages.

Use an asterisk mark to provide emphasis, such as *italics* or **bold**.

Create lists with a dash:

- Item 1
- Item 2
- Item 3

```
Use back ticks to
create a block of code
```

Embed LaTex or MathML equations, $\frac{1}{n} \sum_{i=1}^{n} x_i$

Or even footnotes, citations, and a bibliography. [1]

[1] Markdown is great.↩

Github Markdown

Any .md file on a github repository will be compiled to the HTML document by a compilation engine inside Github.

This is often used to write README.md files in github.

Demonstration (1) : Writing your first README.md on github website.

Github Pages

This is a github service to have a website https://yourname.github.io for every github user/organization.

Demonstration 2 : Creating the github pages website with HTML.

Individual projects (repositories) can have urls as well https://yourname.github.io/project-name

Demonstration 3 : Creating the github pages for individual repositories.

12 Unified Project Documentation

Organization: https://github.com/cepdnaclk

Naming convention: e16-3yp-project-name

The web page would be https://cepdnaclk.github.io/e16-3yp-project-name

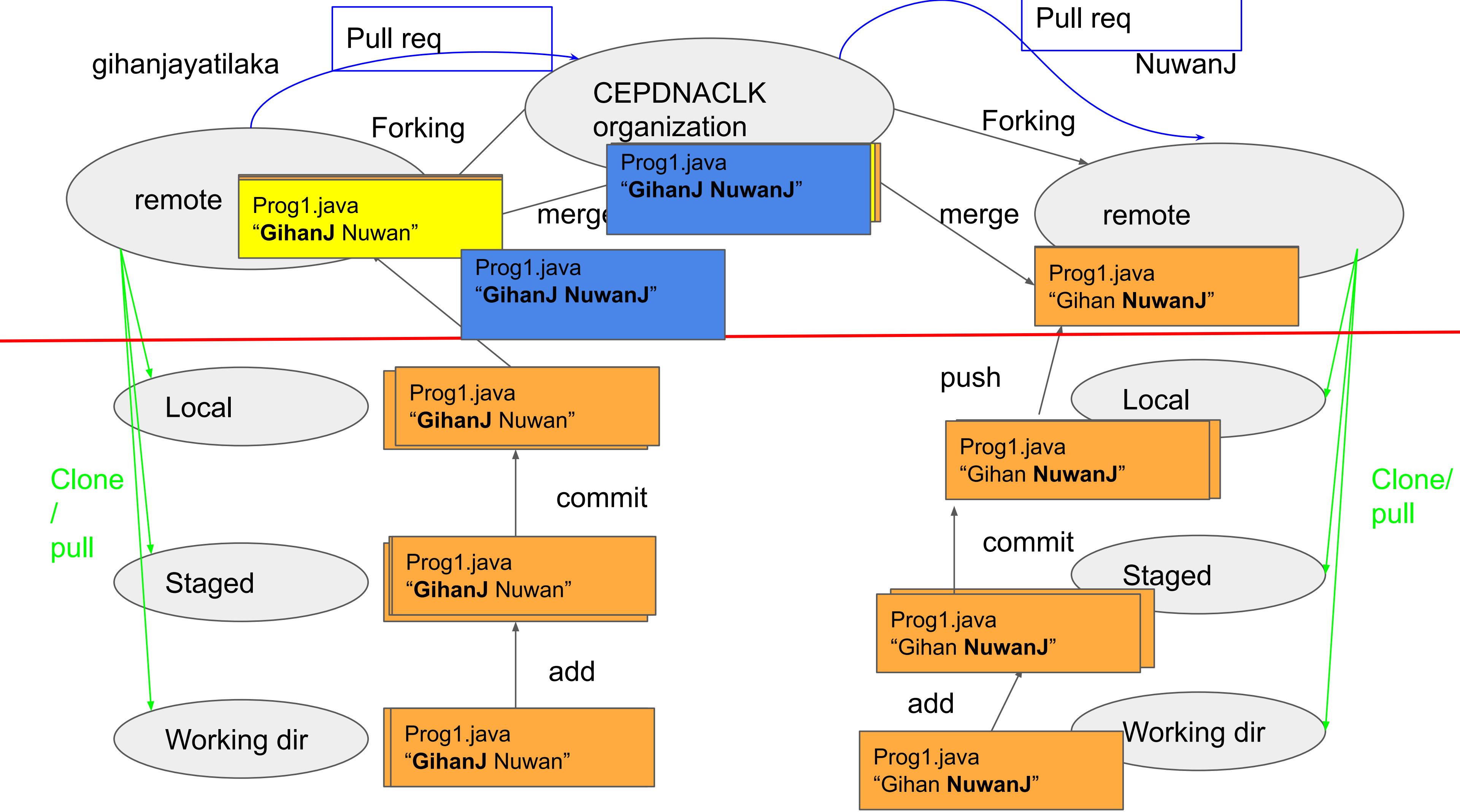
Demonstration 4: How to do documentation for this project.

Requirements

- Title
- Image (photo/drawing of the final hardware)
- Table of Contents (with links to sections lower in the page)
- Introduction (description of the real world problem and solution, impact)
- Solution Architecture (high level diagram + description)
- Hardware & Software Designs (detailed designs with many sub-sections)
- Testing (testing done on hardware and software, detailed + summarized results)
- Detailed budget (all items and costs)
- Conclusion (what was achieved, future developments, commercialization plans)

A simple git workflow for groups

1. Always create the repo in github.com organization/~~group leader’s account~~ (don’t use init)
2. Fork it to your github.com account
3. Clone the repo.
4. Work on your PC, add and commit.
5. Push to your github repo.
6. Then look for pull requests.
7. Merge pull requests in the Github.com GUI (including resolving conflicts)
8. Pull to your local.



gihanjayatilaka

Pull req

CEPDNACLK organization

Pull req

NuwanJ

Forking

Forking

remote

Prog1.java
“**GihanJ** Nuwan”

merge

Prog1.java
“**GihanJ NuwanJ**”

merge

remote

Prog1.java
“**GihanJ NuwanJ**”

Prog1.java
“Gihan **NuwanJ**”

Local

Prog1.java
“**GihanJ** Nuwan”

push

Local

Prog1.java
“Gihan **NuwanJ**”

Clone
/
pull

commit

commit

Clone/
pull

Staged

Prog1.java
“**GihanJ** Nuwan”

Staged

Prog1.java
“Gihan **NuwanJ**”

add

add

Working dir

Prog1.java
“**GihanJ** Nuwan”

Working dir

Prog1.java
“Gihan **NuwanJ**”

Further reading

- Git patches. (the real way of doing pull requests)
- Access tokens instead of passwords.
- Git branches.
- Integrating Git to your IDE / Text editor. (will make your life easy)
- Git cherry picking (going back to a previous version of the code)
- Jekyll Markdown. (create beautiful websites in github pages)
- SVN.
- Github pages and custom domains.
- Arctic code vault (How github is going to save our code for 10000 years?)
- Why master branch became main branch? (history/politics)
- **Github actions**