UNIVERSITY OF THE EAST
COLLEGE OF ENGINEERING
DEPARTMENT OF COMPUTER ENGINEERING

READY. git SET. CODE.

ERROL JOHN M. ANTONIO

Educator. Engineer. Enthusiast

BE STRONG AND COURAGEOUS – Joshua 1:9

MGA RASON NG ESTUDYANTE:

“Na-corrupt po yung files namen”

“Naiwan po yung laptop / flashdrive”

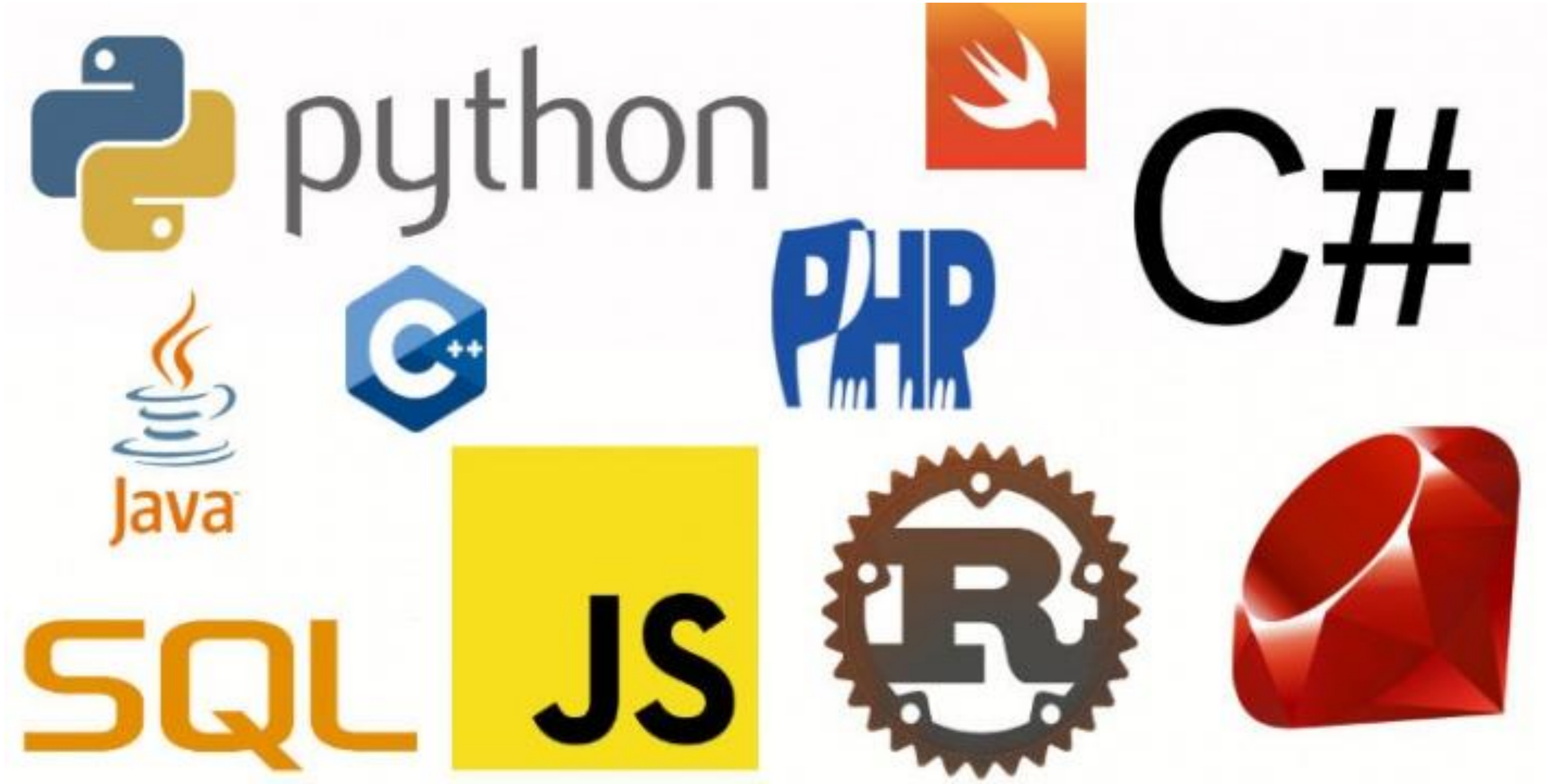
“Hindi po kasi ginawa ni Juan ung kanya”

“Nasira po yung laptop”

“Ako lang po gumawa”

“Moral (financial) support lang po’yung mga ka-grupo ko”

PRE-REQUISITES



UNIVERSITY OF THE EAST
COLLEGE OF ENGINEERING
DEPARTMENT OF COMPUTER ENGINEERING

READY. git SET. CODE.

ERROL JOHN M. ANTONIO

Educator. Engineer. Enthusiast

BE STRONG AND COURAGEOUS – Joshua 1:9

WHAT IS GIT?

GIT

A distributed versioning control system.

Version Control System (VCS)

A software that helps software developers to work together and maintain a complete history of their work.

git

Version Control System (VCS)

All about managing multiple versions of documents, programs, web sites, etc.

WHY VCS?

For working by yourself

Free and open source

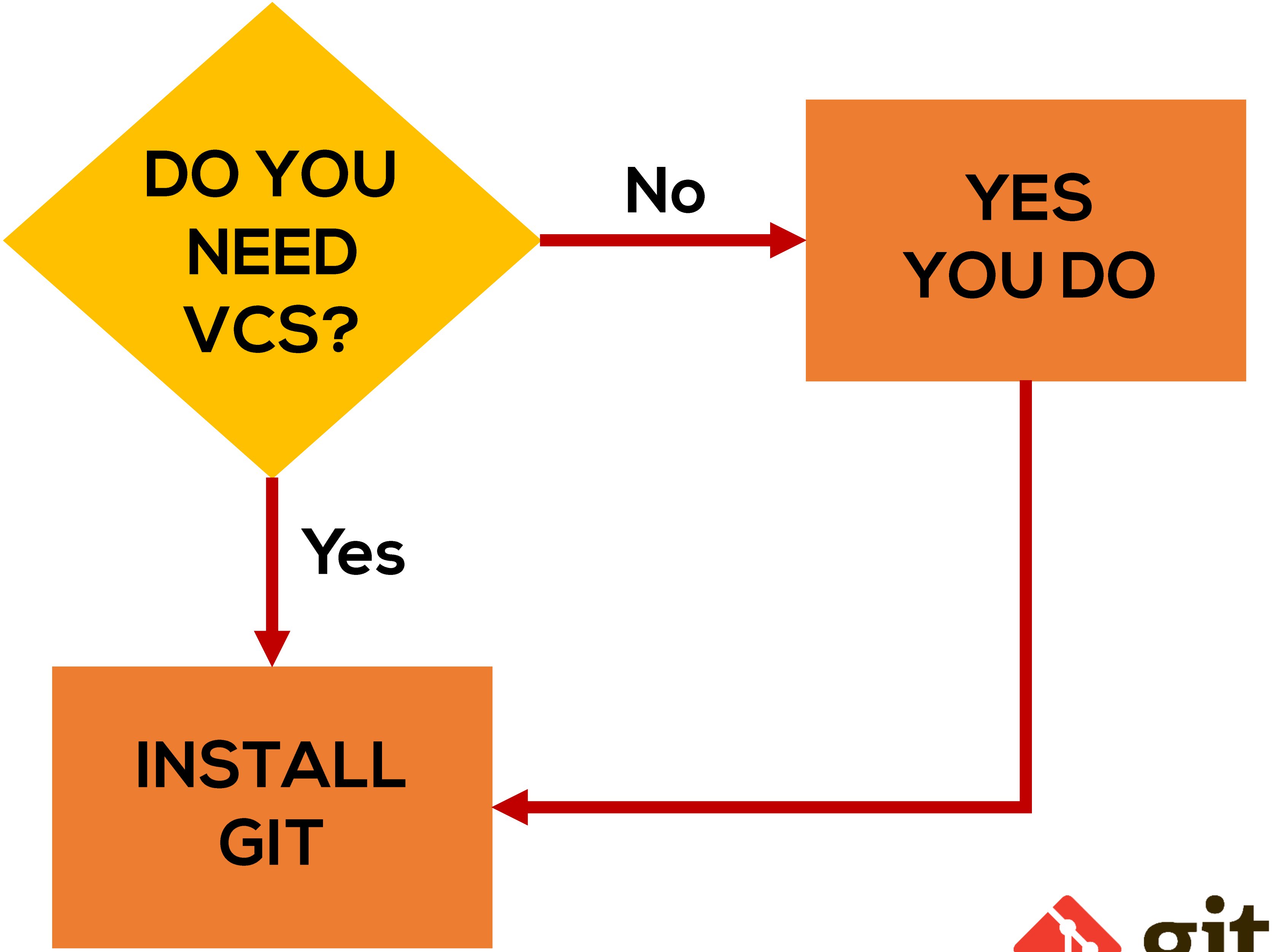
Fast and Small

Implicit Back-Up

Secured

No Need of Powerful Hardware

Easier Branching



DO YOU NEED VCS?

No

YES YOU DO

Yes

INSTALL GIT

UNIVERSITY OF THE EAST
COLLEGE OF ENGINEERING
DEPARTMENT OF COMPUTER ENGINEERING

READY. git SET. CODE.

ERROL JOHN M. ANTONIO

Educator. Engineer. Enthusiast

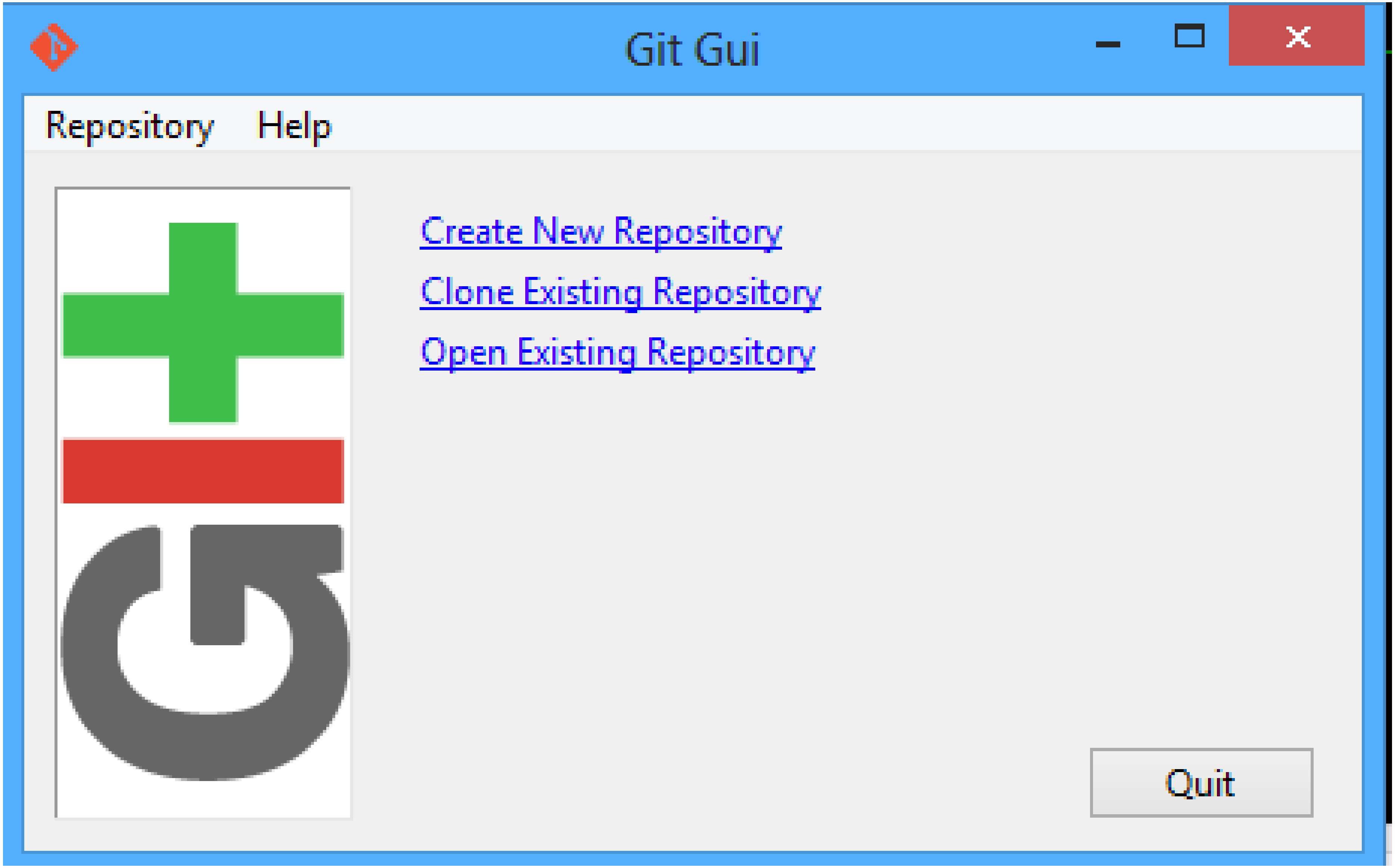
BE STRONG AND COURAGEOUS – Joshua 1:9

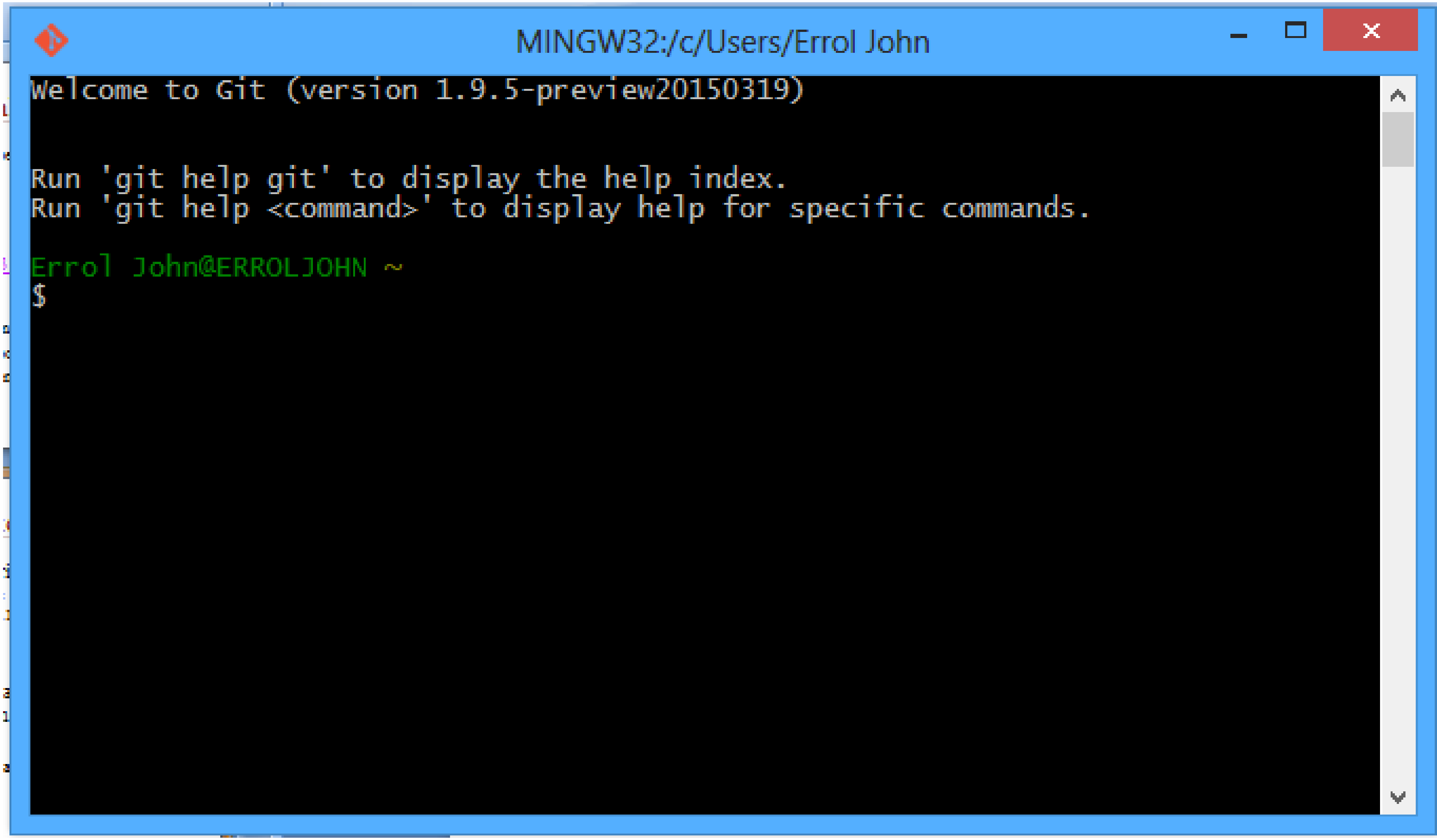
GIT OPERATIONS

DOWNLOAD AND INSTALL GIT

- There are online materials that are better than any that I could provide
- Here’s the standard one: http://git-scm.com/downloads
- Here’s one from StackExchange: http://stackoverflow.com/questions/315911/git-for-beginners-the-definitive-practical-guide#323764

GUI vs CLI

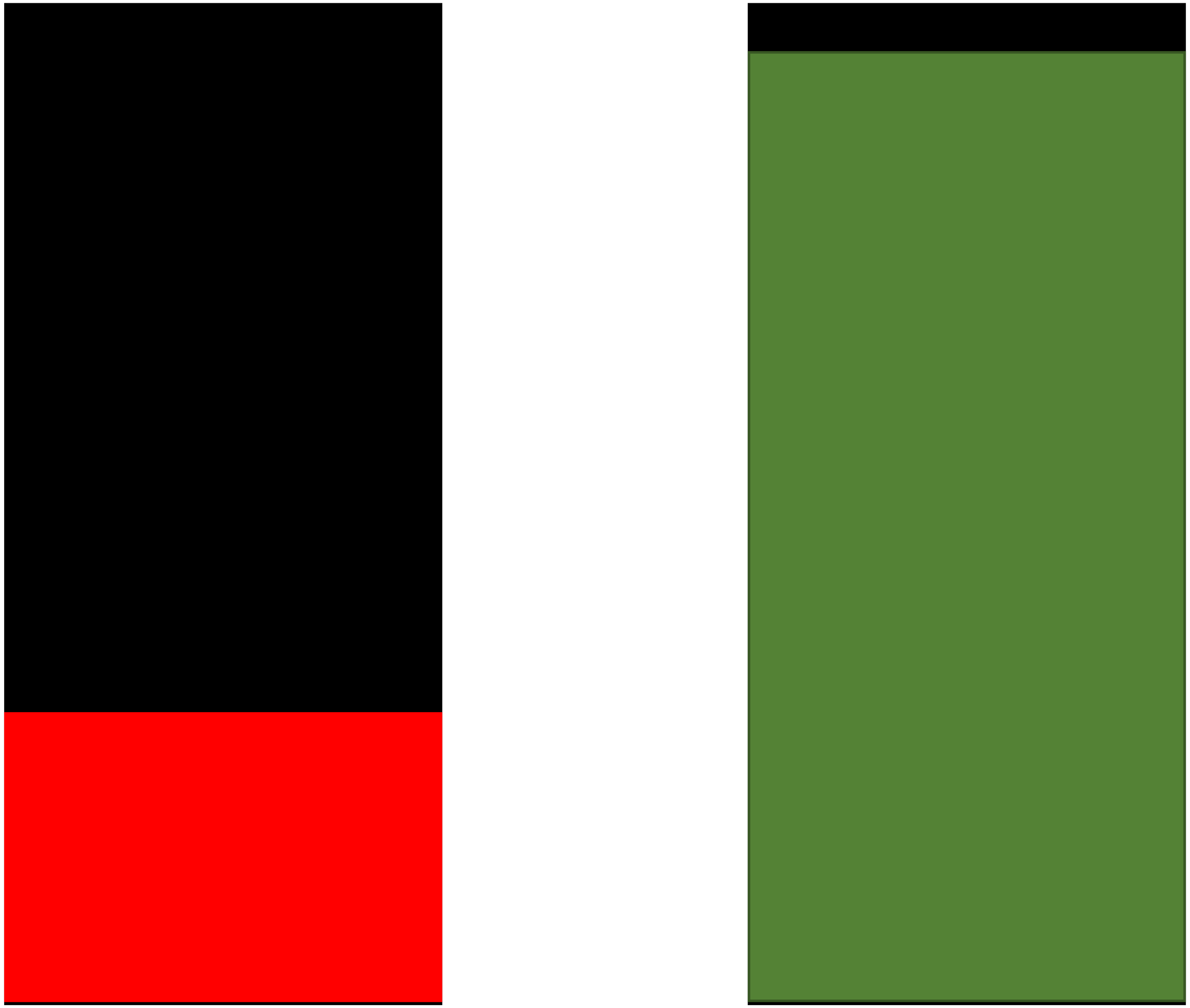

Git Gui
Repository
Help
Create New Repository
Clone Existing Repository
Open Existing Repository
Quit



```
Welcome to Git (version 1.9.5-preview20150319)


Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

Errol John@ERROLJOHN ~
$
```



Power GUI

Power CLI

git

GUI vs CLI

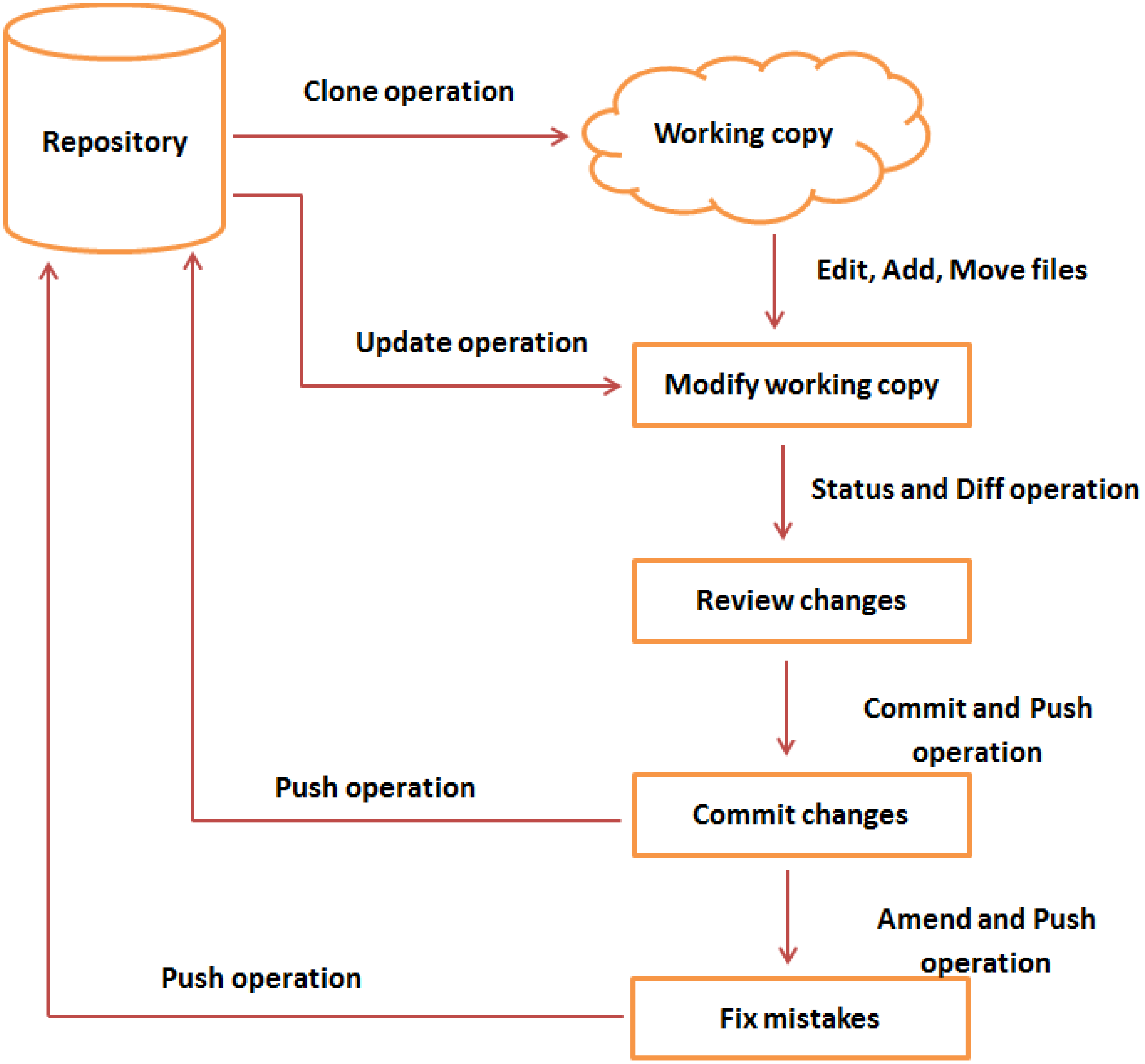
WHO YOU ARE?

Setting username & E-mail

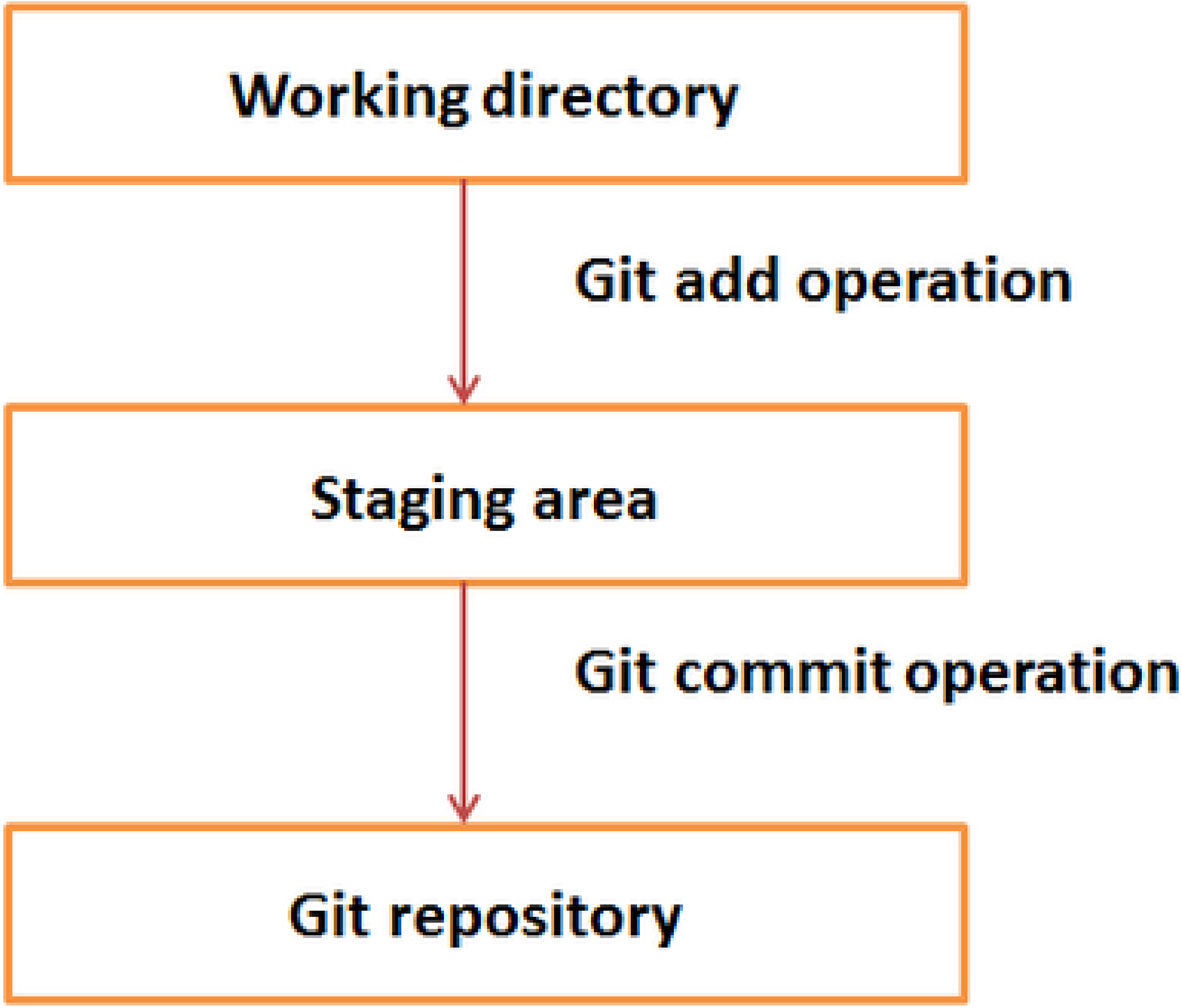
```
git config --global user.name
“NameHere“
```

```
git config --global user.email
“EmailHere“
```

GIT LIFECYCLE


Repository
Clone operation
Working copy
Edit, Add, Move files
Update operation
Modify working copy
Status and Diff operation
Review changes
Commit and Push operation
Push operation
Commit changes
Amend and Push operation
Push operation
Fix mistakes

git



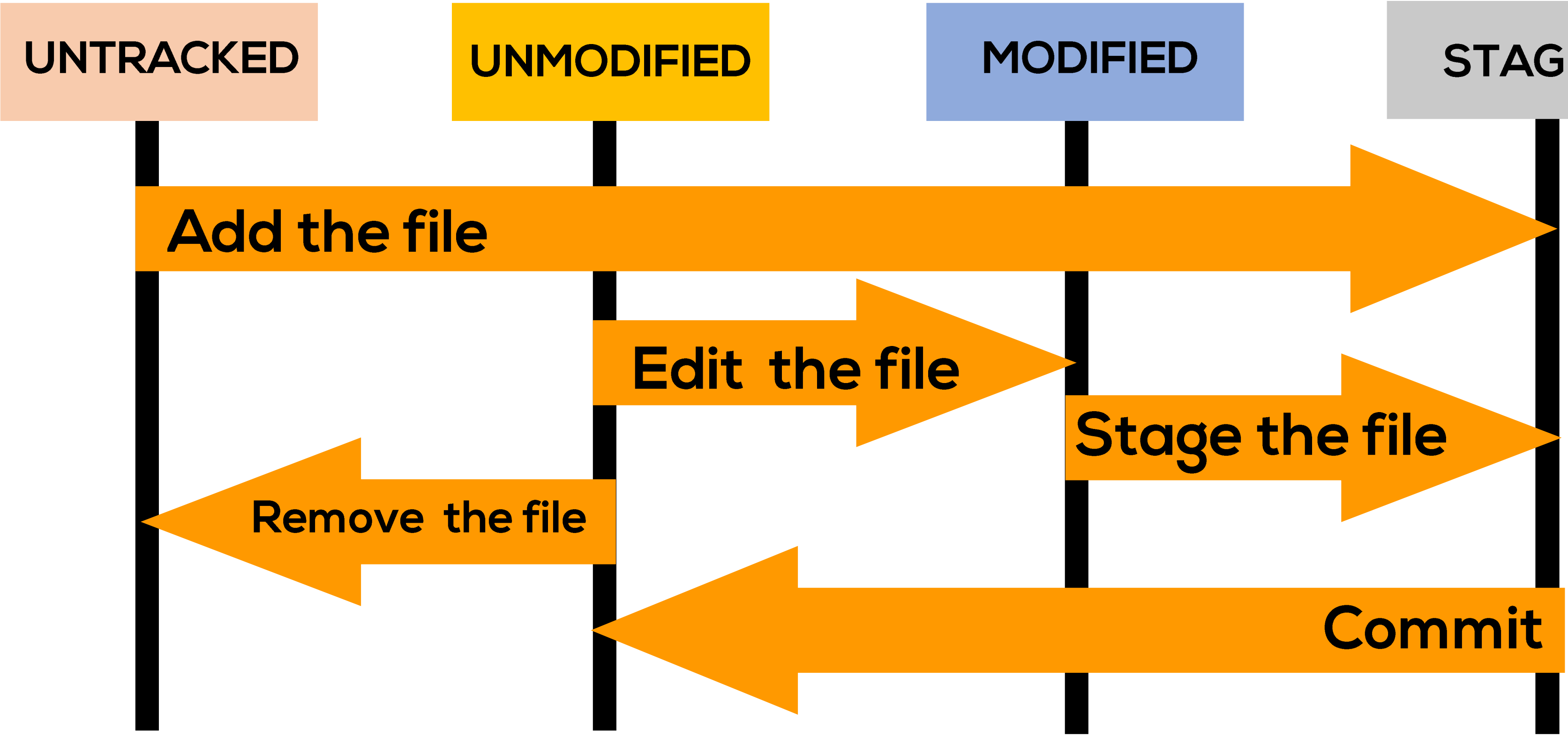
Working directory

Git add operation

Staging area

Git commit operation

Git repository



UNTRACKED

UNMODIFIED

MODIFIED

STAGED

Add the file

Edit the file

Stage the file

Remove the file

Commit

CREATE REPOSITORY:

The git init command creates a new Git repository. It can be used to convert an existing, unversioned project to a Git repository or initialize a new empty repository.

Go to the path of your project:

```
git init
```

REMOTE REPOSITORY:

```
git remote add origin https://github.com/user/repo.git
```

CLONE:

The git clone command copies an existing Git repository.

```
git clone <repository>
```

HOW TO SAVE CHANGES?

ADD:

The git add command adds a change in the working directory to the staging area. It tells Git that you want to include updates to a particular file in the next commit.

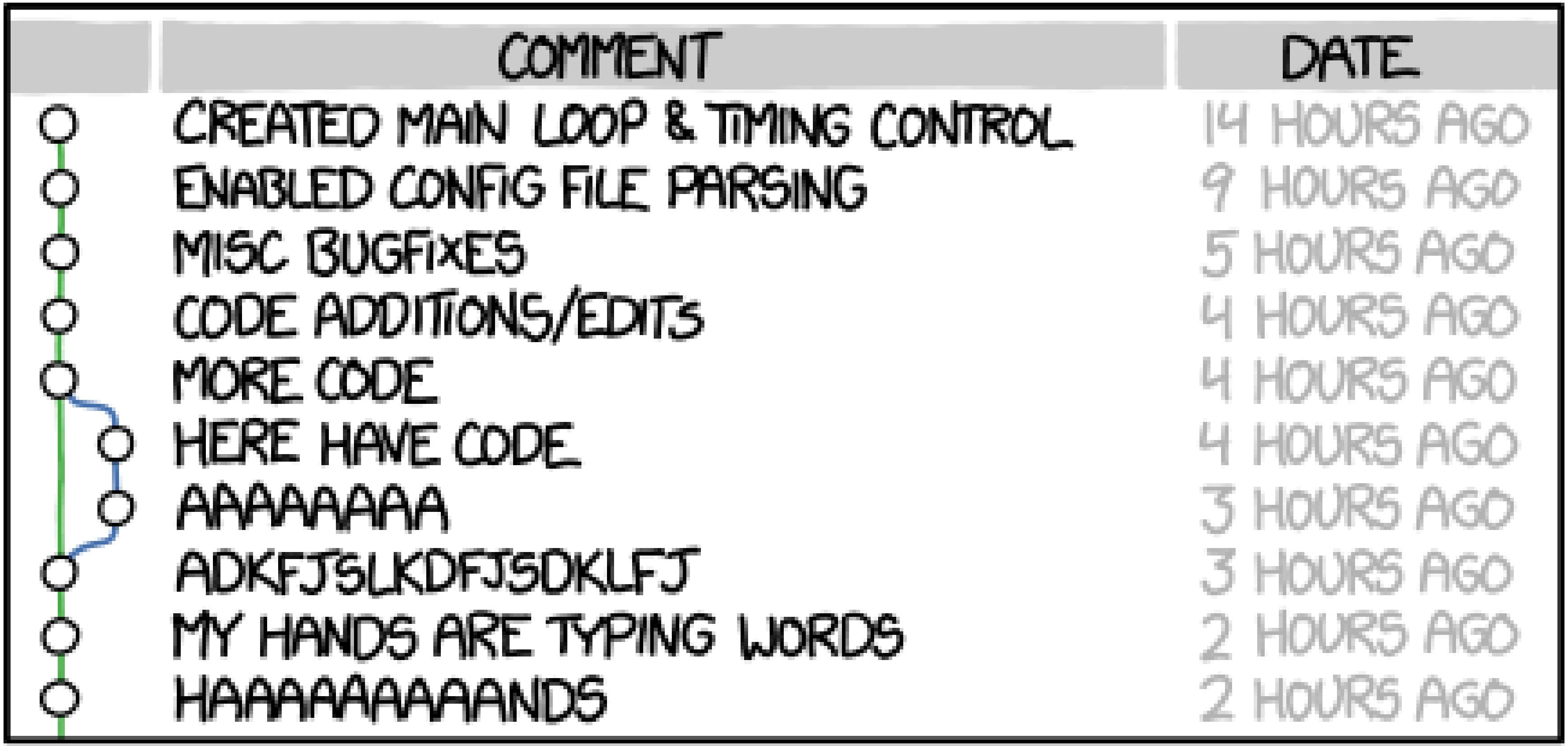
However, git add doesn't really affect the repository in any significant way—changes are not actually recorded until you run git commit.

```
git add <filename>
```

COMMIT:

Changes are recorded at the repository.

```
git commit –m
“<<Git Message Here>>”
```



	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

STATUS:

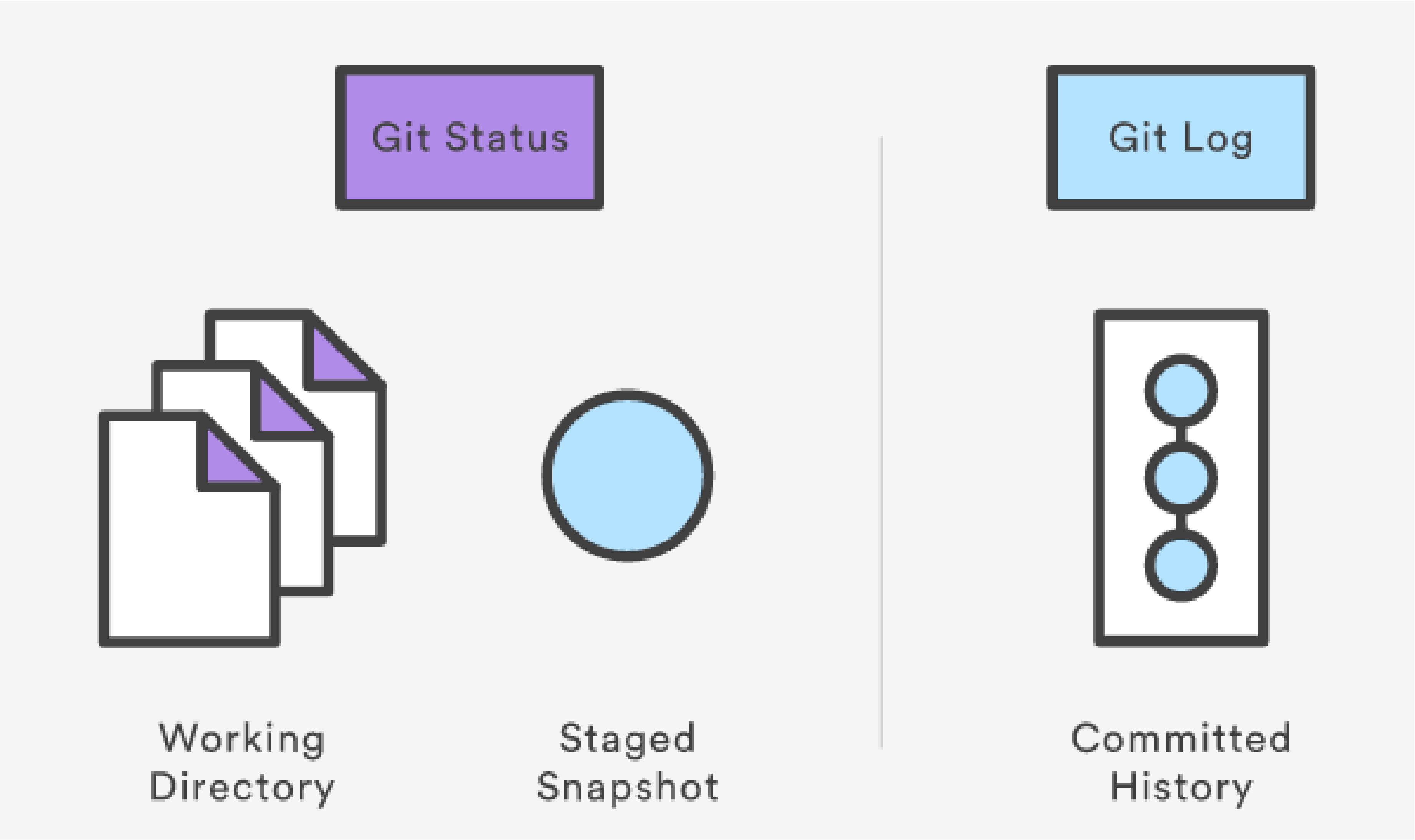
Displays the state of the working directory and the staging area. It lets you see which changes have been staged, which haven't, and which files aren't being tracked by Git.

```
git status
```

LOG:

Status output does not show you any information regarding the committed project history. For this, you need to use:

```
git log
```



Git Status

Git Log

Working Directory

Staged Snapshot

Committed History

PULL:

Merging upstream changes into your local repository is a common task in Git-based collaboration workflows.

```
git pull <origin> <branch>
```

I SEE THAT YOU COMMIT YOUR CODE WITHOUT DOING A PULL FROM GIT TO GET THE LATEST UPDATES
I ALSO LIKE TO LIVE DANGEROUSLY
quickmeme.com

git

PUSH:

Pushing is how you transfer commits from your local repository to a remote repository. This has the potential to overwrite changes, so you need to be careful how you use it.

```
git push<origin> <branch>
```

In case of fire

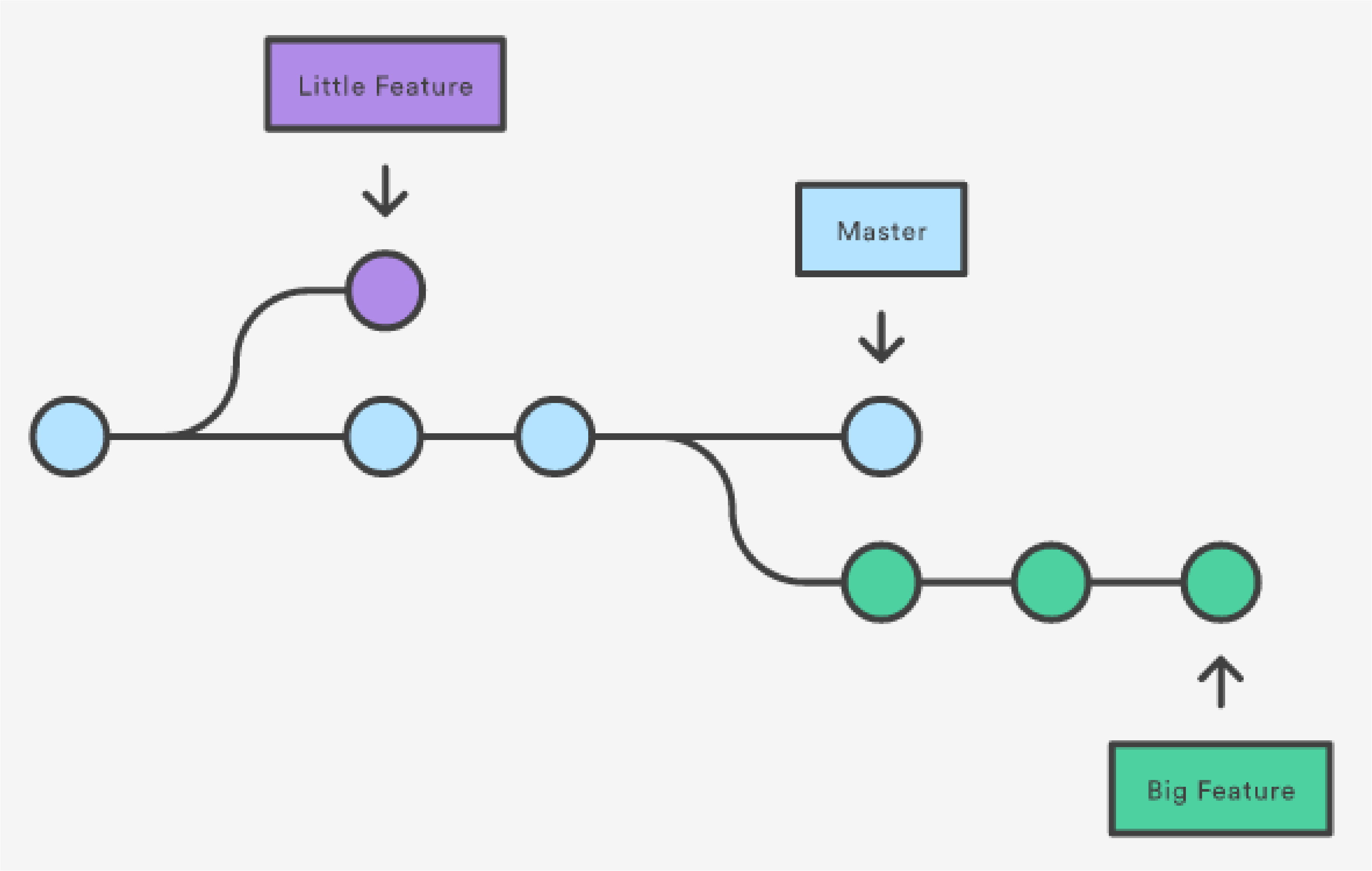
1. git commit
2. git push
3. leave building

git

BRANCH:

A branch represents an independent line of development

git branch <<branchName>>


Little Feature
Master
Big Feature

CHECKOUT:

The git checkout command serves three distinct functions: checking out files, checking out commits, and checking out branches.

```
git checkout
<branch / commit>
```

BRANCH CHECKOUT:

Navigates between the branches created by git branch.

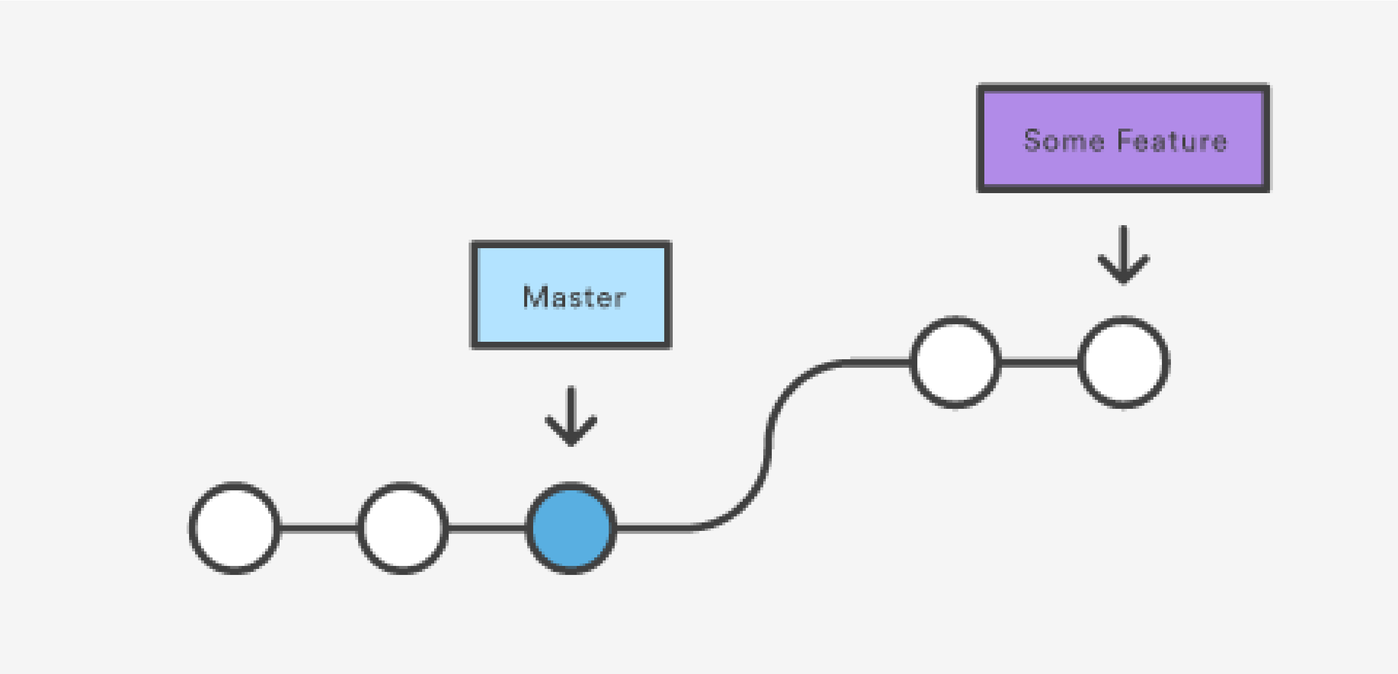
```
git checkout <branch>
```

MERGE:

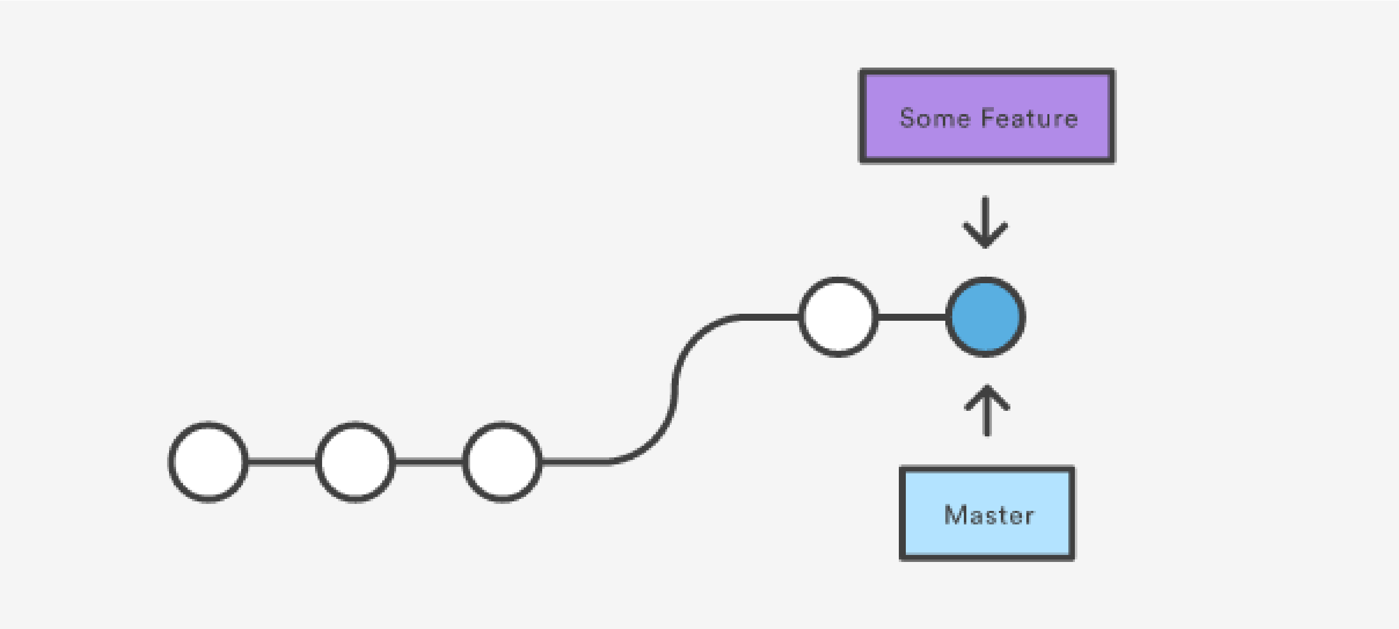
Merging is Git's way of putting a forked history back together again. The git merge command lets you take the independent lines of development created by git branch and integrate them into a single branch.

```
git merge <branch>
```

Before merging:



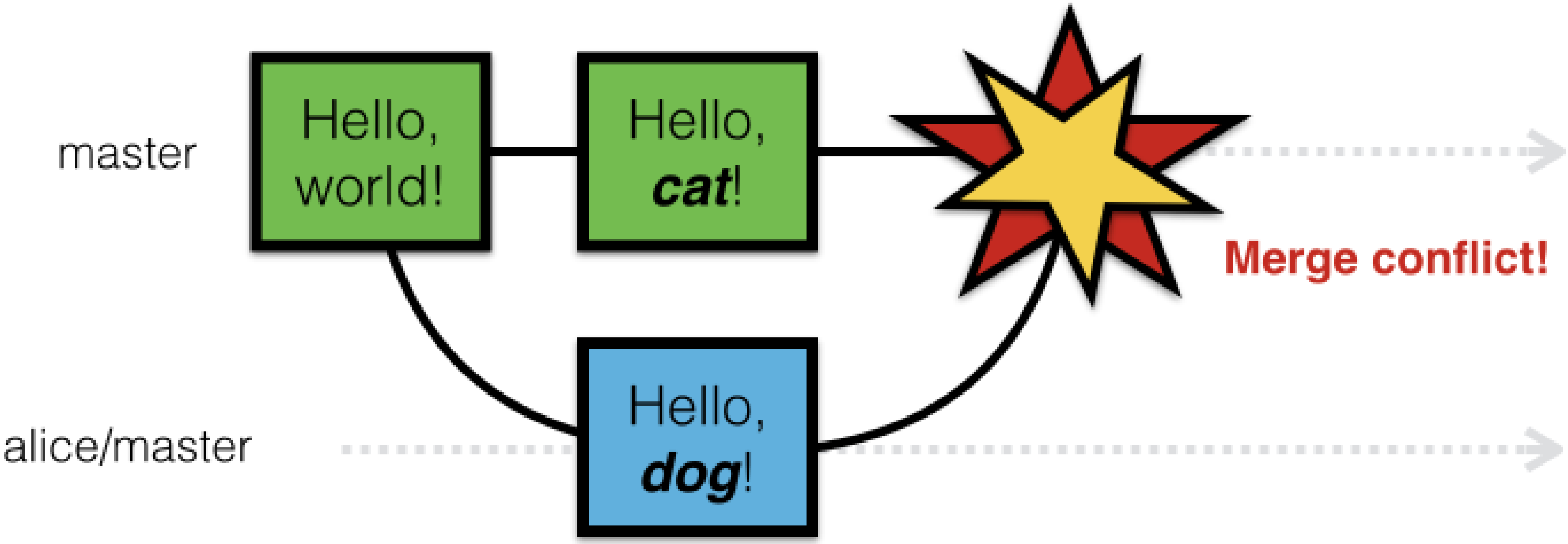
After merging:



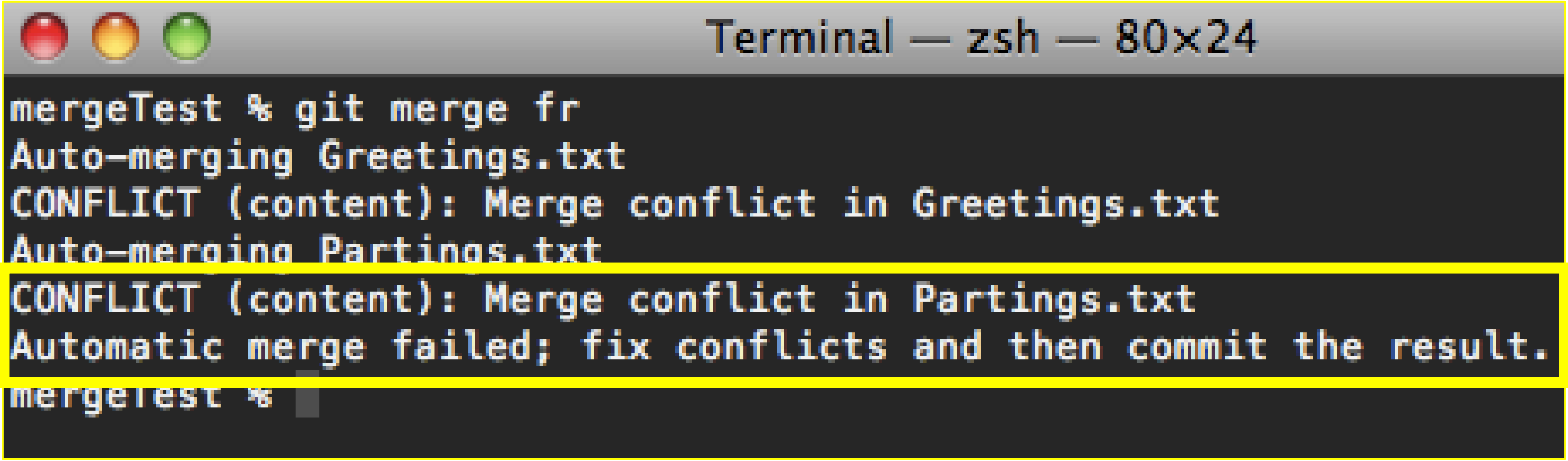
BRACE YOURSELF
MERGE CONFLICTS ARE COMING
memegenerator.net

git

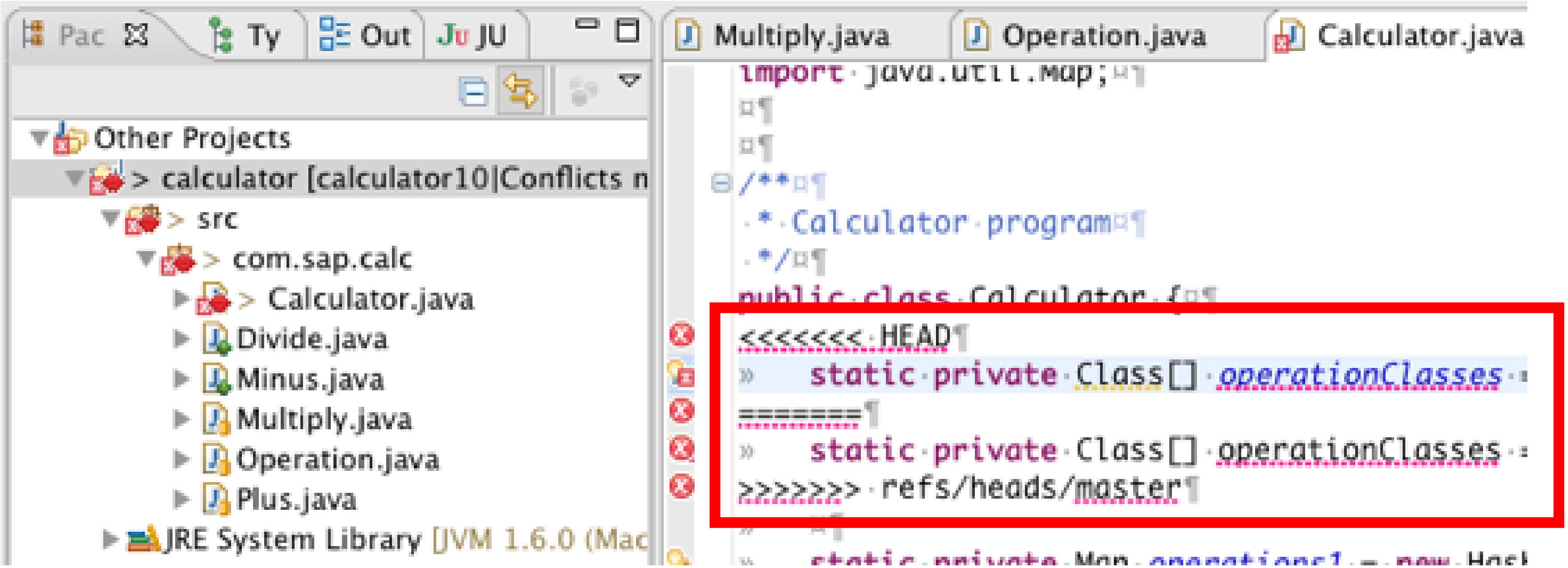
<<git merge GIF>>


master
Hello, world!
Hello, *cat*!
Merge conflict!
alice/master
Hello, *dog*!

git



```
mergeTest % git merge fr
Auto-merging Greetings.txt
CONFLICT (content): Merge conflict in Greetings.txt
Auto-merging Partings.txt
CONFLICT (content): Merge conflict in Partings.txt
Automatic merge failed; fix conflicts and then commit the result.
mergeTest %
```



```
import java.util.Map;


/**
 * Calculator program
 */
public class Calculator {
<<<<<<< HEAD
	static private Class[] operationClasses =
=======
	static private Class[] operationClasses =
>>>>>>> refs/heads/master

	static private Map operations1 = new Hash
```

GIT MERGE
NO CONFLICTS
quickmeme.com

QUESTIONS?

To learn more, visit:

https://www.atlassian.com/git/tutorials/

Send me a message thru messenger:

Errol John Melendez Antonio

THANK YOU,
UE WARRIORS!