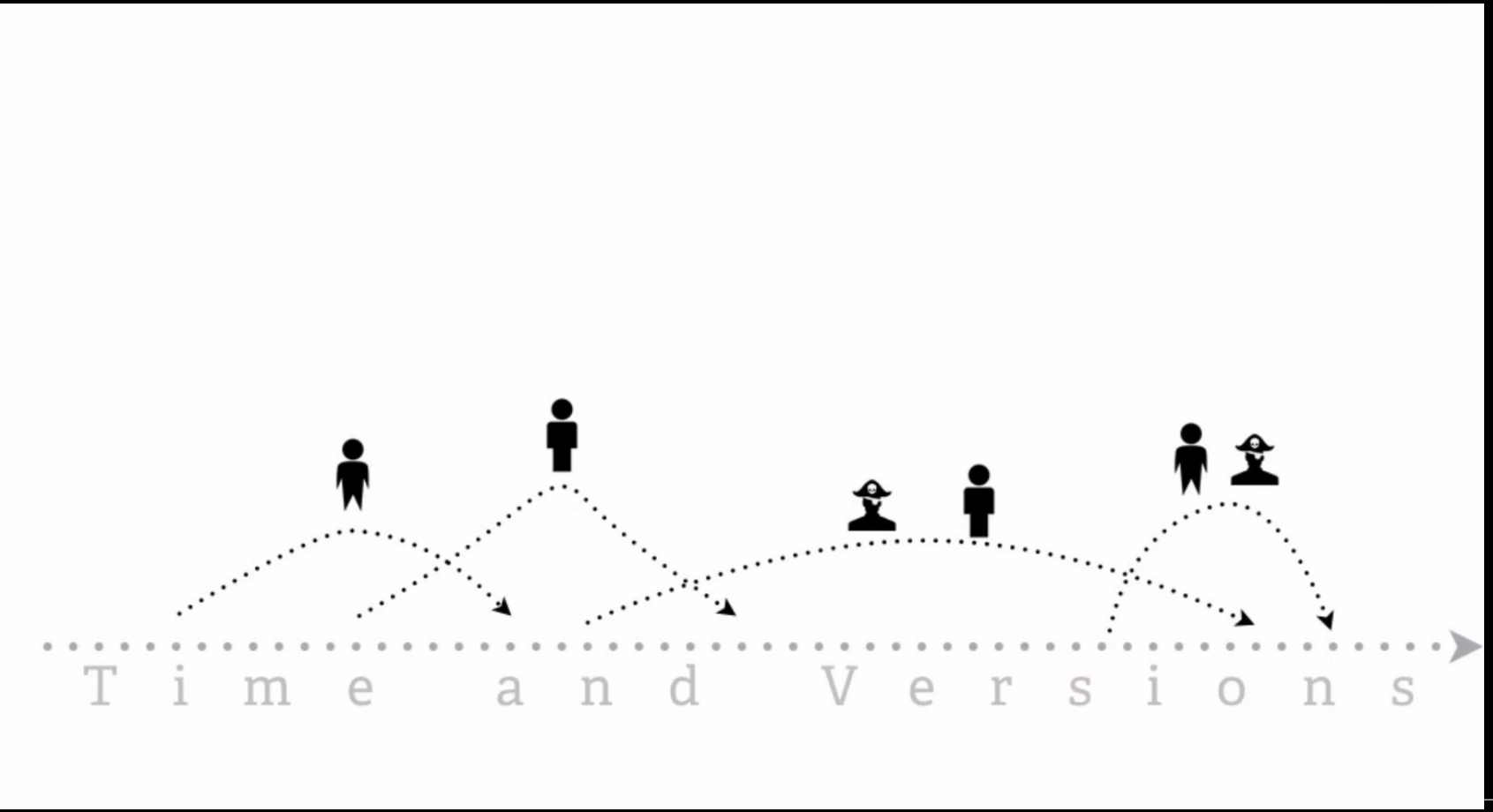
Version Control Systems

Rajesh Verma

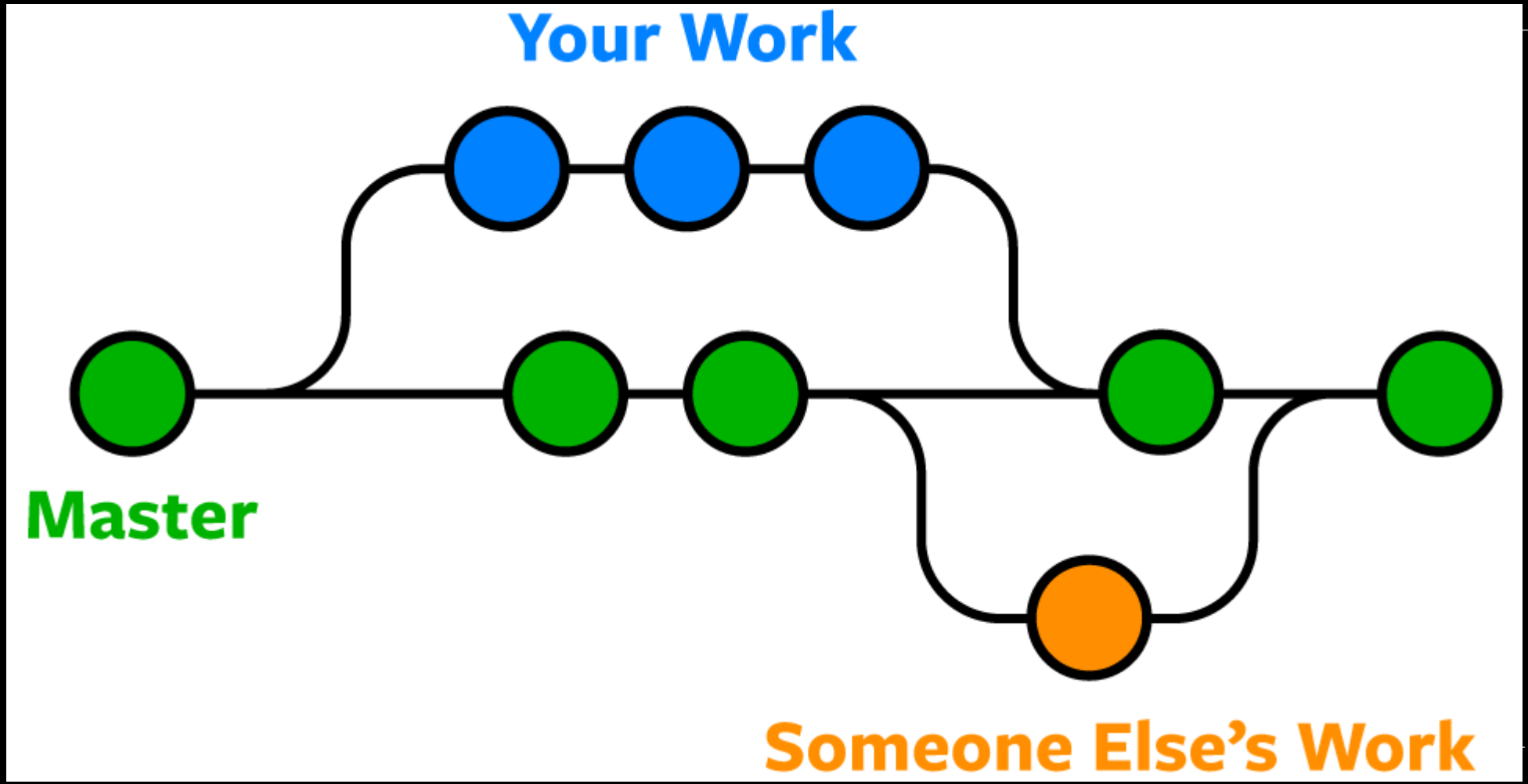
About version control system

- Tracks the history of changes of projects
- Can see the who, when, and why of the changes
- Allows for collaboration on small or large projects

Collaborative History Tracking



Branches



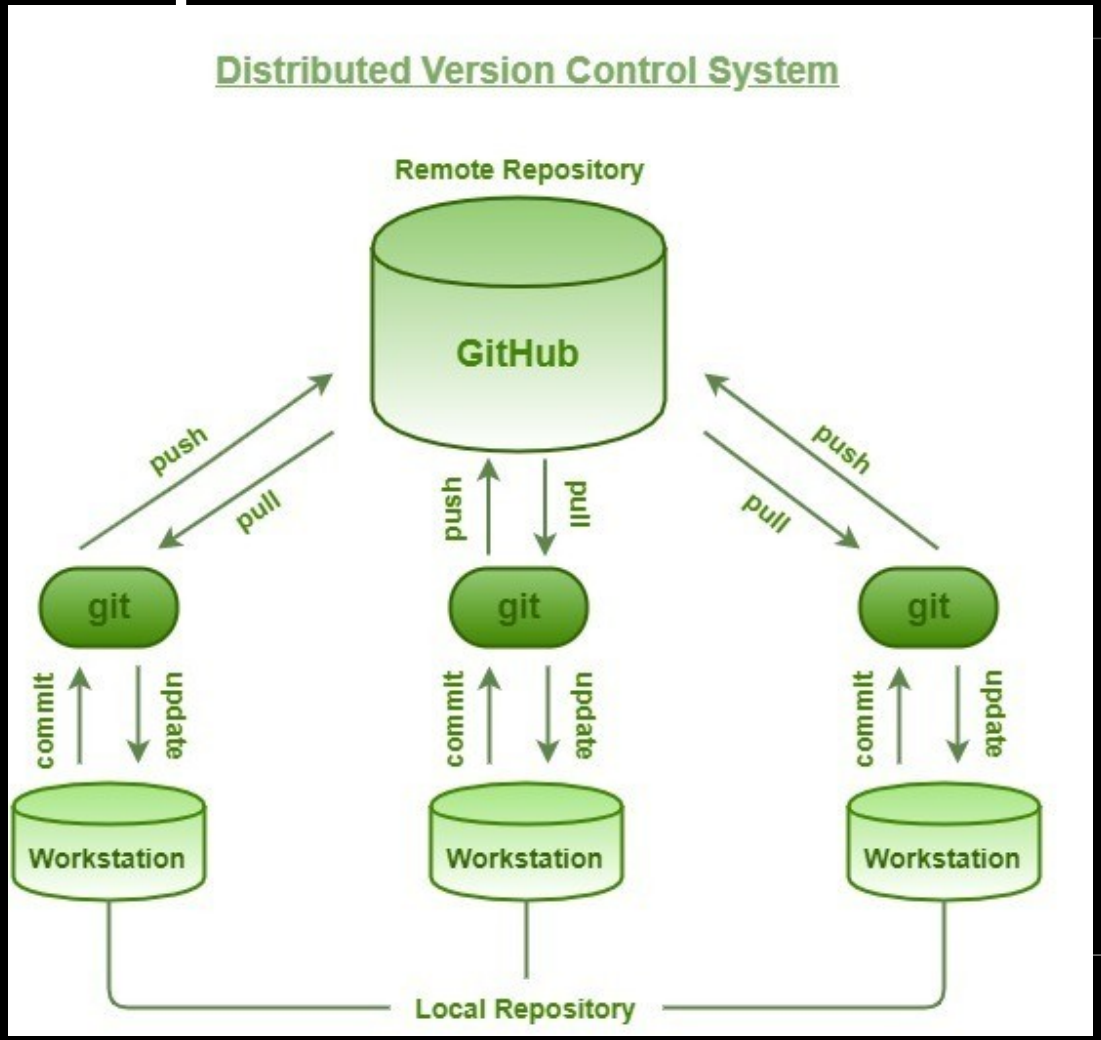
Allows for experimentation

- Try some changes without harming the production environment or your current work

Quick Note

- Git is the tool. GitHub is a repository. There is also Sourcetree, GitLab, etc.
- Repositories provide more features for project management and integrations

Repositories



Different changes can be made on each local repository

Changes are tracked by Commits, pushes, pulls, and updates

Getting and Creating Projects

- git init : Create an empty Git repository or reinitialize an existing one
- git clone: Clone a repository into a new directory

Basic Snapshotting

- git status: show files that have changes
- git diff: show actual changes between different states
- git add: choose files to update in repository
- git commit: record changes to repository
- git reset: reset current head to a specified state

Branching and Merging

- git branch: List, create, or delete branches
- git switch: Switch to a specified branch
- git merge: Join two or more development histories together

Sharing and Updating Projects

- git pull: Apply changes from remote repository into the local repository
- git push: Update remote repository with local repository

Miscellaneous files

- Readme.md: Markdown file that provides documentation for others to use your repo
- .gitignore: A list of paths that will not be committed to a remote repo. Good for configuration files, images, and binaries

Cheat Sheets

- https://git-scm.com/docs (Reference and Book)
- https://ndpsoftware.com/git-cheatsheet.html
- https://training.github.com/downloads/github-git-cheat-sheet/

Live Demo!

- Murphy’s Law: Anything that can go wrong will go wrong