Introduction to Version Control & Git

NHS-R Conference 2022

23/11/2022

Presented by:

Helen Richardson and Joe Wilson

Who we are:

NHS Digital

Helen Richardson:

Joined NHS Digital in 2019 and have been working on training and implementing RAP principles with NHS Digital's data publication teams for the past year. We have published 7 publications' codebases on GitHub with more on the way.

Joe Wilson:

Joined NHS Digital in September 2021 as a Graduate Trainee, after completing a Masters in Data Science. First placement was with NHS Spine. Second placement was working with the Business Intelligence Dashboards team, developing a testing framework for Tableau dashboards. Currently assigned to the RAP Squad, engaging with teams within NHS Digital and helping them learn and utilise RAP techniques and principles.

Schedule

- 9:30 Introduction to version control, Git and GitHub
- 10:00 (roughly) Short break 10 min
- Practical using version control
- 12:00 – 13:00 Lunch break
- 13:00 – 16:00 Git Collaboration with allocated breaks

What is RAP?

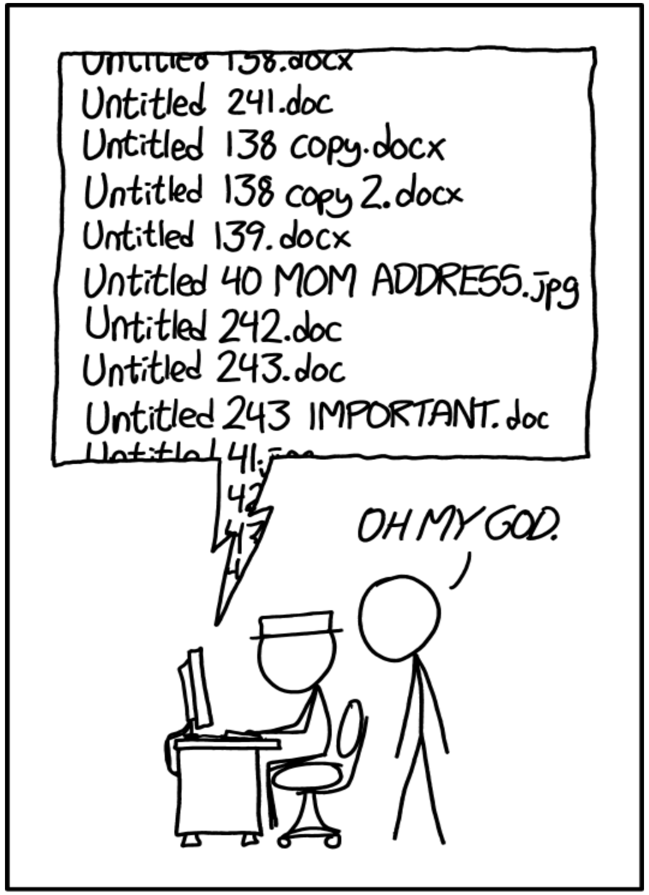
- Reproducible Analytical Pipelines (RAPs) are automated statistical and analytical processes. They incorporate elements of software engineering best practice to ensure that the pipelines are reproducible, auditable, efficient, and high quality.
- RAPs increase the efficiency of statistical and analytical processes, delivering value. Reproducibility and auditability increase trust in the statistics. The pipelines are easier to quality assure than manual processes, leading to higher quality.
- Statisticians and analysts should look to implement the RAP principles in parts of their processes.
- At NHS Digital we strive to publish as many analytical codebases as possible (7 at the moment!).
- **NHS Digital RAP Community of Practice: https://nhsdigital.github.io/rap-community-of-practice/**

Source: https://analysisfunction.civilservice.gov.uk/support/reproducible-analytical-pipelines/

What will we learn today?

- What is version control, Git and GitHub
- What is a repository, branches, README, .gitignore etc.
- How to apply version control to my project using Git commands
- How to create/fork a repository
- How to submit a Pull Request on GitHub
- How to use Git collaboratively (within a team setting)
- How to handle merge conflicts

What is version control?



PROTIP: NEVER LOOK IN SOMEONE ELSE'S DOCUMENTS FOLDER.

There has to be a better way…

https://xkcd.com/1459/

What is version control?

- **Version control is the practice of tracking and managing any changes on project's code, files or folders over time.** This allows you to observe a detailed history of the changes made and enables your team members to collaborate on the same project.

- In Git, each user has the entire **repository** (project's working directory) on their computer (**offline**), hence, they can work separately offline until they opt to push their updated version of the code to the **remote** (**online**) central **repository**.

- **Remote repositories** are versions of your project that are hosted on an online Version Control System. In NHS Digital, we store that in GitLab site for internal users (GitHub site for the public).

Why should we care about version control?

The benefits of using version control include:

- Understanding what happened in the past. E.g., what did we change last time we ran this code.
- Restoring previous versions.
- Tracking changes - avoid accidentally breaking code.
- The ability to review someone's changes and to leave comments.
- The ability to plan development work more effectively through being able to assign small, discrete changes.
- Avoid code being hidden away on someone's machine.
- The ability to set up an approval process for changes.
- The ability to make changes without breaking anything - through running automated tests.
- The ability to try out experiments without the risk of breaking your main code.

Git vs GitHub

- Git is a software
- released in 04/2005
- is maintained by Linux
- manages code and file history
- is installed locally on your computer
- focuses on version control and code sharing
- no user management ability

- GitHub is a service/platform
- released in 2008
- is maintained by Microsoft
- hosting service for Git repositories
- available online
- focuses on centralised source code hosting
- has user management features

Other Git hosting services:

- GitLab
- Bitbucket
- SourceForge
- GitBucket
- AWS Elastic Beanstalk
- Etc.

tl;dr:
Git is the tool to make things appear on GitHub (basically).

What is a repository?

- **Example of a remote (online) repository:**

https://github.com/nhs-r-community/git_training

What is a branch?

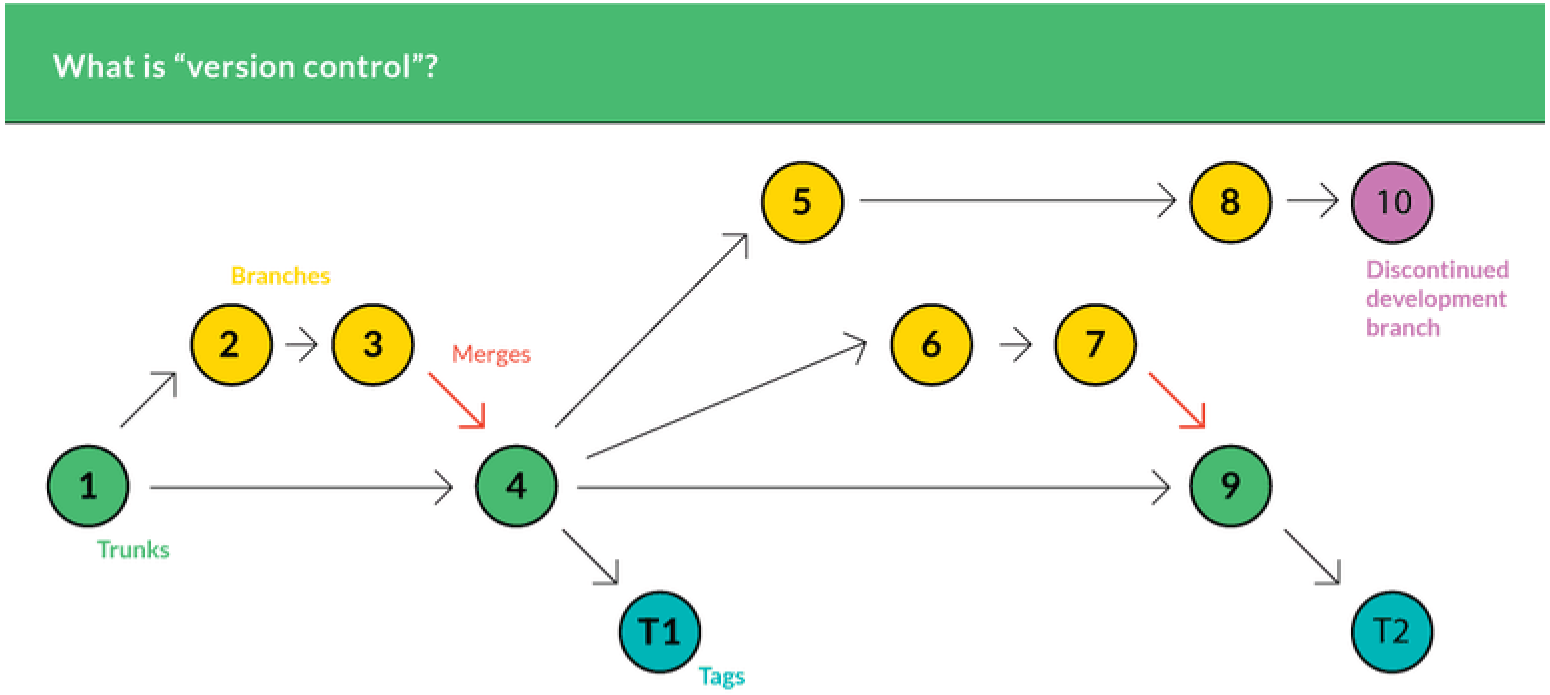
Source: https://www.gotreequotes.com/branch-vs-limb/

What is a branch?

"Branch" is another word for "version".

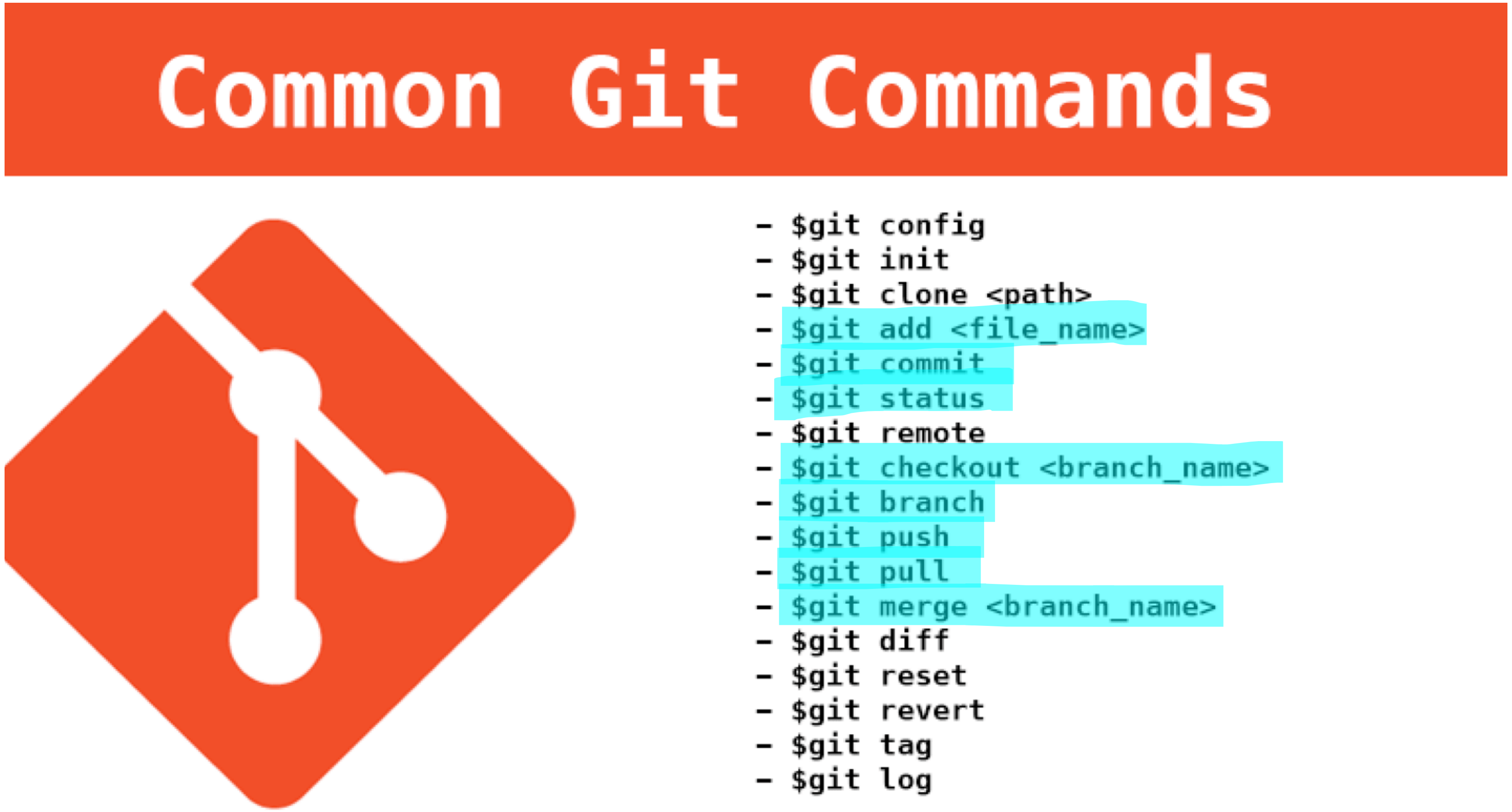
Usually when developing a document or slides for a presentation, we could have numerous versions of the same slides, with small or major differences. By using different branches (or versions) of the same code, we can safely work and test without breaking the publication code, that resides in the default master branch of the repository.

Git branches



Source: https://www.webinerds.com/blog/version-control-systems-keep-your-code-in-order

Git commands – why learn them?



Common Git Commands

```
- $git config
- $git init
- $git clone <path>
- $git add <file_name>
- $git commit
- $git status
- $git remote
- $git checkout <branch_name>
- $git branch
- $git push
- $git pull
- $git merge <branch_name>
- $git diff
- $git reset
- $git revert
- $git tag
- $git log
```

Source: GitLab, https://dev.to

What’s next?

- Log into Posit cloud (or use your local Rstudio desktop if you wish).
- Further instructions: https://github.com/nhs-r-community/git_training/blob/main/README.md
- Break time! (10 min)

Fun time (but also 10min break)!