



Building software: Version control with Git

Data Sciences Institute
University of Toronto

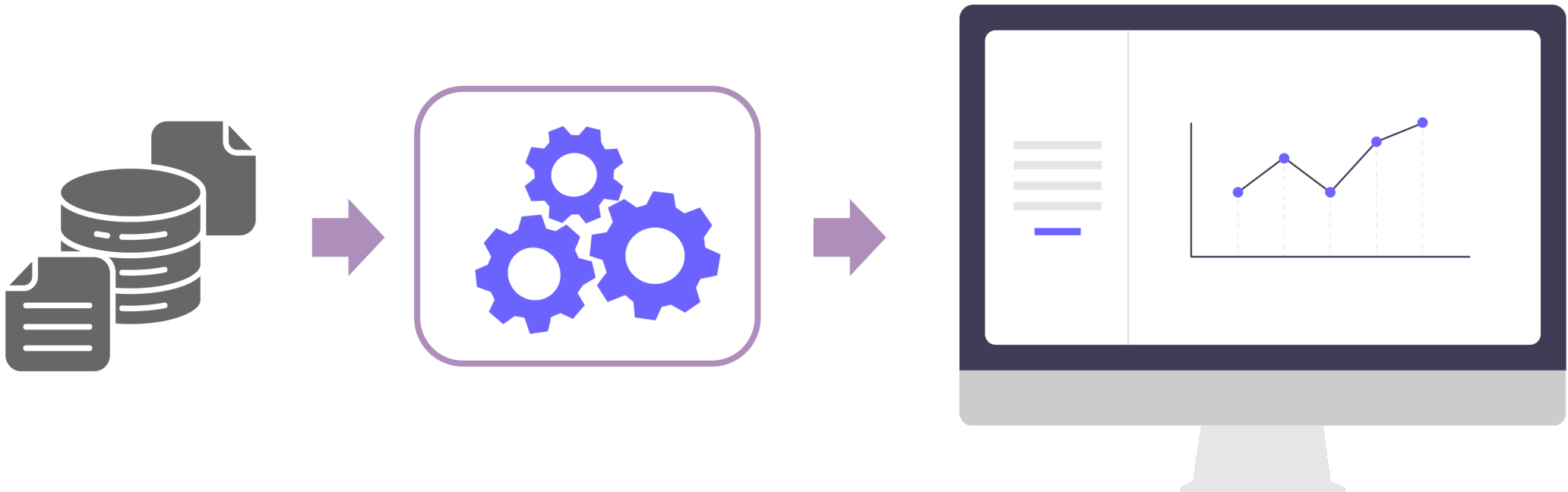
Course objective

How to write **robust software** in a **team** that we, our colleagues, and the public can **trust** and **use with confidence**.

Git: Branching

Alex's new data pipeline

- Alex is a data engineer at a mid-sized company working on a new data processing pipeline and BI dashboard module



Git: Branching

Alex's new data pipeline

- Alex is a data engineer at a mid-sized company working on a new data processing pipeline and BI dashboard module



Git: Branching

Alex's new data pipeline

- Alex is a data engineer at a mid-sized company working on a new data processing pipeline and BI dashboard module
- Alex has a basic data pipeline and most of the BI module written
- Alex is currently working on expanding the data pipeline with more features. The expanded pipeline is not yet working, but.....
- She has a big client meeting coming up and they want a demo!

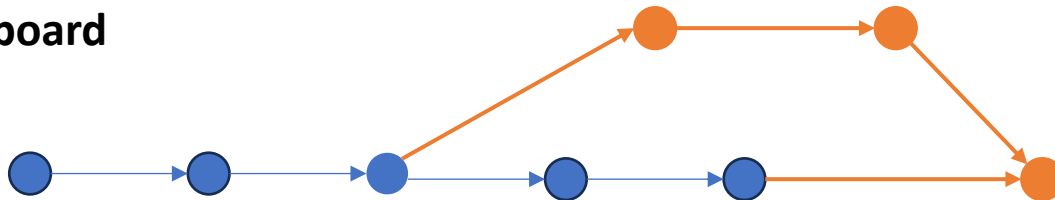
Git: Branching

Alex's new data pipeline

- Alex can use Git to go back to the last working state of her basic data pipeline
- Alex can finish up the BI module on another branch
- Present the amazing new BI module and wow then client
- Then merge her dashboard work back into the main branch incorporating both the in-progress pipeline and finished BI module

finish_dashboard

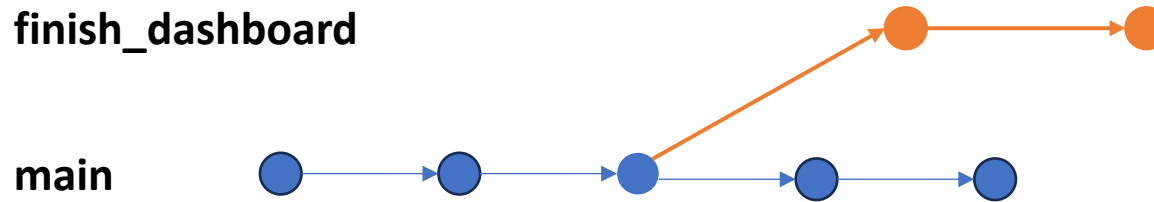
main



Git: Branching

\$> Interactive live coding

Finish the dashboard on a separate branch.



1. Clone https://github.com/dtxe/DSI_branch_demo

```
git clone https://github.com/dtxe/DSI_branch_demo
```

2. Switch to good commit

```
git switch -c finish_dashboard 6539845acab60c73ab50b65d58d9e39fd4a10119
```

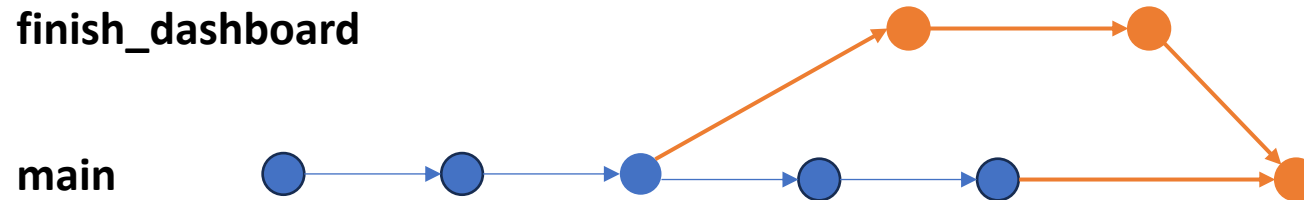
3. Finish the dashboard

```
touch dashboard ; git add -A ; git commit -m "finish dashboard"
```

Git: Branching

\$> Interactive live coding

Merge the dashboard work into the main branch.



1. Switch to main

```
git switch main
```

2. Merge finish_dashboard

```
git merge finish_dashboard
```


Git: Branching

Tracking changes with Alex

- Follow along as Alex uses Git to simplify her work
 - Create a new branch from a commit `git switch`
 - Merge changes from another branch `git merge`

Questions?

Git: Branching

Listing branches

- List branches in your repo with

```
git branch -v
```

\$> Let's try it now!

Git: Branching

Deleting branches

- Delete branches in your repo with

`git branch -d <branch name>`

- Git will warn you if your branch contains work that hasn't been incorporated into the main branch yet
 - But it is best practice to check before deleting anyways!

Pop-quiz: How do we check what commits are in a branch?

Git: Branching

Deleting branches

- Delete branches in your repo with

`git branch -d <branch name>`

- Git will warn you if your branch contains work that hasn't been incorporated into the main branch yet
 - But it is best practice to check before deleting anyways!

\$> Let's try it now!

- Create a branch, make a commit, try deleting, merge, try deleting again

Git: Branching

\$> Deleting branches

```
git switch -c newbranch
```

```
touch newfile ; git add -A ; git commit -m "newfile"
```

```
git switch main
```

```
git branch -d newbranch
```

```
>>> Error: branch is not fully merged
```

```
git merge newbranch
```

```
git branch -d newbranch
```

```
>>> Success!
```

Git: Branching

Branches on GitHub

The screenshot shows the GitHub repository page for 'DSL_git_assignment'. The 'main' branch is selected, and the file list is visible. A yellow banner at the top indicates that 'feature1' had recent pushes 42 minutes ago. The file list includes 'Analyze.py', 'LICENSE.txt', 'README.md', and 'ttc-bus-delay-data-2023.csv'. The README section is partially visible, showing the text 'Load, analyze, and visualize TTC bus delay data'.

The screenshot shows the 'Branches' page in the 'DSL_git_assignment' repository. The 'Overview' tab is selected, and a table lists the branches. The 'main' branch is the default. The 'feature1' branch is 0 commits behind and 2 commits ahead of 'main'. The 'bugfix1' branch is 0 commits behind and 1 commit ahead of 'main'. A red box highlights the 'Behind' and 'Ahead' columns, with a red arrow pointing to the text 'Quick overview of branch content relative to main'.

Branch	Updated	Check status	Behind	Ahead	Pull request
main	12 hours ago		Default		
feature1	43 minutes ago		0	2	
bugfix1	12 hours ago		0	1	

\$> Let's try it now!

Git: Branching

Git fetch

- Ask git to download from remote repositories
- Does not change your current working directory
- Enables subsequent merge from or switch to remote branches

`git fetch upstream newfeature`

THEN `git merge upstream/newfeature`

OR `git switch -c newfeature upstream/newfeature`

Git: Branching

Pull = Fetch + Merge

- The combined fetch and merge happens very often
- Combined into the verb pull

`git pull upstream newfeature`

essentially performs:

`git fetch upstream newfeature`

`git merge upstream/newfeature`

Questions?

Git: Branching

Git in VSCode

- Basic git commands are built-in with VSCode
 - Staging files, Commits, Branches
- View relationship between git commits intuitively with Git Graph

\$> Let's try it now!

Questions?

Git: Branching

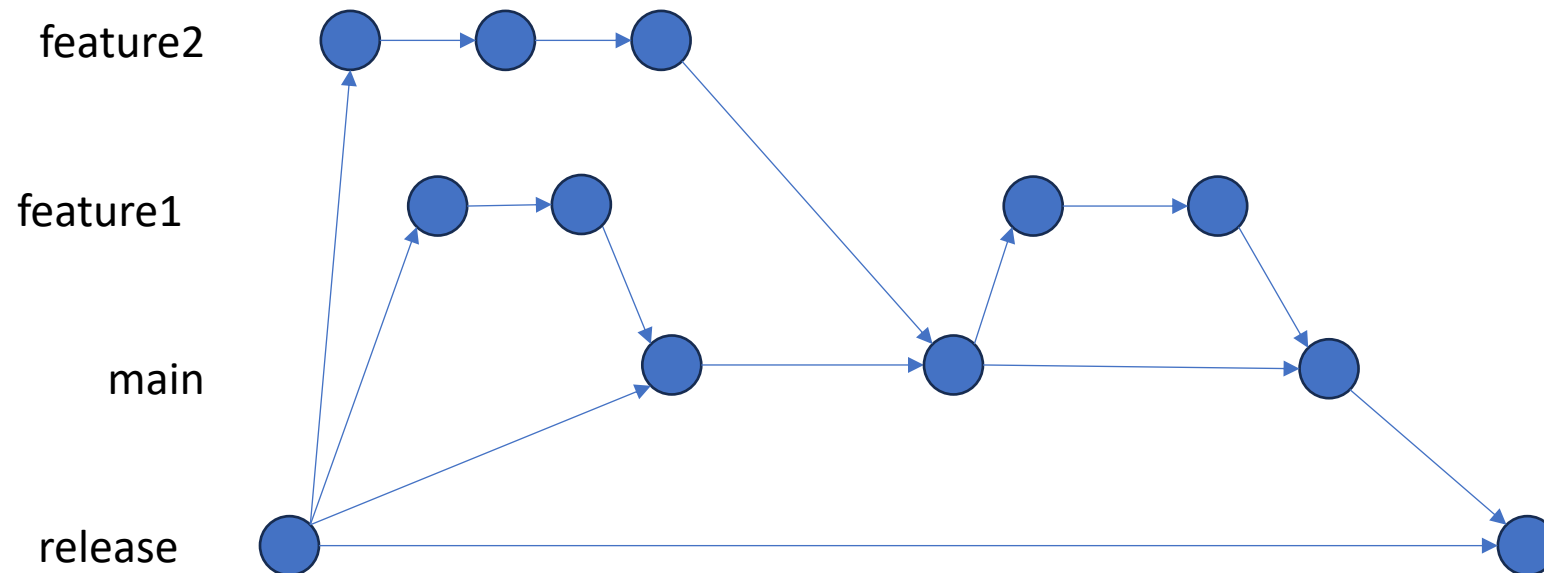
Everything is a branch

- Including forks!
- We can merge from local branches, forks, remote branches, etc...

Git: Branching

Branch workflows

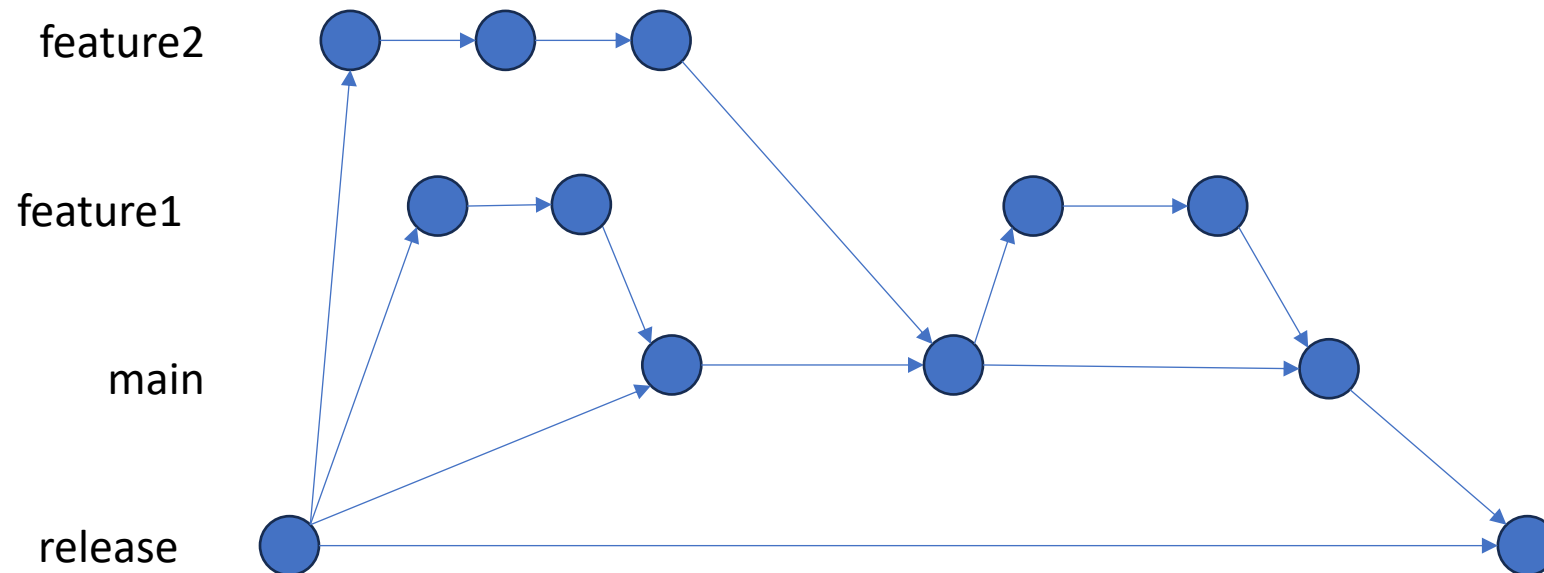
- Multiple long-running branches are helpful for large projects
- Features are developed in their own branches, based on release commits



Git: Branching

Branch workflows

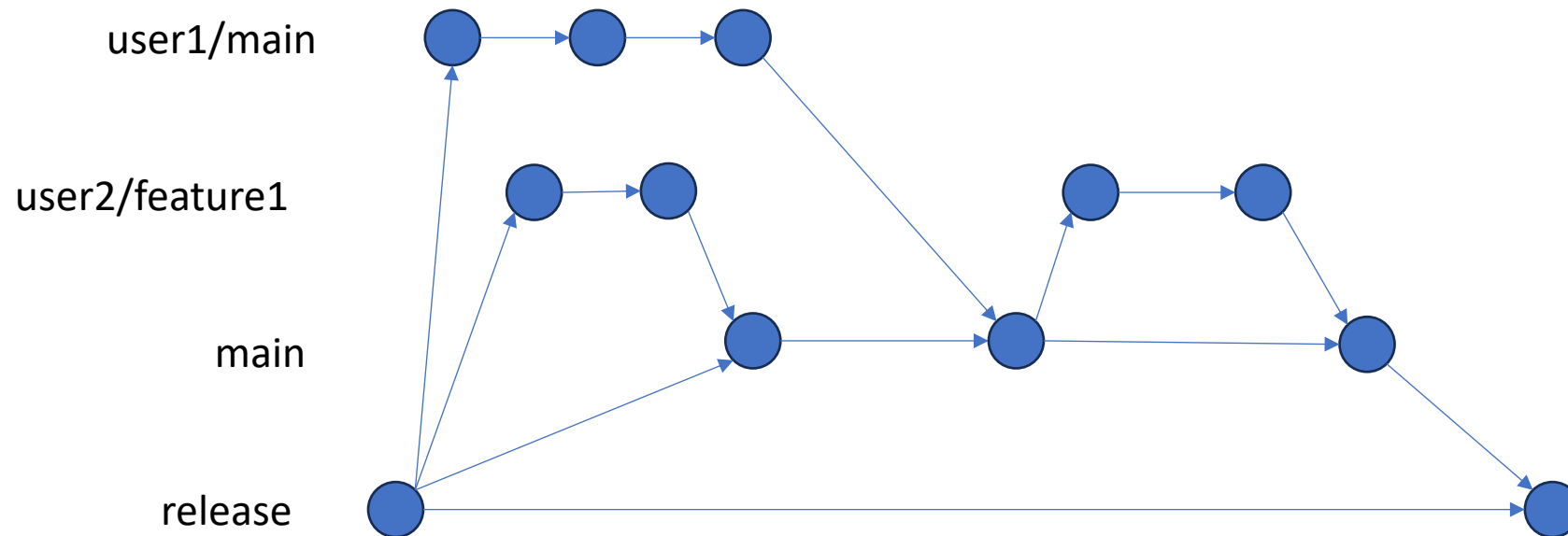
- Branches can have various levels of stability
 - Code can graduate/merge from feature/topic branches when stable
 - Features can be developed in parallel and merged into main



Git: Branching

Branch workflows

- Branches can be user forks too



Questions?

Git: Ignoring files

.gitignore

- Why?
 - Large data files, intermediate output, secret keys, etc...
- Defined in the .gitignore file
- Specify path with wildcards
- **Best practice:** use existing .gitignore templates
 - Find them on GitHub: <https://github.com/github/gitignore/blob/main/Python.gitignore>

Questions?