

Version Control & Dev Workflow

Let's make a deal



- You can just chill if you know what I'm doing, or helping your classmate!
- Help me make this class less boring.
- There is no stupid question, there is no stupid answer.
- We have time, raise your hand for anything.
- I might be wrong, you might know something better than me, so tell me.

Hi, I'm Lookkid

CPE29 KMUTT | Open Society Technology @ Tallinn University, Estonia

PREVIOUS EXP.



Wongnai
Start up (~100 people)



Thoughtworks
International Tech Consult
(Client size: 1k+ people)

CURRENT ROLE



Punch Up / WeVis
DataVis Studio / Civic Tech
(15 People)



HATO
Long term side project
(2 core team, solo dev)

My goal



Understand the concept

+

Hands on experiment

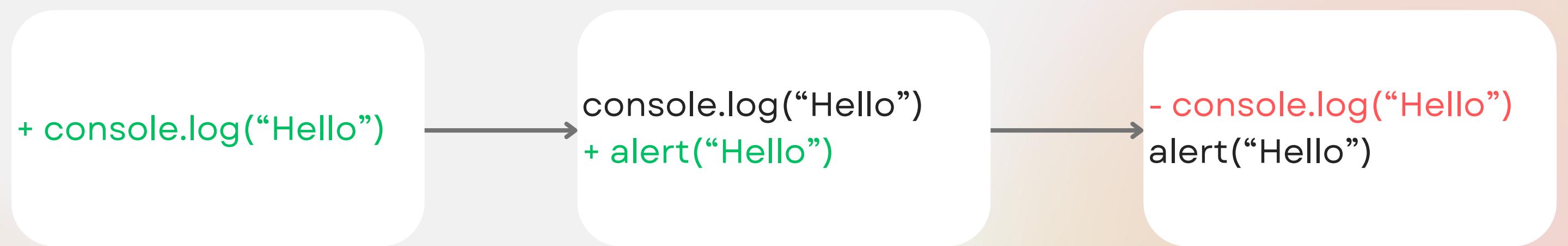
+

Real world experience

Version Control ?

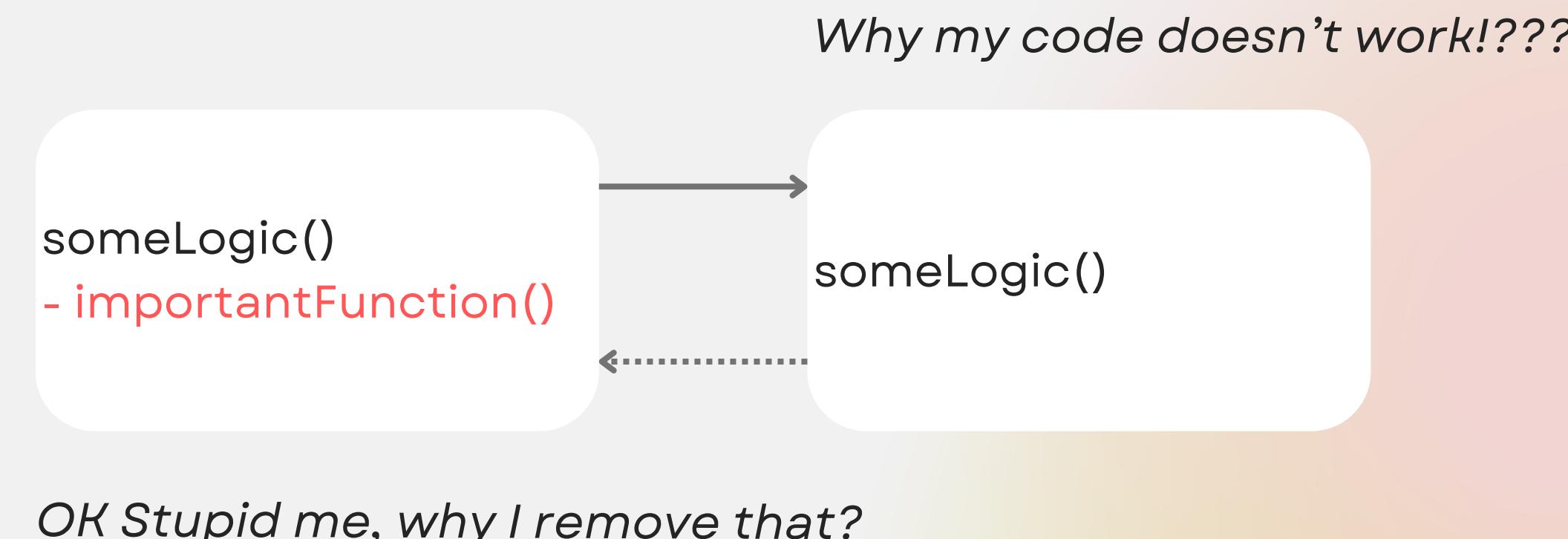
Version Control (n.)

The practice of tracking and managing changes to software code.
Also known as “Source Control”



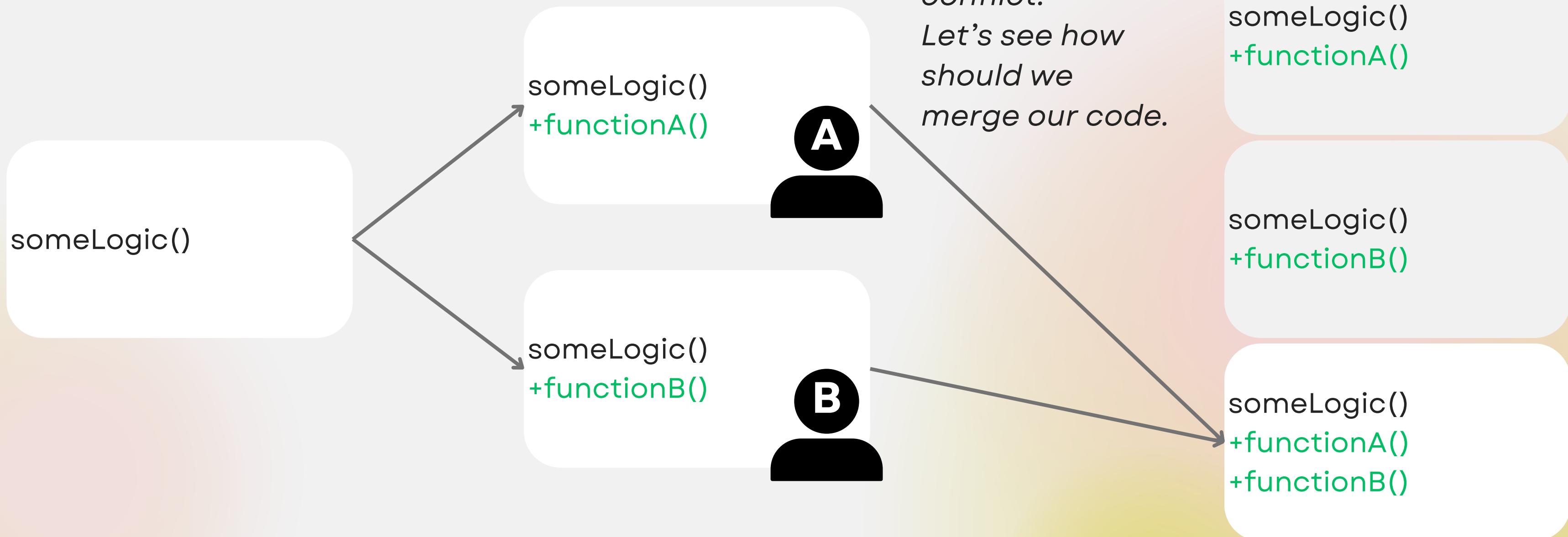
Why Version Control?

1. Software is fragile → We need traceability



Why Version Control?

2. Most software aren't made alone



Version Control System (n.)

Software tools that help you do the version control [1]



BITKEEPER
Scalable Version Control



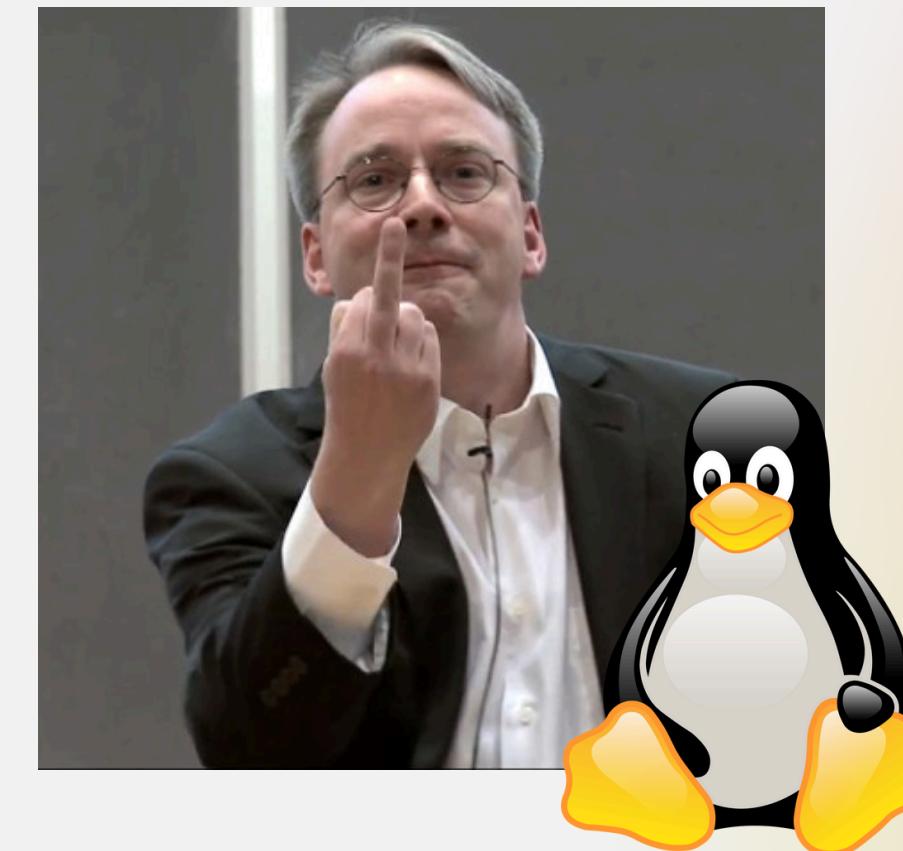
mercurial



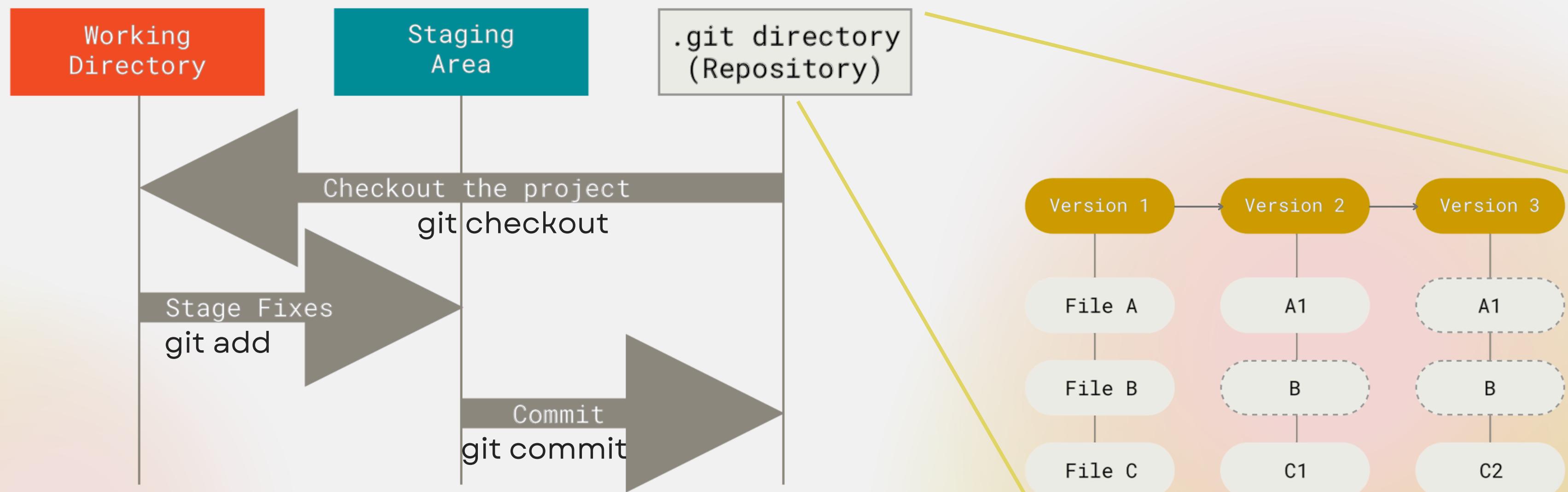
git

From Linux to Git

- 1991 Linux code are passed around as patches and archived files
- 2002 Linus start using BitKeeper for free.
- 2005 BitKeeper want to charge Linux project money, so Linus created git.
- 2018 Final released of BitKeeper

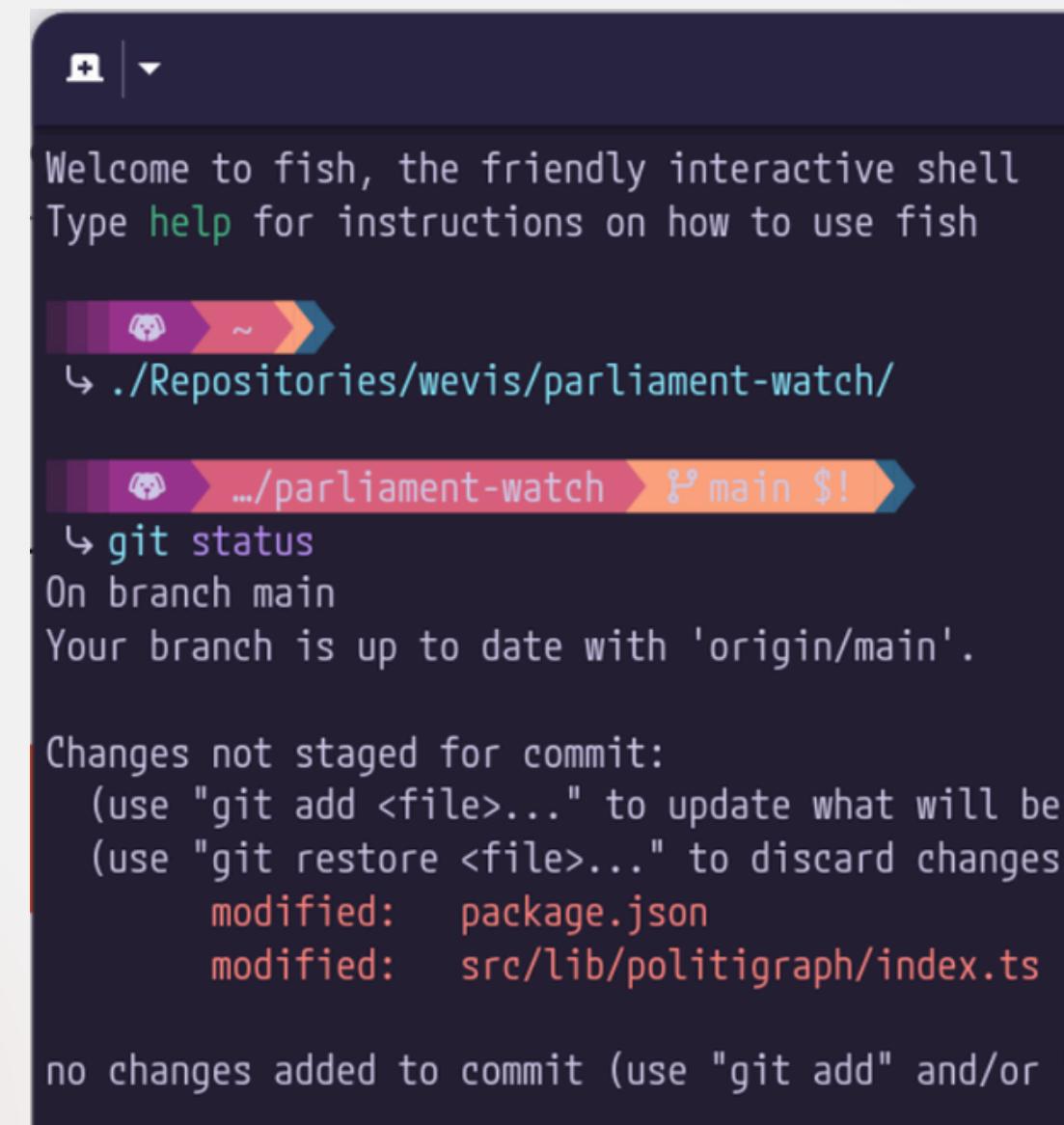


How git do version control?



Let's use git

Command
line



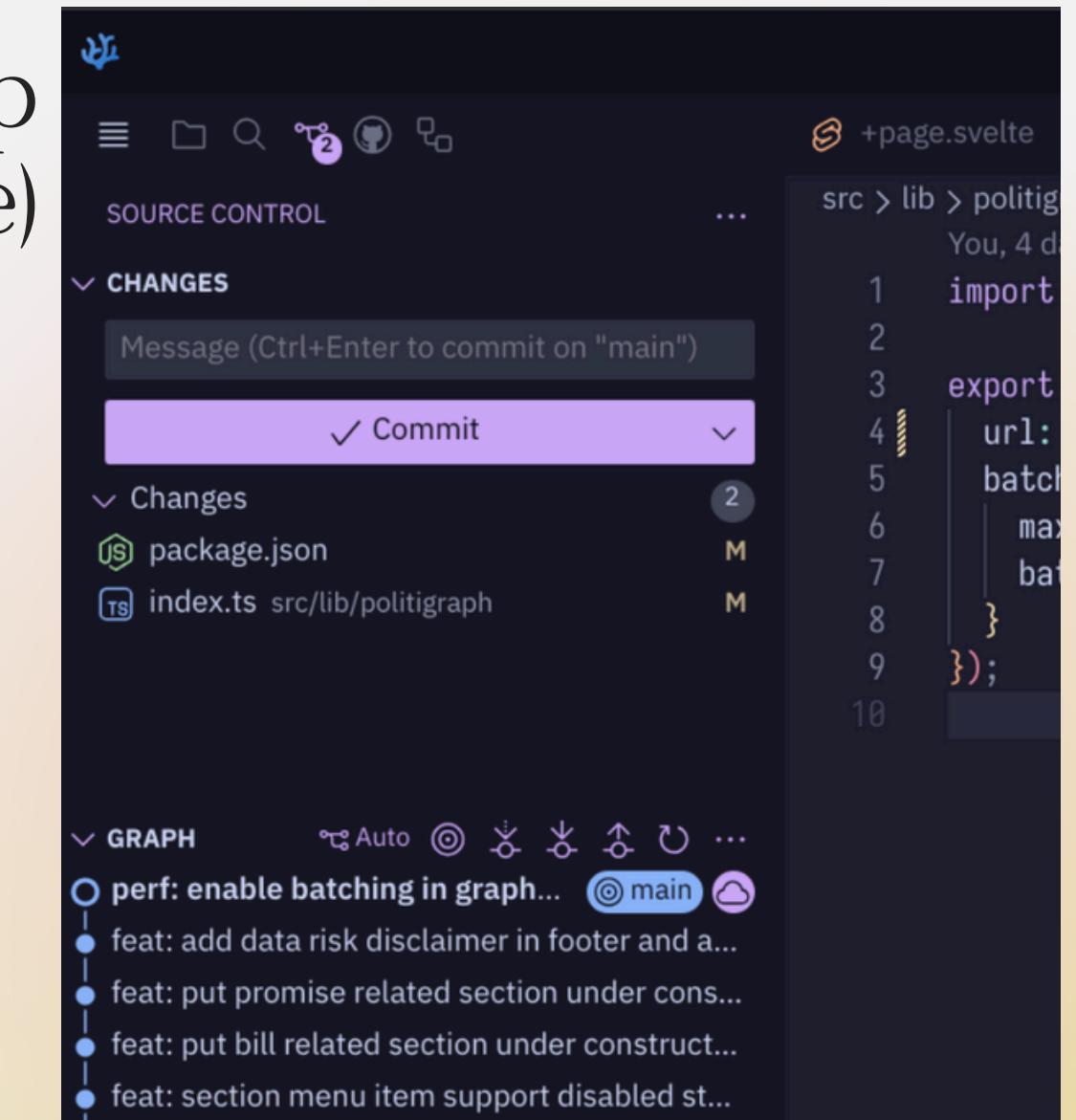
```
Welcome to fish, the friendly interactive shell
Type help for instructions on how to use fish

↳ ./Repositories/wevis/parliament-watch/
↳ .../parliament-watch ➔ ⌂ main $!
↳ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in the working directory)
    modified:   package.json
    modified:   src/lib/politigraph/index.ts

no changes added to commit (use "git add" and/or "git commit -m")
```

UI App
(eg. VSCode)



<https://education.github.com/git-cheat-sheet-education.pdf>

<https://code.visualstudio.com/docs/sourcecontrol/intro-to-git>

Config git user

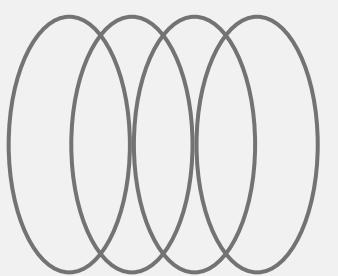


- ★ `git config --global user.email “your@email.com”`

- ★ `git config --global user.name “Your Name”`

*Should use the same email/name with your GitHub account

Hands On!



★ Write code

★ Stage

★ Commit

EXP: What to Commit?

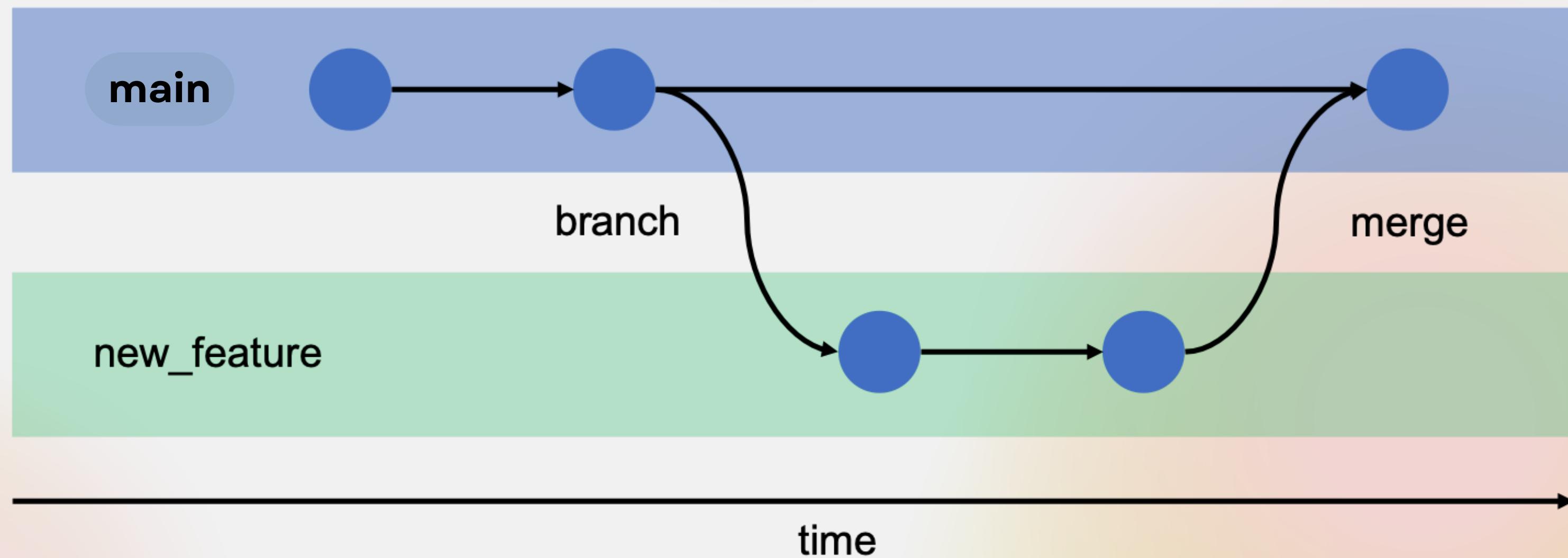
- o- Commits on Aug 20, 2025
 - refactor: use person name instead of first and last name field in vote event editor
 - Th1nkK1D committed on Aug 20 · ✓ 2 / 2
 - fix(schema): mark person name field as not null since firstname and lastname can't be null
 - Th1nkK1D committed on Aug 20
- o- Commits on Aug 18, 2025
 - docs: support node en field and improve typo
 - Th1nkK1D committed on Aug 18 · ✓ 2 / 2
 - feat(schema): add organization name_en field
 - Th1nkK1D committed on Aug 18
 - fix(schema): name custom resolver return null if every name part are not exist
 - Th1nkK1D committed on Aug 18
 - feat(schema): add option_en custom resolver field
 - Th1nkK1D committed on Aug 18

“Smallest set of working code
that can be described with a
simple sentence”

“Others should know what is this
comment about just by reading
commit message”

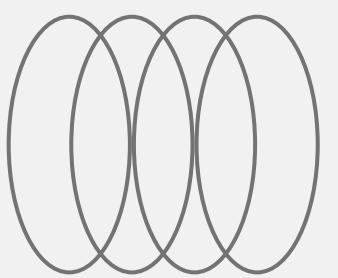
[EXTRA] Conventional Commit
<https://www.conventionalcommits.org>

Branch



<https://gitbookdown.dallasdatascience.com/branching-git-branch.html>

Hands On!



★ Create branch

★ Switch branch

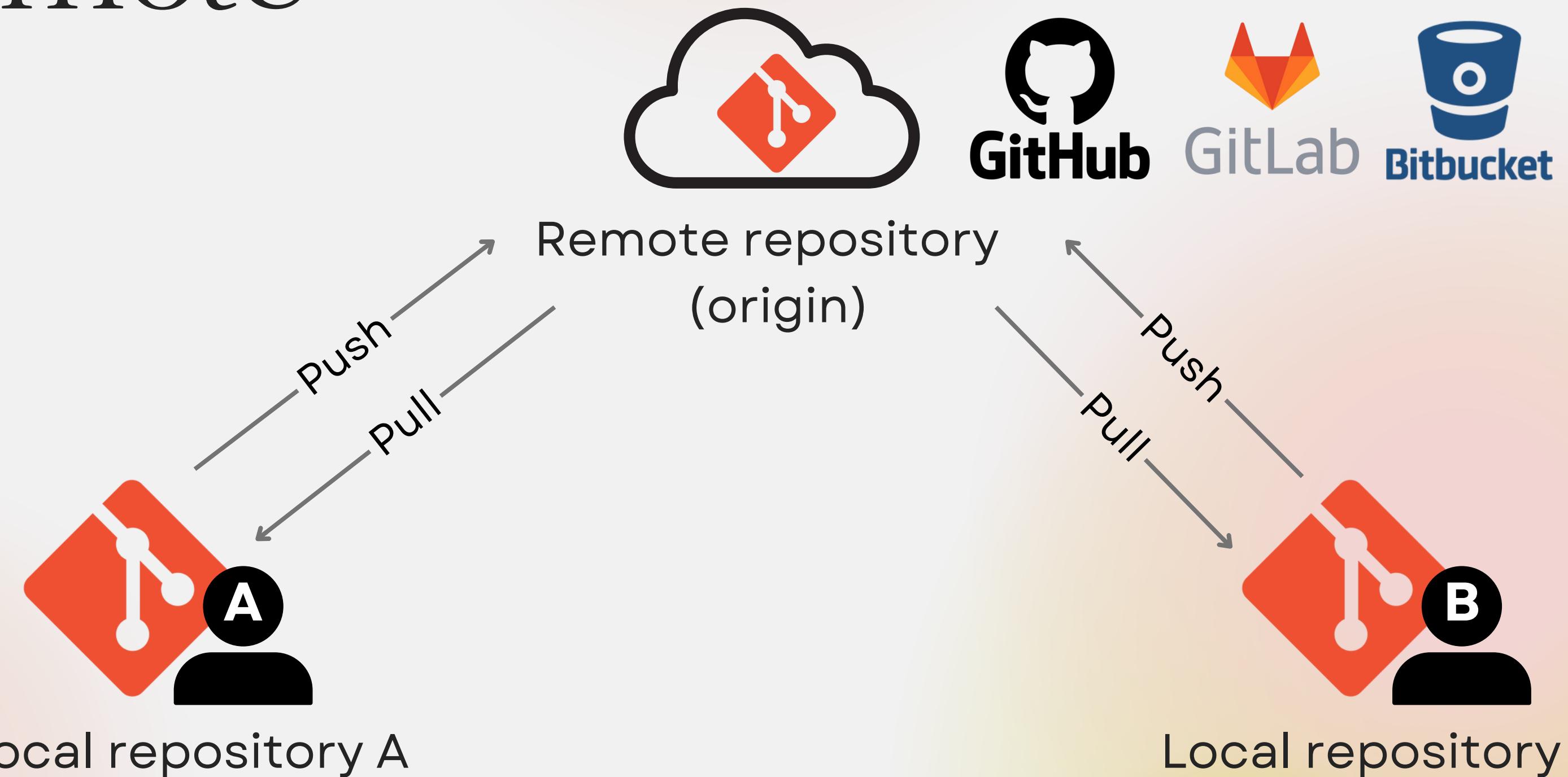
★ Merge

EXP: What to Branch?

- Experiment / WIP feature
- Group of commit you want to review before merging
- Your team stratgie (will talk about it later)



Remote



GitHub

2008 Launched
2018 Acquired by Microsoft

The screenshot shows the GitHub repository page for the 'react' repository, which was originally owned by 'facebook'. The repository has 810 issues, 233 pull requests, and 20,876 commits. It includes sections for Code, Issues, Pull requests, Actions, Models, Security, and Insights. The repository is public, has 6715 watchers, 49.3k forks, and 239k stars. It features 637 branches and 147 tags. The 'Code' tab is selected, showing a list of recent commits from various contributors, including EugeneChoi4, Update Code Sandbox CI to Node 20 to Match .nvmrc (#34474), move devtools notify to different channel (#34476), Playground applied configs (#34474), Release Activity in Canary (#34374), Update Flow to 0.265 (#34270), Fix some DevTools regression test actions and assertions..., Fix some DevTools regression test actions and assertions..., Remove trim_trailing_whitespace from editorconfig (#31...), Update Flow to 0.263 (#34269), [DevTools] Allow inspection before streaming has finishe..., Add run prettier commit to .git-blame-ignore-revs, .gitattributes to ensure LF line endings when we should, [forgive] Init (#31918), and updates mailmap entries (#19824). The repository is described as 'The library for web and native user interfaces.' and is associated with react.dev, react, javascript, library, ui, frontend, and declarative. It also links to Readme, MIT license, Code of conduct, Contributing, Security policy, Activity, Custom properties, 239k stars, 6.7k watching, 49.3k forks, and Report repository. There are 105 releases, with the latest being 19.1.1 (July 28, 2025).

Repository Visibility

Public

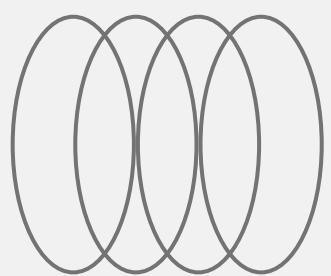
- Open-source
- Support external contributor
- Get many free features

Private

- Close-source
- Only internal team
- Some features need to be paid

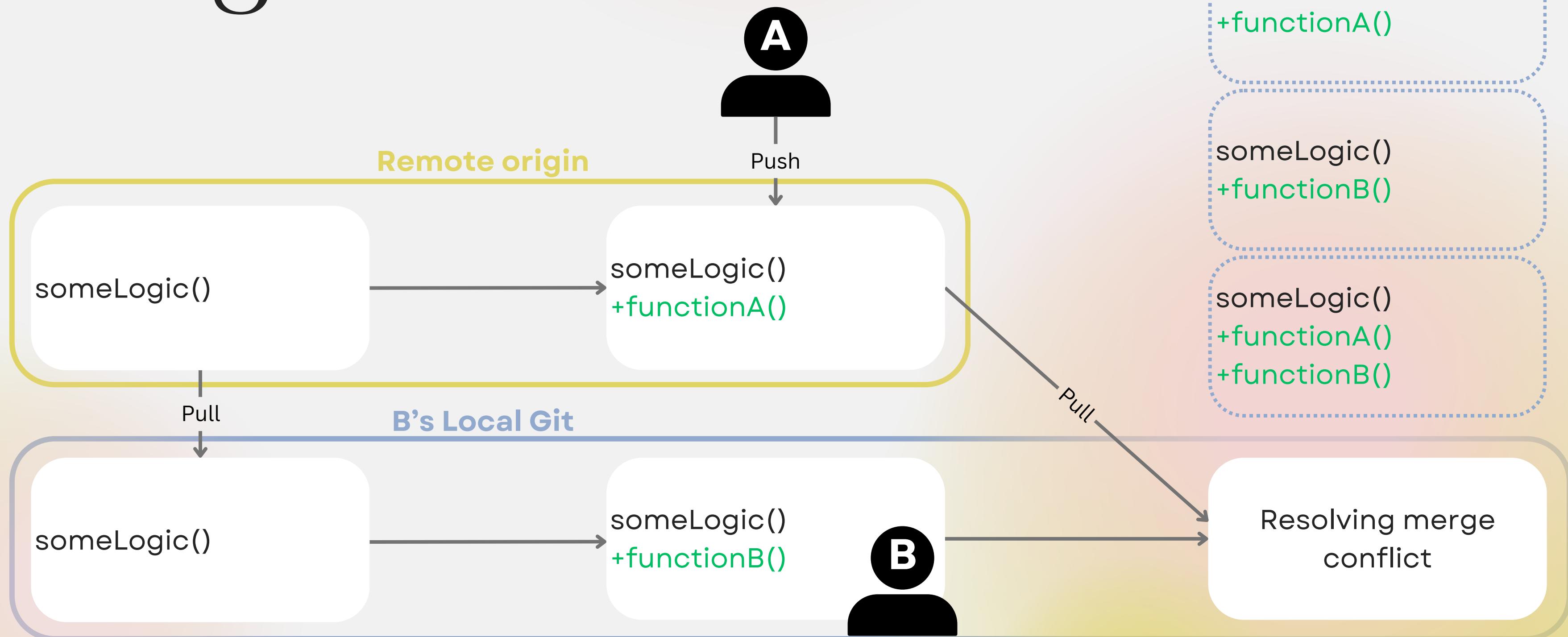
Hands On!

(in pair)



- ★ Set up remote repository
- ★ Invite team member
- ★ Clone, push, pull

Merge Conflict!



CI/CD ?

CI: Continuous Integration (n.)

A practice of encouraging development teams submit code changes to the remote repository more frequently

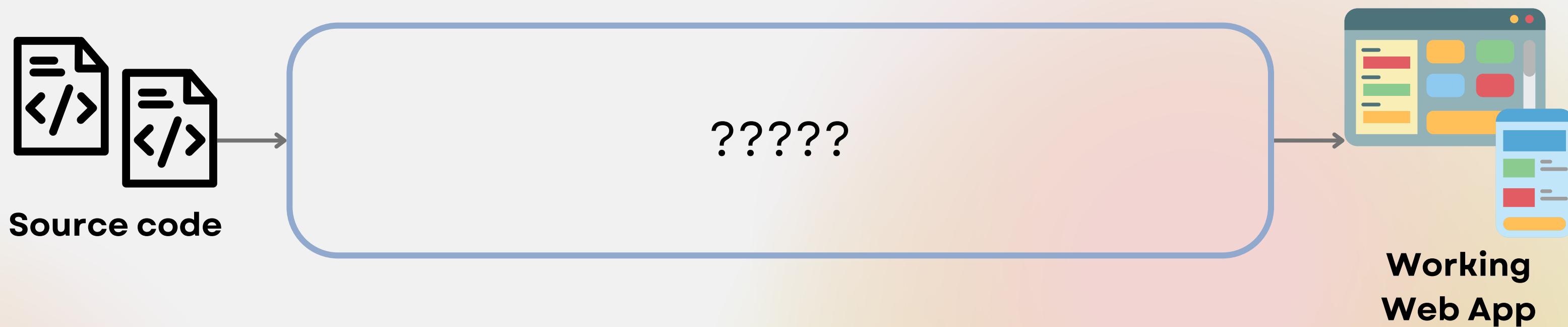
→ *Less merge conflict, everyone are more on the same page*

CD: Continuous Delivery (n.)

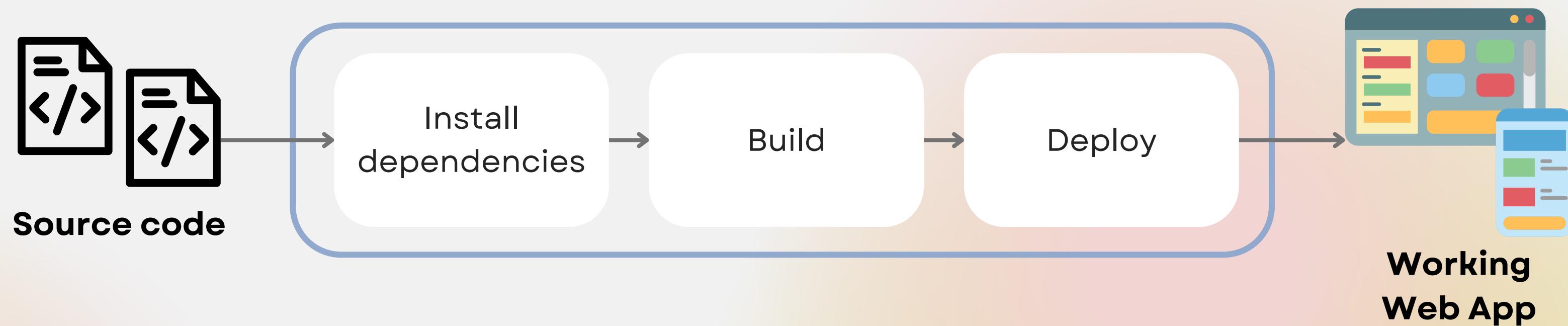
A practice of making an update to a live production environment as soon as the source code in remote repository has changed

→ *Internal team / users can see the changes as soon as possible*

What we need to do?

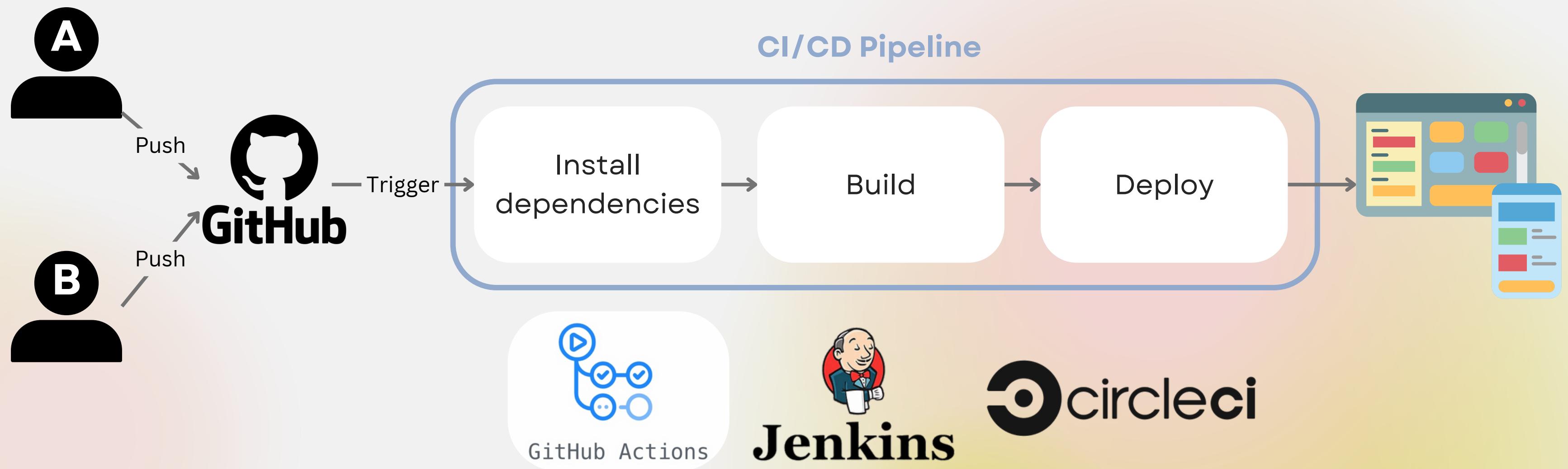


What we need to do?

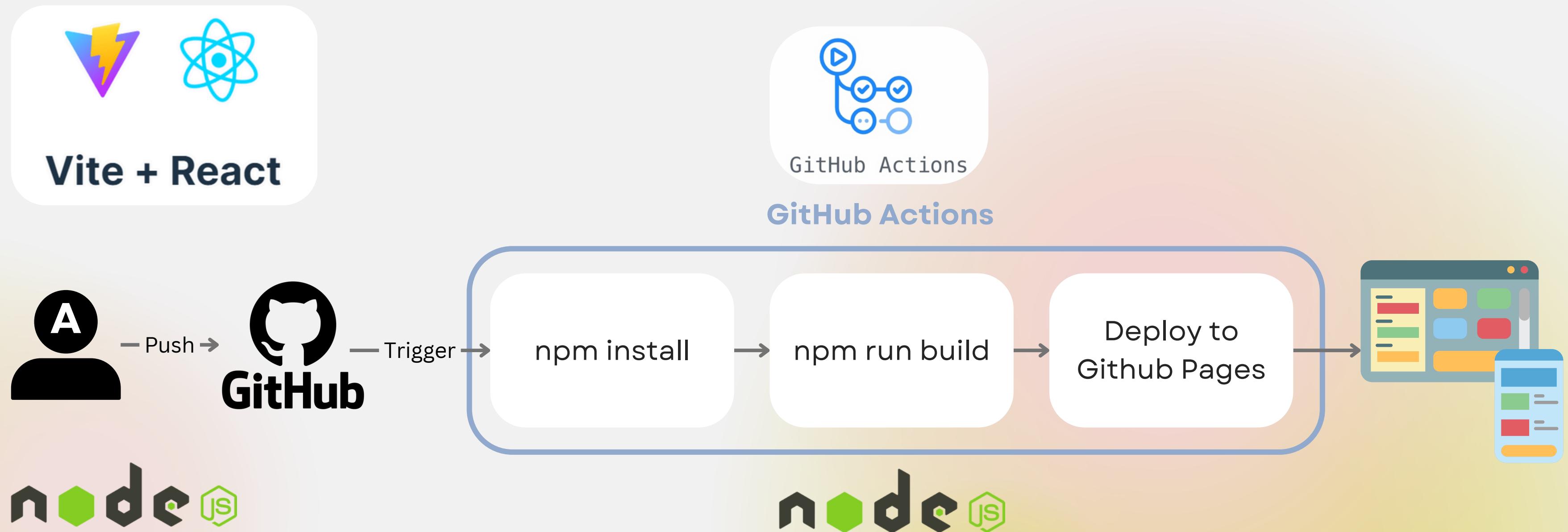


CI/CD Pipeline (n.)

An automation workflow, consist of several steps, to achieve CI/CD practice.



Hands on: Vite+React project



Set up Vite+React

- ❖ Open terminal on VSCode
- ❖ npm create vite@latest

```
> npx
> create-vite
|
| Project name: .
|
| Select a framework: React
|
| Select a variant: JavaScript
|
| Scaffolding project in /home/lkz/Repos
|
| Done. Now run:
  npm install
  npm run dev
```

GitHub Pages

❖ Commit and push to GitHub

❖ Enable in repository setting

The screenshot shows the GitHub repository settings page for a repository named "GitHub Pages". The top navigation bar includes links for Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings, with Settings being the active tab.

The left sidebar lists repository settings categories: General, Access, Collaborators, Moderation options, Code and automation, Branches, Tags, Rules, Actions, Models, Webhooks, Copilot, Environments, Codespaces, and Pages. The Pages category is highlighted with a blue bar at the bottom.

The main content area is titled "GitHub Pages" and contains the following text: "GitHub Pages is designed to host your personal, organizational, or company websites." Below this, there are two sections: "Build and deployment" and "Source".

The "Source" section is currently set to "GitHub Actions". A callout box highlights the "GitHub Actions" option, which is described as "Best for using frameworks and customizing your build process". Below this, there is a "Deploy from a branch" section with a "Configure" button and a note about dependencies being preinstalled.

At the bottom of the main content area, it says: "Workflow details will appear here once your site has been built." There is also a "Custom domain" section.

GitHub Actions

❖ <https://tinyurl.com/viteaction>

❖ .github/workflows/deploy.yml

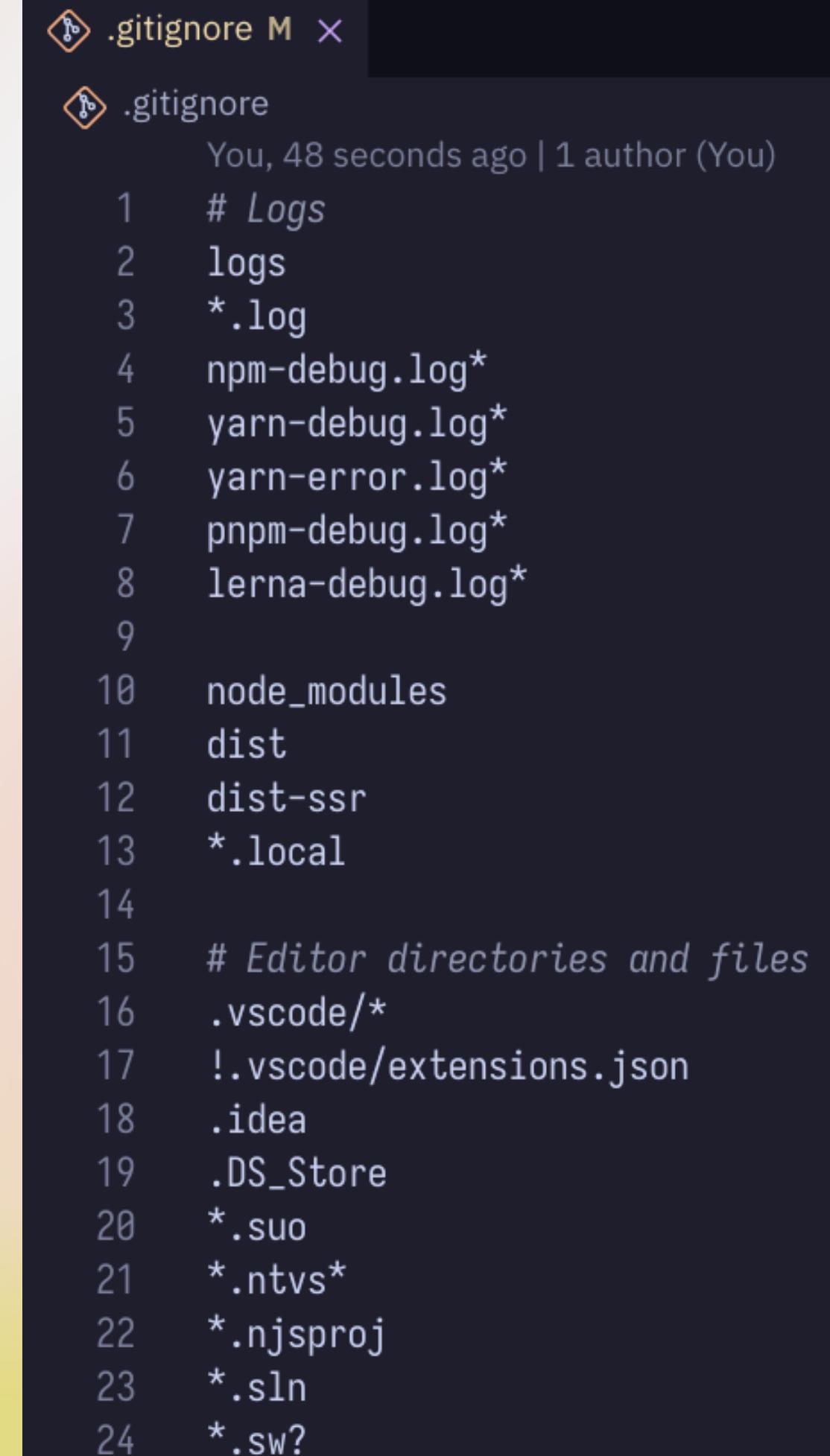
```
1  name: Deploy to GitHub Pages
2
3  on:
4    push:
5      branches: ['main']
6
7  # Sets the GITHUB_TOKEN permissions for GitHub Pages
8  permissions:
9    contents: read
10   pages: write
11   id-token: write
12
13 jobs:
14   build-and-deploy:
15     runs-on: ubuntu-latest
16     steps:
17       - name: Checkout
18         uses: actions/checkout@v5
19       - name: Set up Node
20         uses: actions/setup-node@v4
21         with:
22           node-version: 22
23           cache: 'npm'
24       - name: Install dependencies
25         run: npm install
26       - name: Build
27         run: npm run build
28       - name: Setup Pages
29         uses: actions/configure-pages@v5
30       - name: Upload artifact
31         uses: actions/upload-pages-artifact@v4
32         with:
33           path: './dist'
34       - name: Deploy to GitHub Pages
35         id: deployment
36         uses: actions/deploy-pages@v4
37
```

.gitignore

What should be ignored by git

- Logs
- External dependencies (node_modules)
- Build output (dist)
- Environment variable and secrets (.env)
- etc.

*“No redundant / sensitive information
should be kept in git”*



```
 .gitignore M ×  
 .gitignore  
 You, 48 seconds ago | 1 author (You)  
 1 # Logs  
 2 logs  
 3 *.log  
 4 npm-debug.log*  
 5 yarn-debug.log*  
 6 yarn-error.log*  
 7 pnpm-debug.log*  
 8 lerna-debug.log*  
 9  
10 node_modules  
11 dist  
12 dist-ssr  
13 *.local  
14  
15 # Editor directories and files  
16 .vscode/*  
17 !.vscode/extensions.json  
18 .idea  
19 .DS_Store  
20 *.suo  
21 *.ntvs*  
22 *.njsproj  
23 *.sln  
24 *.sw?
```

Env. Variables and Secrets

- .env file with Vite
 - <https://vite.dev/guide/env-and-mode>
- GitHub Actions' Secrets
 - <https://docs.github.com/en/actions/how-tos/write-workflows/choose-what-workflows-do/use-secrets>

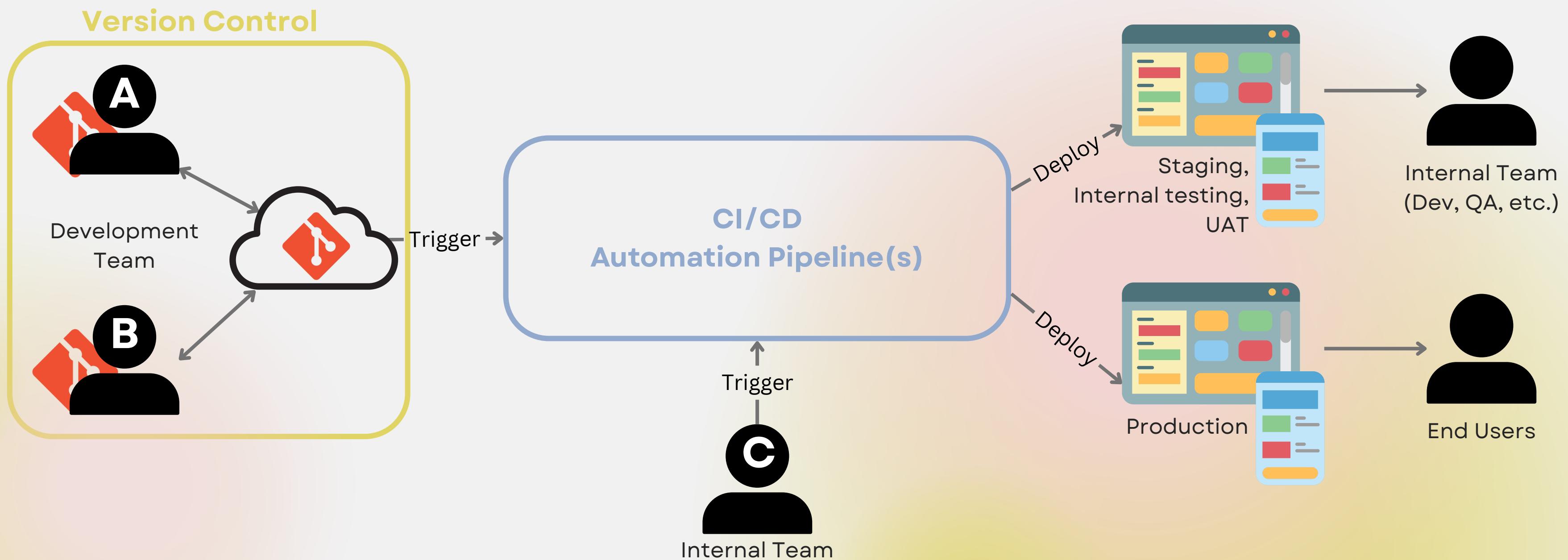
The image shows a code editor interface with three tabs open:

- .gitignore**: Contains two entries: ".gitignore" and ".env".
- .env**: Contains one entry: "SOME_SECRET=liadsgyfilYAG9173T91".
- deploy.yml**: A GitHub Action configuration file with the following content:

```
jobs:
  build-and-deploy:
    steps:
      - name: Build
        run: npm run build
        env:
          SOME_SECRET: ${{ secrets.SOME_SECRET }}
      - name: Setup Pages
        uses: actions/configure-pages@v5
      - name: Upload artifact
```

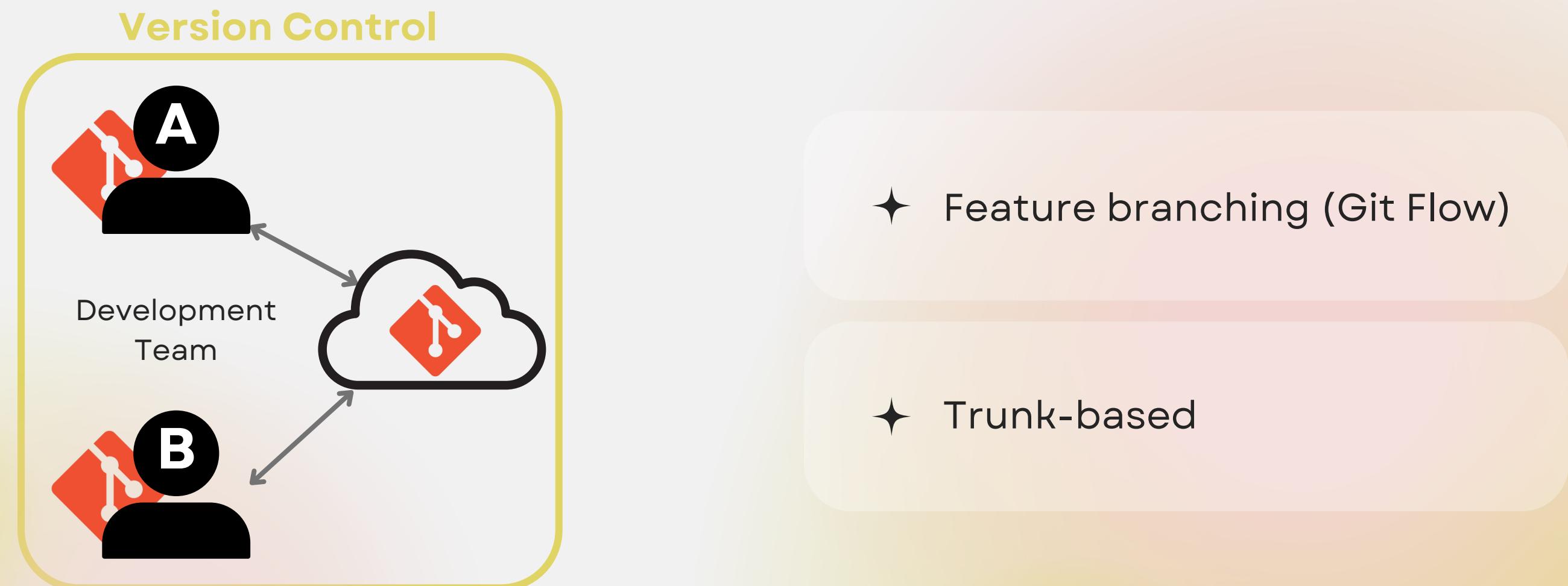
The Development Workflow

We do it for the agility → Fast feedback loop → Agile!



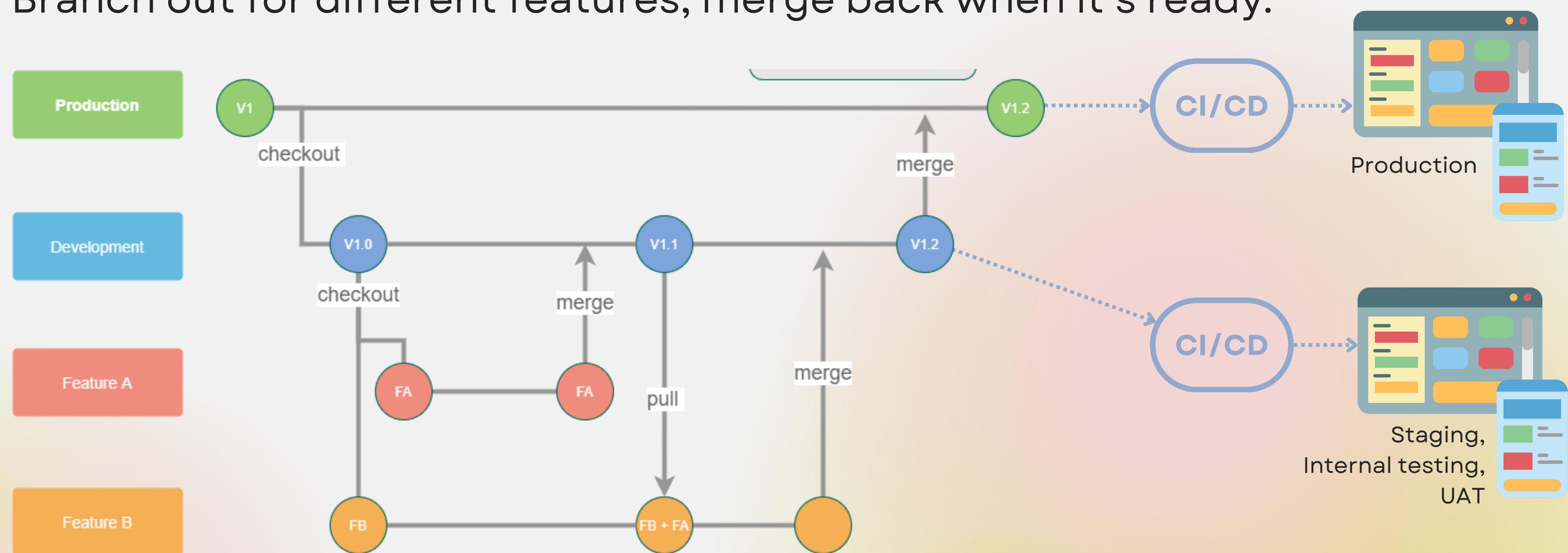
Branching Strategies

A set of rules or guidelines that defines how developers interact with a shared codebase [1] focusing on git branch.



Feature Branching

Branch out for different features, merge back when it's ready.



Feature Branching

Branch out for different features, merge back when it's ready.

Pros

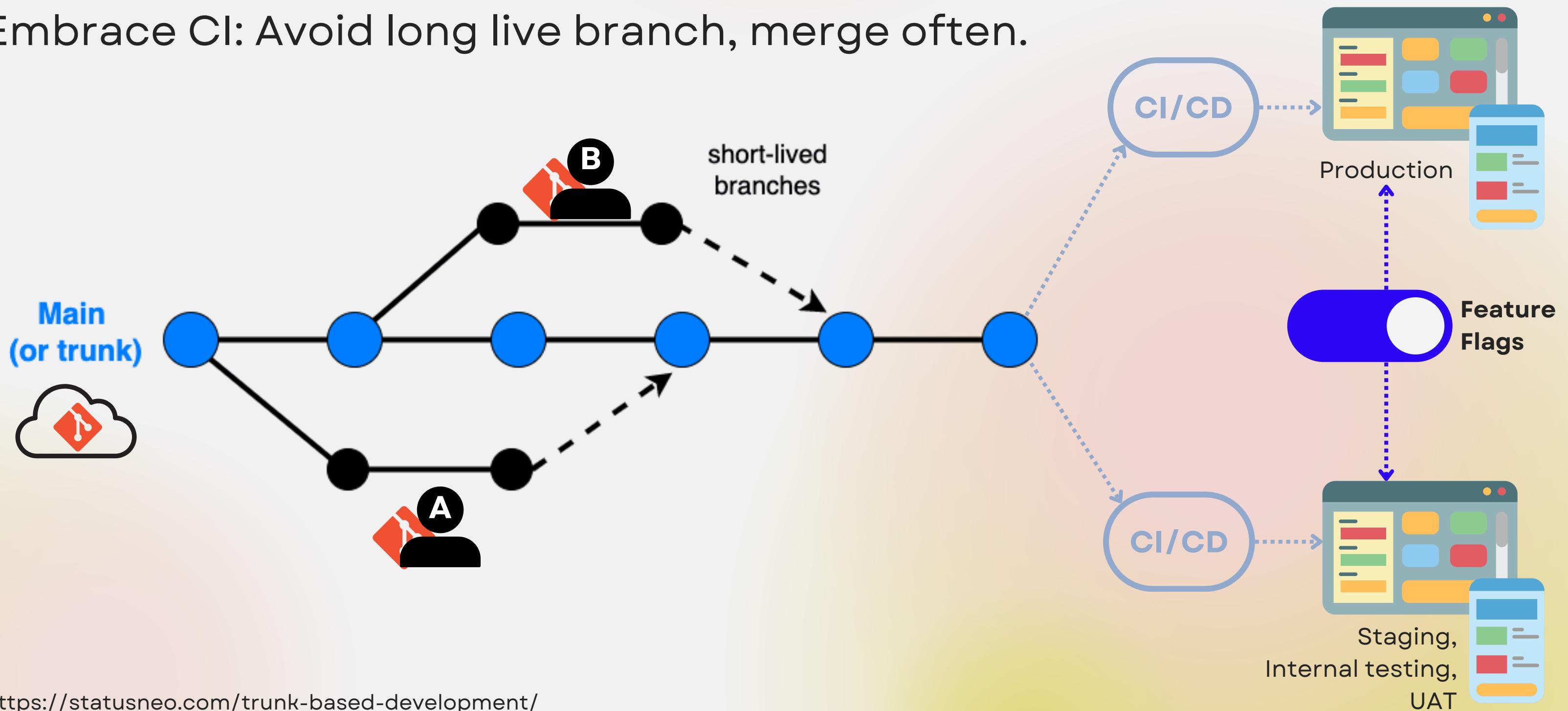
- Easy to control the environments / release
- Easy to review code

Cons

- Not so Continues Integration

Trunk-Based

Embrace CI: Avoid long live branch, merge often.



Trunk-based

Embrace CI: Avoid long live branch, merge often.

Pros

- Great Continues Integration

Cons

- Require effort on feature flags
- Code review is harder

Mixed Strategy?

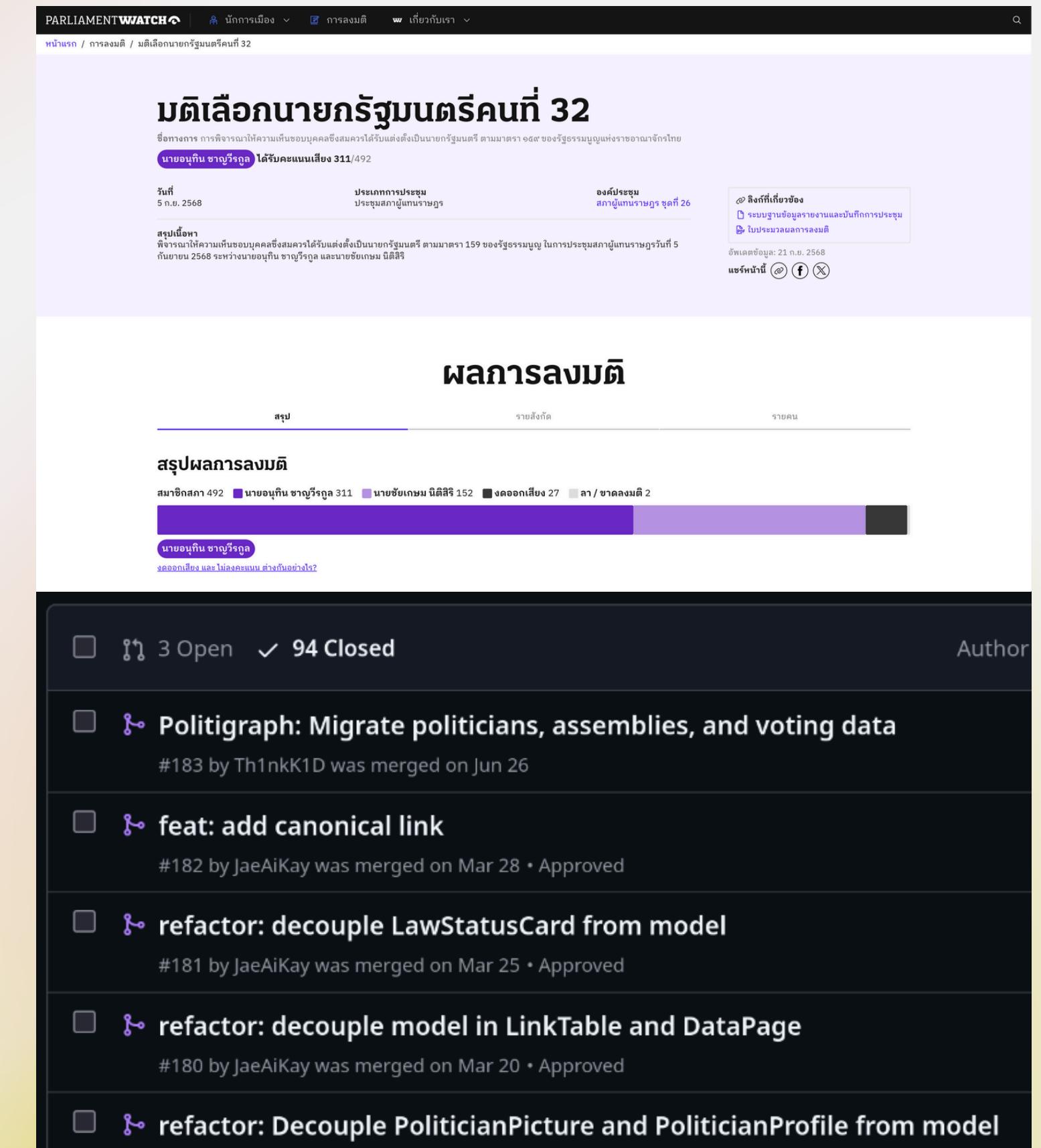
WeVis's Parliament Watch's strategy:

1. Trunk-based for internal team

- We want continues integration

2. Feature branching for external contributor

- We want to review codes from external contributor



Software Engineering

=

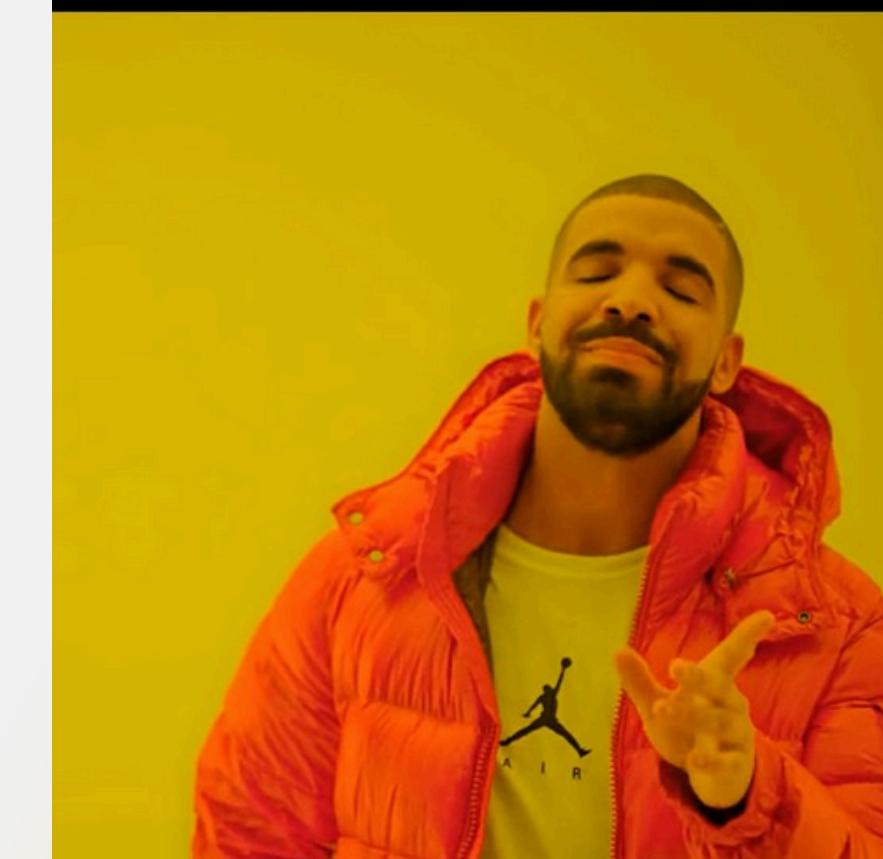
Understanding options

+

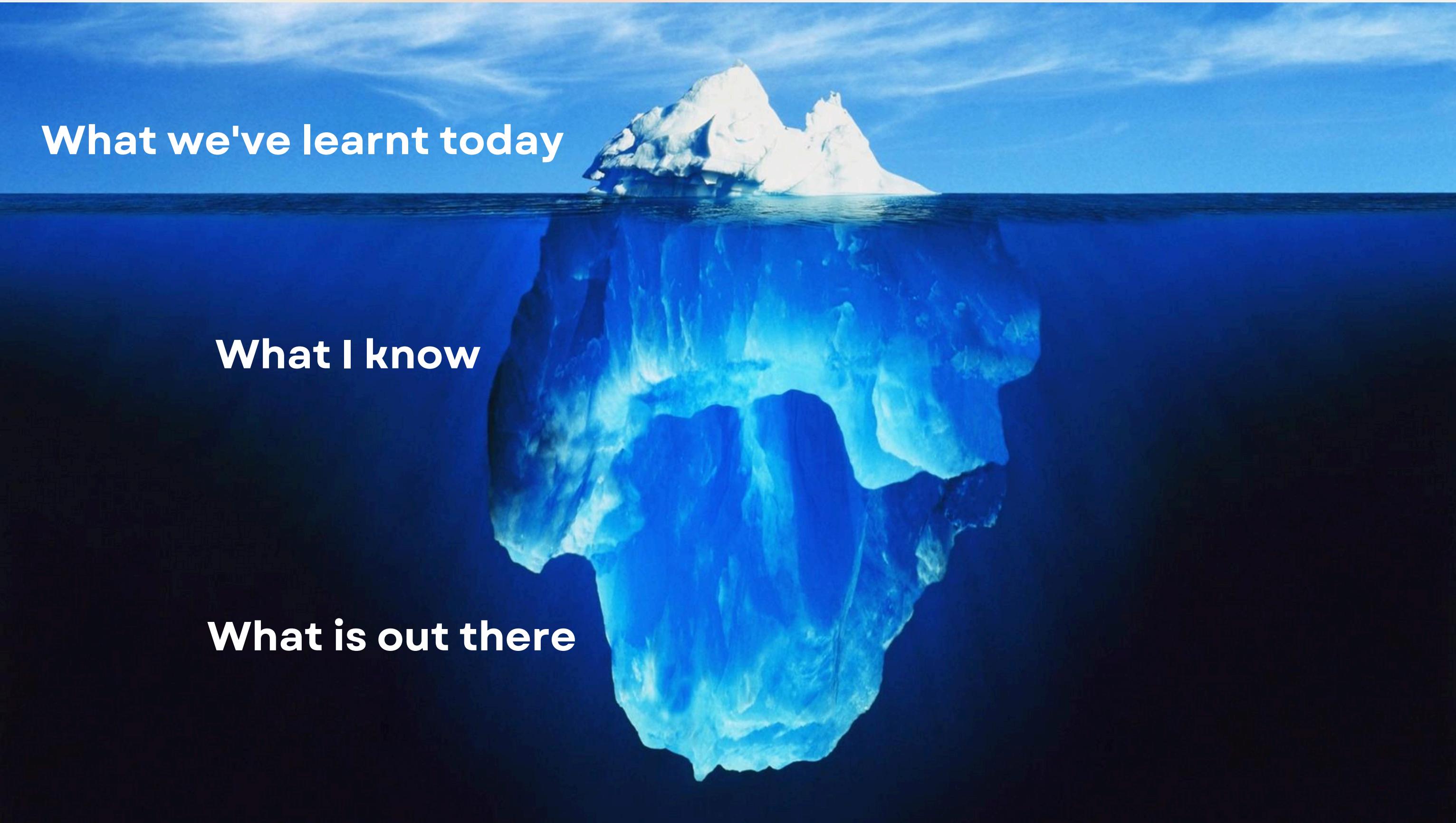
adapting to your context



Best practice



It depends



Thank You

(FB) Lookkid Withee Poositasai
(Email) witheep@gmail.com
(GitHub) Th1nkK1D

Wanna clean up repo on GitHub? Go to settings:

