Git for data professionals

SQLbits

Els van Vessem - 18th of June 2025

Mentimeter

Els van Vessem

they/them

Data engineer + Scrum Master

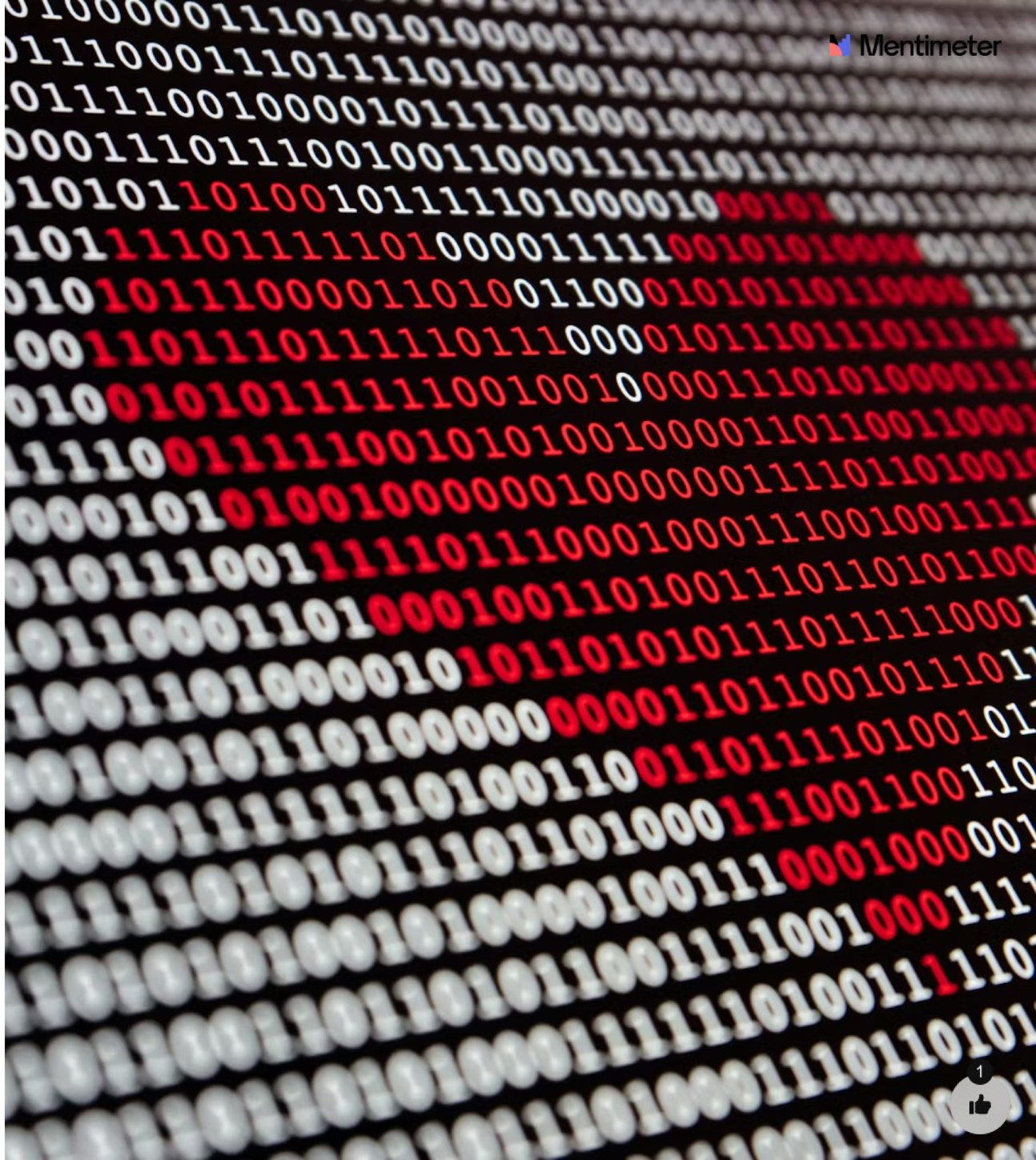
@geekingthething.bsky.social

elsvanvessem@gmail.com

ifthenels.com

Tweede Kamer der Staten-Generaal

https://opendata.tweedekamer.nl/



Mentimeter

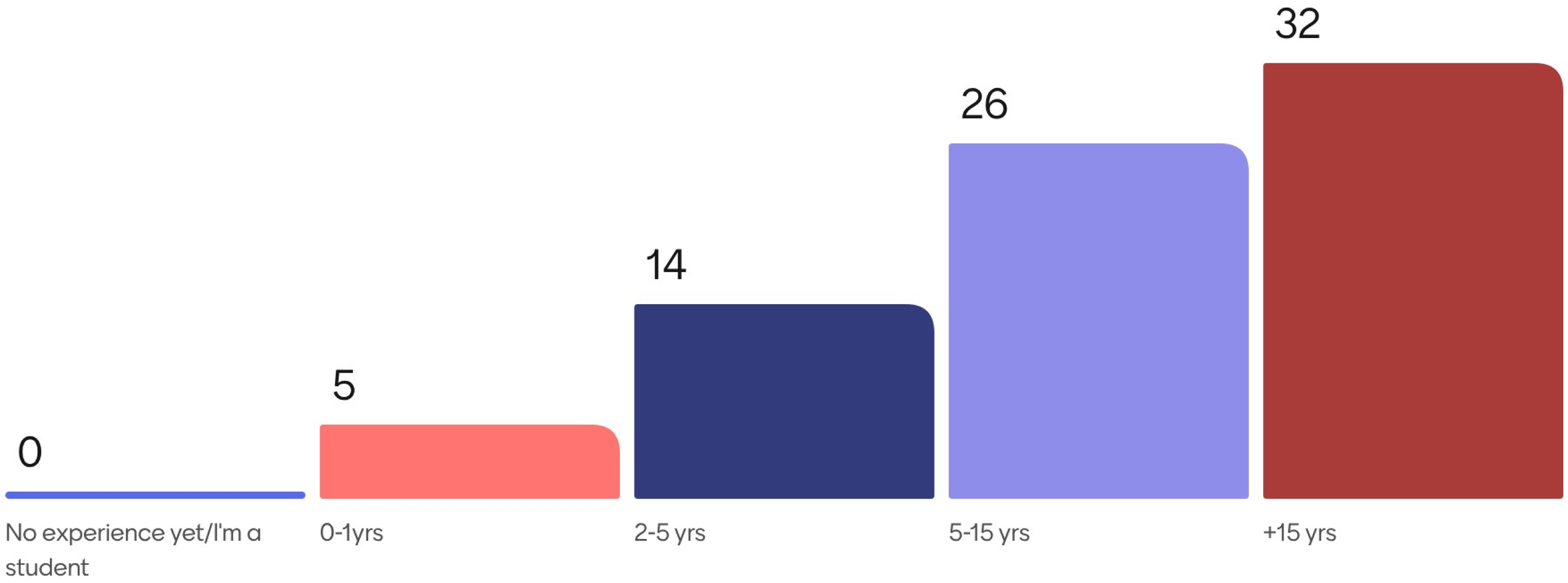
Mentimeter

Instructions

54

Mentimeter

How long have you been working in data?



11

77

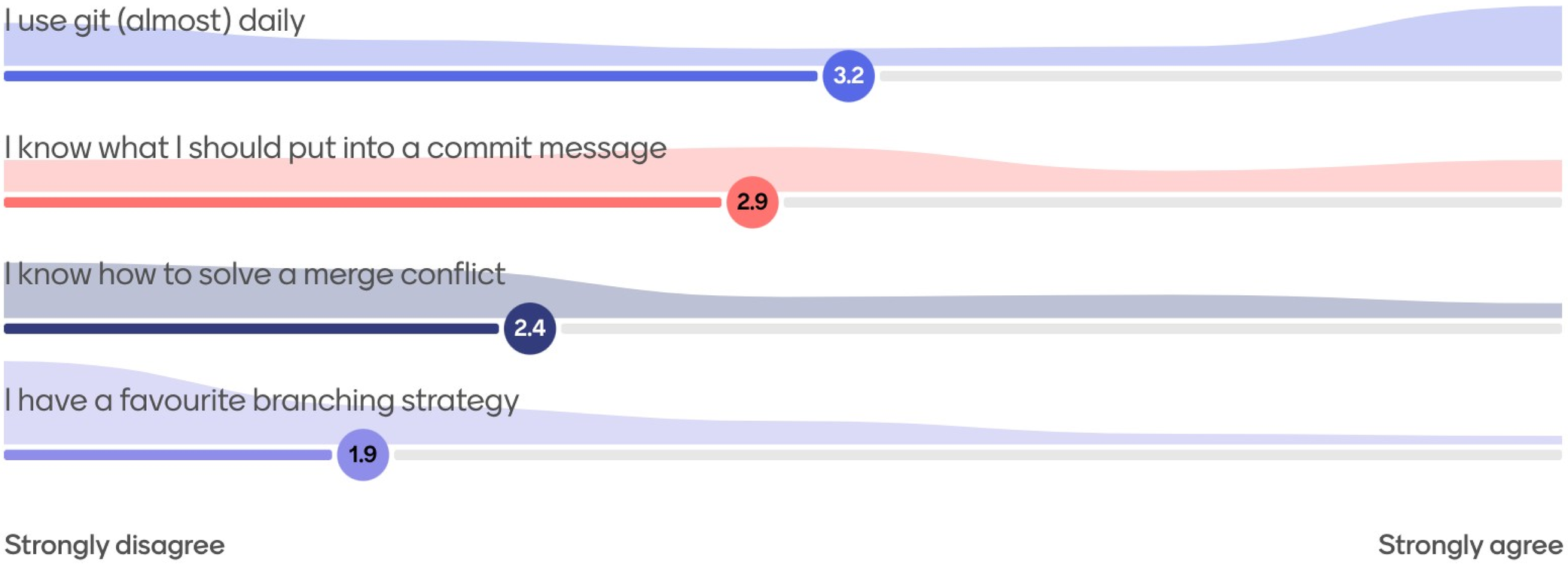
Mentimeter

What do you think about when you hear the term "git"?

5 72

Mentimeter

What's your experience with git?



15

77

Plan of the session

- Why version control?
- How to set it up
- The basics: clone, commit, push, pull
- Break (I think? timing is hard!)
- Quiz time!
- More on commits
- Branching
- Merge conflicts
- If everything fails...

Why would we want git/version control?

- Parallel development
- Debugging
- Control
- Keep changes organised and revertable
- Prevent model failures
- Rollbacks
- Time travelling
- Safety

4

62

Mentimeter

Why would we want git/version control?

Minimise issues

debugging

Safety

Team collaboration

Keep an audit trail

Control

Code History

Undo

4

62

Mentimeter

Why would we want git/version control?

Safety

Version control and collaboration

Quality

Proper release strategy

Safety

Auditing

Catalogue Changes

No duplicate effort

4

62

Mentimeter

Why would we want git/version control?

Rollbacks

CI/CD automation

Change history

Effective Teamwork

A safe place

Possibility to go back to working version

control

Safety Safety

4

62

Why would we want git/version control?

Track all changes made in our team code

Control

History

Automate production rollout

Version control

Tracking

Change history

Consistency

4

62

Mentimeter

Why would we want git/version control?

Deployment

Change history

To control stupid decisions because that's what Linus thought to when he named it git

History

Release versioning

Protect code

Version control

Collaboration and version control

4

62

Mentimeter

Why would we want git/version control?

Because my colleagues have 20 versions of the same PBI report on my disks

To make sure we work on right versions

Team collaboration

Joint working

History

Joint development

Ability to role back, storage of code, history and logging

Keep previous versions

4

62

Mentimeter

Why would we want git/version control?

Collaboration

History

Backup

To secure our codebase

History

Control history

Because we make big mistakes

Avoid problems

4

62

Mentimeter

Why would we want git/version control?

Code history

Code history

Risk management

Code review

CI/CD

Control

Version control and collaboration

Release control

4

62

Mentimeter

Why would we want git/version control?

Who did what(no blaming)

Blame

Traceability

Team working

Know who messed up...

Blame

Reduce heart attack

Because humans

4

62

Mentimeter

Why would we want git/version control?

Roll back to previos working version

Review the idiots code

Peer review

Answer you changed something and why

To know who wrote this code

App and data code in the same place

Confidence of what is in a branch is in line with our environments

'Documentation'

4

62

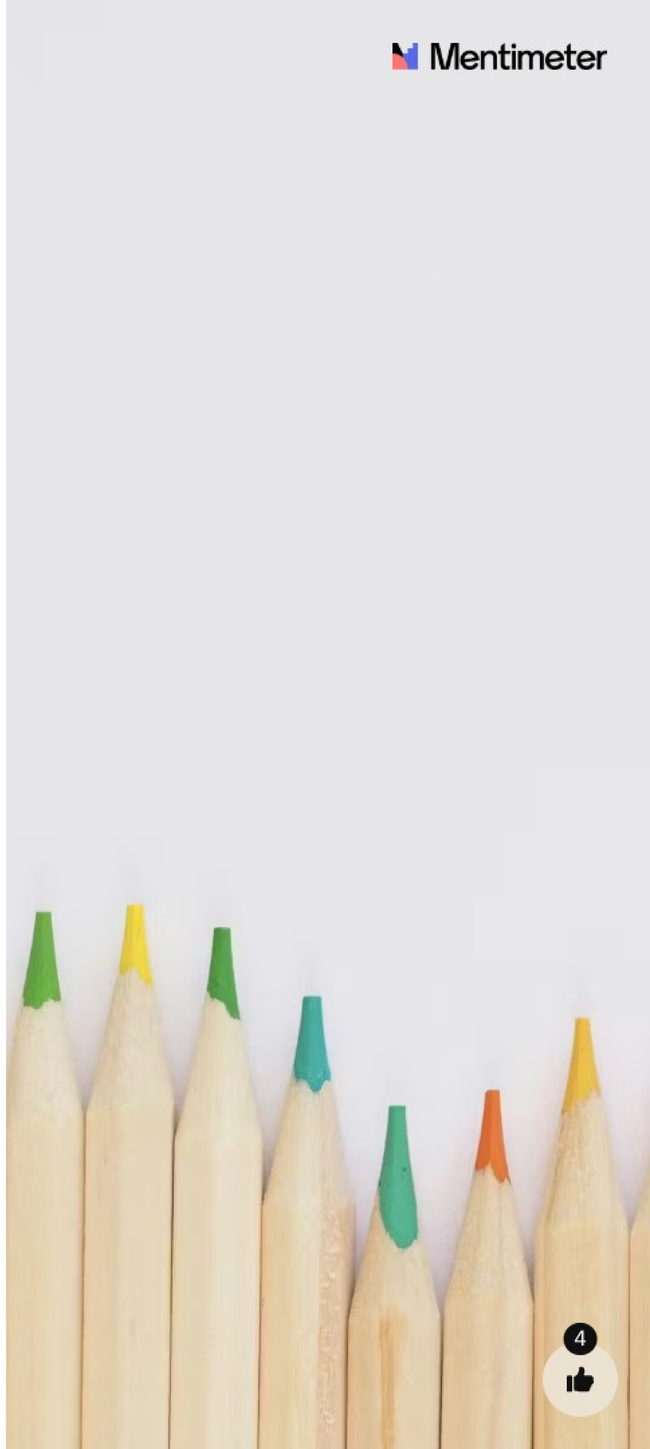
The power of version control systems like Git, lies in merging many code changes from multiple developers smoothly, while documenting the entire timeline and context .

Rob Sewell: saturday 12:20, room 8

4

Which version is which?

- final_query.sql
- now_really_final.sql
- now_really_really_final.sql
- final_query2.sql
- new_final_query2.sql



Mentimeter

Working on the same code: issues

- Endless emailing of queries
- Different versions with different people
- Working on the same code and creating a similar solution
- Continuing on an old version

4

Working on the same code: merging

- Two, or more, separate scripts/codebases
- The longer the code stays with separate from other people, the more changes there will be
- And how much harder it will be to merge the scripts

How did this script develop?

1. ✅select * from dbo.orders
2. ✅SELECT * FROM dbo.orders
3. ✅SELECT orderid, customer, company, date FROM dbo.orders
4. ✅SELECT orderid, orderdetailid, customer, company, date, product FROM dbo.orders o
   LEFT JOIN dbo.orderdetails od ON o.orderid = od.orderid
5. ❌SELECT orderid, orderdetailid, customer, company, date, product FROM dbo.orderdetails od

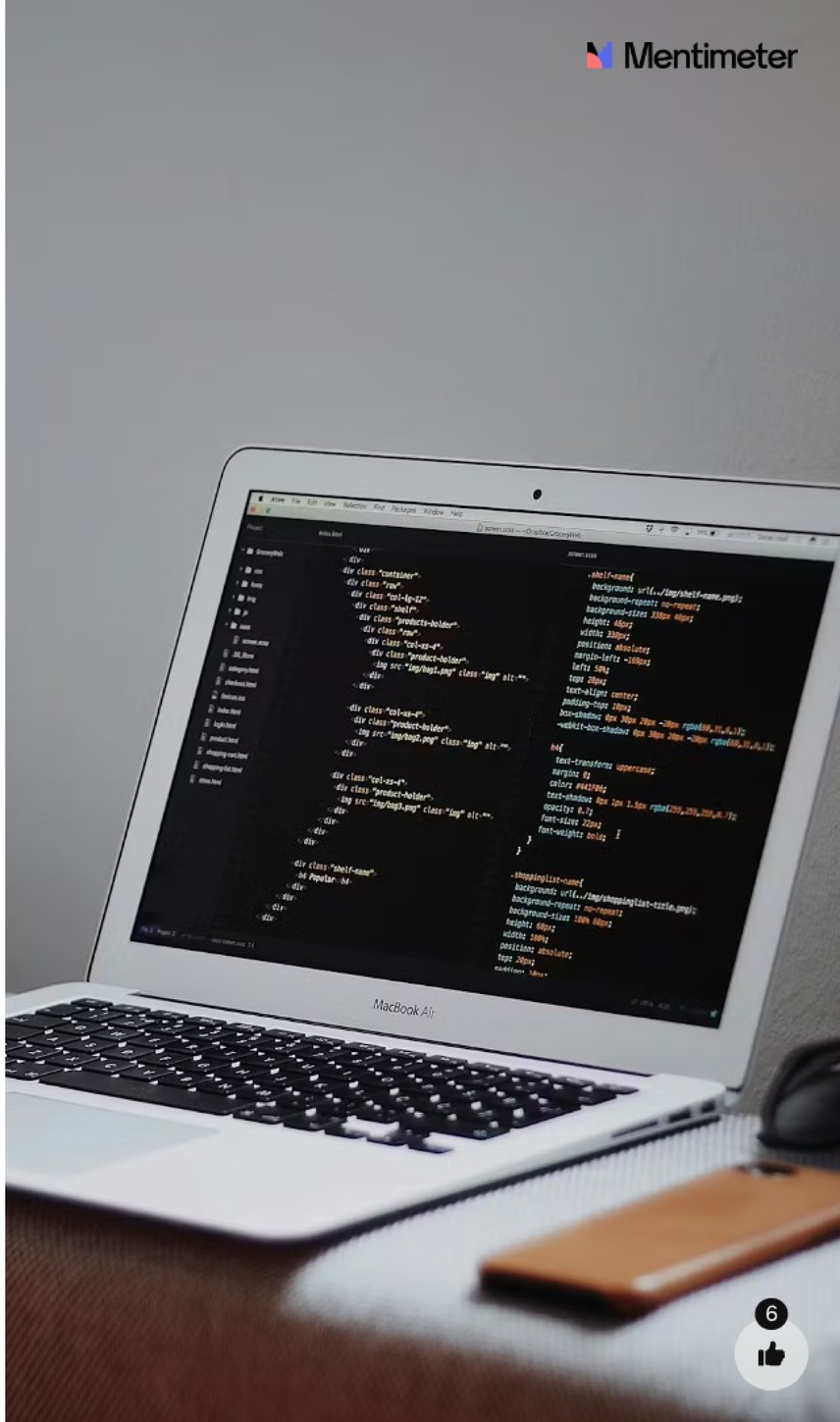
3

The power of **version control** systems like Git, lies in **merging many code changes** from **multiple developers** smoothly, while documenting the **entire timeline** and **context** .

Rob Sewell: saturday 12:20, room 8

1

Demo time!



Summary: what do you need to get started?

- install git on your machine
  - https://git-scm.com/downloads
  - don't just "next next finish!" Read the options
- get a place for your repositories
  - github, gitlab, azure devops repos, etc etc etc.
- get an application to write your code
  - I recommend Visual Studio Code

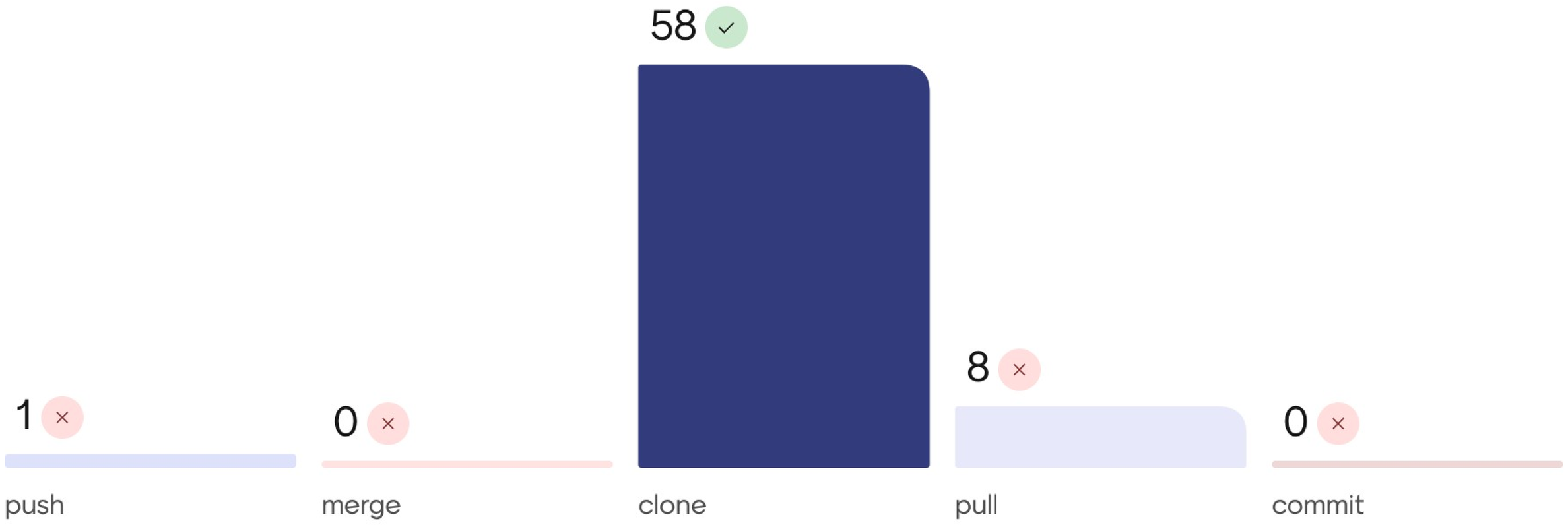
3

Summary: what do you need to get started?

- Set up your identity:
- open "git bash" and use the following commands
  - git config --global user.name "SQL bits
  - git config --global user.email sqlbits@sqlbit.com
  - git config --global credential.helper store (optional)
- Check your settings with
  - git config --global --list

Mentimeter

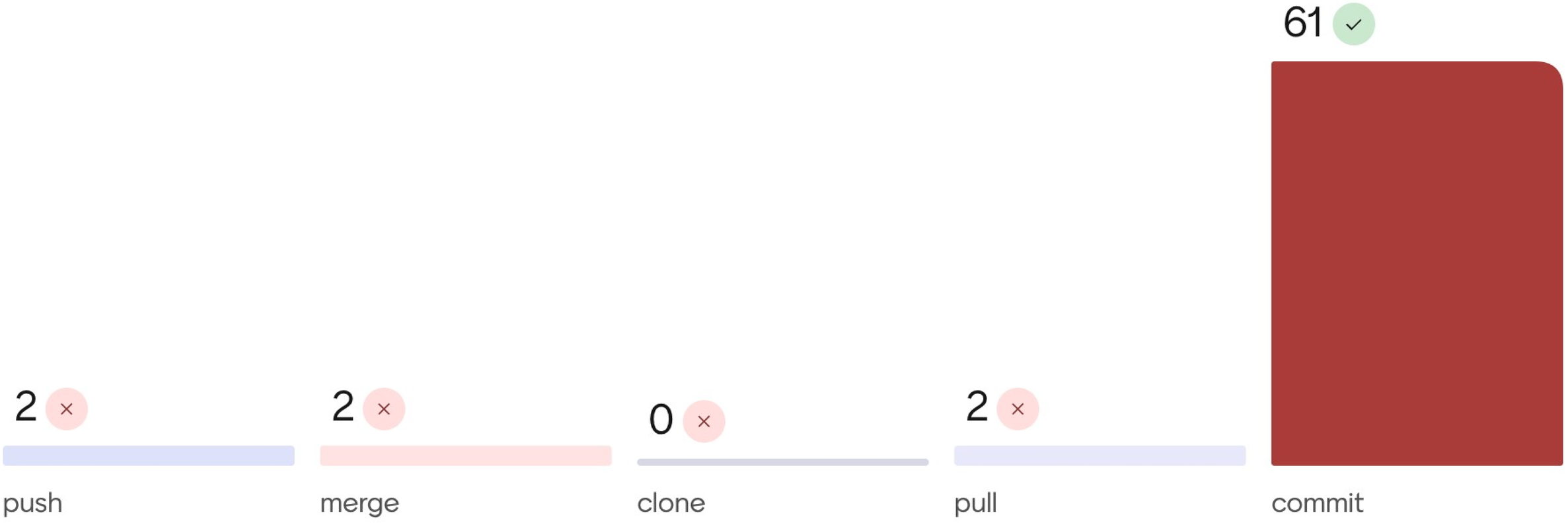
Getting a personal/local copy from the main codebase (for the first time)



33

Mentimeter

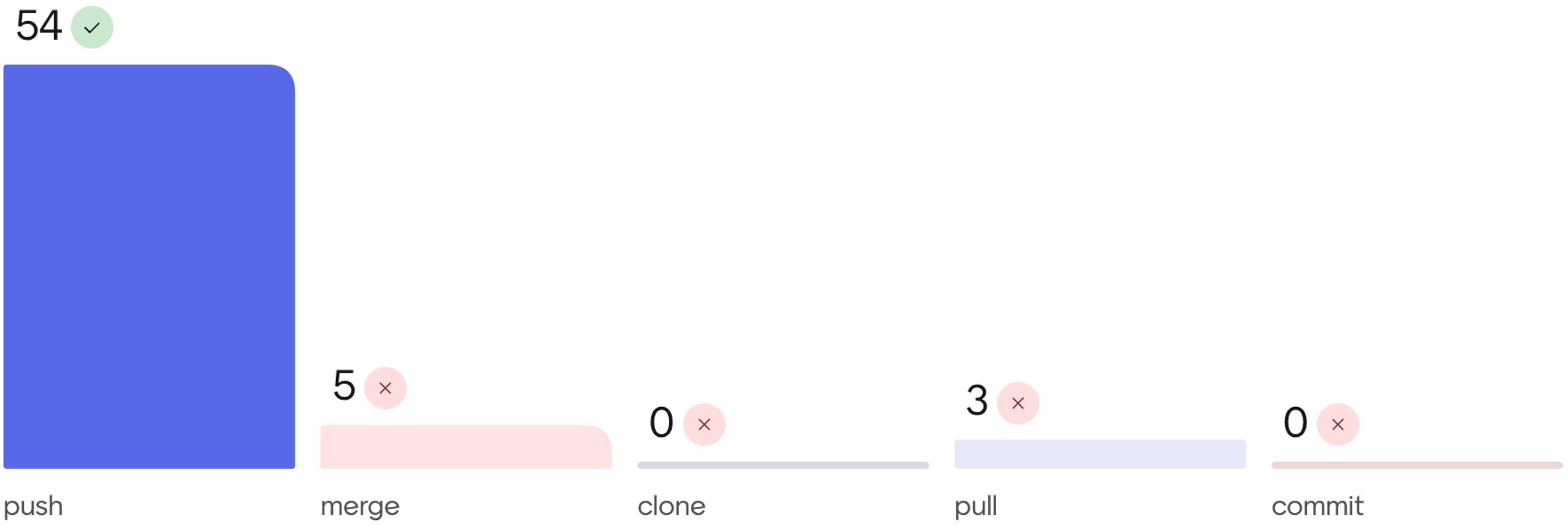
"Saving" your changes in your local copy



24

Mentimeter

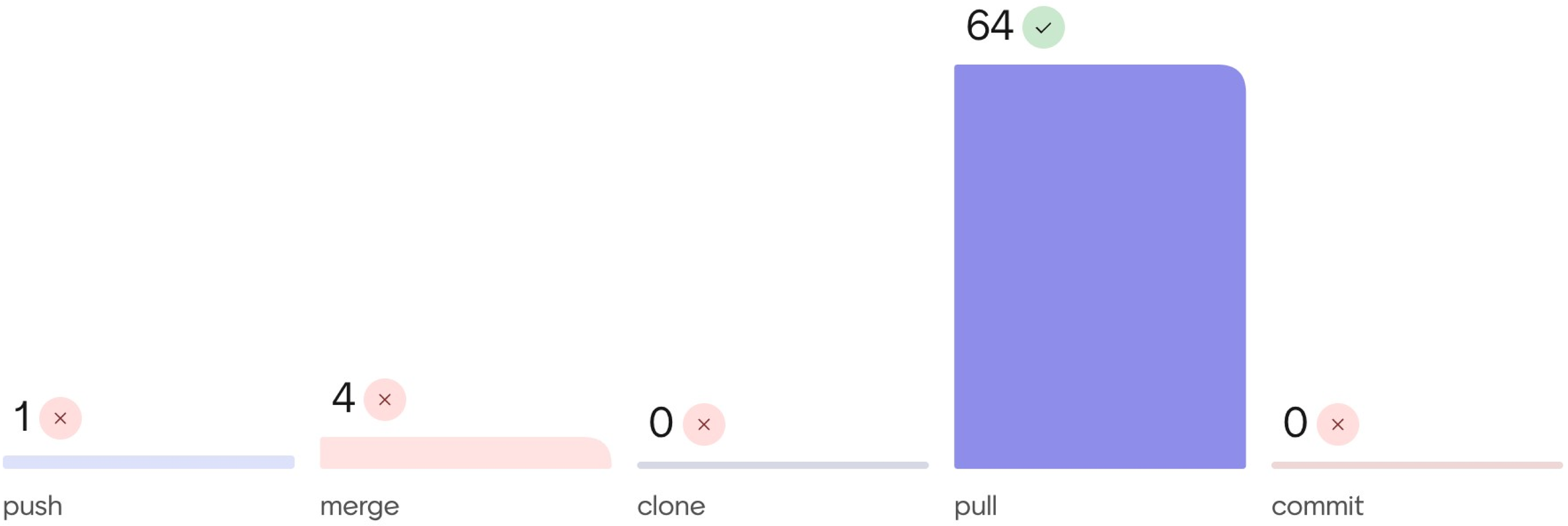
Adding your local commits to the main codebase



17

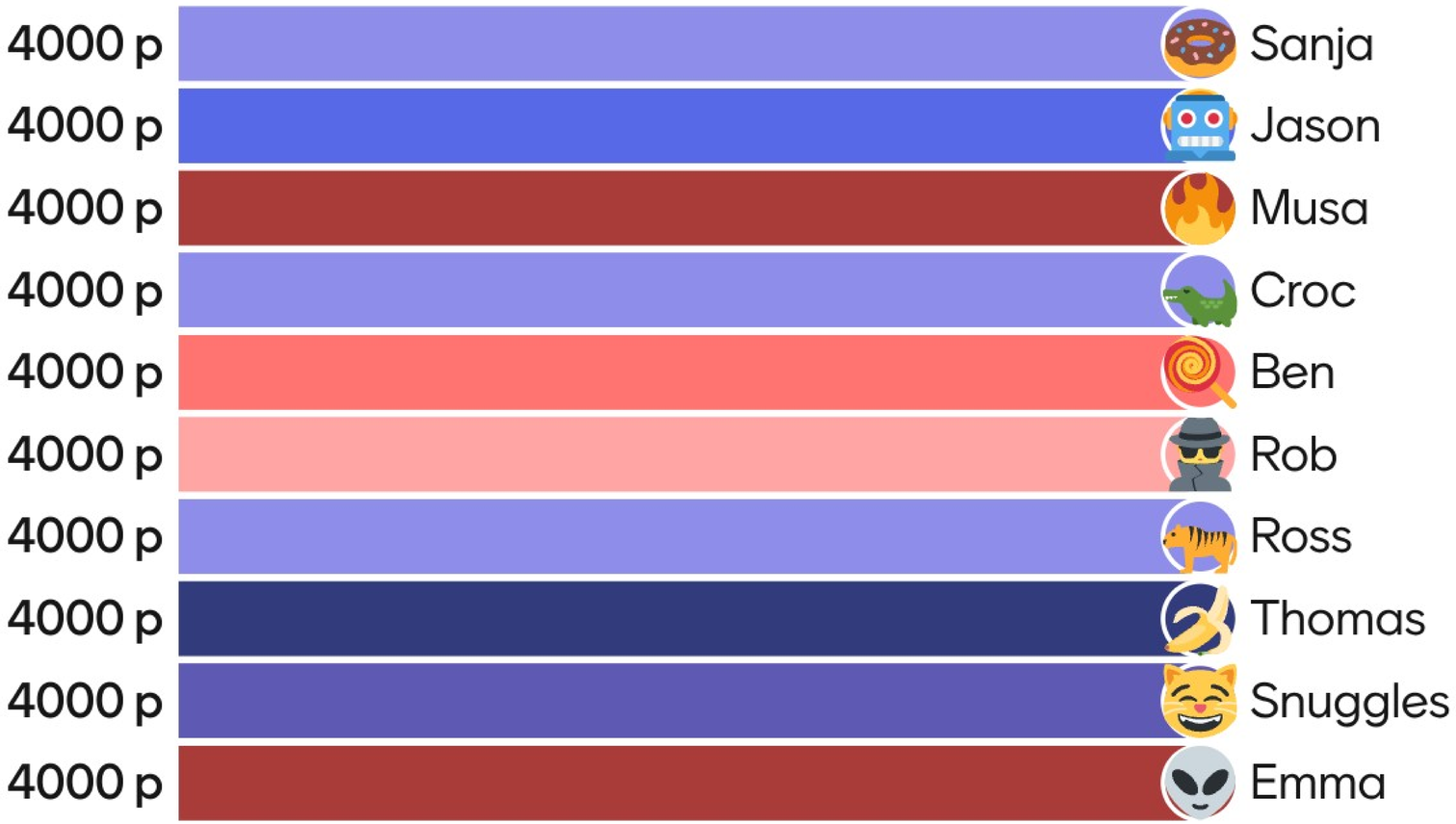
Mentimeter

Getting the (possible) changes from the main codebase to your local version



23

Quiz leaderboard



On commit messages

- What does the commit intend to change?
- Which item/user story/ticket is this related too?
- Remember the commitwhale ->
  - "this commit will"
- verb + thing + ticket

(and go to Rob Sewell's talk on commit messages on Saturday before lunch!)

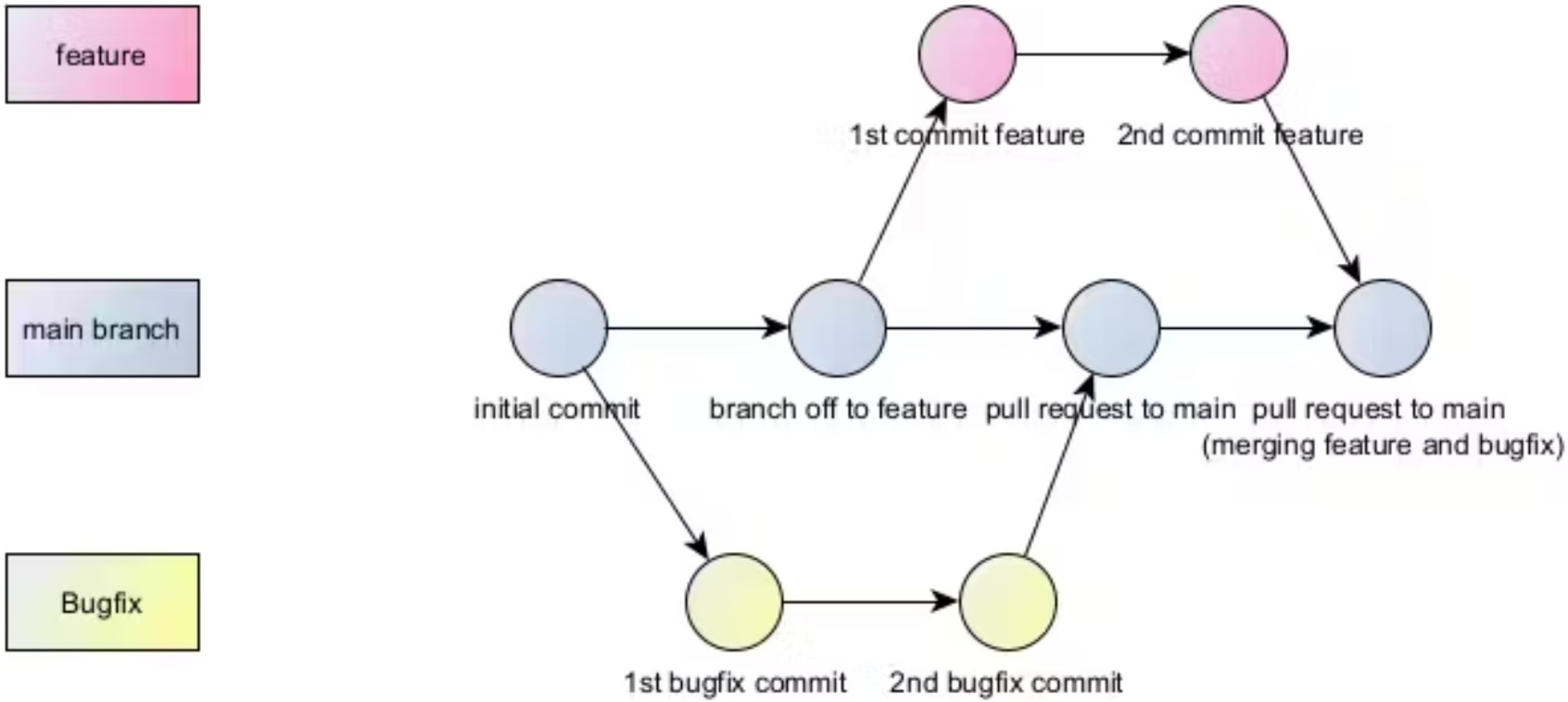
3

Branching

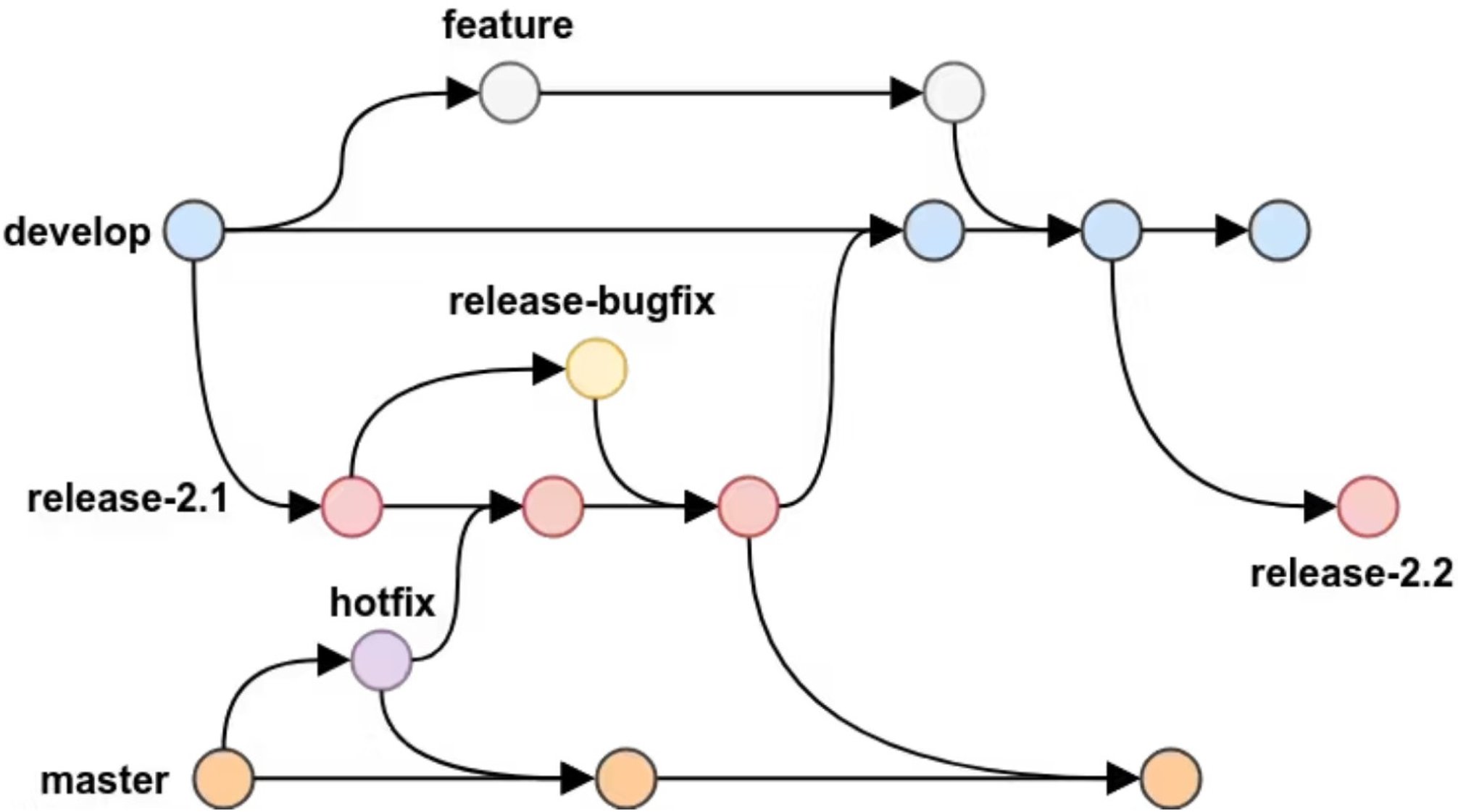
- Central part of version control, for all aspects
  - timeline
  - many changes
  - by multiple people
  - context
- Many different strategies, based on:
  - amount of developers
  - the kind of product
  - maturity of the team/organization
  - history
- CI/CD

Mentimeter

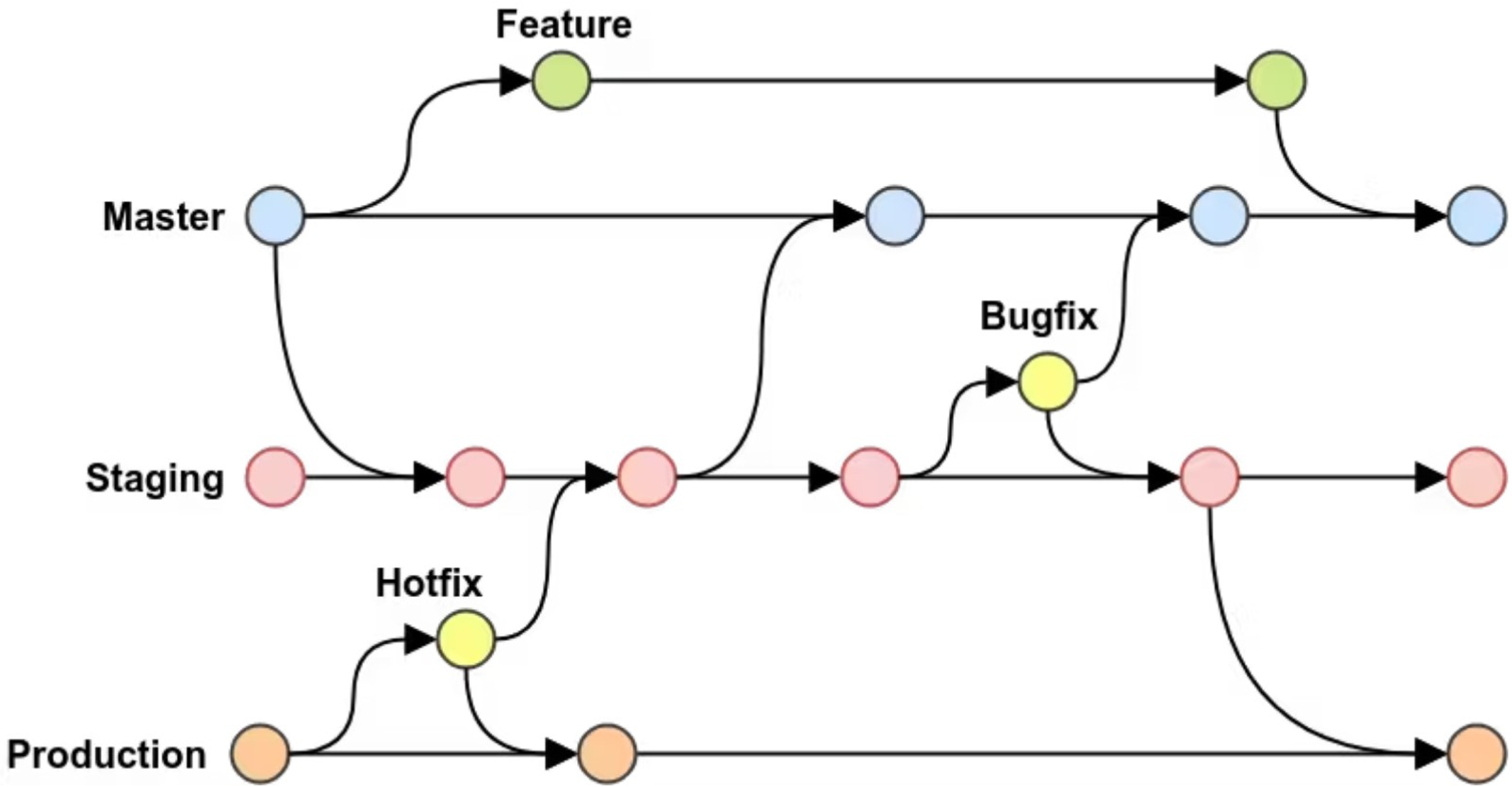
Mentimeter


feature
main branch
Bugfix
1st commit feature
2nd commit feature
initial commit
branch off to feature
pull request to main
pull request to main
(merging feature and bugfix)
1st bugfix commit
2nd bugfix commit

Mentimeter


feature
develop
release-bugfix
release-2.1
release-2.2
hotfix
master

Mentimeter



Pull requests

- How does code end up from a branch to the main branch?
- Demo time!

2

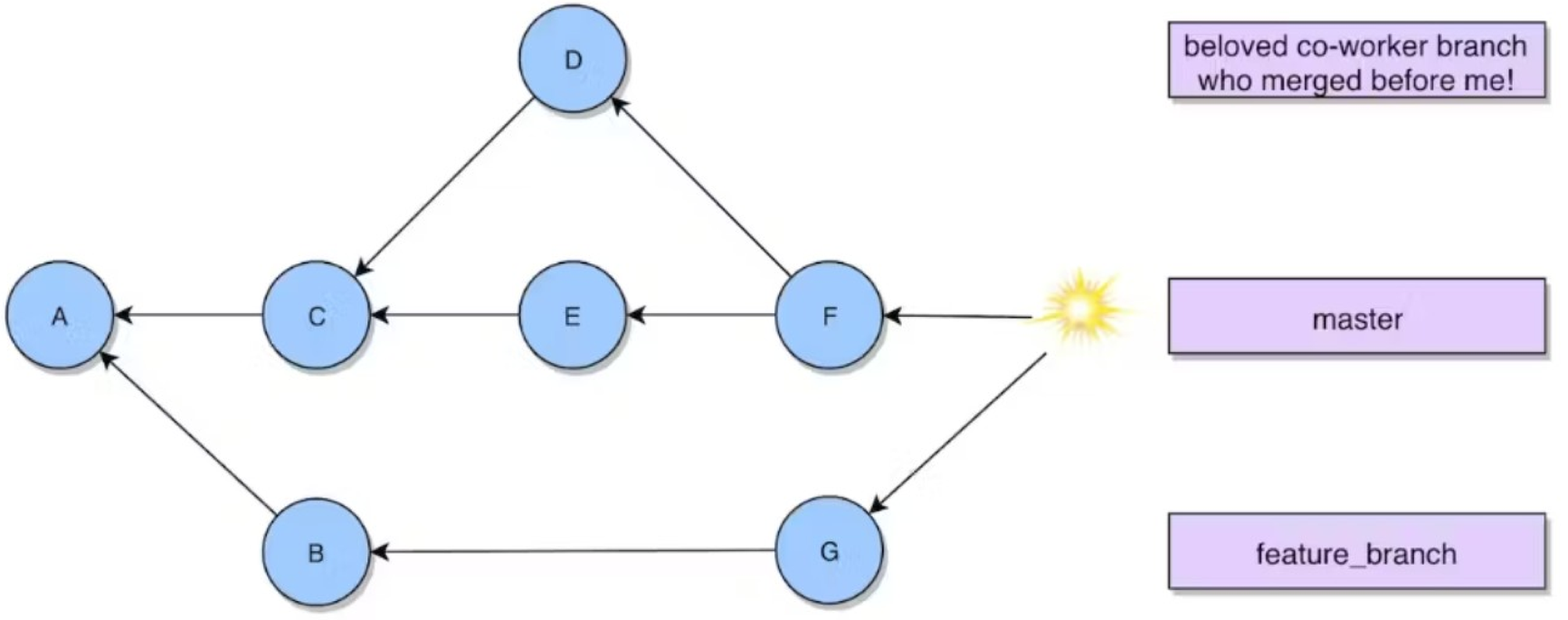
Mentimeter

Why do merge conflicts happen?

Mentimeter

Merge conflict

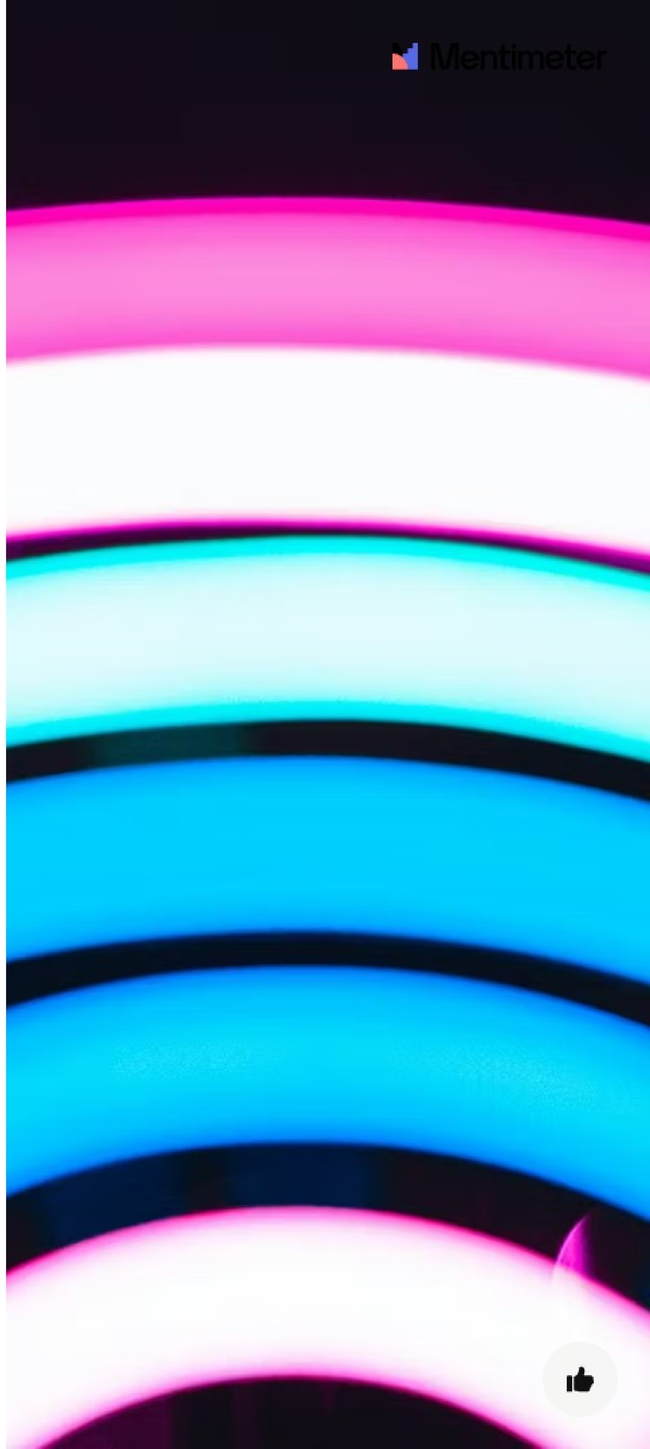
- can happen during a pull request
  - (or a push or pull on a branch, when your teams git hygiene is not great)
- happens when there is a difference between:
  - your branch and the branch you're pulling to
  - in the same line/area of code



itle of Quick Form

Actions to take when a merge conflict happens:

- diff!
- Either:
  - choose a branch to "win"
  - choose which lines of code to keep



What if everything goes wrong?

- Breathe
- Ask for help
  - or just an extra look at the conflict/error
- Revert
- Reset

0 questions

0 upvotes

What should you check, or decide in your team?

- Branching strategy
- Pull requests
  - who reviews what code? Within what timeframe?
  - merge type
- Commit messages
  - language (English? or your local language?)
  - specific format or free-form?
  - if/where/when do you link the ticket?

Mentimeter

In summary:

- Clone
- Commit
- Push
- Pull
- Branch
- Main branch vs. feature branch
- Checkout
- Pull request
- Merge
- Revert
- Reset

But:

we're still humans, so

- be kind
- show compassion
- continue to learn

Mentimeter

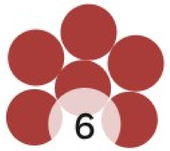
Return On Time Invested

Excellent: High Value. Very useful. Worth more than the time I spend on it.

Above Average: Good value. I gained more than the time I spend

Average: Value. Gained enough to justify the time I spend.

Meh: Useful, but it wasn't worth 100% of the time I spend on it; I lost time.



Useless: I gained nothing.

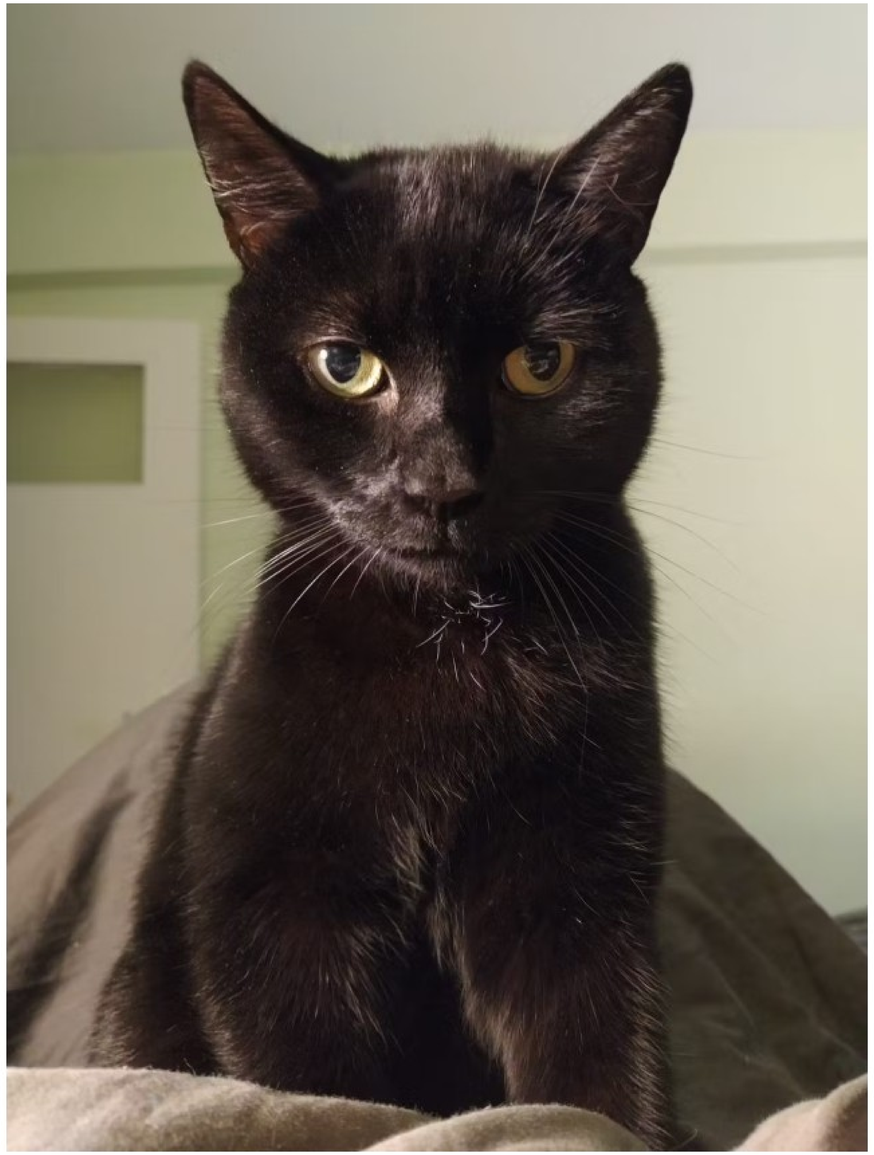
13

49

Mentimeter

Thank you, and have a lovely rest of your day!

(and don't forget to fill out the evaluation form in the SQLbits app!)



Els van Vessem:
(they/them)
@geekingthething.bsky.social
elsvanvessem@gmail.com
ifthenels.com
https://www.linkedin.com/in/els-van-vessem/

Extra resources (for sqlbits2)

- Database projects in Visual Studio with SSDT,by Kamil Nowinski:
  - https://learn.azureplayer.net/?coupon=els10 (affiliate and discount link!)
  - through this link you'll get a 10% discount if you buy the course; there are also free lessons!
- Extra tool you can use for git, with a clear visualization of branches etc:
  - htps://www.gitkraken.com
  - short explainer videos: https://www.youtube.com/@GitKraken
- Session by Heini Ilmarinen on git (20 minutes)
  - https://hub.sqlbits.com/hub/events/5f7e1f6d-8f7a-485d-9332-06a2bb2f7988/sessions/0902ad98-4b07-47e4-ae54-1807333bb2cc
- Session of Rob Sewell on commit messages (50 minute):
  - https://hub.sqlbits.com/hub/events/5f7e1f6d-8f7a-485d-9332-06a2bb2f7988/sessions/015a2228-f91d-4329-8fe8-242acdc53bfe