

Working with Git



Workshop module II

Structure of Git tooling

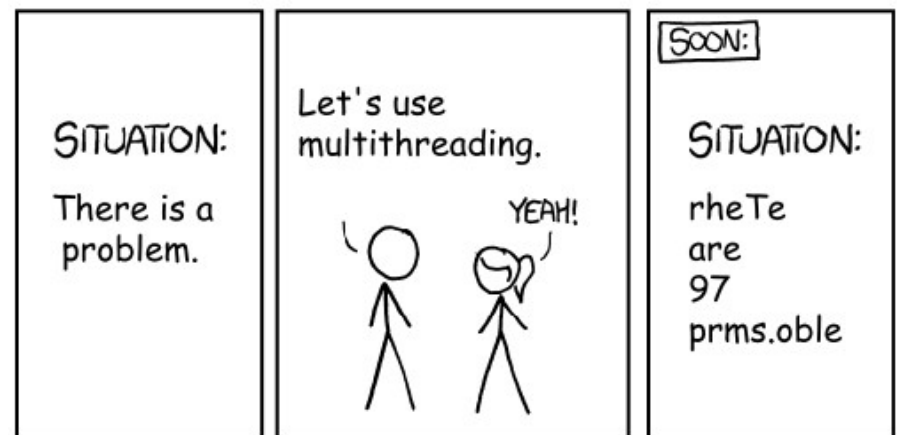
1. Introduction – what, why Git, expectations
2. Concepts – the hidden life of commits, files, heads and remotes
3. Real life example – lifecycle files, logs, distributed, social Git
4. Open stage – oh-my-git, Anki, more advanced topics, questions

Can we live without version control?

- Analogy: container ships
- Archive function
- Sole developer: don't need version control more than a trapze artist needs a safety net

Can we live without version control?

- Analogy: container ships
- Archive function
- Sole developer: don't need version control more than a trapze artist needs a safety net
- What type of complexity is Git trying to solve → inherent



Some words on origin

- BitKeeper started to charge money
- Development took 2 week till first kernel commit was hosted (2005)
- Main objectives: fast, distributed, learn from previous attempts of what not to do
- And also security, i.e. sign changes, collisions etc.



Some words on origin

- BitKeeper started to charge money
- Development took 2 week till first kernel commit was hosted (2005)
- Main objectives: fast, distributed, learn from previous attempts of what not to do
- And also security, i.e. sign changes, collisions etc.
- After two month: Junio Hamano

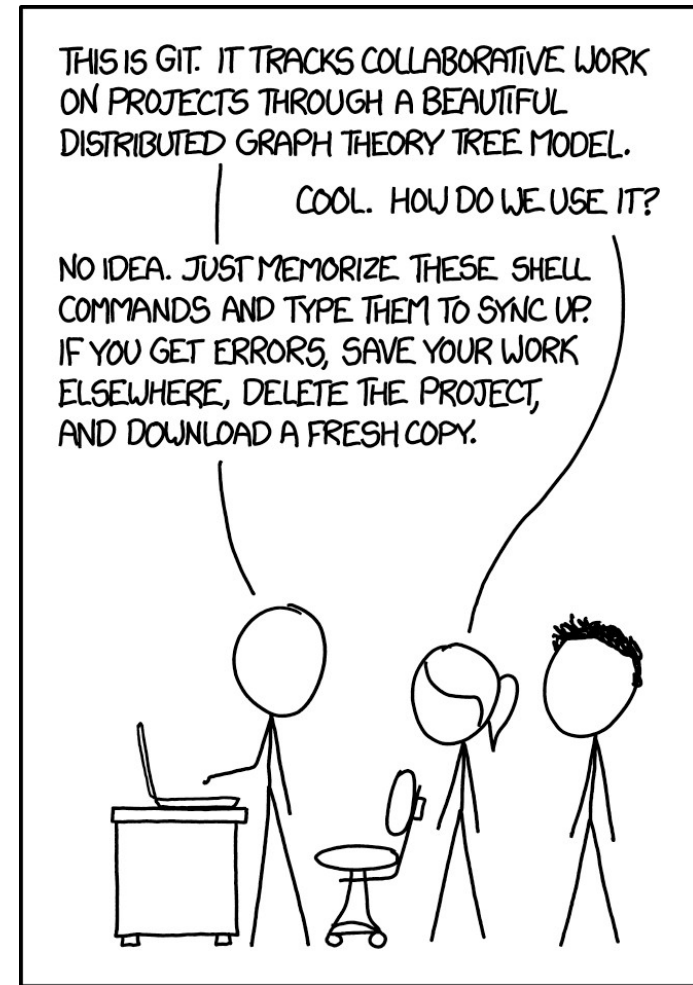


What do expect after class?

- Main goal: setting you in the right mind-set
- If you learn some commands, the better
- Able to read git man-pages without too much confusion
- You can also use an Anki deck prepared for you

What are the Elements of Git?

- Git structures
- Structuring without understanding is stupid
- So let's discuss some graphs ...



Open stage - Outro

- Read about rebase, case for linear history
- Read about notes
- Read about three main workflows in Git



An open source game about learning Git!