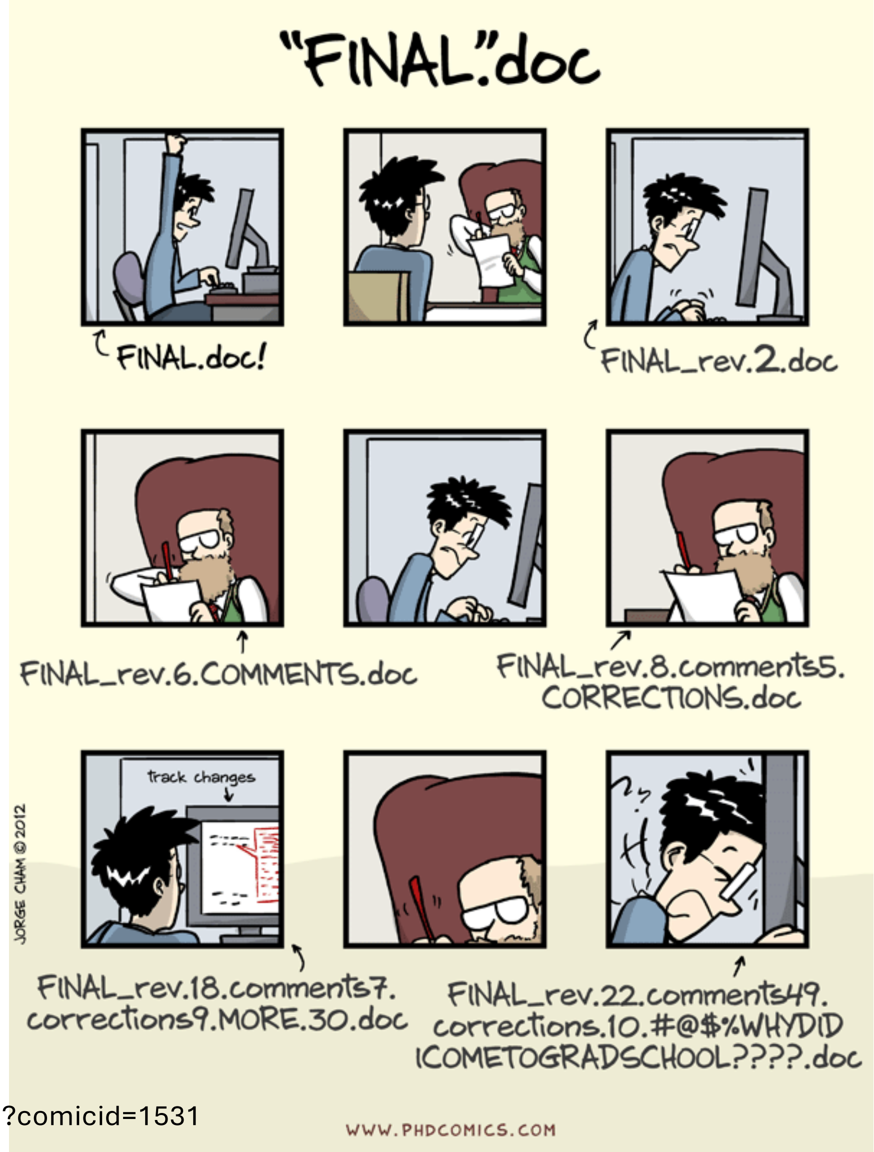
Any Questions From Day 1’s Material?

Version Control with Git

2025-12-17-Clemson



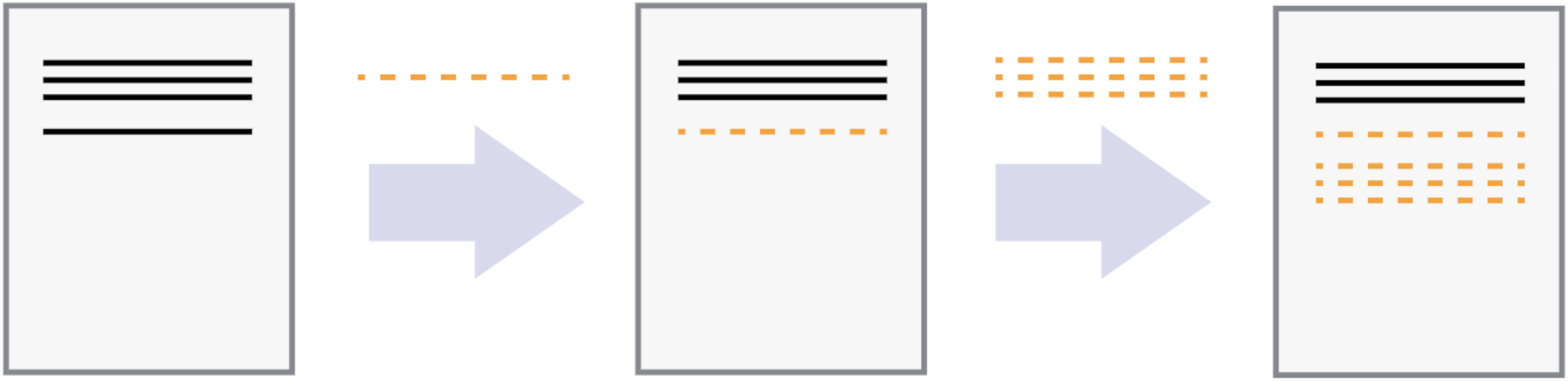
What NOT to do!

https://phdcomics.com/comics/archive.php?comicid=1531

Introduction to Version Control

- Method to manage files
- One or multiple users
- Trackable and mergeable

Example Workflow – One User

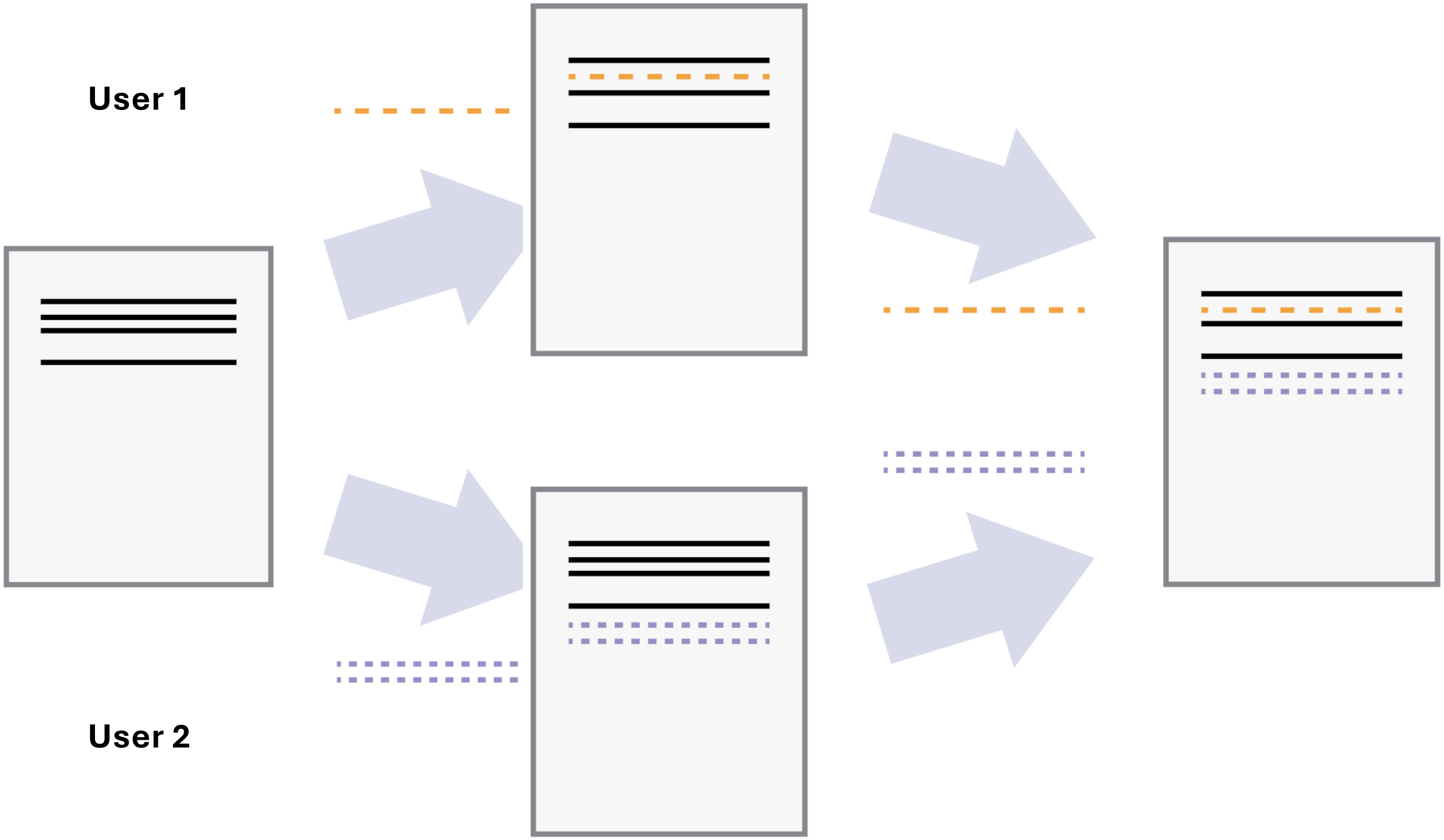


Base Document

V2

V3

Example Workflow – Two Users

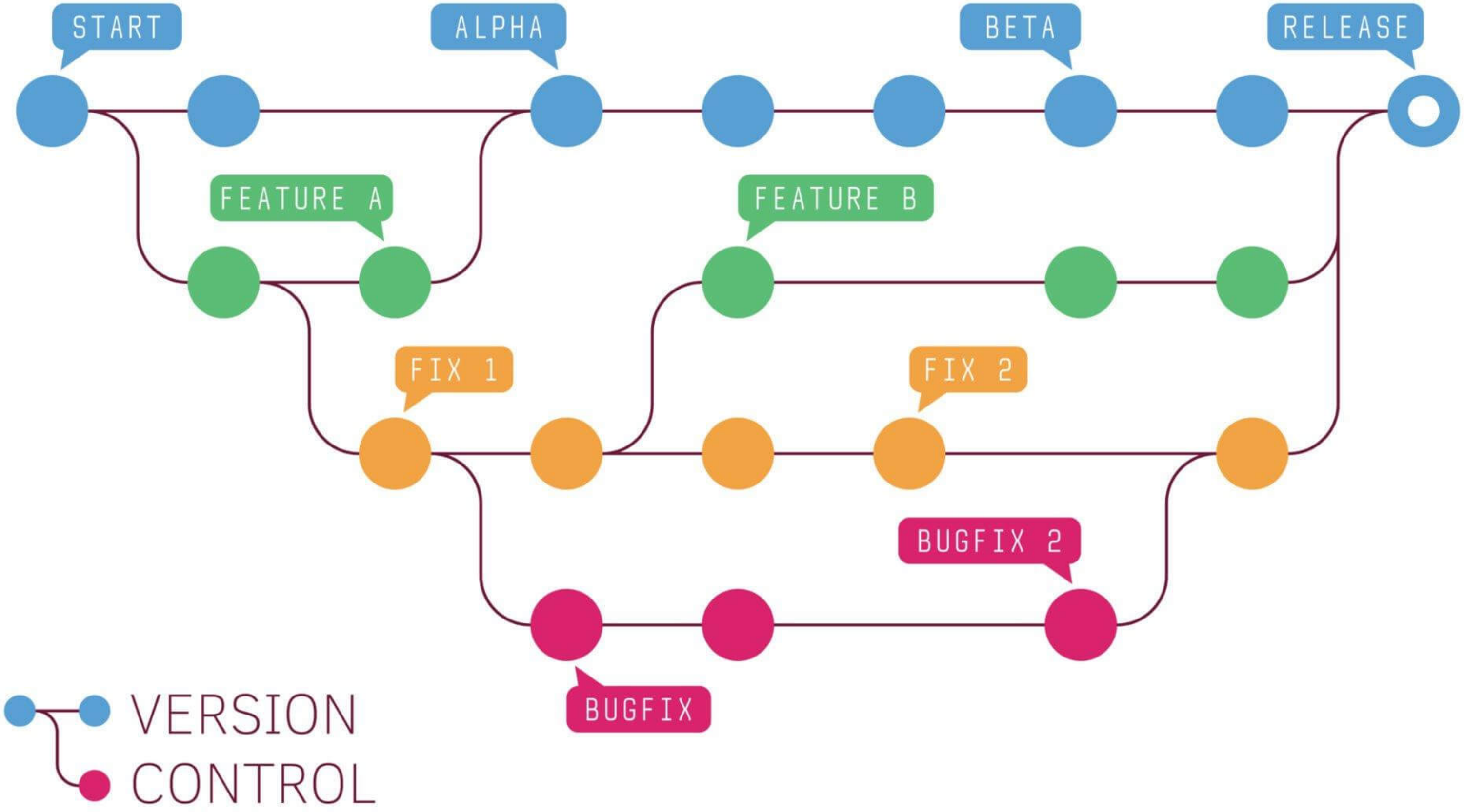


Examples You May Already Use

- Google Drive
    - Revisions (“file”->“version history”)
        - Docs, Sheets, Etc.
- Word
    - Track changes tool

Meaning of Git

- Fast, scalable, distributed revision control system
- Pronounceable 3-letter word (maybe “get” misspelled)
- “global information tracker”



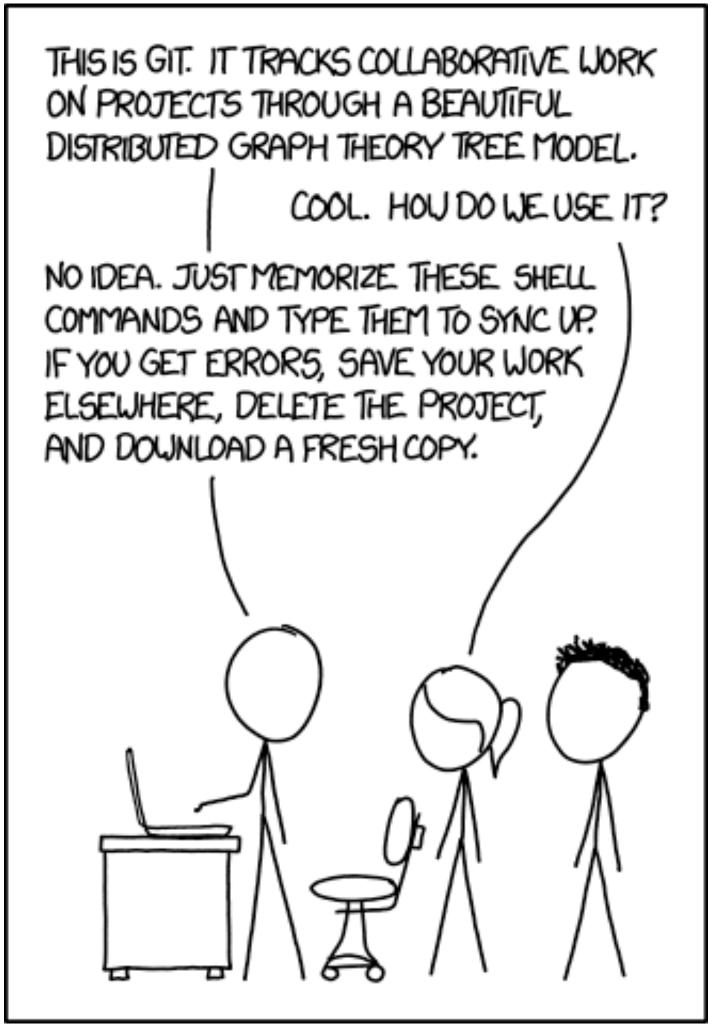
https://cssanimation.io/blog/what-is-version-control-and-how-does-it-help-you-collaborate-and-manage-code/

What is Git?

- Handles version control
- Transparent and history can be queried
- Revertible changes
- Reproducibility
- Collaborative
- Works for simple to complex projects
- Public distribution

KEY POINTS

- Version control is like an unlimited ‘undo’.
- Version control also allows many people to work in parallel.


THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.

https://xkcd.com/1597/

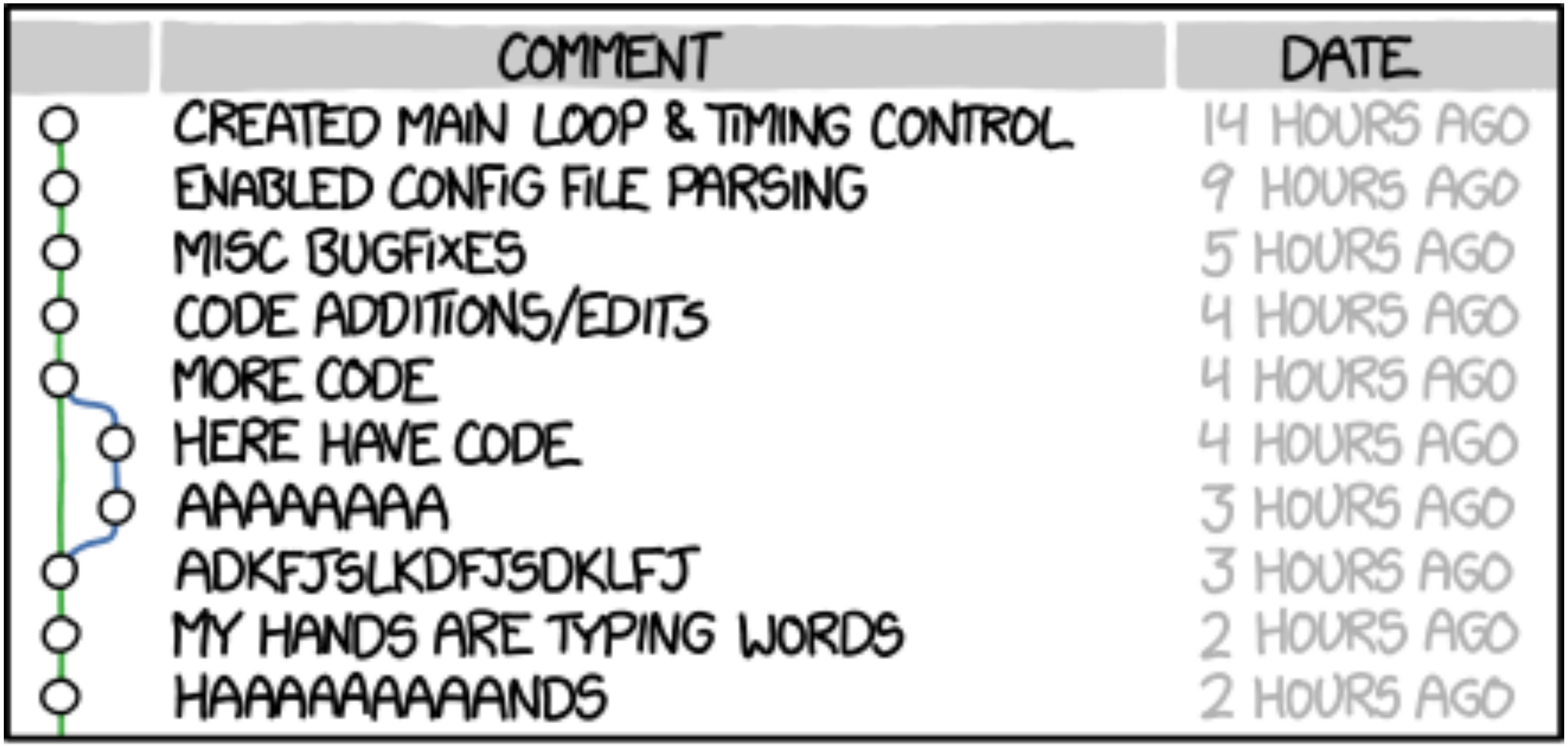
Terms to Know

- **Git** – “a version control system that intelligently tracks changes in files.”
- **GitHub** – "a cloud-based platform where you can store, share, and work together with others to write code.”
- **Repository** (**repo**) – “a place where you can store your code, your files, and each file's revision history.”
- **Branches** – “allow you to develop features, fix bugs, or safely experiment with new ideas in a contained area of your repository.”

https://docs.github.com/

Commands (git **<option>**…)

- **init** – make a new repository
- **status** – reports status of the project
- **add** – track a new/modified file
- **commit** – records changes
- **log** – lists recent commits
- **diff** – review changes before a commit
- **restore** – revert to a previously saved version of a file
- **show** – see changes made and commit message



COMMENT	DATE
CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
ENABLED CONFIG FILE PARSING	9 HOURS AGO
MISC BUGFIXES	5 HOURS AGO
CODE ADDITIONS/EDITS	4 HOURS AGO
MORE CODE	4 HOURS AGO
HERE HAVE CODE	4 HOURS AGO
AAAAAAAA	3 HOURS AGO
ADKFJSLKDFJSDKLFJ	3 HOURS AGO
MY HANDS ARE TYPING WORDS	2 HOURS AGO
HAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

https://xkcd.com/1296/

Helpful Links

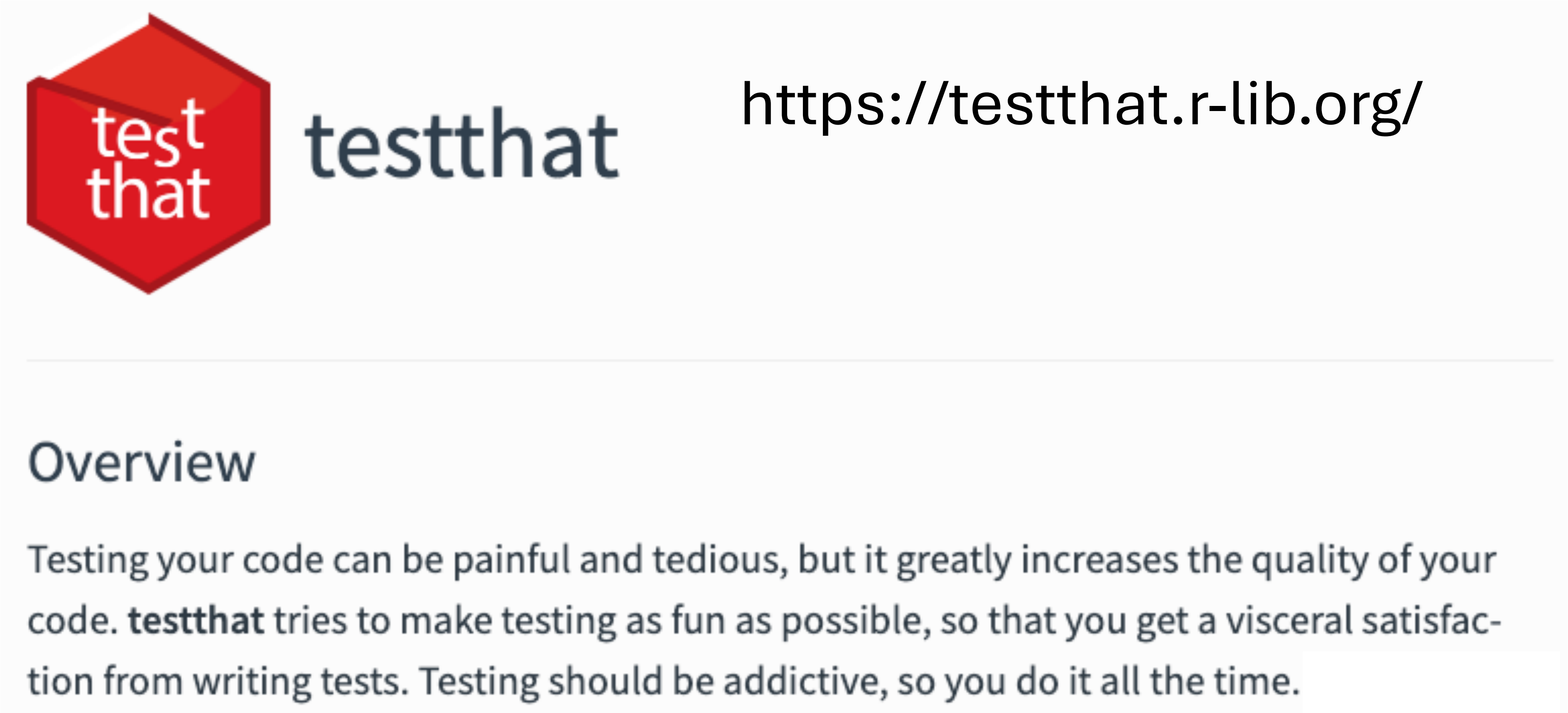
- **Markdown:** https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet

- **Visual Git Reference**: https://marklodato.github.io/visual-git-guide/index-en.html

- **Interactive Tutorial on GitHub**: https://github.com/skills/introduction-to-github

- **More on Git Branching**: https://learngitbranching.js.org/

Continuous Integration (works in github)



https://testthat.r-lib.org/

testthat

Overview

Testing your code can be painful and tedious, but it greatly increases the quality of your code. **testthat** tries to make testing as fun as possible, so that you get a visceral satisfaction from writing tests. Testing should be addictive, so you do it all the time.