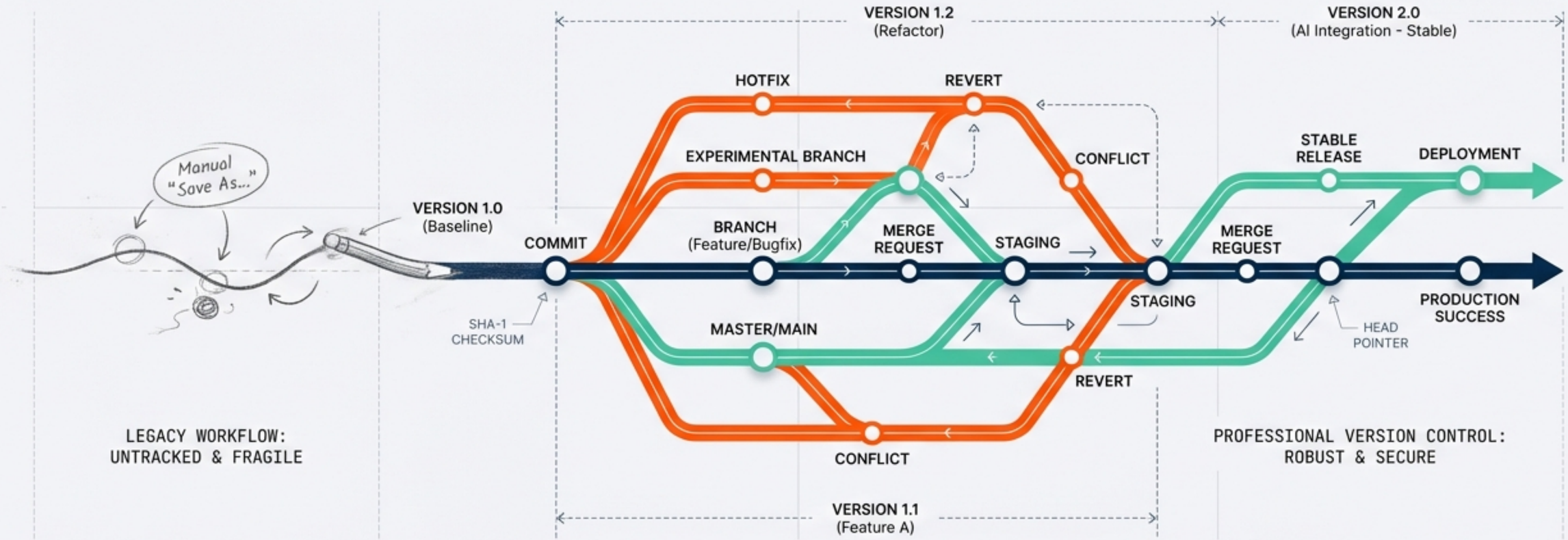
Git Foundations for the AI Era

From ‘Ctrl+Z’ to Time Travel: How to protect, experiment, and showcase your work.

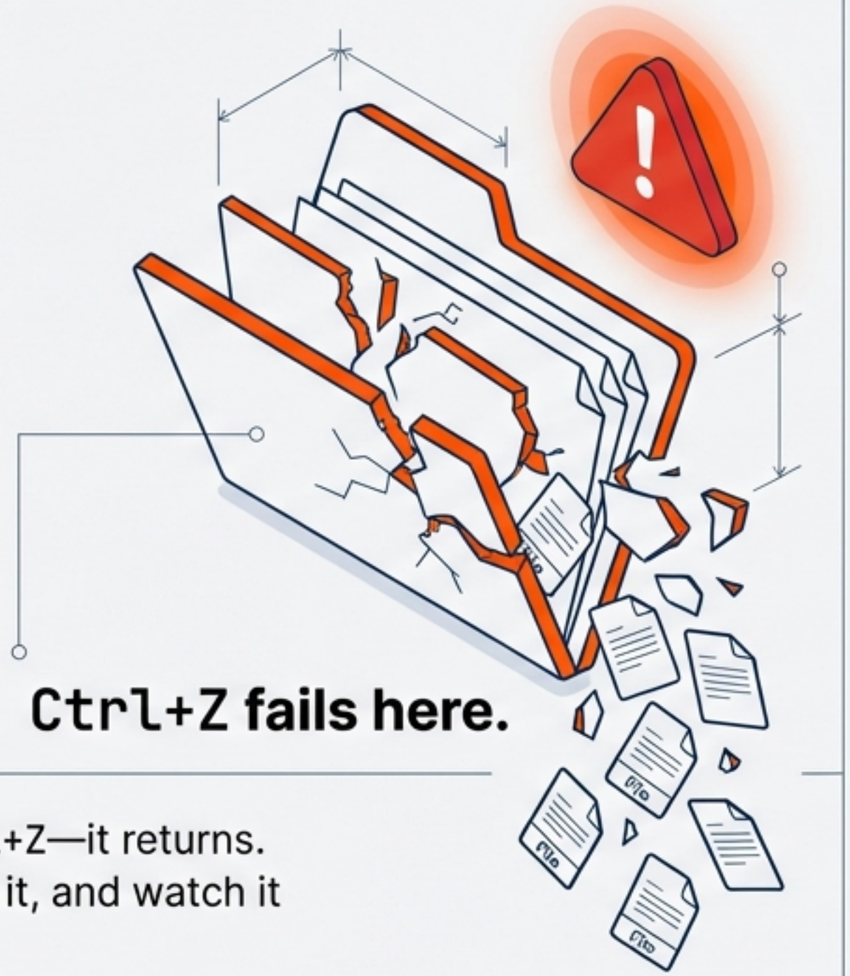


SCALE: 1:1
PROJECT: AI_FOUNDATIONS
DATE: 2024.05.15

Module 01 / Reading Deck

A Project Folder Has No Undo Button

`Ctrl+Z` works here.



`Ctrl+Z` fails here.

Open a Google Doc, delete a paragraph, press `Ctrl+Z`—it returns. Open a project folder, ask an AI agent to reorganize it, and watch it mangle 12 files. Press `Ctrl+Z`—nothing happens.

Case File

The Toy Story 2 Disaster (1998)

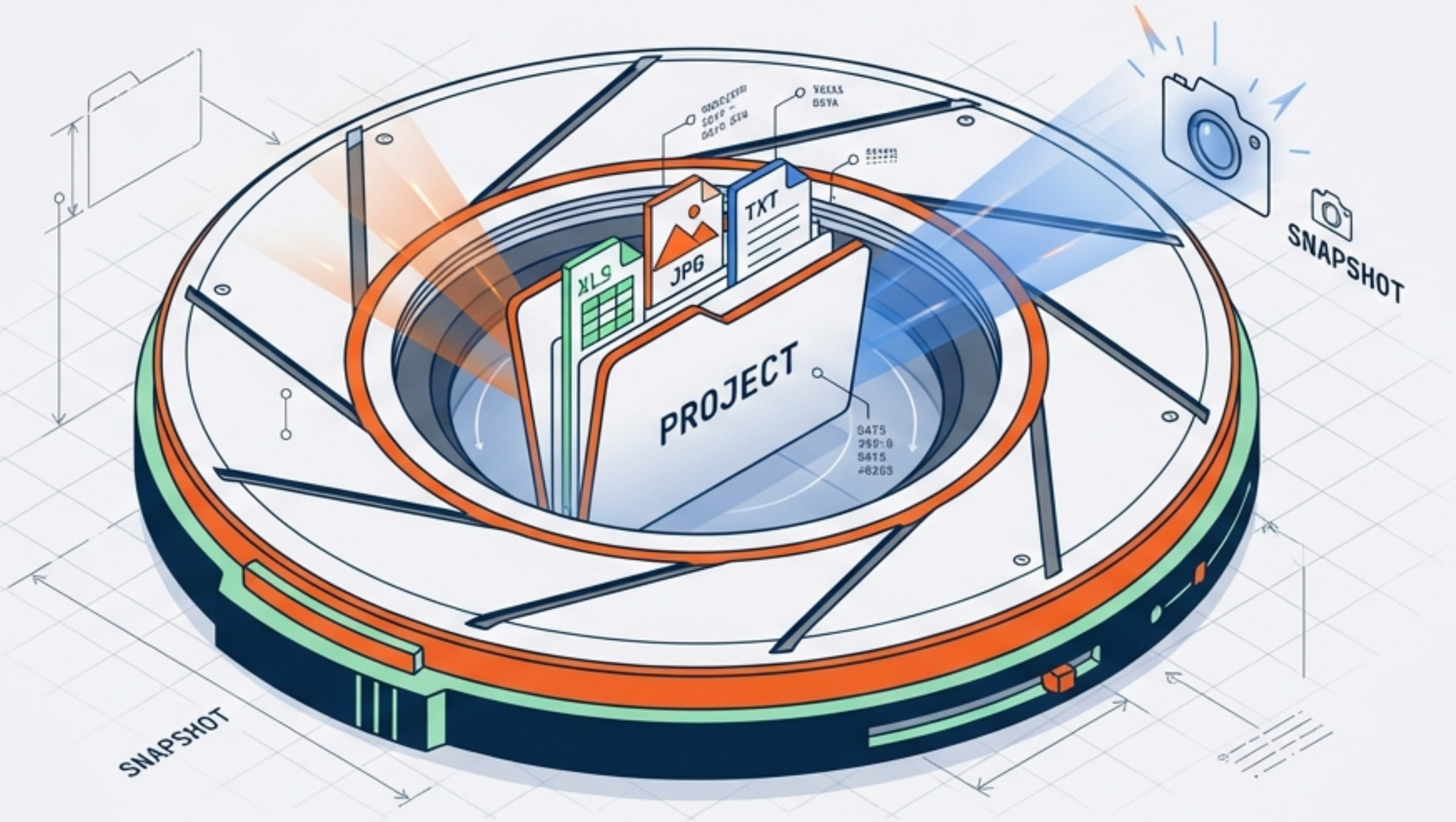
A Pixar employee accidentally ran a `delete` command. 90% of the movie vanished in seconds.

The backup system failed. The film was only saved because one employee, Galyn Susman, had a copy on her home computer to care for her newborn.

“Ctrl+Z works for one file. Git works for your life’s work.”

The Camera: Commits Are Snapshots, Not Saves

A standard “Save” overwrites the previous version. A “Commit” is a photo-graph of your *_entire_* project folder at a specific moment.



Git photographs your entire project, not individual files.

The Scenario:

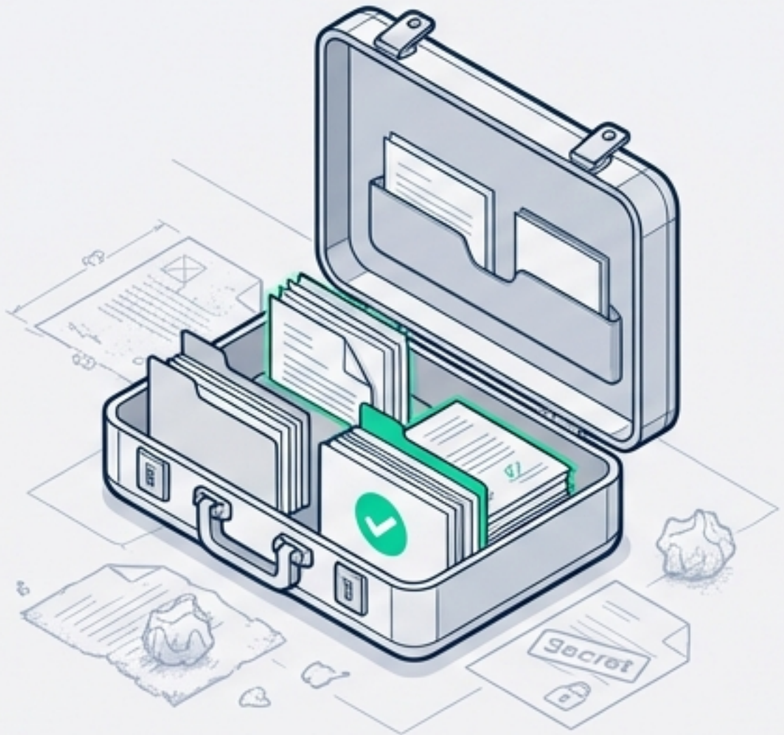
Sarah spends two hours on a budget spreadsheet. She tries a new layout and accidentally overwrites the file.
Without Git, the old version is gone forever. With Git, she simply rewinds to the previous snapshot.

The Suitcase: Why We Don’t ‘Add Everything”

You might have personal notes, privacy-sensitive data, or broken code on ‘the bed.” You only commit what is ready for the permanent record.

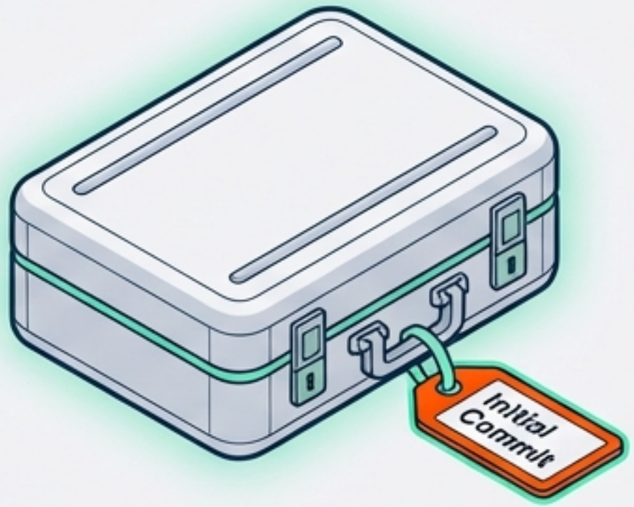
The Bed
(Working Directory)

`git add`



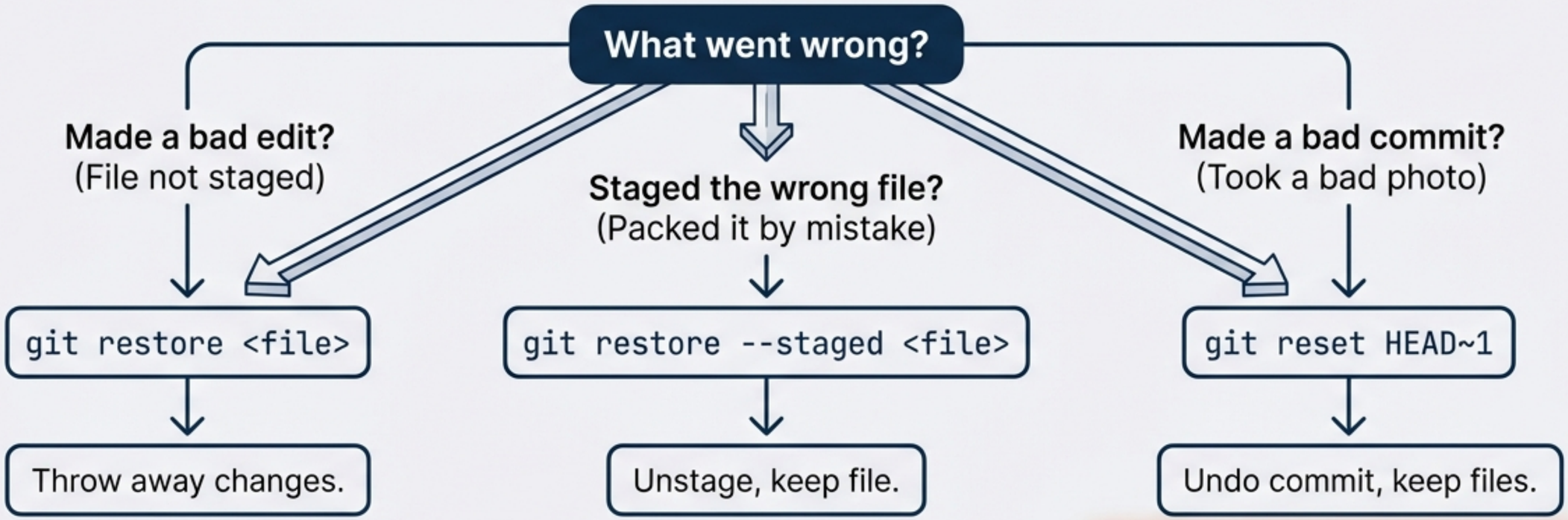
The Open Suitcase
(Staging Area)

`git commit`



The Closed Suitcase
(Commit)

The Panic Button: Three Levels of Undo

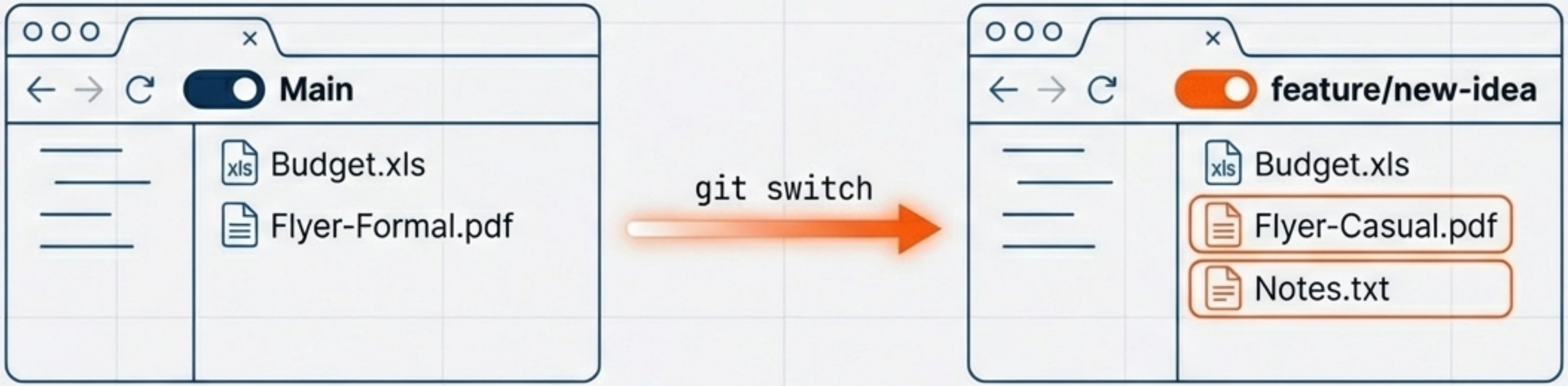


The Nuclear Option:
`git reset --hard` deletes the commit AND the files. It has no confirmation dialog. Use with extreme caution.

Parallel Universes: Testing Risky Ideas Safely

Sarah needs a "Formal" flyer for sponsors and a "Casual" one for social media. Instead of making file copies like `flyer_final_v2.txt`, she creates a Branch. She can mess up the casual version without ever touching the formal project.

The Magic Trick: Switching Contexts



Main Branch in Inter

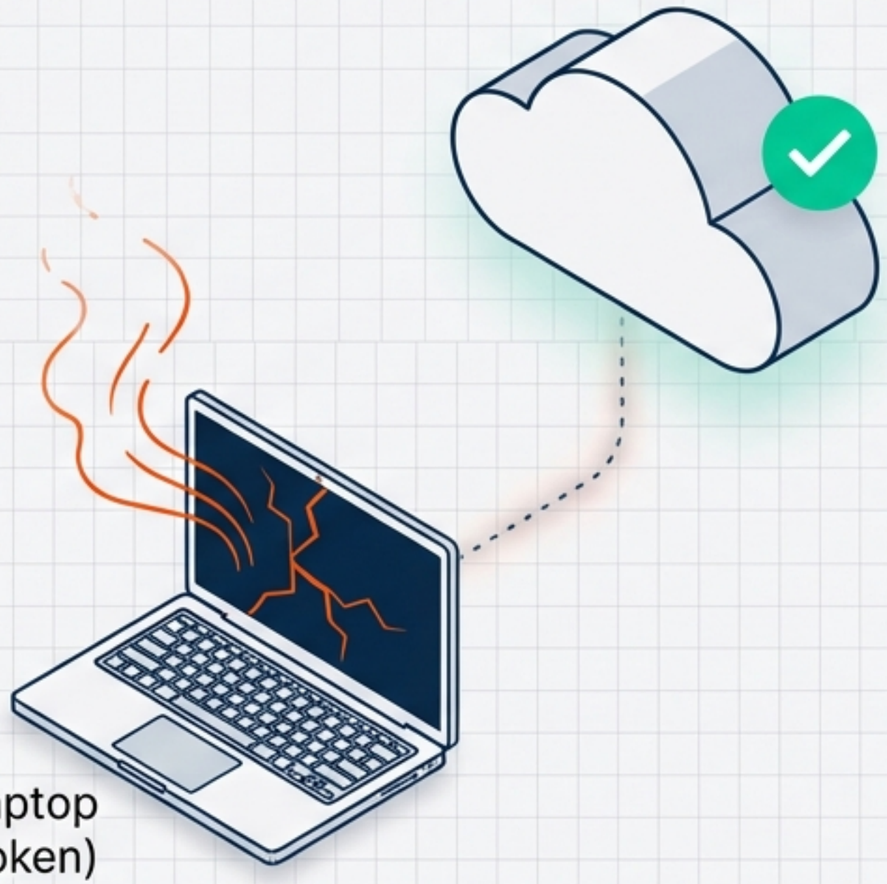
Feature Branch in Inter

Workflow Steps

- **Create Branch:** `git switch -c feature/new-idea`
- **Make Changes:** Edit files safely.
- **Switch to Main:** Return to safety.
- **Merge:** Bring the winner into the main timeline.

Branches let you live in two timelines at once—and only keep the one that works. Files literally disappear and reappear when you switch.

The Cloud: A Backup You've Never Tested Doesn't Exist



GitHub
Cloud Vault (Safe)

Sarah's Laptop
(Broken)

The GitLab Database Disaster (2017)

An engineer accidentally deleted the production database. Five different backup systems failed. They lost 6 hours of user data.

Git is the tool on your computer.
GitHub is the cloud vault.

If Sarah's project is only on her laptop, it's gone.
If it's on GitHub, she can recover it in seconds.

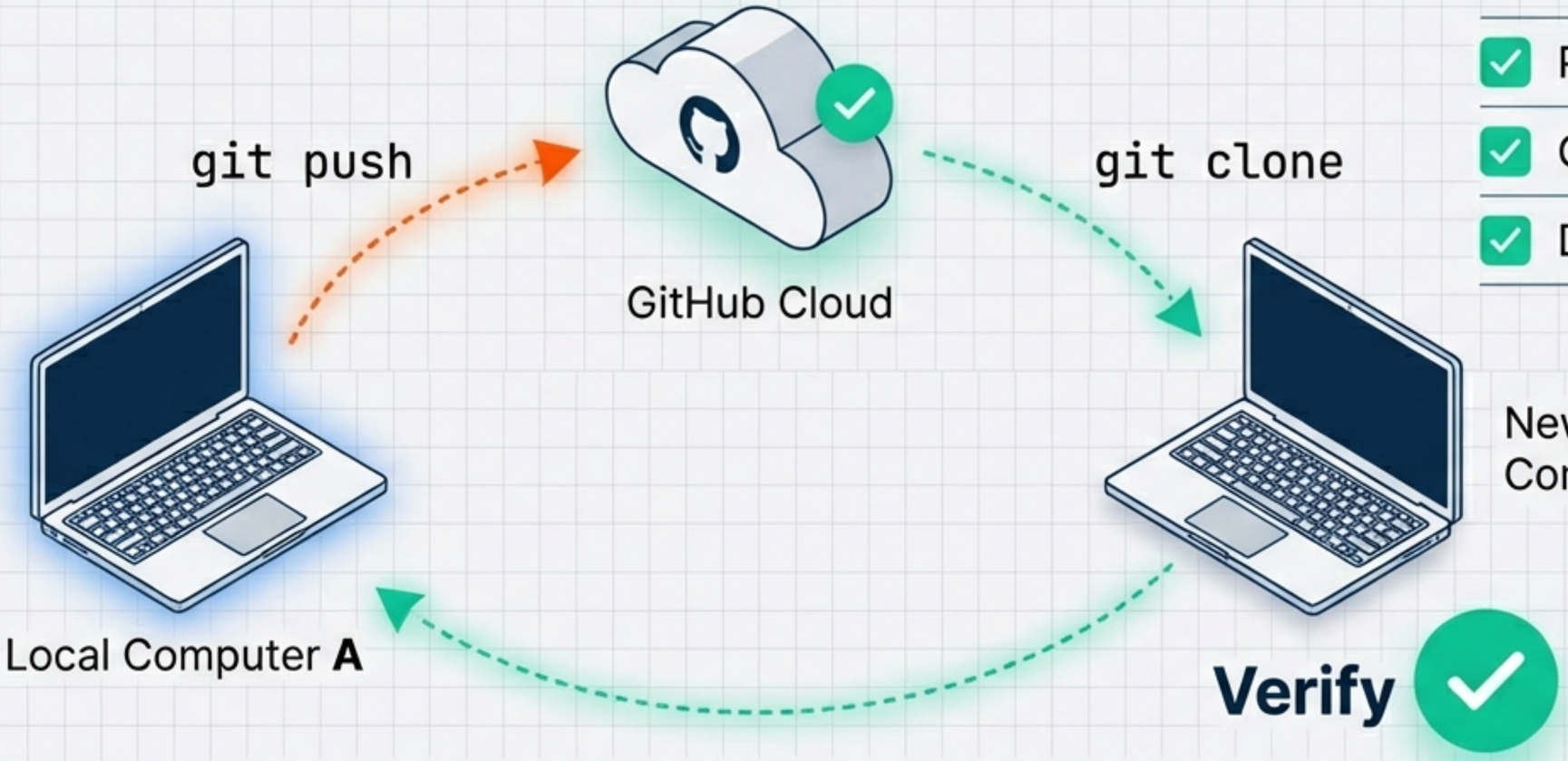
Security Critical: The .gitignore Rule

The Danger: Sarah accidentally committed her Stripe API key. Even if she deletes the file later, the key remains in the history for anyone to find.

The Rule: Configure .gitignore BEFORE your first push.

- **What to ignore:** API keys, system files, passwords.

Verification: The Clone Test



- [x] Push snapshots to GitHub.
- [x] Clone to a **NEW** folder.
- [x] Does the project run?

The push/clone cycle is the verification step most people skip.

The ‘Are You Sure?’ Dialog: Pull Requests

The Problem

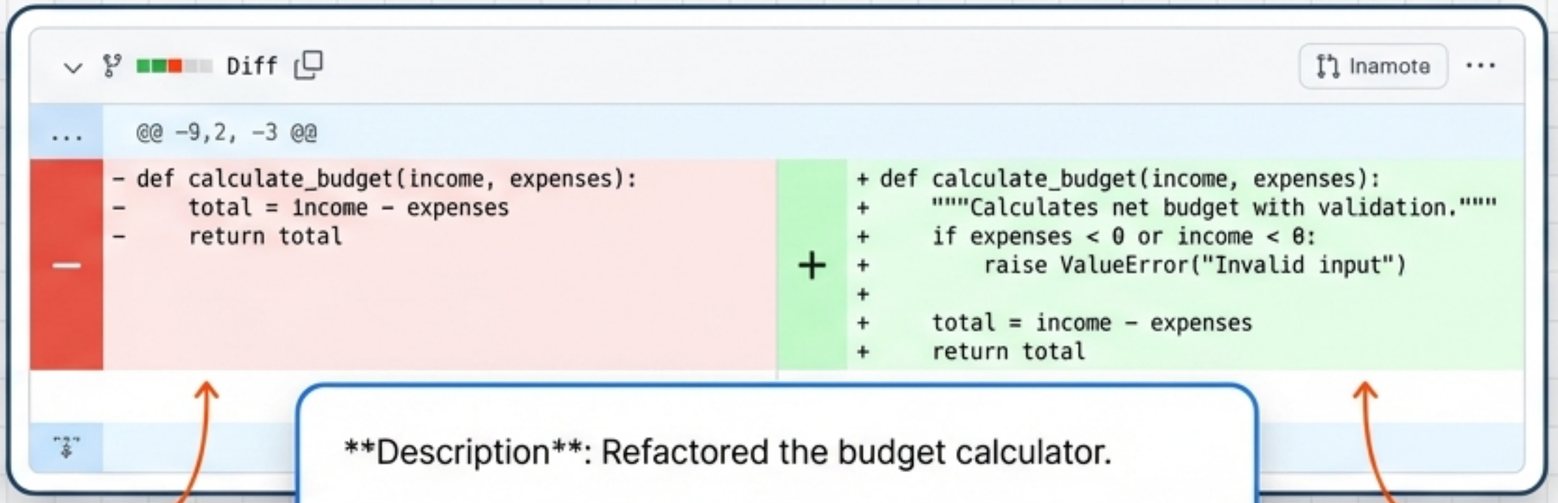
Sarah overwrote Maya’s formatting because she merged without looking. “I wrote it, so it must be fine” is a dangerous instinct.

Solution

A Pull Request (PR) forces a pause. It asks: “Here are the changes. Review them before they become permanent.”

Context Note:
The 2012 Knight Capital incident (massive financial loss) was caused by unreviewed deployment changes.

Reading the Diff & Declaring AI Intent



Green = Added

Red = Removed

Green = Added

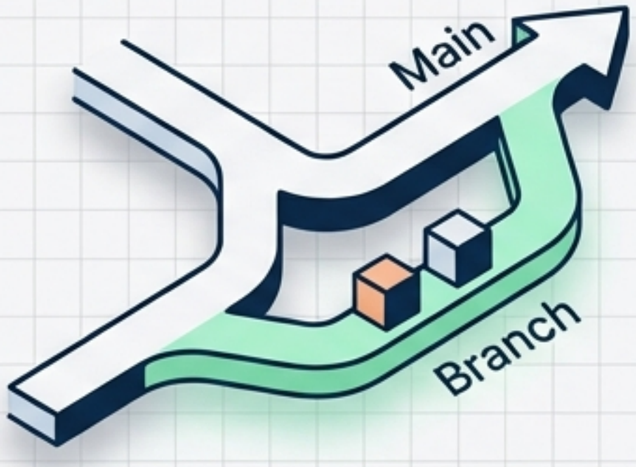
The Golden Rule —
Never merge what you don't understand—even if you wrote it.

- What changed?
- How was it tested?
- What role did AI play? (Transparency is a professional strength).

The 3 Professional Patterns

Commit Before Experimenting

The Safety Net. Save state before asking the agent to try something risky.



Branch-Test-Merge

The Sandbox. Isolate messy work so main stays clean.

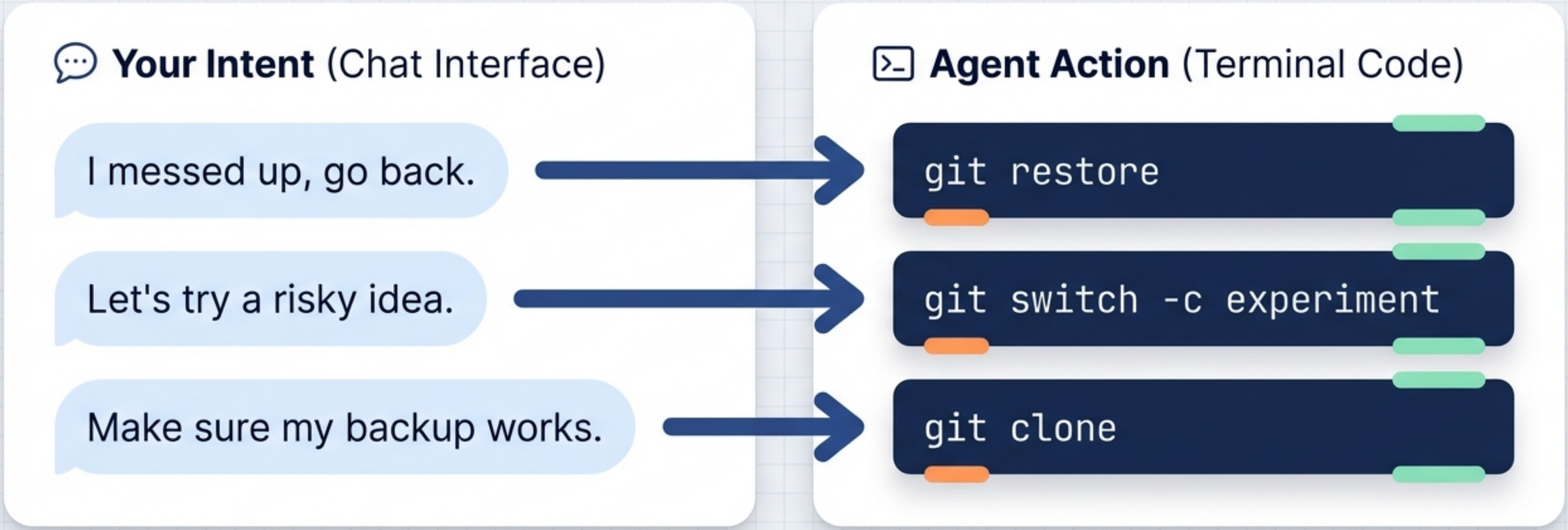
Push for Backup

The Vault. Push after every meaningful step.

Synthesizing everything into daily habits.

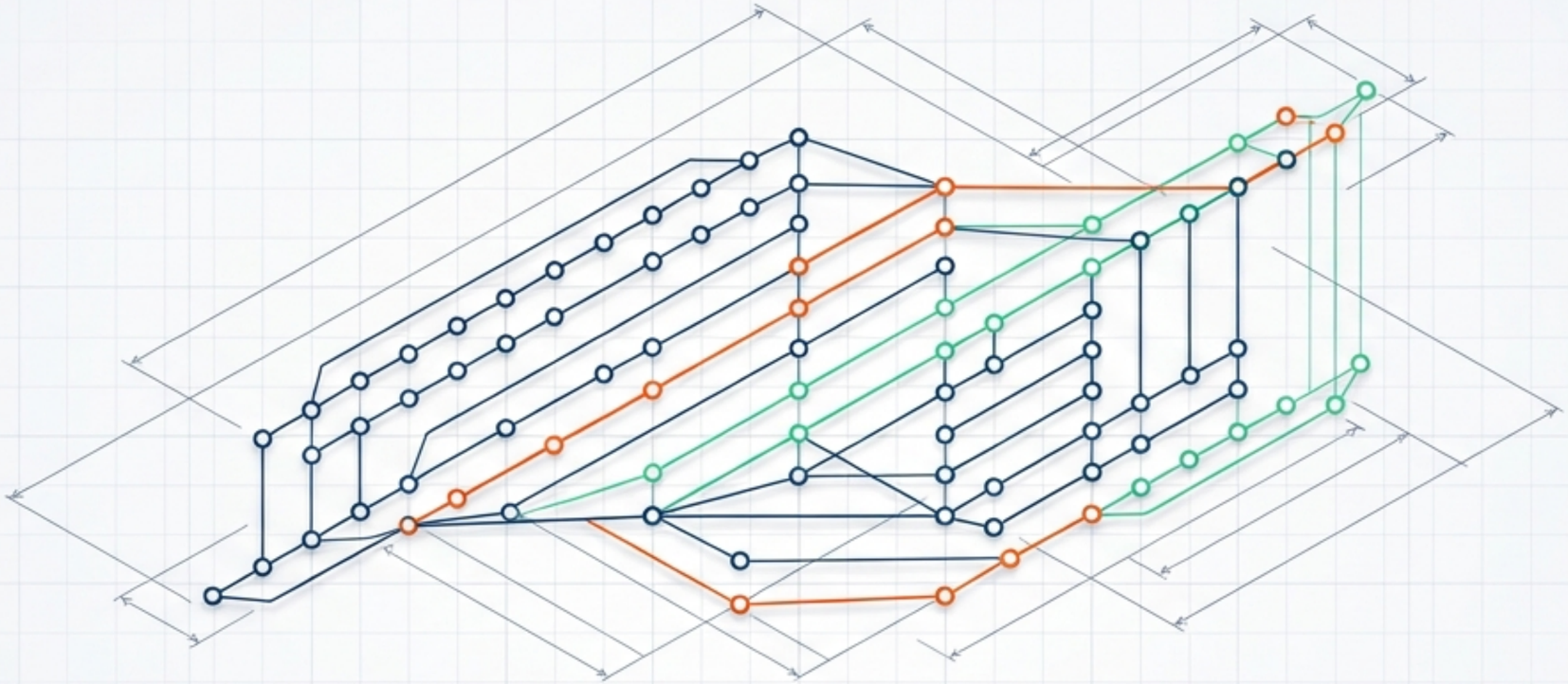
Speaking ‘Git’ to Your Agent

You don’t need to memorize syntax. You need to communicate intent.



The agent handles the command. You provide the strategy.

From Fear to Time Travel



You started hoping Ctrl+Z works. Now you have a system to Snapshot history, Create parallel universes, and Teleport work to the cloud.

Trust the system, not your memory.