AI AUTOMATION

Six coding agents at once, no two share a file

One git worktree per issue, one terminal tab per worktree, and a file-overlap check before anything launches.

Sytze P.
Solution Engineer

The problem

Running several coding agents at once sounds like free throughput. In practice they share one checkout, so two of them editing the same working tree overwrite each other mid-run.

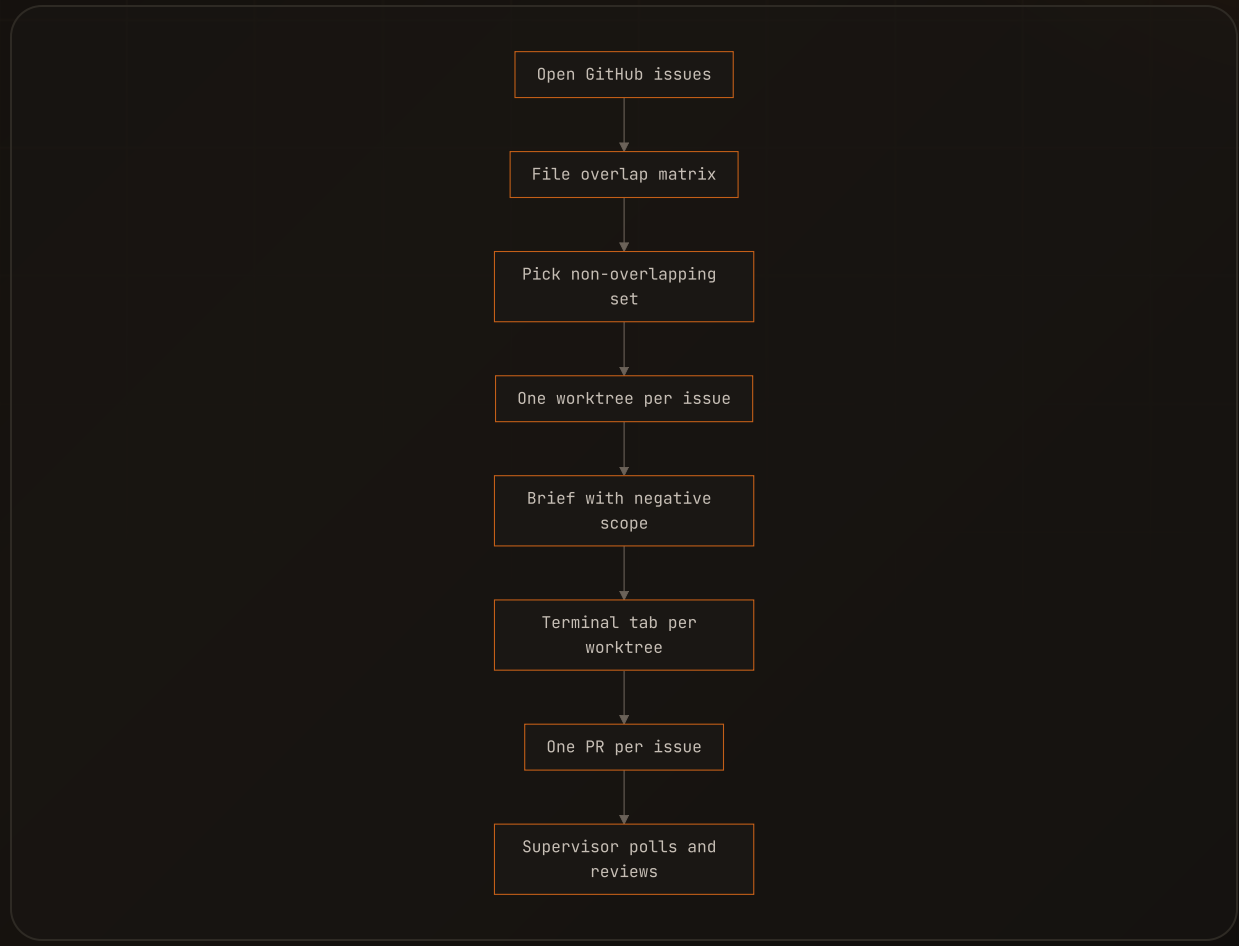
Separate branches only move the collision later. When two issues touch the same file, the conflict surfaces at merge, after both agents have finished and neither can say which half was right.

APPROACH

Issue selection to merged PR

Isolation is the whole design. Issues are not picked by priority, they are picked by file overlap. The tool reads each open issue body, builds an overlap matrix over the files each one claims, and takes the largest set of at most six where no two issues share a file.

Each surviving issue then gets its own git worktree and its own branch, so no two agents write into the same directory. Each agent's opening brief also carries a negative scope, the union of the other selected issues' files, marked off limits. Sessions run as visible terminal tabs rather than headless jobs, so the operator can watch and steer any one of them.



HIGHLIGHTS

How it works

ISSUES ARE PICKED BY FILE OVERLAP

Selection runs as a conflict check over the issue bodies. If two candidates claim the same file, one is dropped and the conflicting pair is printed for the operator to settle.

EVERY BRIEF SAYS WHAT NOT TO TOUCH

Each agent receives the union of the other worktrees' files as an explicit off-limits list, so the negative scope travels with the work instead of living in the operator's head.

A SUPERVISOR TAB RUNS THE REVIEW PASS

One extra tab polls every fifteen seconds. When a worker's pull request opens, it spawns a fresh review session for that branch, then queues a squash merge.

RESULTS

Results & impact

six sessions, hard capped

the spawn refuses a seventh worker

no shared working tree

one git worktree and branch per issue

resumes a partial spawn

only the tabs that died relaunch

STACK

Tech stack

PowerShell 7
spawn and supervisor scripts

git worktree
per-issue isolation

GitHub CLI
issues, PRs, merge

Claude Code
the agent in each tab

Windows Terminal
one tab per worker

WMI Win32_Process
tab liveness checks

Sytze P.

Solution Engineer

One git worktree per issue, one terminal tab per worktree, and a file-overlap check before anything launches.