

Ensemble Programming

Put the whole team on one problem at one keyboard, rotating roles on a short timer, so code is written and reviewed simultaneously and knowledge spreads as it's produced.

DURATION

2

hours

GROUP SIZE

—

people

WHAT YOU BRING

A real task suited to group work, a shared keyboard with a big screen, a visible timer, and 3-6 people for two hours.

WHAT YOU LEAVE WITH

- Committed, reviewed code on a real task
- Shared knowledge of the area across everyone present
- A rotation length the team has calibrated for next time
- Captured follow-ups (tech debt, missing tests) with owners

WHO TO INVITE

- **Facilitator.** Runs the timer, enforces role discipline, coaches new navigators and drivers; doesn't code on a first session.
- **Driver.** Hands on the keyboard, types what the navigator says; translates intent into code at the right level of abstraction.
- **Navigator.** Directs the driver and decides the next small step; the only person who speaks to the driver.
- **Ensemble members.** Think ahead, raise concerns, suggest ideas, spot bugs; speak to the navigator, never to the driver.
- **On-call engineers for this code.** Join so the people who'll be paged at 3am have already seen the code, not just its author.

USE WHEN

Entering unfamiliar territory – new language, framework, service, or domain

Knowledge is trapped in one head and needs to spread across the team

Code review is a bottleneck and feedback arrives days after the code

A tangled legacy area produces more bugs than progress when worked solo

AVOID WHEN

The work is trivial and one person will finish it in ten minutes

The team hasn't agreed to try it – forced ensembles breed resentment

People need deep individual focus on separate tasks right now

The group would be larger than six and rotations turn glacial

How the session runs

● Phase 1 – Orientation (10 min)

Recap the driver-navigator-ensemble roles and the one-voice-to-the-driver rule, set the goal at a high level (shape, not full design), and pick a confident first driver and a middling-experience first navigator.

● Phase 2 – Working rotation, block 1 (40 min)

Start the timer and enforce the rotation out loud at every handoff, mid-sentence if needed. Coach silent navigators, redirect ensemble chatter back to the navigator, and aim for something running by the end of the block.

- **Phase 3 – Break (10 min)**

A real break – stand up, leave the room, recover from the social load of continuous presence. Run a thirty-second check-in on rotation length before stopping; don't let the break become a meeting.

- **Phase 4 – Working rotation, block 2 (40 min)**

Same rules, same timer; this is when the rhythm clicks. Watch for late-block energy drops, the just-one-more-thing trap, and solo drift to personal laptops, and aim for a clean commit point before time.

- **Phase 5 – Commit, retro, wrap-up (20 min)**

Stop coding when the timer ends, commit cleanly or stash on a clearly labelled branch, then run a short retro on the format – what felt good, what felt awkward, what rotation length to try next time.