

ARTIFICIAL INTELLIGENCE

INTERMEDIATE

2 DAYS

Cursor for Teams: Rules, Review and CI/CD

What changes when a whole team adopts Cursor rather than a few enthusiasts. Covers shared rules as a versioned team asset, onboarding onto unfamiliar codebases, review standards for AI-generated code, automated pull request review, and integrating agents into CI/CD without weakening your gates.

Why Cursor for Teams?

Consistency across the team

Shared rules so output quality does not depend on who wrote the prompt.

Faster onboarding

New joiners productive on unfamiliar systems in days rather than weeks.

Review is the bottleneck

Practical techniques for handling more generated code without drowning reviewers.

Gates stay intact

Integrate agents into CI/CD without quietly lowering your quality bar.

What you'll be able to do

- ✓ Design shared project rules and maintain them as a versioned team asset
- ✓ Use Cursor to onboard engineers onto unfamiliar or legacy codebases
- ✓ Define review standards specific to AI-generated code
- ✓ Configure automated pull request review and interpret its findings
- ✓ Handle autofix suggestions and control false positives
- ✓ Integrate the headless CLI into CI/CD pipelines and pre-merge checks
- ✓ Establish a team definition of done for AI-assisted work
- ✓ Measure whether team throughput has genuinely improved

Who this is for

- Engineering teams adopting Cursor collectively
- Tech leads and staff engineers setting standards
- DevOps engineers owning pipelines and gates
- Engineering managers accountable for delivery quality

Curriculum

01 Shared Rules and Team Conventions

- Project rules as a shared, reviewed, version-controlled asset
- Encoding architecture decisions, conventions and forbidden patterns
- Scoping rules by directory so a monorepo behaves sensibly
- Reviewing and updating rules as the codebase changes
- **Hands-on:** build a rule set for a real repository as a team exercise

02 Onboarding and Codebase Understanding

- Getting productive on an unfamiliar system quickly
- Asking the codebase questions instead of interrupting colleagues
- Generating architecture notes and onboarding documentation
- Mapping dependencies and tracing behaviour through unfamiliar code
- **Hands-on:** onboard onto a codebase nobody in the room has seen

03 Code Review for AI-Generated Code

- What changes about review when volume increases
- Failure modes specific to generated code and how to spot them
- Attribution and labelling so reviewers know what they are looking at
- Review checklists that catch what humans reliably miss
- Protecting reviewer attention as the scarce resource

04 Automated Pull Request Review

- Enabling automated review across repositories
- Interpreting findings and what the tool catches that humans do not
- Autofix suggestions and deciding what to accept
- Managing false positives so the team keeps trusting the signal
- Using the review trail as a compliance artefact

05 CI/CD and Pipeline Integration

- Running the headless CLI inside pipelines
- Pre-commit and pre-merge checks that use agents
- Where agents belong in a release path and where they do not
- Vulnerability scanning and security checks in the flow
- **Hands-on:** wire an agent check into a working pipeline

06 Team Standards and Measurement

- Defining done for AI-assisted work at team level
- Throughput against review load: measuring the real bottleneck
- Escaped defects and quality signals worth tracking
- Retrospective cadence as the tooling changes underneath you
- **Hands-on:** agree a team working agreement for AI-assisted development