

CI/CD and Quality Gates

- Run automated tests on every significant change
- Block merges on critical quality failures
- Keep pipelines fast to preserve developer flow

References: <https://docs.github.com/en/actions> |
<https://www.atlassian.com/continuous-delivery/principles/continuous-integration-vs-delivery-vs-deployment>

GitHub Actions for QA

- Define workflows as code in YAML
- Split jobs by test type and parallelize
- Publish reports/artifacts for visibility

References: <https://docs.github.com/en/actions>

Branching and Integration Strategy

- Prefer short-lived branches and frequent merges
- Use trunk-based practices to reduce merge risk
- Keep main branch releasable

References: <https://trunkbaseddevelopment.com/>

Release Confidence and Versioning

- Use semantic versioning for clear change signals
- Tie release decisions to evidence, not intuition
- Track rollback readiness and known risks

References: <https://semver.org/>

Shift-Left and Shift-Right Together

- Shift-left: prevent defects earlier
- Shift-right: monitor and validate in production
- Use both for complete quality feedback loops

References: <https://dora.dev/> | <https://opentelemetry.io/docs/>

