

8 Tips for Effective Branch Management



Branching is what makes version management so important to developers. It allows teams to work in parallel, to experiment and to ensure the right features are released at the right time. Good branch management is key to effective development. Bad branching will cause risk and rework.

1	Keep Things Simple	Ideally everyone would use a single main codeline (or “mainline”) with no branches, but that’s not often possible. Keep your branching model simple with branches lasting only as long as necessary.
2	Have Well-Defined Policies	Each branch should have a single policy for who can make changes, when CI builds happen, what tests to run, etc.
3	Give Codelines an Owner	Appoint a single owner for each codeline or branch to resolve any ambiguities in the policy.
4	Don’t Retain Development or Task Branches	Creating branches for a team or task/feature helps many teams, but keep these short-lived and remove them at the earliest opportunity. Having too many “dormant” branches adds complexity and increases the risk of using the wrong branch.
5	Use Branch for Release/Milestones	In the real world, releases have to be made while development continues. Create branches to separate a release while final testing is running or if you have multiple releases in production.
6	Make Streams a Key Workflow	Streams implement a simple workflow to control the behavior of development (usually unstable) and release (stable, restricted change) branches. They are also good for visualizing where a change needs to flow from developer to release and for componentizing larger projects.
7	Protect Your Mainline	A common pattern for Continuous Delivery is to have a single mainline and to ensure that mainline is always buildable and potentially releasable. That means all commits have to be high quality. Use code reviews and pre-commit CI build and test.
8	Merge Down/Copy Up	Streams encourage the simple “merge down” (merge hot fixes from release into refactored development stream) and “copy up” (take the entire release from dev to release) after verification.