

KOMOLAFE, EZEKIEL D.

Branching Strategy: Git Workflow for Multi-Developer Projects

In working on a large project with multiple developers, maintaining code quality and stability is crucial. One strategy that has really helped my team achieve this is the Gitflow Workflow. It is a branching strategy that makes collaboration easier, keeps our codebase stable, and help us maintain our deadlines without sacrificing quality.

Branch Types

In Gitflow, we mainly use five types of branches:

- i. Main (master or main) Branch:* in this type of branch the following are achieved
 - a. Where the production-ready code lives.
 - b. Only official versions are released from this branch.
 - c. It's crucial that this branch stays stable, so no one pushes directly to it.
 - d. Codes only go to it after they have passed the test, and code reviews
- ii. Develop Branch:* in this type of branch the following are achieved
 - a. It is the playground for integrating features.
 - b. All the latest changes that are still in development are stored in this branch.
 - c. Once a feature branch merged into develop, it goes through testing.
- iii. Feature Branches:* in these type of branches the following are achieved
 - a. This branch is created off develop.
 - b. Each one focuses on a single feature or task.
 - c. They are usually named feature/<feature-name>. e.g feature-ezekiel
 - d. After finishing a feature, it is merged back into develop, assuming it passes the developer's own tests and code review
- iv. Release Branches:* in these type of branches, the following are carried out
 - a. When there are set of features ready for release, a release branch is created from develop.
 - b. This branch is for final testing and last-minute bug fixes.
 - c. They are named release/<version-number>.

- d. Once it is stable, they are merged into both main and develop.
- v. **Hotfix Branches**: in this type of branch;
 - a. Hotfixes are for critical issues in production, created directly from main.
 - b. These branches allows addressing urgent problems without disrupting the ongoing work in develop.
 - c. They are named hotfix/<issue-description>.
 - d. After applying the fix and testing it, they are merge back into both main and develop.

Conclusively, the Gitflow Workflow branching strategy, combined with thorough code reviews and CI pipelines, has brought about great development for managing complex projects with multiple developers. It keeps our codebase stable, makes releases predictable, and allows us to work efficiently without stepping on each other's toes.

This approach has not only improved the quality of our code but also reduced the stress around release times. For release, tools like CodeMagic are utilized to build the code directly from a github branch whenever it is ready for release