WEBT-VT | ADV | GIT Expert

User Story 1

As a quality assurance engineer, I want to use git blame to track down the introduction of bugs, so that I can aid in faster resolution.

Acceptance Criteria

- The QA engineer can use git blame to determine when a problematic line was introduced.
- The QA engineer can correlate the line's introduction with specific bugs or issues.
- The process allows for tagging or noting bugs in the version control system.
- The QA engineer can communicate findings effectively to the development team.
- Documentation is available to guide QA engineers in using git blame for bug tracking.

User Story 2

As a developer, I want to use git clean to remove temporary files and build artifacts, so that my working directory remains clean and organized.

Acceptance Criteria

- The developer can identify and list files that are not tracked by git.
- The developer can execute git clean to remove all untracked files and directories.
- The working directory is free from unwanted build artifacts after the operation.
- The developer receives a confirmation message when the clean operation is complete.
- The operation does not affect any tracked files or directories.

User Story 3

As a quality assurance engineer, I want to use git revert to undo problematic changes during testing, so that the test environment remains stable.

Acceptance Criteria

- The QA engineer can identify and revert specific commits that introduce test failures.
- The revert process is documented and communicated to the development team.
- The test environment accurately reflects the state before the problematic changes.
- The QA engineer can confirm that the reversion resolves the issues encountered during testing.
- The QA engineer provides feedback to the development team on the nature of the reverted changes.

User Story 4

As a developer, I want to know different ways of resetting a branch using git reset with the different options, so that I can undo local commits while keeping changes staged for further modifications.

Acceptance Criteria

- The developer can execute git reset option (–soft, –mixed, –hard).

- The command moves the current branch pointer to the specified commit without altering the working directory.
- Changes from the undone commits remain staged in the index.
- The developer receives a confirmation message indicating the reset operation is complete.
- The repository history reflects the reset, with local changes ready for new commits.

User Story 5

As a developer, I want to use git reflog to recover deleted commits, so that accidental data loss is minimized and changes can be restored.

Acceptance Criteria

- The developer can view the reflog to see a history of changes to HEAD.
- Deleted commits are listed in the reflog with their respective hashes.
- The developer can use the commit hash to reset the branch and restore the deleted commit.
- The developer confirms that the restored commit is accurately reflected in the branch history.
- The recovery process is documented to assist in future instances of data loss.

User Story 6

As a release manager, I want to create tags in the repository, so that I can mark specific points in the project history for releases.

Acceptance Criteria

- The release manager can create a lightweight tag using git tag .
- The release manager can create an annotated tag with descriptive metadata.
- The tag accurately points to the desired commit in the project history.
- The tag name follows the project's versioning scheme or tagging conventions.
- The release manager verifies the tag creation with git show .

User Story 7

As a quality assurance engineer, I want to test release packages generated from git archive, so that I can verify they are complete and function as expected.

Acceptance Criteria

- The QA engineer receives release packages generated using git archive.
- The engineer verifies that the package includes all files necessary for testing.
- The QA team runs tests to ensure the release package functions correctly in the target environment.
- Any discrepancies or issues found in the package are documented and communicated to the development team.
- Test results are recorded and used to validate the release process.