Git Flow Internal (Veda Studios):

General Workflow

Reference at:
[Introduction to GitFlow]

Naming Convention:

Master - master

Development - development/staging (both based on complexity of the project dev for lower-level integration testing and staging for the higher-level integration testing)

Release/hotfixes - release/+version_code+"." +subcode (for hotfix)

Features - feature/+work_id+"-" +issue_keyword(4words max)

Release tags - release/+clint_alias+version_code+"." + subcode (for hotfix)

Branching scopes:

Master

- Consists last stable code (generally in production)
- Only hotfixes or release branches can make the merge request
- Checkouts allowed only for the hotfixes

Hotfixes

- For production bugs
- Checkouts allowed only from master
- Perform the fix in the production code and then merge back to the master and then make release update and make the tag

Release

- Release level code for the testing of production level code and then merged to the master once tested and verified
- accepts merge requests only from the develop branch

Develop/Staging

- Consists final integrated code from the feature branches (Unit and UI tested)
- Is the final environment for the dev/staging level testing to perform integrated testing(/UAT)
- Checkout point for every feature branches
- Once fully tested and verified in staging environment is then merged back into release branch (of specific release version)

Features (Work Item Specific)

- Issue/Task/Work-Item level branch
- Number depends on number of features/issues

- Checked out from development as per feature requirement
- Once tested then merged back to development branch and deleted

PR (Pull Requests) Strategy

What is a Pull Request?

A pull request is a development process that provides a platform for discussion and review of a completed feature. Its purpose is to notify team members that the feature has been completed and it is open for discussion or modifications. The discussion usually occurs to improve the quality of the code; it is basically a code review process.

Note: In some sites or tools, you may find the term merge request instead of pull request. Both terms mean the same thing.

What needs to be done before creating a pull request?

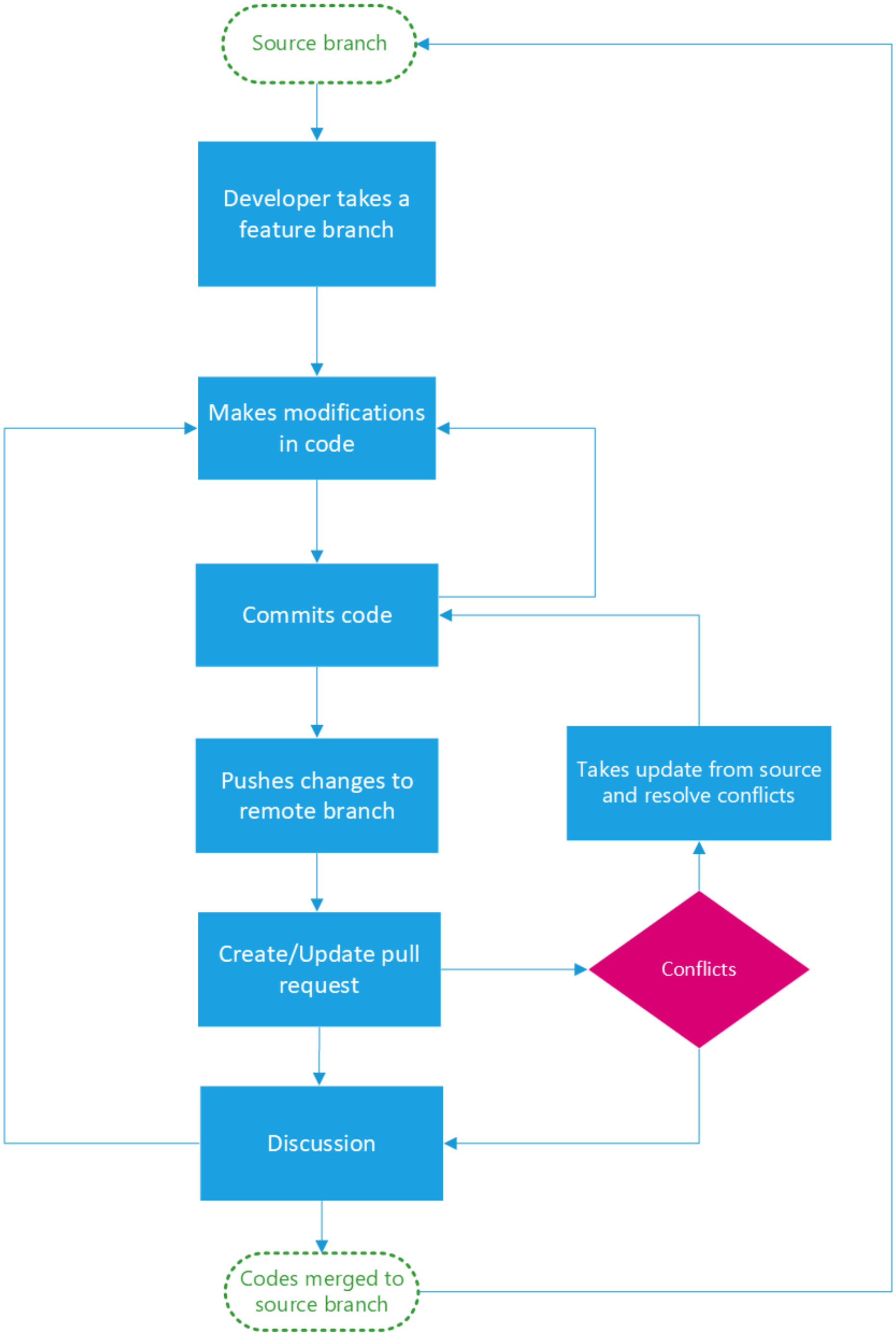
1. Commit code in your feature branch. Note that a feature can have any number of commits.
2. Commits made in the branch only reflect changes on the local machine (i.e., your machine). So, the commits need to be pushed to the remote branch.
3. Then, you're ready to rebase your branch. Rebasing is required if any new pull requests are merged after you have taken the feature branch.
4. After rebasing, any conflicts that arise need to be resolved, and the code needs to be pushed back to the remote branch.
5. Finally, it's time to create a pull request.

Note: Pull requests require two distinct branches. There must be some difference in the code between the taken branch and source branch to create a pull request.

Pull Request Process

1. When creating a pull request, you add a brief overview of your feature, select the branch to which the code needs to be merged, and select the assignee who will be reviewing it.
2. Once a pull request is created, it is open for discussion or modifications.
3. Sometimes conflicts occur after creating a pull request, and you must resolve these conflicts. Probable causes of conflicts are discussed later in this article.
4. Usually, the assigned person reviews the code, but it is not mandatory that only the assignee performs the review. Any team member can participate in the review process and give their feedback or discuss potential modifications to the code.
5. Any feedback or modifications are added in the form of comments near the code line.
6. The developer resolves comments and replies to the reviewer.
7. This process continues until all comments are resolved.
8. Once all discussions are resolved, the code is merged into the branch that was selected when the pull request was created.

Pull Requests flow


Source branch
Developer takes a feature branch
Makes modifications in code
Commits code
Pushes changes to remote branch
Takes update from source and resolve conflicts
Create/Update pull request
Conflicts
Discussion
Codes merged to source branch

Happy Coding!