Continuous Delivery – Using Feature Branch

Table of Contents
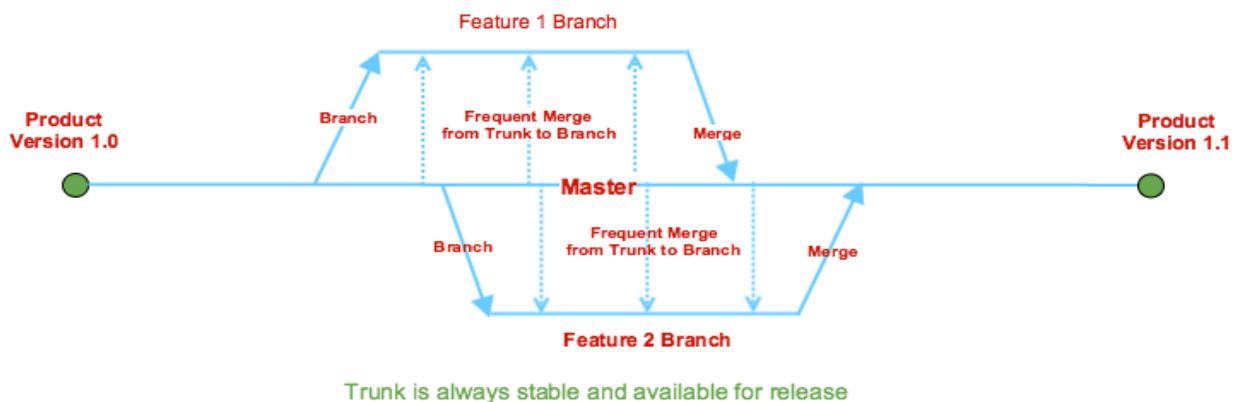
Continuous Delivery – Using Feature Branch

Important tenets of Continuous delivery

1. Feature branch and stable Master
2. Fully automated unit and functional testing
3. Continuous integration on Master and all Feature branches
4. Automated merging and merge issues alert mechanism

Feature Branch and stable Master

1. Add new feature means new feature Branch from Master.
2. Frequent merges from Master to feature Branch.
3. When feature is ready – merge Master into feature Branch.
4. Run Unit and functional test cases on feature Branch.
5. If tests pass on the feature Branch, merge feature Branch back to Master.
6. On merged code on Master – run Unit and functional test cases.
7. If tests pass on Master code, commit changes to Master.

Automate - Feature Branch and stable Master requirements

In addition to creating the feature branch user will have to trigger a Jenkins job with feature Branch URL and email recipients as the argument.

This Jenkins Job will do following tasks
1. Create a code stability build job.
2. Create a code quality build job.
3. Create a code merge Job from Master to feature Branch
4. Create a code merge Job from feature Branch to Master
5. Code build and code deploy Job.

Jenkins Jobs:

Master/Feature Branch	On Demand Job	Scheduled Job
Master	Code Stability build Job	Code Stability build Job
	Code Deploy	Code Quality build Job
Feature Branch 1	Code Stability build Job	Code Stability build Job
	Code Deploy	Code Quality build Job
	Merge - Master to Feature Branch	
	Merge - Feature Branch to Master	
Feature Branch 2	Code Stability build Job	Code Stability build Job
	Code Deploy	Code Quality build Job
	Merge - Master to Feature Branch	
	Merge - Feature Branch to Master	

Brief description of Jobs

Code stability build Job

This job will run at comparatively lower frequency (depends upon the overall build time required). This job will do following tasks

1. Compile the code and check if there are any compilation issues.
2. Run Unit Test cases (PHP Unit)
3. Report issues if any to predefined list of email recipients.

Code quality build Job

This job will run at comparatively higher frequency say twice a day and will do following

1. Build the code
2. Build the necessary infrastructure
3. Deploy the code
4. Run Unit Test cases (PHP Unit)
5. Functional Test Cases (Sahi)
6. Run Static Code quality analyzers (PMD etc.)
7. Reports issues/status to predefined list of email recipients.

Code Deploy Job

This Job will build the complete deployment infrastructure and deploy the build and make the environment available for end User testing.

1. Build the necessary infrastructure
2. Deploy the code (assumes build done)
3. Report issues/status to predefined list of email recipients.

Merge - Master to Feature Branch Job

This job will pick up all commits in Master, which are missing in branch and attempt to merge them in Feature Branch.

1. Checkout/Update the Feature Branch code.
2. Merge changes from Master to Feature Branch code.
3. Run all Unit and Functional test cases on the Feature Branch.
4. On success commit the changes to Feature Branch in version control.
5. Reports merge conflicts/test failures/status to predefined list of email recipients.

Merge - Feature Branch to Master

This is a more protracted process.
1. Run the Merge Master to Feature Branch Job.
2. Checkout/update the master branch code.
3. Merge the Feature Branch with Trunk.
4. Run all Unit and Functional test cases on the Master code.
5. On success commit the changes to Master in version control.
6. Reports merge conflicts/test failures/status to predefined list of email recipients.

Note:
Please note that to ensure that all merges and changes in code also keeps the DB in sync we will use DB deploy plugin of Maven. This will ensure that the DB remains in sync with the code changes.