# **CI & CD – High level Documentation**

**Flow**

Master Branch

PROD Env

Higher Environments

Create Release branch off Master

Merge Release branch into Master with unique Tag

CI/CD Pipeline

QA Env

DEV Env

CI/CD Integration

GIT CHECK OUT

SONAR

Release Branch

CHECKMARX

Merge feature/hotfix into Release

JUNIT

Create feature/hotfix branch off Release

Feature Branch - 1

DEPLOYMENT

NEXUS

Feature Branch - 2

SELENIUM TEST

Hot Fix/Bug Fix

BUILD PROMOTION

# **Branch Details:**

# **Master Branch:**

This branch will contain the current **PROD** code changes.

# **Release Branch:**

This branch will be checked out from **MASTER branch** by the developer which will contain the code changes for the current release.

~~This is a single branch and will not be created for every release~~.

We can have develop branch for each release (like, release-1.0, release-2.0). In this way, we can maintain repository for every release and have the track of repository/changes made for each release.

# **Feature Branch:**

This branch will be checked out from **DEVELOP branch** by the developer for developing the feature/story for the current release and it will be merged to release branch of current release and moved sequentially from DEV to PROD.

**Naming convention**: feature/{task name}

**Example**: feature/flex-okta-edge-service.

# **BugFix or HotFix Branch:**

This branch will be checked out from **DEVELOP branch** by the developer for bugs/issues identified in DEV or QA Environment and it will be merged to release branch of current release and moved sequentially from DEV to PROD

**Naming convention**: hotfix/{bug name}

**Example**: hotfix/flex-okta-login-auth-issue.

# **CI/CD Process Details:**

1. Current projects from SVN should be moved to Git. This will be the Master branch.
2. Develop branch will be created off from Master.
3. Feature branch for every feature will be created off from Develop branch.
4. Once code changes are done for the feature, feature branch should be merged to Develop branch.
5. CI/CD Engine will checkout the code changes from Develop branch and does the automated build and deployment with below process.
   1. SONAR will be executed, and it will check for code quality issues/warnings.
   2. Checkmarx will be executed, and it will run a security scan for checking any vulnerability in the code.
   3. Junit will be executed, and it will run the unit tests.
   4. Once all the above process is successful, changes will be deployed into the DEV Environment.
6. Once Unit Testing is completed in DEV Environment, Nexus Repository Manager should be triggered, and it does the below process.
   1. Automated Selenium scripts should be executed.
   2. Once the above step is successful, war will be copied from DEV to QA Environment by Nexus.
   3. Once QA Validation is completed, Nexus script should be triggered again which will copy the war from QA and deploy it in PROD Environment.
7. Once PROD Deployment is completed, Changes from develop should be Up merged to Master with a unique Tag.