Overview-Git process used for collaborative development,

This documentation describes the Git process used for collaborative development, focusing on the use of a development branch (**develop**), feature branches, and the creation of a release candidate (RC) branch. The RC branch is created with a XX.YY version and serves as a staging area for final integration and testing in staging environment before merging into the **Main** branch.

Branching Strategy

1. **Main Branch (main)**: The **main** branch represents the production-ready code. It should always contain stable and tested code. Direct commits to the **main** branch are restricted (the branch is protected), and changes should be made through pull requests.
2. **Dev Branch (develop)**: The **develop** branch is the integration branch for ongoing development. It acts as a staging area for merging feature branches and undergoing further testing.
3. **Feature Branches (CHEET-XXX)**: Feature branches are created off the **develop** branch for implementing specific features or fixes. Each feature branch should have a descriptive name that reflects its purpose.
4. **Release Candidate (RCXX.YY) Branch**: The RC branch is created every 2 weeks to represent a release candidate with a specific XX.YY version. It serves as the final testing and integration stage before merging into the **main** branch.

Workflow

1. **Clone the Repository**: Start by cloning the Git repository to your local machine using the git clone command.
2. **Create a Feature Branch**: Create a new feature branch from the latest **develop** branch using the command git checkout -b CHEET-number-ticket-name.
3. **Implement the Feature**: Make changes, write code, and commit your work to the feature branch using git commit.
4. **Push the Feature Branch**: Periodically push your feature branch to the remote repository using git push origin CHEET-number-ticket-name. This step ensures your work is backed up and visible to other team members.
5. **Submit a Pull Request**: When the feature implementation is complete and tested locally, submit a pull request on GitHub to merge your feature branch into the **develop** branch.
6. **QA Review**: The QA team reviews the pull request, tests the feature, and provides feedback or identifies issues that need to be addressed.
7. **Code Review:** Other developers on the team review the pull request, examine the code changes, and provide feedback or suggestions. At

least two reviews are required for each feature branch.

8. **Address Feedback**: Make any necessary adjustments or fixes based on the feedback received during the QA and code reviews. Commit and push the changes to the feature branch.
9. **Merge to develop**: Once the feature branch is approved by QA and the code reviews, it is merged into the **develop** branch using the "Squash and Merge" option on GitHub.
   Open image-20230628-102909.png

10.

11. **Create Release Candidate (RC) Branch**: Every two weeks, create a new release candidate (RC) branch with a specific XX.YY version from the latest **develop** branch using the command git checkout -b RCXX.YY.
12. Creation of the rc-XX.YY branch means a feature **freeze for the XX.YY release**. Only bugfix PRs should be allowed.
13. **Continuous Integration/Testing**: The CI/CD pipeline automatically triggers the build and testing process on the RC branch to ensure the integration of new features does not introduce regressions.
14. **Testing and QA on RC**: The QA team performs extensive testing on the RC branch, focusing on the integrated features and overall stability.
15. **Bug Fixes and Updates**: If issues or bugs are identified during testing on the RC branch, fixes and updates should be made on the respective feature branches, followed by merging those branches into the **develop** branch and re-merging them into the RC branch as needed.
16. **Merge to main**: After thorough testing and validation on the RC branch, it is merged into the **main** branch, marking it as a new stable release.

Steps to get there :

OPDE-3777: DCS - CI on develop branch
RELEASE DONE

- change cloud build jobs : develop trigger
- configure develop to be protected like rc-* branches
update documentation

other repos : FEM, wordpress,
People to inform
Release manager
Iron Dev