**Problem Statement**

You work as a Devops Architect in Zendriix Softwares. The company has been struggling to manage their product releases. The releases should happen on 25th of every month.

Suggest a Git Workflow Architecture for this requirement.

Simulate this workflow, by creating a pseudo code files and branches, and upload the same to your GitHub Account. As a part of solution, share the link to your GitHub repository.

**Solution Approach**

To address the problem of managing product releases on a fixed schedule (e.g., the 25th of every month), I recommend the following **Git Workflow Architecture**.

**Proposed Git Workflow Architecture:**

1. **Branches:**
   * **main branch:** Contains production-ready code. This is the branch where final releases are tagged.
   * **develop branch:** Acts as an integration branch for feature development. It contains all the features that are ready for the next release.
   * **Feature branches:** Used for developing individual features (feature/<feature-name>).
   * **Release branches:** Created for preparing the code for the scheduled release (release/<release-version>).
   * **Hotfix branches:** For urgent bug fixes on production code (hotfix/<issue-name>).
2. **Workflow:**
   * Developers work on **feature branches** and merge them into the develop branch once the feature is complete.
   * On or before the 25th, a **release branch** is created from develop. This branch is used for final testing, bug fixing, and polishing.
   * After testing, the release branch is merged into both main and develop, and a **release tag** (e.g., v1.0.0) is created on the main branch.
   * Urgent fixes to production are handled via **hotfix branches**, merged into both main and develop.

**Solution Repository:** <https://github.com/vyombhatt/zendriix-softwares-workflow.git>

**Simulation of Workflow:**

Here’s how we have simulated the workflow:

**Step 1: Initialize Repository and Create main and develop Branches**

1. mkdir zendriix-release-workflow

2. cd zendriix-release-workflow

3. git init

4. echo "Zendriix Project" > README.md

5. git add README.md

6. git commit -m "Initial commit on main branch"

7. git push

**Step 2: Create Feature Branches**

1. git branch develop

2. git checkout develop

3. git checkout -b feature/feature1

4. echo "Code for Feature 1" > feature1.txt

5. git add feature1.txt

6. git commit -m "Add Feature 1"

7. git push

8. git checkout develop

9. git merge feature/feature1

**Step 3: Create Release Branch**

1. git checkout -b release/v1.0.0

2. echo "Release Notes: Version 1.0.0" > release\_notes.txt

3. git add release\_notes.txt

4. git commit -m "Prepare release v1.0.0"

5. git push

**Step 4: Merge Release Branch to main and develop**

1. git checkout main

2. git merge release/v1.0.0

3. git tag v1.0.0

4. git checkout develop

5. git merge release/v1.0.0

**Step 5: Handle Hotfix (if needed)**

1. git checkout -b hotfix/fix-critical-bug main

2. echo "Fix for critical bug" > fix.txt

3. git add fix.txt

4. git commit -m "Fix critical bug"

5. git push

6. git checkout main

7. git merge hotfix/fix-critical-bug

7. git checkout develop

8. git merge hotfix/fix-critical-bug

**GitHub Repository Link:**

Final GitHub Repository Link:

<https://github.com/vyombhatt/zendriix-softwares-workflow.git>