**Suggested Git Workflow Architecture for Scheduled Releases**

For managing scheduled product releases on the 25th of every month, I recommend using a **Git Flow** workflow with additional considerations for timed releases. Here's how it would work:

1. **Master Branch**: Holds the production-ready code that is released on the 25th of every month.
2. **Develop Branch**: Serves as the integration branch where all new features and fixes are merged.
3. **Feature Branches**: Created from the develop branch for working on individual features.
4. **Release Branches**: Created from the develop branch around the 15th of the month to stabilize the code for the scheduled release.
5. **Hotfix Branches**: Created from the master branch for urgent bug fixes that need to be released immediately.

**Workflow Steps**

1. **Feature Development**: Developers create feature branches from develop, work on features, and merge them back into develop.
2. **Stabilization**: On the 15th of the month, a release branch is created from develop to begin final testing and bug fixing.
3. **Final Release**: On the 25th, the release branch is merged into master, tagged with the version number, and pushed to production.
4. **Hotfixes**: If urgent issues are found post-release, a hotfix branch is created from master, fixed, and merged back into both master and develop.

**Simulating the Workflow**

I'll create a simulated workflow using pseudo code files and branches to represent this architecture.

**Steps to Simulate and Push to GitHub**

# Initialize a new Git repository

mkdir zendrix\_gitflow\_simulation

cd zendrix\_gitflow\_simulation

git init

# Create the main branches

git checkout -b develop

git checkout -b feature/new-feature develop

git checkout -b feature/another-feature develop

# Simulate feature development

echo "Feature 1 code" > feature1.txt

git add feature1.txt

git commit -m "Add feature1.txt with new feature code"

echo "Feature 2 code" > feature2.txt

git add feature2.txt

git commit -m "Add feature2.txt with another feature code"

# Merge features into develop

git checkout develop

git merge --no-ff feature/new-feature

git merge --no-ff feature/another-feature

# Create release branch on 15th

git checkout -b release/v1.0 develop

# Simulate final tweaks

echo "Final tweaks before release" > final\_tweaks.txt

git add final\_tweaks.txt

git commit -m "Final tweaks before release v1.0"

# Merge release into master on the 25th

git checkout master

git merge --no-ff release/v1.0

git tag -a v1.0 -m "Release v1.0 on 25th"

git push origin master

git push origin v1.0

# Merge release back into develop

git checkout develop

git merge --no-ff release/v1.0

# Simulate a hotfix

git checkout master

git checkout -b hotfix/urgent-fix

echo "Hotfix for urgent issue" > hotfix.txt

git add hotfix.txt

git commit -m "Add hotfix for urgent issue"

# Merge hotfix into master and develop

git checkout master

git merge --no-ff hotfix/urgent-fix

git checkout develop

git merge --no-ff hotfix/urgent-fix

git branch -d hotfix/urgent-fix

# Push all branches to GitHub

git push origin develop

git push origin master

**GitHub Repository**

git remote add origin https://github.com/Shahid199578/zendrix-gitflow-simulation.git

git push -u origin --all