Development Cycle

30 March 2021 11:12

Cycle

1. Planning
2. Design
3. Development
4. Testing
5. Staging
6. Publish

Planning

1. Plan upcoming sprint on Fridays and divvy out tasks

Design

1. If the change is complicated, review design and plans with the team or a partner
   a. A partner can be someone working on similar tasks
   b. Separate meetings are okay for complicated tasks/design
2. Briefly review the plan with the team during daily stand up

Development

1. Pull from develop branch to make sure it's up to date
   a. git checkout develop
   b. git pull
2. Create a feature branch off of develop
   a. git branch <branchname>
3. Checkout your new branch and push it to the remote repository
   a. git checkout <branchname>
   b. git push --set-upstream origin <branchname>
4. Complete development and push to GitHub
   a. git add *
   b. git commit -m "commit message"
   c. git push

Testing

1. Do one round of testing:
   a. Run 'docker-compose up -d --build' and verify there are no build errors
   b. Visit http://localhost:3000/ and verify changes are as expected
   c. Complete the relevant test script (in progress)
2. If change is complex/integrated, have another person repeat testing.
3. If issues are found, fix them and restart testing.
4. Switch to develop branch
5. Do one round of testing to make sure develop works before your change.
   a. If that doesn't work, contact person who made last commit
6. Do a local merge
   a. git checkout develop
   b. git pull
   c. git merge <branchname>
7. Do one round of testing:
   a. Run 'docker-compose up -d --build' and verify there are no build errors
   b. Visit http://localhost:3000/ and verify changes are as expected
   c. Complete the relevant test script (in progress)
8. If complicated and integrated, have another person repeat testing.

9. If issues are found, fix issue in feature branch and restart testing.

Staging

1. In Github, create a pull request merging your feature branch into the development branch.
2. Delete feature branch

Publish

1. Merge development into main on a bi-weekly basis.
2. Use main for demonstrations