**Pull Request Checklist**

**Pull requests not ready for review**

* Create the PR as draft
* Do not assign a reviewer
* When ready for review
  + Publish pull request
  + Assign a reviewer

**Pull requests ready for review**

* Create
* Assign a reviewer

**If a pull request has changes**

* Reviewer adds all comments
* Reviewer sets status to Wait for author
  + If there is some chance that the reviewer will not be available to approve the changes after they are made, this step should be skipped
* Developer makes changes
* Developer sets the comment status to pending
* Developer notifies the reviewer when it is ready for review again (outside of DevOps)
* Reviewer either
  + resolves comment
    - The fix is good to go
  + sets as will not fix
    - There is a problem but not going to address now
    - Create a task to address later **??????????**
  + Sets as closed
    - Turns out that it really wasn’t an issue
  + adds additional comments and sets back to active

**Pull request needs significant additional work**

* Developer changes the pull request to a draft

**Once a code review is complete**

* Reviewer merges into develop/master
* If there are additional changes like merging in updates from develop
* The reviewer approves and sets auto-complete
* Either the reviewer or original developer can do the merge

**Assigning a reviewer**

* If you are the architect or architect is not around, assign the lead developer group
* Otherwise assign architect group (this is the default)
* Additional reviewers may be assigned if you believe that their insights will bring additional value to the pull request

**Exceptions**

* Anyone can create and complete the following PRs and do not need to be reviewed
  + master-->develop
  + develop-->master
  + Pull requests not involving develop or master
* If a high priority hotfix needs to go out the waiting for a reviewer can be skipped but must still be reviewed ASAP