Development summary

- Use Branch permissions -> Only pull request allowed

General

1. Work on feature branch
   - Writes own unit-tests
   - Gets updates from development branch when needed
   - Push code to repository -> CI
2. Pre-Publishing flow from feature to development branch
   - Final update from development branch
   - unit-tests & build are ok
3. Pull request to development branch
   - Pull request unit, integration and regression tests -> Development Build

Release

Once it's time for release a release branch is created from development

- Semantic versioning is used
- Comments review
- Cleanup
- Branch gets tagged + merged into Development & Master branches -> Release Build

Hotfix

Used when there is a urgent fix to do in production

1. Merge of from development(Up to date with master)
2. Fix bug
3. Merge back directly into master with the right semantic versioning

Sources

- https://medium.com/devsondevs/gitflow-workflow-continuous-integration-continuous-delivery-7f4643abb6
- https://hackernoon.com/gitflow-is-a-poor-branching-model-hack-d46567a156e7
- https://www.agile-im.de/2017/01/10/continuous-deployment-gitflow/
- TOP ressource: https://hackernoon.com/how-we-transitioned-from-traditional-agile-to-agile-ci-cd-2ef9b7a
- https://blog.seibert-media.net/blog/2016/03/09/der-continuous-delivery-workflow-fuer-gitflow-projekte/

Alternative flows

- https://blogs.technet.microsoft.com/devops/2016/06/21/a-git-workflow-for-continuous-delivery/

Overview

- https://jp-lambert.me/git-gitflow-and-continuous-integration-for-dummies-5e4300148