Github link:

* Delt repo:

<https://github.com/MathiasGeCh/Calculator-shared>

* Egen:

https://github.com/tobias-roy

Gitlab/GitHub

CI:

Continuos Integration, er automatisering af test af kode. I Gitlab kan der med hver push af noget kode kan man lave scripts der automatisk tester den kode der er. Det vil altså sige at hvis der bliver pushet noget ny kode, opdager man hvis det resultere i fejl på det kode der er.

Hver ændring bliver altså bygget og testet automatisk.

GitHub / Gitlab

Gitlab har continuos integration og tilbyder altså built in at man kan benytte den service. Dette er ikke noget GitHub tilbyder, he skal man selv integrere med en 3-part som eks. Jenkins eller lignende.

Gitlab giver den fordel at det er en komplet løsning, GitHub kan du selv vælge hvilken 3-parts løsning du vil bruge hvis du vil bruge CI.

Workflowet er også forskelligt hos lab/hub. GitHub ligger op til at man skal merge sine egne branches med en main branch, for hurtigt at kunne spytte det nye ud.

GitLab ligger mere vægt på at lave mange stabile branches og altid benytte en pre-production og production branch.

GitLab tilbyder den komplette software løsning for at komme fra HelloWorld til kundens udgave.

Der er også forskel på hvilke tredjeparts integrationer de er med til at samarbejde omkring. GitLab har eks. Jira integration, det har Github i sig selv ikke, men her skal man benytte 3-part services.

Versionsstyrings Strategi:

Branches:

* Produktion
* Test
* Udvikling

1. Præsenteret med et problem
2. Design din løsning
3. Design review af tredjepart for at godkende/afvise/optimere design
4. Lav din kode
5. Test din kode
6. Codereview af tredjepart for at godkende/afvise/optimere kode
7. Commit og merge til udvikling
8. Merge Udvikling branch til Test branch
9. Test alt der er merged
10. Merge til Produktion
11. Udgiv