

Way of working

Contents

1	Inledning	1
2	Sprinter	1
3	Projektplanering.....	1
4	Code review	1
5	Definition of done	1
6	Git.....	1

1 Inledning

Detta dokument beskriver vårt arbetssätt.

2 Sprinter

Vi arbetar i en-veckas-sprinter från fredagar till nästföljande torsdag.

Varje torsdag ska all fungerande kod från veckans sprint vara pushat på develop-branch.

Sprint review och retrospective sker därefter. Efter detta planerar vi nästa veckas sprint.

3 Projektplanering

Projektplanering sker på vårt clubhouse-workspace.

Alla stories sorteras som "Unscheduled" i lämpliga epics beroende på område. Detta är vår produkt-backlog.

Under sprint-planering taggas alla stories med label för kommande sprint och flyttas från Unscheduled till "Ready for development" alternativt "Blocked" om de är beroende av andra stories i samma sprint.

Blocked-status kan även användas under pågående sprint om förhinder dyker upp på en story man arbetar med.

När arbete påbörjas på story sätter man sitt namn som Owner och flyttar storyn till "In Development".

När storyn är klar (se Definition of done), flyttas den till Completed.

4 Code review

Före merge mot development ska all kod vara granskad mot våra Code guidelines.

Granskning sker av annan person än den som skrivit koden.

Granskare lämnar en kommentar på storyn efter godkänd code review, med commit-id.

Exempelvis "Code review commit 40cf8313ed5 OK".

Om granskare och ansvarig utvecklare inte är överens kring tolkning av Code guidelines så har SCRUM master sista ordet.

5 Definition of done

En story anses vara klar när:

- Kod som uppfyller kraven för storyn är skriven.
- Koden är granskad och godkänd
- Koden är committad på development-branchen

6 Git

Efter färdig sprint pushas development-branchen till master.

Under pågående sprint sker allt arbete mot development-branchen.

1. Skapa branch från development
2. Skriv kod.
3. Committa kod på samma branch
4. Be kollega granska koden (se Code review)

Vid behov så kan man göra flera commits i samma branch utan att granska koden. All kod som mergeas med development måste dock vara granskad.

5. Mergea din branch med development-branchen
6. Deletea branchen du jobbade i (lokalt och på repot om du pushat den)

Det rekommenderas att committa och mergea mot development dagligen, men detta är inget krav.

*Mellan punkt 3 och 4: Pusha upp kod till respektive GitHub repo.