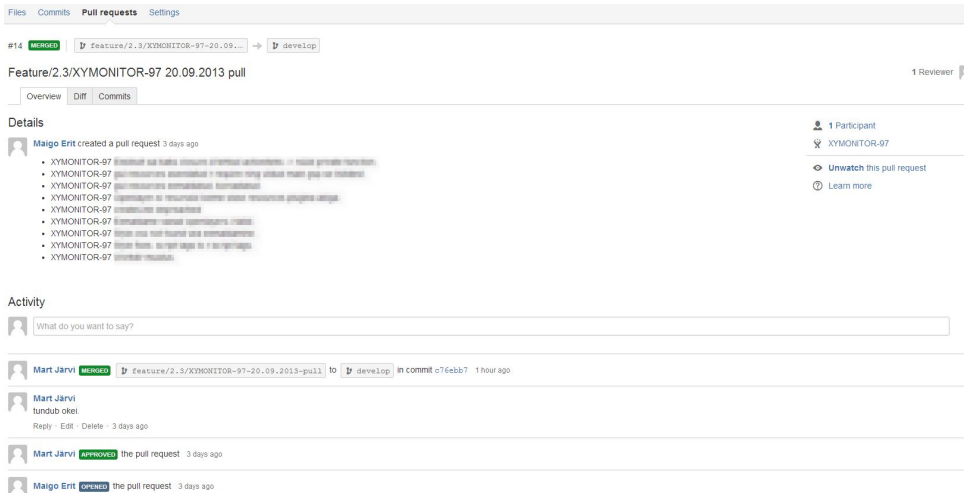
Lähtekoodi halduse ja ehitamise nõuded

Versioonihalduse kasutamine

- Tarkvara lähtekoodi halduseks tuleb kasutada https://source.smit.sise aadressil asuvad GIT repositooriumi (ligipääs antakse projekti põhiselt).
- Tarkvara versioonihalduses jälgitakse üldises mõistes ***Git-flow*** protsessi (http://nvie.com/posts/a-successful-git-branching-model/) või ***Feature-branche-workflow*** (https://www.atlassian.com/git/tutorials/comparing-workflows/feature-branch-workflow) protsessi.
- **Git-flow** lähenemist on sobilikum kasutada siis, kui tarkvara versioonide väljastamine on aeglasem ja harvem ning versioonides toimuvad stabiliseerimisperioodid, samuti kui on vaja mitut erineva funktsionaalsusega versiooni pikaajaliselt toetada. Teine mudel sobib neile kes on rohkem automatiseerinud oma paigaldus ja tarneprotsessid ning kasutavad kas ***Continuous Delivery*** või ***Continuous Deployment*** töövooge.

- Iga JIRA pileti realiseerimise alguses loob arendaja JIRA abil (kasutades JIRA sees pileti juures käsku create branche) ***"develop"*** nimelisest harust endale vastava konvensiooni (vt. joonis) põhiselt ***"feature/xxx"*** nimelise haru, kus arendust tehakse.
- Konfliktide vältimiseks peab kesksest harust järjepidevalt enda harusse muudatused mestima.
- Piletis realiseeritud lähtekoodi üleandmiseks SMIT-ile tuleb minna versioonihaldus keskkonna (https://source.smit.sise) vastava tarkvara ruumi ning valida sealt tab ***pull-requests*** ning luua uus ***pull-request***, kus tuleb määrata lähteharuks enda tehtud arendusharu ning lõppharuks keskne haru. Ülevaatajaks tuleb valida SMIT poolne kontakt ning sisuks täiendavad kommentaarid, mida silmas pidada antud tarnes (näiteks et muutus konfiguratsioon või baas vms).
- Kui meeskond on kasutusele võtnud koodi ülevaatustes, AI assistendi siis tuleks esimesena tähelepanu pöörata automaatsest tagasisidest, mis peaks tekkima 30s jooksul peale pull-requesti tegemist.



- Tarne üleandmise eelduseks on, et üleantav kood vastab kõikidele kokkulepitud nõuetele, mis arenduse alguses on fikseeritud või on konkreetsed puudused toodud välja *pull-requesti* kirjelduses;
- Tarne loetakse vastuvõetuks, kui on toimunud vastavad tegevused:

    a. Bamboo ehitusserver on edukalt üleantava haru ära ehitanud
    b. SonarQube analüsaator näitab, et SMIT poolt kasutatav ***Quality Gate*** on läbitud
    c. SMITi poolne vastuvõtja on tarnele teinud koodi analüüsi ja need heaks kiitnud
    d. Kood on ilma konfliktideta süsteemi poolt mestitud ***"develop"*** harusse (***pull-request*** on täidetud)
    e. Kõik tarnes sisalduvad ***commitid*** on seotud konkreetsete JIRA piletinumbritega kujul XXXX-YYY

- ***Pull-request****-i* kinnitamisel automaatselt kustutatakse vastav arendaja poolt tehtud haru ära, kui mestimine on olnud edukas;
- ***Pull-request***-i võib SMIT tagasi lükata, kui seal esineb puudusi või alustada seal sees dialoogi puuduste kõrvaldamiseks (koodi ülevaatuse tegemisel lisatakse kommentaarid otse koodi ridade vahele);
- ***Pull-request***-i kirjeldus peaks kokkuvõtlikult selgitama üleantava tarne sisu (AI olemasolul saab kasutada vajadusel tarne sisu genereerimiseks)

Bamboo (CI/CD) kasutamine

- Igal tarkvaral on bamboos defineeritud 1 ehitusplaan, mis ehitab ennast "***develop***" või "***master|main***" haru pealt automaatselt (haru valik sõltub, kumba protsessi kasutatakse koodi halduseks).
- Arendajad peaksid oma arendusi tegema "***feature"*** harudes, mida automaatselt Bamboo on võimeline ehitama. Samuti saab vajadusel seadistada Bambood ehitama kõik harusid, mida git-ist leitakse (https://docs.atlassian.com/bamboo-specs-docs/9.4.1/specs.html?yaml#plan-branches)
- Bamboo ehitusplaan ehitab koodi, teeb koodile staatilist analüüsi, võimalusel turvaanalüüsi, jooksutab testid ning olenevalt harust siis paigaldusplaan paigaldab lõpuks rakenduse määratud keskkonda (feature harude puhul paigaldusi ei toimu).
- ***Git-flow*** puhul on reeglina ehitusplaan liidestatud "***develop"*** haruga ning paigaldatakse tulem arenduskeskkonda, testi ja toodangu jaoks versioonid tekivad harude pealt ("***release***" harud reeglina), mida paigaldatakse Bamboo kaudu käsitsi.

- **Feature-branche-workflow** protsessi puhul on ehitusplaan liidestatud "***master|main"*** haruga, mille tulemus paigaldatakse automaatselt testkeskkonda. Võimalus on selle kõrvale luua ka täiendavaid harusid ja siduda neid konkreetse keskkonnaga - näiteks **"*develop*"** paigaldatakse automaatselt arenduskeskkonda.
- Toodangu keskkonda paigaldus tehakse käsitsi Bamboo sees ning sinna paigaldatakse sama tulem, mis läks testi.
- Bamboo paigaldusprojekt luuakse eraldi git-i spec repona (nimekuju xxx-deploy) ning Bamboo paigaldusprojekti sees on iga keskkond defineeritud eraldi Bamboo "env"-ina (https://docs.atlassian.com/bamboo-specs-docs/9.3.0/specs.html?yaml#environments).