# PGR301 DevOps eksamen 2021

# Oppgave – DevOps

* *Beskriv med ord eller skjermbilder hvordan man kan konfigurere GitHub på en måte som gir bedre kontroll på utviklingsprosessen. Spesielt med tanke på å hindre kode som ikke kompilerer og feilende tester fra å bli integrert i main branch.*
* *Beskriv med ord eller skjermbilder hvordan GitHub kan konfigureres for å sikre at minst ett annet medlem av teamet har godkjent en pull request før den merges.*
* *Beskriv hvordan arbeidsflyten for hver enkelt utvikler bør være for å få en effektiv som mulig utviklingsprosess, spesielt hvordan hver enkelt utvikler bør jobbe med Brancher i GitHub hver gang han eller hun starter en ny oppgave.*

For å få god kontroll på arbeidsprosessen kan man sette opp «Branch protection rules», i Settings > Branches > Add Rule.

«Require a pull request before merging» betyr at man ikke kan pushe rett til main, men at man må gå gjennom en annen branch og så lage pull requests. Man kan sikre at andre medlemmer har godkjent pull requesten ved å velge «Require approvals».

NB: Man må også velge «Include administrators» om det skal gjelde for absolutt alle.

Text

Description automatically generated

For å forhindre at man får feilende tester i main branchen, kan man sette opp å kreve status checks før man merger. I bildet under er det valgt at «build» is Github Actions skal bli godkjent før man kan merge en pull request. I eksempelet her vil koden prøve å builde Maven med ‘mvn -B package –file pom.xml’. Om det failer, kan man ikke merge. Man kunne også *kun* kjørt tests, f.eks ‘npm test’ i et prosjekt som bruker npm, f.eks en nettside i React.

A screenshot of a computer

Description automatically generated with medium confidence

For å få en effektiv arbeidsflyt, burde nye oppgaver ha egne branches, som igjen er navngitt med beskrivende navn. Man kan også knytte potensielle github issues opp mot branch ved å bruke github issues ID i navnet på branchen. Her kan forskjellige firmaer ha forskjellige konvensjoner på hvordan de navngir branches, men et eksempel vil være «tests/added-tests-to-cover-doSomething/issueID». Ikke «fix-new-test».

Drøft:

*SkalBank har bestemt seg for å bruke DevOps som underliggende prinsipp for all systemutvikling i banken. Er fordeling av oppgaver mellom API-teamet og «Team Dino» problematisk med dette som utgangspunkt? Hvilke prinsipper er det som ikke etterleves her? Hva er i så fall konsekvensen av dette?*

Her trenger vi litt drøfting