

IN1000-Seminar

Før innledning:

Nå som dere har satt dere sammen er det på tide å bli kjent med hverandre som et team. Ta en navnerunde der dere også nevner hvilken linje dere går, hvor mye programmeringserfaring dere har og hvor mye erfaring dere har med slikt gruppearbeid.

I team er det viktig at alle er enige om hva målet er, diskuter derfor disse spørsmålene med hverandre:

- Tenker dere å jobbe videre etter seminaret?
- I såfall, hvor ofte vil dere møtes?
- Skal det være noe form for sanksjon hvis noen kommer for seint til neste møte eller ikke gjør det de lover? Det trenger ikke være noe stort, bare nok til at det er en liten konsekvens.
- Tenker dere å dra på gjennomgang av oppgaven 29. November og diskutere løsningsforslag?
- Når er dere ferdig med oppgaven, hva er deres *definicion of done*?
- Skal dere ha en retrospekt i løpet av prosjektet? (vi anbefaler å ta 10-15 min til det før pizza, men man kan også ta det ved en senere anledning)

Hvis dere vil kan man skrive ned dere kommer frem til så man har dokumentert hva man ble enige om, dette er lurt for store prosjekter da det kan forekomme konflikter eller misforståelser rundt de tingene dere har diskutert.

Før Programdesign:

Når man jobber med store problemer og flere mennesker er det lurt å bli enige om hvordan programmet skal se ut før man går løs på programmeringen. Les gjennom Programdesign i fellesskap og gjør **oppgave a**.

Før b) Antall prosessorer og minnekrav:

Etter at dere har laget en datastruktur, er det på tide å planlegge hvordan dere skal gå frem under programmeringen og hvem som skal begynne med hvilke roller, her kan det være lurt å lese gjennom hele oppgaveteksten før man begynner planlegging også. Man kan både parprogrammere og bruke mob-programming For mob-programming kan det være lurt å sitte på et kollokvierom. Når dere har lagt en plan er det bare å gå løs på **oppgave b, c og d!**

Parprogrammering går ut på at to personer jobber med samme kode på samme maskin, der en av dem skriver koden og den andre forteller hva som skal skrives, eller navigerer. Still hverandre spørsmål når dere er usikre eller uenige. Dette kan virke som en tungvint arbeidsmetode noen ganger, men gjør at man får tenkt mer grundig gjennom koden før, og når den skrives. Bytt på rollene!

Mob-programming foregår likt som parprogrammering men her er det flere med navigatør-rollen. Her har man fordelene av input fra flere personer, men pass også på at alle får delta i diskusjonen, og at man holder seg til tempoet til den som skriver, så ikke de blir stresset. Bytt på rollene!

Hvis du jobber alene kan du prøve deg på *rubberduck debugging* i **oppgave c**. Dette er en form for debugging der man finner noe å forklare koden til, dette kan f.eks. Være en badeand. Poenget her er at man gjennom å forklare koden, tvinger seg selv til å se på koden med nye øyne, og dermed kan oppdage feil eller forbedringspotensialer. Badeanden er der mest for at man lettere skal kunne leve seg inn i øvelsen.

Etter oppgaveløsning:

Hvis dere har parprogrammert eller jobbet individuelt, orienter hverandre om hvordan de forskjellige delene av programmet ble. Har dere gjort noen avvik fra datastrukturtegningen? Hvorfor?

Her kan det være lurt med en siste retrospektiv om prosjektarbeidet. Under lange prosjekter brukes tilbakemeldingene fra retrospektiv for å se hvordan man kan forbedre teamarbeidet underveis, men det er også nyttig for alle i teamet å ta med seg erfaringer videre til neste prosjekt.

Et par nyttige spørsmål for retrospektiv:
Hvordan hånderte dere evt. endringer i planen?
Hvordan var tidsforbruket deres?
Hvordan brukte dere personene på gruppen?
Hva kan forbedres?
Hva gikk bra?

