

Individuell rapport

Stud nr. 561791, Gr nr. 27

Tema: 00 Arbeidsvaner

Antall ord: 1507

6. desember 2022

Innhold

1	Innledning	1
2	Gruppens Arbeidsvaner	1
2.1	Gruppemøter	1
2.2	Scrum-modellen	2
2.3	Parprogrammering	2
2.4	Ugunstige arbeidsvaner	3
2.5	Fletting av kode	3
3	Individuelle Arbeidsvaner	4
4	Tiltak for fremtiden	4
	Referanser	5

1 Innledning

Gode arbeidsvaner er viktig for å kunne utføre en oppgave effektivt og godt. De fleste opparbeider seg sine egne rutiner og arbeidsvaner, etter erfaringer man gjør seg når man løser og gjennomfører ulike oppgaver. Så når man samarbeider med andre, kan man av og til støte på utfordringer hvis man angriper situasjoner anderledes. I boken *A Mind For Numbers* [1] forteller Barbara Oakley om hva som er gode arbeidsvaner når man skal arbeide med problemer innenfor matematiske og teknologiske fag. Mange av teknikkene Oakley presenterer som chunking, jevnlig repetisjon, og å variere problemløsningsteknikker er vaner som bevist og ubevist har vært viktige for vårt arbeid med prosjektet.

Denne rapporten tar for seg individuelle arbeidsvaner, og arbeidsvaner vi som gruppe har blitt enige om.

2 Gruppens Arbeidsvaner

Ved starten av semesteret skrev vi som gruppe en kontrakt der vi blant annet protokollførte noen forventninger om hvilke arbeidsvaner vi skulle ha; både individuelt og som gruppe. I kontrakten avtalte vi minimum ett møte i uka, torsdager kl. 09.00 - 11.00. Dette var for å sikre at gruppen møttes hver uke, og legge opp til å kunne møtes mer enn det. I praksis ble denne møtetiden alltid utvidet etter kl 11.00, men den ble da uforpliktende slik at vi kunne gå og komme som vi ville.

Utover i semesteret, etterhvert som omfanget av arbeidet ble større, utvidet vi uformelt antall avtalte møter i uka til to, med 4-8 timer avsatt tid til felles arbeid. Grunnen til det var at behovet for direkte samarbeid økte på linje med arbeidsmengden, og at vi utfylte hverandres oppgaver bedre i lag.

2.1 Gruppemøter

Gruppemøtene hadde ingen bestemt fast struktur, men startet som regel med at vi gikk gjennom eventuelt arbeid vi hadde gjort på egenhånd, for å flette kode-grenene våre sammen. Hvis vi ikke hadde blitt ferdig, eller det var noe vi satt fast med, jobbet vi sammen to, tre eller av og til hele gruppen, med å løse blokaden.

Videre utover i møtene diskuterte vi hva som sto igjen av arbeid, og delte opp arbeidet i små, håndterlige, utviklingsoppgaver. Disse utviklingsoppgavene fordelte vi innad i grup-

pen ut ifra hva vi kunne håndtere og følte vi kunne løse. Et mål for møtene ble da å ikke øke backloggen betraktelig mer for hver gang vi møttes, uten å ha løst eller ferdigstilt noen av oppgavene før vi dro.

2.2 Scrum-modellen

Scrum-modellen har vært en viktig del for arbeidet til gruppen, og har hjulpet oss med å få oversikt over, dele opp, og fordele arbeidet i gruppen. I henhold til retningslinjene og anbefalingene fra faget har vi ikke delegert hvert av gruppemedlemmene en egen rolle, som er å finne i Scrum guiden [2]. Hvert medlem har derimot fått prøve ut hvordan det er å være henholdsvis produkteier, Scrum master, og en team arbeider. De gangene det har vært nødvendig å ha en av disse rollene har gruppen som fellesskap samarbeidet om rollen, slik at alle har fått bidra på de ulike områdene.

Vi har arbeidet aktivt etter tre sprinter, en for hver del-innlevering av prosjektet. For hver sprint har vi i fellesskap opparbeidet en backlog med utviklingsoppgaver som inneholder arbeidet vi må gjennomføre for å fullføre sprinten, og dermed også innleveringen.

Noen utfordringer vi har hatt med tanke på Scrum-modellen er å greie å beregne arbeidsmengden til en utviklingsoppgave. En del av arbeidet har vært nytt for mange, så det å beregne arbeidsmengde og tidsbruk på oppgavene for å kunne jevnt fordele de innad har ikke alltid vært like enkelt. Løsningen vår på denne utfordringen har vært å tidlig informere de andre på gruppen når vi merker noe er mer belastende enn noe annet, og initiere samarbeid om oppgaven.

2.3 Parprogrammering

Parprogrammering var en arbeidsvane ingen i gruppen hadde benyttet seg særlig av før prosjektet, men ble en stor favoritt blant alle sammen. Gjennom arbeidet benyttet vi oss i stor grad av parprogrammering, både under de oppsatte møtetidene, og når vi møttes uten om faste tider. Parprogrammeringen hjalp oss å lettere sette oss inn i hva den andre tenkte når de løste et problem, og gjorde arbeid med vanskelige oppgaver mer effektivt. Noe som kanskje ikke er så rart siden parprogrammering har vist seg å være gunstig, særlig blant studenter [3].

Rent praktisk gjennomførte vi parprogrammering ved å sitte sammen to stykker (eller av og til fler) og diskuterte ulike løsninger mens vi byttet på å skrive kode. Vi benyttet oss

også av utvidelsen «Live Share» i Visual Studio Code, som gjorde at begge parter lett kunne ha et tastatur tilgjengelig, og vi kunne også samarbeide når vi ikke har fysisk på samme sted.

2.4 Ugunstige arbeidsvaner

Alt har dernest ikke gått helt plettfritt, og vi har bemerkt oss noen ugunstige arbeidsvaner som gikk utover gruppearbeidet. Den største, som vi også raskt greide å gjøre noe med, var at vi ikke greide å håndtere alle oppgavene som måtte gjøres på en gang. Dette var noe alle kjente på før første innlevering, da vi alle sammen måtte sitte i lengre og lengre perioder for å få ferdigstilt alt vi skulle gjøre før innleveringen. Vi erfarte raskt at dette var noe som fungerte dårlig og som over tid kunne gå kraftig utover både arbeidskvaliteten og arbeidsmoralen til alle på gruppen. Vi planla derfor grundigere før de to neste innleveringene hva som måtte gjøres, når utviklingsoppgavene måtte bli ferdig, hva som måtte gjøres først, og i hvilken rekkefølge arbeidet måtte gjøres. Vi satt også deadlines i forkant av innleveringsdatoene der vi skulle ha «kodestopp» og ikke kunne implementere nye funksjoner i prosjektet.

Vi merket raskt at disse tiltakene bedret kvaliteten på arbeidet, både for oss mentalt men også i det faktiske produktet. Vi fikk satt av tid til å implementere det som var viktig, og stresset ikke med å legge til nødvendige funksjoner på korte tidsfrister, som kunne ført til uoversiktlig kode og dårlige løsninger.

2.5 Fletting av kode

Som i alle andre større kodeprosjekter har fletting av kode av og til bydd på enkelte utfordringer. Når flere jobber på samme område og er innom og redigerer de samme filene oppstår det gjerne konflikter i koden som, hvis man ikke er forberedt, kan føre til ekstra mye unødvendig og frustrerende arbeid. Vi erfarte derfor veldig tidlig at all fletting av kode skulle skje i fellesskap, og man skulle alltid ha mist én makker som var med i prosessen; å flette sin egen kode uten å informere andre førte som regel til konflikter lengre frem i tid.

En positiv bi-effekt av at vi alltid hadde oppsyn av en annen under en fletting var at alle fikk se over hva de andre hadde gjort, og fikk innsyn og lærdom i hvordan de andre løste oppgavene sine. Dette gjorde at vi fikk bedre oversikt over sammenhengen til koden, og lærte nyttige triks og knep av hverandre.

3 Individuelle Arbeidsvaner

Som avtalt i kontrakten har vi alle arbeidet på egenhånd i tillegg til det planlagte gruppearbeidet. I gruppekontrakten ble vi enige om å arbeide minst 4 timer på egenhånd, noe alle greide å holde à jour. Individuelt har vi hatt litt ulike arbeidsvaner, så jeg kommer bare til å gå inn på mine egne arbeidsvaner.

Alt arbeidet har, som nevnt tidligere, blitt delt opp i mindre utviklingsoppgaver som vi har fordelt jevnt ut i gruppen. Vi har jobbet med disse oppgavene på litt ulike måter, men min strategi har som regel vært liknende den man finner i Scrum-modellen [2], bare i litt mindre skala. Den fungerte slik:

1. Planlegge løsning på en del av oppgaven
2. Implementere løsningen på delen
3. Teste del-løsningen
4. Gå tilbake til steg 1 og implementer neste del av oppgaven

Denne prosessen fortsatte helt til oppgaven var ferdig implementert. Som regel ble ikke stegene repetert veldig mye; 1-3 ganger var vanlig for en håndterlig utviklingsoppgave. For større oppgaver som strakk seg over lengre tid kunne jeg også finne på å dele den opp i mindre deler som en etter en bygget på hverandre, helt til de til slutt dannet en hel løsning.

4 Tiltak for fremtiden

Som en oppsummering vil jeg trekke frem at gruppen håndtere prosjektet ganske bra, og prøvde etter beste evne å løse de problemene som oppstod underveis. Som et tiltak for fremtiden kunne vi enda tidligere vært nøye på hvordan oppgavene skulle fordeles, og satt flere mindre frister iløpet av hver sprint. Dette kunne senket arbeidspresset, og gjort at arbeidet ble jevnere fordelt iløpet av sprinten.

Det å faktisk fordele de ulike Scrum-rollene til hvert gruppemedlem kunne også vært et godt tiltak. Det kunne ført til at vi brukte mindre tid på «det administrative», og da ville sluppet situasjonene der alle skulle være en del av rollen.

Referanser

- [1] B. Oakley, *A Mind For Numbers*. TarcherPerigee, 2014.
- [2] K. Schwaber og J. Sutherland. «The 2020 Scrum Guide.» (2020), adresse: <https://scrumguides.org/scrum-guide.html>.
- [3] L. Williams og R. L. Upchurch, «In Support of Student Pair-Programming,» *SIGCSE Bull.*, årg. 33, nr. 1, s. 327–331, feb. 2001, ISSN: 0097-8418. DOI: 10.1145/366413.364614. adresse: <https://doi.org/10.1145/366413.364614>.