

# **XPed – Prosjektorientert undervisning: Et forsøk med å bruke eXtreme Programming som pedagogisk metode**

**Mari-Ann Akerjord  
Terje Samuelsen  
Håkon Tolsby  
Børre Stenseth**

**Høgskolen i Østfold  
Rapport 2010:10**

Online-versjon (pdf)

Utgivelsessted: Halden

Det må ikke kopieres fra rapporten i strid med åndsverkloven og fotografiloven eller i strid med avtaler om kopiering inngått med KOPINOR, interesseorgan for rettighetshavere til åndsverk.

**Høgskolen i Østfold har en godkjenningsordning for publikasjoner som skal gis ut i Høgskolens Rapport- og Arbeidsrapportserier.**

Rapporten kan bestilles ved henvendelse til Høgskolen i Østfold.  
(E-post: [postmottak@hiof.no](mailto:postmottak@hiof.no))

Høgskolen i Østfold. Rapport 2010:10

© Forfatterne/Høgskolen i Østfold

ISBN: 978-82-7825-326-7

ISSN: 1503-2612

## Forord

Denne forskningsrapporten har sitt utgangspunkt i et pedagogisk utviklingsarbeid støttet av Norgesuniversitetet. Utgangspunktet for utviklingsarbeidet var å skape bedre rammer for prosjektbaserte undervisningen i programmeringsfag. Vi ønsket en pedagogisk tilrettelegging som styrket samhandling og kunnskapsdeling. Videre ønsket vi å utforske verktøy og teknikker som kunne gjøre veiledningsoppgaven bedre, enklere og mer oversiktlig.

Paradoksalt nok er programmeringsfaget selv den viktigste arenaen for utvikling av nye metoder og teknologier for samarbeid, kunnskapsdeling og kunnskapsproduksjon. De fleste sosiale Internettjenester, og det som refereres til som web 2.0, har sitt utgangspunkt i miljøer som arbeider med åpen programvareutvikling, som for eksempel Linux.

I dette utviklingsarbeidet har vi sett på hvordan innovative metoder for programutvikling, slik som XP (eXtreme Programming), og verktøy som brukes til å koordinere programmeinsprosjekter, slik som Trac og Subversion, kan brukes i en pedagogisk sammenheng.

Prosjektet startet høsten 2007 og ble avsluttet vinteren 2009. I denne perioden fulgte vi åtte prosjektgrupper i to forskjellige programmeringskurs ved Høgskolen i Østfold. Prosjektgruppene ble introdusert for nye verktøy og metoder for å styrke samarbeid og veiledning. Forskningsarbeidet har vært basert på en eksplorativ utviklingsprosess med kontinuerlig evaluering og endring av praksis. Resultatene av disse eksplorative studiene er samlet i denne rapporten.



# Innhold

<b>Forord</b>	<b>1</b>
<b>Innhold</b>	<b>3</b>
Hvorfor	5
Hva	5
Hvordan	6
<b>XP-metoden</b>	<b>7</b>
Noen metodiske perspektiver	8
Noen didaktiske perspektiver	9
<b>Verktøy</b>	<b>13</b>
Blogg	13
Wiki	14
Trac	14
Subversion	17
<b>Forsøk</b>	<b>21</b>
Kurs og veiledning	21
Premisser	21
<b>Undersøkelsen</b>	<b>25</b>
Metodikk	25
Intervjuer med studenter og veiledere	25
<b>Analyse</b>	<b>27</b>
Gruppe1	27
Gruppe2	29
Gruppe3	33
Gruppe4	38

Gruppe 5	43
Gruppe 6	44
Gruppe 7	45
Gruppe 8	50
Veileder 1	53
Veileder 2	56
<b>Diskusjon</b>	<b>61</b>
Oppdragsgiverrollen	61
Rytme , tempo og struktur.	62
<b>Referanser</b>	<b>65</b>

## **Hvorfor**

Forfatterne har alle bakgrunn i IT-faget, og har lang erfaring med prosjektbasert undervisning. Prosjektbasert undervisning har tradisjonelt en sentral plass i informasjonsteknologi. Faget er en blanding av teori og ferdigheter som vanskelig kan læres uten en god del mengdetrening og praksis.

De problemene vi har erfart med prosjektbasert undervisning er neppe spesielle for dette faget. Vi opplever at:

Studentene underpresterer fordi de ikke greier å strukturere sitt eget arbeid, og spesielt samarbeidet innenfor en prosjektgruppe. Arbeidsfordelingen er skjev, både over tid og innen gruppa.

- Vi som faglærere og veiledere underpresterer fordi vi ikke lykkes med ønsket progresjon i arbeidet. Det betyr at det er vanskelig å synkronisere undervisningen med prosjektrelaterte problemstillinger og det betyr at vi ikke får satt inn veiledningsinnsats på tidspunkter der den har størst effekt.
- Som veiledere går vi rett og slett trøtte av å legge arbeid i prosjekter som ikke har progresjon. Dette er ikke nødvendigvis en høyverdig pedagogisk eller faglig reaksjon, men den er høyst menneskelig.

## **Hva**

Vi har følgende mål med prosjektet:

- Utvikle en metodikk ved Avdeling for informasjonsteknologi for å oppnå en bedre programmeringsundervisning, basert på XP (eXtreme Programming).
- Utvikle en faguavhengig undervisningsmetodikk basert på prinsippene i XP.
- Utvikle et sett med erfaringsbaserte retningslinjer og råd for bruk av metodikken, både for lærere og studenter.
- Rapportere erfaringene våre i et teoretisk perspektiv

## ***Hvordan***

Prosjektarbeid innen IT-faget involverer kompleksitet, sterk kontroll av detaljer, god dokumentasjon og tett samarbeid. Det er derfor en tradisjon i faget å anvende ulike metoder for å håndtere slikt prosjektarbeid. I grove trekk kan vi si at vi har, og har hatt, to fundamentalt forskjellige angrepsvinkler.

- "Fossefallsmetoden" som beskriver en progresjon med høy grad av formell spesifisering på ulike nivåer, fra kravspesifisering til programkode.
- Iterative metoder som er basert på mindre rigide formalkrav, og som beskrives som en serie iterasjoner med testing og korreksjoner fram mot et ønsket mål.

Den metodikken vi har tatt fatt i, eXtreme Programming (XP), hører hjemme i den siste kategorien, og er iterativ i sitt vesen.



# XP-metoden

XP er en iterativ metodikk som er beskrevet i litteraturen (for eksempel Steiberg and Palmer, 2004; Astels, Miller and Novak, 2002; Niemeyer, 2003) og på en rekke vevsteder (for eksempel Jeffries, 2001; Wells, 1999). Den skiller seg fra andre iterative metoder ved at den stiller noen spesifikke krav til organisering og arbeidsform. Den skiller seg fra metoder i "fossefallstradisjonen" ved at den har langt mindre formalkrav til spesifisering og dokumentasjon. Det er organiseringen og arbeidsformen, som er i fokus for vår interesse.

Navnet, eXtreme Programming, er på mange måter lite dekkende. Assosiasjonene går lett i retning av en eller annen ekstrem arbeidsform med dedikerte individer som programmerer døgnet rundt. I virkeligheten er metoden på mange måter det stikk motsatte. Metoden beskriver en arbeidsform som er tilpasset en fornuftig, jevn innsats med tydelige krav til samarbeid og felles ansvar. I beskrivelsen nedenfor har vi beholdt den engelske terminologien slik den er beskrevet av Sharma (2004) for ikke å miste viktige nyanser:

1. **The planning game.** Systematisk planlegging med korte etapper. En slik planlegging må holdes opp mot en planlegging av totalprosjektet med de tidsrammer, mål og ressurser som utgjør rammene for arbeidet.
2. **Small releases.** Små, oversiktlige endringer fra gang til gang. Løsningen skal være oppe og stå fra dag en, med de mangler den til en hver tid har.
3. **Metaphors.** Prosjektet, komponenter og aktiviteter skal omtales i et språk som alle forstår og som det er lett å assosiere riktig til. Riktig i den forstand at alle involverte forstår det samme med de begrepene som brukes. Med alle menes da både prosjektgruppa, oppdragsgiver og andre involverte, for eksempel testpaneler.
4. **Simple design.** Design og implementasjon skal aldri være mer komplisert enn det som er nødvendig for neste planlagte versjon.
5. **Testing.** Tekniske løsninger skal testes. Testene skal helst lages før løsningene.
6. **Refactoring.** Kast og restrukturer. Alle komponenter skal kunne skrives om og eventuelt forkastes. Dette er en naturlig og nødvendig konsekvens av "Small releases" og "Simple design".
7. **Pair programming.** All kode skal skrives sammen av to personer.
8. **Collective ownership.** Alle (i gruppa) står kollektivt og personlig ansvarlige for alt som er produsert.

9. **Continuous integration.** Endringer integreres kontinuerlig i den til enhver tid eksisterende løsningen.
10. **No overtime.** Planleggingen skal være slik at en unngår "skippertak".
11. **On-site customer.** Kunden (oppdragsgiveren) skal være tilgjengelig til enhver tid. Metoden argumenterer sterkt for et mye tettere forhold mellom oppdragsgiver/kunde og prosjektgruppe enn det som er vanlig.
12. **Coding standards.** Grappa enes om noen standarder som skal følges, men metoden i seg selv angir ikke slike standarder.

## ***Noen metodiske perspektiver***

Selv om XP er utviklet innen informatikkfagets tradisjon, og har en form og terminologi som er tilpasset dette faget, er det etter vår mening en åpenbar overføringseffekt til andre typer prosjektorientert arbeid enn utvikling av IT-prosjekter.

XP er beskrevet som en metode uten angivelse av støtteverktøy. For oss er det både naturlig, og langt på vei nødvendig, å se metoden i sammenheng med noen verktøy som støtter opp under arbeidet. De viktigste støttefunksjonene er

- Prosjektadministrasjon, med dokumentasjon av milepæler og deloppgaver. Vi har i vårt arbeid brukt et relativt enkelt system, Trac (Edgewall Software, 2003).
- Versjonshandtering for vedlikehold av de dokumentene som inngår i prosjektet. Det innebærer at et dokument kan vedlikeholdes av mange personer fra forskjellige steder. Wiki kan dekke denne funksjonaliteten for ordinære dokumenter av typen rapporter. Vi har brukt Subversion (Tigris.org, 2001) som er bedre egnet til vedlikehold av ulike typer dokumenter, som f.eks. programfiler og filer i annet format enn tekst. Subversion er dessuten integrert i Trac.

XP formaliserer en fornuftig rollefordeling (On-site customer).

Kunde/oppdragsgiver er utgangspunktet for beskrivelse av en lærerrolle i en prosjektorientert setting. Det er en rekke mulige nyanser i beskrivelsen av rollen og den er avhengig av prosjektet. For våre prosjekter har rollen som "teknisk kyndig oppdragsgiver" vært nærliggende. Den muliggjør en arbeidsform som både stiller krav og samtidig kan bidra i faglige problemstillinger. En kan lett se andre situasjoner der den faglige veiledningen skal skilles klart fra oppdragsgiverrollen. Rollefordelingen forplikter både arbeidsgruppa og oppdragsgiver på en konkret måte. Studentene har konkrete arbeidsmål og

oppdragsgiver er forpliktet til å gi tilbakemeldinger. Rolleoppfattelse er nærmere analysert i rapportens erfaringsdel.

XP realiserer en formalisert "pacemaker"-funksjon (the planning game, small releases, no overtime). Både lærer (oppdragsgiver) og studenter (prosjektgruppa) er forpliktet i forhold til planleggingsprosessen, og begge parter må forberede seg til planleggingsmøtene. Det lages i fellesskap milepæler for hvert møte. En milepæl uttrykker oppdragsgivers ønsker for neste steg. Et slikt ønske beskrives ofte som en "user-story". I dette ligger det at oppdragsgiver i prinsipp forteller hva han/hun ønsker å kunne gjøre. I en slik beskrivelse er bruk av aksepterte metaforer viktig. Vi vil bruke det litt mindre presise begrepet delmål for "user-story" i teksten.

En milepæl tar utgangspunkt i oppdragsgiverens ønsker, men det er avgjørende viktig at arbeidsoppgaven aksepteres av gruppa. Den må oppfattes som rimelig i omfang og kompleksitet og den bør tilpasses andre oppgaver på gruppas dagsorden.

XP formaliserer samarbeidet innen prosjektgruppa (pair-programming, collective ownership, coding standards). Både gruppa selv og oppdragsgiver kan opprettholde en oversikt over hvem som gjør hva, og hvordan arbeidsfordelingen er beskrevet.

XP formaliserer en arbeidsform der status på prosjektet alltid er synlig, og det er mulig å følge og korrigere oversiktlige steg i utviklingsarbeidet. (small releases, refactoring, continous integration, simple design)

XP innebærer for et programutviklingsprosjekt at all kode skal utvikles i par, og at man hele tiden skal bytte på hvem som arbeider sammen (pair-programming). En av fordelene med denne måten å arbeide på er at alle får kjennskap og eierskap til alle deler av prosjektet, og dermed at erfaringer som gjøres i en del av prosjektet kan utnyttes i en annen sammenheng.

XP som metode innebærer at både faglærere og studenter involverer seg og deltar i en kontinuerlig arbeidsprosess (on-site customer). Dette vil være med på å forebygge plagiater, som er en problemstilling som ofte reises i forbindelse med mappevurdering og evaluering av prosjektarbeider.

### ***Noen didaktiske perspektiver***

Alt veiledningsarbeid er basert på noen mer eller mindre aksepterte rolleoppfatninger. Ofte er disse lite eksplisitte og koker ned til en litt diffus oppfatning av at veileder kan mye og vil gjerne fortelle om det, mens arbeidsgruppa kan mindre og gjerne vil veiledes. Det kan godt hende at dette

ofte er en god nok rolleoppfatning, men det er en del problemstillinger som ikke finner et godt nok svar uten litt mer presisjon.

## **Prosessveiledning**

Det er et kjent problem at arbeidsbelastningen i grupper ofte er skjevt fordelt. Det kan skyldes at medlemmene har forskjellig ambisjonsnivå eller forskjellig prestasjonsevne. I en del tilfeller er det slik at en gruppe arbeider sammen på flere kurs og at de så å si fordeler kursene mellom seg. Det siste er en rasjonell arbeidsdeling, men den kan være ganske destruktiv når det gjelder læringsutbytte.

Uansett årsak er arbeids- og ansvarsfordeling internt i en prosjektgruppe et tema som må tas opp i gruppa. Dette er ofte en ganske vanskelig problemstilling, dels fordi studentene kvier seg for å karakterisere andres arbeid og dels fordi problemene manifesteres relativt sent i prosjektarbeidet. XP bidrar metodisk på flere måter til å ta fatt i gruppedynamikken:

- Den stegvise utviklingen med klare avsjekningspunkter avslører progresjon eller mangel på progresjon tidlig. Verktøyene som det er rimelig å bruke i prosessen (planlegging og versjonering) plasserer dessuten ansvaret for deloppgaver tydelig, hos enkeltpersoner eller deler av gruppa.
- Rollefordelingen i XP skal være eksplisitt og avklart mellom veileder og gruppemedlemmene. Kjernen er at veileder skal være i en posisjon der han/hun har en rolle som legitimerer å etterlyse delresultater, og sette fingeren på ansvaret for manglende resultater.

## **Progresjon**

XP beskriver en arbeidsform som er iterativ i den forstand at detaljert planlegging ikke skal strekke seg lenger enn til neste milepæl, typisk om to uker. På den annen side er det åpenbart at det må foreligge en tidsplan som sikrer at prosjektet blir ferdig til et bestemt tidspunkt. Dette "planning game" inneholder en potensiell konflikt. På den ene siden den begrensede gjennomføringshorisonten som typisk er to uker, og på den andre siden den langsiktige ferdigstillelsen som er prosjekts mål.

Disse to perspektivene må vedlikeholdes samtidig og må kommuniseres i samspillet mellom veileder og gruppe. Det går an å betrakte dette som en potensiell svakhet med metoden, og det går an å betrakte det som en åpning for en konstruktiv planleggingsdiskusjon. Det siste gir en fin mulighet til å diskutere

gruppas arbeid i forhold til tilgjengelig tid og ønsket om en jevn arbeidsbelastning.

## **Deling**

XP overlater mye ansvar til gruppas egne vurderinger når det gjelder bestemmelse av standarder, skrivemåte osv. Dette kan innebære at gruppa velger en anerkjent standard eller at de så å si finner opp sin egen. Begge deler kan være akseptabelt, avhengig av prosjektets art. Det viktigste er at man eksplisitt må enes og at man forplikter seg til å følge de retningslinjene som er bestemt. Alle skal kunne lese de andres bidrag og bygge videre på disse.

XP beskriver parprogrammering som en obligatorisk arbeidsform når det gjelder programmering. Det viktige er de effektene på språk og bruk av vedtatte standarder som et nært samarbeid kan ha. I programmering er selvsagt også effekten på algoritmiske løsninger og effekten på ren leselighet spesielt viktig (variabelnavn, kommentarer osv.).

## **Sensur**

Et tilbakevendende problem i all prosjektbasert undervisning er blandingen av rollene som sensor og veileder. Dette er uheldig av flere årsaker. Veileder vil påvirke resultatet, ikke bare når det gjelder kvalitet i detaljene, men også design og problemavgrensning.

På den ene siden kan en hevde at det er en del av alle læreres ansvar å holde styr på dette, og være med på å sette karakter. Ofte tillegges også veileders oppfatning av selve prosessen vekt ved sensur.

På den annen side er det åpenbart at dobbeltrollen skaper forstyrrelser i prosjektgjennomføringen. Det er ikke nødvendigvis bra eller enkelt at veileders synspunkter og råd tillegges vekt fordi vedkommende skal sensurere, både resultatet og prosessen. Studenters evne til rasjonalitet med tanke på resultatmaksimering er stor, og veiledere/lærer/sensorer er kanskje ikke alltid så uhildet og saklig som en ideelt sett skulle ønske.

Det er derfor viktig at avklaring av roller også inkluderer en avklaring av sensuroppdraget. Praksis er her forskjellig, og denne delen av rollediskusjonen drøftes ikke ytterligere i rapporten. Ideelt sett burde et XP-prosjekt være helt løst fra sensurproblematikken. Både veileder og studenter burde ha som mål å lage et så godt resultat som mulig, og i fellesskap sende det til ekstern sensur.



## Verktøy

Selv om XP, i motsetning til en del andre sammenlignbare metoder, ikke stiller store formalkrav til dokumentasjon så er det åpenbart mye å hente i å finne en egnet form for administrasjon av dokumenter. De dokumentene vi lett kan identifisere er:

- Dokumentasjon av utviklingsprosessen. Med dette mener vi beskrivelse av hva som skal gjøres/er gjort i hvert steg. Vi trenger å kunne lage og tidfeste det målet som er definert for neste arbeidsteg. Ofte beskrives dette delmålet som en "user story" for å understreke at det er et ønske eller et krav som stilles av brukeren, og som beskriver hva brukeren ønsker å kunne gjøre.
- Selve produktet, uansett hvilken form dette har. I et utviklingsprosjekt innen IT-faget vil dette typisk være en samling filer med programkode, designdokumenter, websider, prosjektkataloger etc. I en mer generell kontekst er kanskje produktet et eller flere dokumenter av mer generell karakter.

Disse dokumentene utgjør en form for felles hukommelse for alle involverte parter. XP som metode angir at denne hukommelsen står sentralt i prosessen. Vi skal ha små utviklingssteg beskrevet ved klare delmål, vi skal ha kontinuerlig integrasjon og implementering av seneste versjon osv. Dokumentene skal være åpent tilgjengelig for alle involverte.

En slik støttefunksjon lar seg realisere på mange måter. Vi vil nedenfor se på noen alternativer, først et par generelle verktøy og til slutt et par mer spesialiserte verktøy.

## **Blogg**

Det finnes mange typer og varianter av blogger. Sett i vårt perspektiv ville en blogg kunne brukes av et prosjekt. Artikkene i bloggen vill for eksempel kunne beskrive en iterasjon i utviklingsprosessen, og den kunne kommenteres fra de enkelte deltagerne i prosjektet, kunde/veileder så vel som utvikler/student. Selve artikkelen skulle altså kunne være en tekst som er en definisjon på målet for neste arbeidssteg, mens en kommentar kunne for eksempel være at en deloppgave var ferdigstilt, med en referanse til det som var gjort.

Blogg har flere av de mekanismer som en ofte tilstreber i læringsprosesser som for eksempel samarbeid og produksjon av tekst. Blogg er et "Participatory media" [5] det vil si der brukeren kan lese, skrive, utveksle meninger og lage nytt innhold i felleskap med andre. Et annen viktig poeng med bruk av blogg er

at samtidig med at studentene anvender blogg som et virkemiddel i læringsprosessen, kan det bidra til bygging av digital kompetanse for studenten.

Blogger kan opprettes som en åpen eller lukket tjeneste for grupper eller klasser. Ved en lukket tjeneste bruker brukerne en id og passord for å få tilgang. Dette gir mulighet for at studentene ikke føler seg så utsatt for at noen utenfor gruppen ”titter en over skulderen”. Det kan også bidra til at det som blir skrevet i et kurs ikke kan brukes av de som tar kurset neste gang, dersom dette ikke er ønskelig. Dette gir igjen større mulighet for gjenbruk av oppgaver. Samtidig som bloggen kan være lukket er det som skrives transparent for alle som har tilgang til bloggen. Dette gjør at veileder kan følge med på produksjonen til studentene. Med flere av blogg verktøyene er det også funksjoner som dokumenterer hva den enkelte bidrar med.

Blogger har ofte funksjonalitet som gjør den enkel å bruke og enkle å tilpasse forskjellige undervisningsopplegg.

Fordelene med en blogg sett i et XP-perspektiv er at den er allment kjent, lett tilgjengelig og tilpassbar og den er allment tilgjengelig via en nettleser. Ulempene er at den ikke har noe begrepsapparat som direkte eller indirekte understøtter en prosjektorientert tankegang, for eksempel en tidslinje. For ikke-trivielle prosjekter vil vi dessuten måtte basere oss på linker til andre lagringsområder for å vise og bearbeide dokumenter.

## **Wiki**

En wiki er i sin natur og etter sin hensikt et format hvor mange personer i fellesskap vedlikeholder de samme dokumentene. Wikipedia er det mest kjente av slike systemer. Det er åpenbart at vi kunne bruke en eller flere wikisider til å administrere et XP-prosjekt. Trolig vil dette være den minst byråkratiske måten å legge opp en struktur for et XP-prosjekt. Det eneste som kreves av deltagerne er at de i noen grad behersker de editeringsreglene som gjelder for wiki.

Wikisider kan på samme måte som en blogg adgangsbeskyttes.

Fordelene med wiki er fleksibiliteten, den lave inngangsterskelen og tilgangen via en nettleser. Ulempene er de samme som for en blogg, mangel på eksplisitt tidslinje og håndtering av andre dokumentformater enn tekst.

## **Trac**

Hvis vi ønsker et mer tilpasset verktøy som har innebygd et begrepsapparat for prosjektstyring, beveger vi oss inn i en annen kategori programmer. På den ene



siden ønsker vi et enkelt verktøy med lav brukerterskel og på den andre siden vil vi ha så mye som mulig gratis i forhold til arbeidsformen som XP foreskriver.

Vi ønsker dessuten et verktøy som i størst mulig grad er uavhengig av plattformer og helst er tilgjengelig via en nettleser.

Vårt valg, som vi har brukt i den virksomheten som beskrives i denne artikkelen, er Trac (Edgewall Software, 2003)

Som utgangspunkt for å beskrive mulighetene i verktøyet bruker vi et klipp av en typisk webside:



Figur 1; Typisk eksempel på en oppstartside i et undervisningsprosjekt.

Øverst i figuren er det en samling med funksjoner som finnes i Trac. Denne delen vises hele tiden, mens resten av siden brukes til å vise de data som er knyttet til den valgte funksjonen.

I toppmenyen finner vi følgende:

- **Wiki.** Dette er forsiden til prosjektet. Det naturlige innholdet er en kort beskrivelse av prosjektet, aktuelle linker og kontaktinformasjon for alle involverte.
- **Timeline.** Gir en beskrivelse i punktvis form av hva som har skjedd i prosjektet fram til nå.

- **Roadmap.** Her listes Milestones eller milepæler. Milepæler, er et viktig begrep i metoden og i verktøyet. En typisk milepæl inneholder et delmål og et tidspunkt for når dette målet skal være implementert.
- **Browse Source.** Dette er tilgang til alle versjonerte dokumenter. Mer om dette nedenfor under avsnittet Subversion.
- **View tickets.** Tickets, arbeidsoppgaver, er det andre viktige begrepet i vår metodiske angrepsvinkel. Det normale er at arbeidet som må gjøres for å fullføre en Milepæl brytes ned i tickets. En ticket er altså koplet til en milepæl, og ansvaret for å fullføre en ticket kan eksplisitt koples til en eller flere navngitte personer. Vi vil bruke begrepet ”trac-oppgave” for ticket i det følgende.
- **New ticket**

. Opprettelse av en ticket.

Dette betyr at de sentrale begrepene som brukes for å omtale et prosjekts status er Milestone (milepæl) og Ticket (Trac-oppgave). Det er i seg selv viktig at dette begrepsapparatet er enkelt og forståelig. Det finnes noen enkle mekanismer for å regulere tilgang til en slik Trac-side. For det første kan selve tilgangen beskyttes med brukernavn og passord. For det andre kan editering av milepæler koples til en administratorrolle. En måte å bruke dette på er å gi prosjektlederen eller kunden slik administrativ rolle. I undervisningssammenheng vil et typisk eksempel på milepæler være delinnleveringer i et prosjektorientert kurs. For å få en tettere oppfølging av progresjonene i arbeidet til studentene vil det ofte være naturlig å ha flere prosjektmøter underveis mot en milepæl.

Et typisk prosjektmøte kan foregå ved at alle har tilgang til trac-siden. Møtet begynner med at kunden (vi bruker denne betegnelsen her selv om vi vil diskutere modifikasjoner av rollen senere) går gjennom den de aktuelle Trac-oppgavene som er planlagt fullført til møtet. Dette vil innen vårt fag typisk innebære en test av løsningen. Dersom det er avvik forklarer prosjektgruppa dette. Avvikene diskuteres og resultatet av denne diskusjonen, enten det er en ren videreføring eller en modifikasjon tas med til neste møte.

Denne situasjonen er helt transparent i den forstand at alle ser hvilke oppgaver den enkelte skulle ha gjort til dagens møte. Dette er selvsagt en innfallsvinkel til mange typer diskusjoner og det nødvendig å ta opp en del problemstillinger som ofte forsvinner i mer overflatisk prosjektstyring. Dette vil reise spørsmål av gruppodynamisk karakter, ulike ferdighetsnivåer, ulik arbeidsmoral osv. Hvordan disse spørsmålene skal håndteres henger tett sammen med den rollefordelingen som er avtalt mellom kunden og prosjektgruppa. Det er blant annet i denne sammenheng kundebegrepet må analyseres nøye.

Et annet moment som setter kundebegrepet under lupen er i hvilken grad møtet skal brukes til konkret faglig veiledning.

Møtet avsluttes med at kunden formulerer et forslag til hva som skal gjøres til neste prosjektmøte, altså typiske Trac-oppgaver og et tidspunkt for ferdigstilling. Begge deler gjøres til tema for en åpen diskusjon på møtet. Det er to avgjørende viktige momenter i denne diskusjonen:

- Alle må bli enige om både oppgavene og tidsfristen.
- Progresjonen knyttet til neste prosjektmøte må avstemmes mot den totale tiden prosjektet har til rådighet.

Dersom disse to momentene ikke lar seg forene, må det nødvendigvis ha konsekvenser for ambisjonsnivået.

Vi har prøvd Trac på flere nivåer i studiet fra første semester til hovedprosjektet i 6. Semester. Det er særlig enkelheten i funksjonaliteten som gjør det kan være egnet for bruk i forbindelse med forskjellige didaktiske opplegg. Det er viktig at verktøyet oppfattes som lett tilgjengelig og at det faktisk bidrar til en arbeidsprosess som oppleves positivt.

## ***Subversjon***

De aller fleste studenter har jobbet med dokumenter som flere bidrar til. Som regel skriver de forskjellige deler og så lappes det sammen på slutten. Noen er også vant med å sende biten de har skrevet til andre og få tilbakemelding på teksten. Disse arbeidsformene medfører ofte at dokumentene ikke har den konsistens i form og språk som kan forventes på høgskole/universiteter.

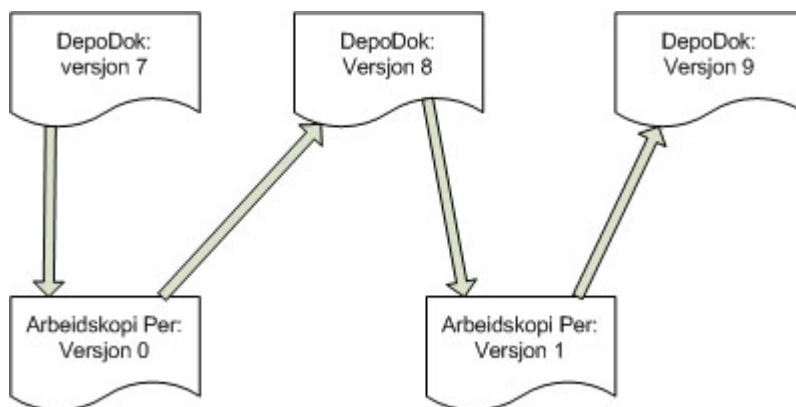
Problemer av denne art er svært tydelige når det som skal lages er programkode, der problemstillingen ikke bare går på stil og konsistens, men direkte på funksjonalitet.

Tankegangen i versjonering er at versjonskontroll ideelt skal kunne gjøre det mulig at flere skal kunne jobbe simultant på samme dokument og at en enkelt ved hvilket som helst tidspunkt kan se hva dokumentet inneholder samt spole tilbake til gamle versjoner.

Vi skal her prøve å skrive en forenklet versjon av strategien systemet virker etter i undervisningsperspektiv. Vi ser på hovedtrekkene og hvorledes disse kan passe i et undervisningsopplegg. For at leseren skal ha en oppfatning av hvorledes denne type verktøy brukes har vi først en introduksjon til arbeidsgangen før vi ser litt på bruken relatert til noen pedagogiske begreper. Teksten beskriver to scenarier på arbeidsgangen. Det første er basert på at Per utvikler et dokument

uten å samarbeide med noen. Det andre illustrerer to personer på samme dokument.

Scenario en (se Figur 1): Når jeg (Per) skal jobbe med et dokument så henter jeg fra et felles depot som alle som skal jobbe med dokumentet har tilgang til. Dette felles depotdokumentet (DepoDok: versjon 7) kan ligge på en server og jeg henter en arbeidskopi (Per: versjon 0) til min lokale maskin. Så jobber jeg på denne lokale kopien. Når jeg sender denne lokale kopien (Per: versjon 0) til det som ovenfor er omtalt som depotdokumentet flettes mine endringer inn i dette på riktig plass. Denne nye versjonen får et nytt versjonsnummer (DepoDok: versjon 8). Slik fortsetter det. Fordelen med denne arbeidsformen når du arbeider alene, er at du kan få tak i og kan bearbeide dokumentet fra mange steder, typisk på skole/arbeid og hjemme.

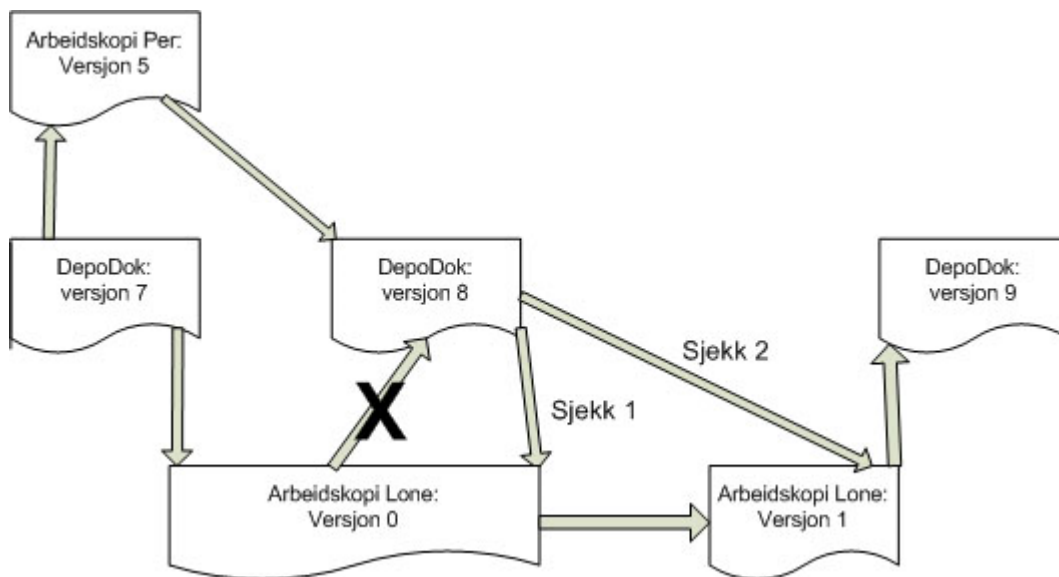


Figur 1, Arbeidsgang med en person

Scenario to (se Figur 2): Revisjonskontrollsystemet gjør at flere kan jobbe på samme dokument tilnærmet simultant. Dette skjer ved at det er et depotdokument som fungerer som et fellesdokument. La oss si at en annen person (Lone) har hentet en arbeidskopi (Lone: versjon 0) av depotdokumentet (DepoDok: versjon 7) mens jeg jobbet med min lokale versjon slik jeg har beskrevet i forrige avsnitt. La oss nå si at jeg appliserer min arbeidskopi slik jeg har beskrevet i forrige avsnitt, slik at vi har depotdokument (DepoDok: versjon 8). Nå vil Lones arbeidskopi (Lone: versjon 0) være forskjellig fra det som er depotdokumentet.

Dersom Lone nå appliserer sin kopi til hoveddokumentet vil hun skrive over de endringer jeg har gjort (merket med X på figuren). Derfor må hun sjekke om det er noen konflikter mellom hoveddokumentet og hennes arbeidskopi. Dette gjør hun ved å sammenligne (sjekk 1) siste versjon av depotdokumentet (DepoDok: versjon 8) mot sin arbeidskopi og sjekke om det er noen konflikter. Dersom det er noen må hun fikse disse. Dette gjør at hun nå har (Lone: versjon 1) som

arbeidskopi. Så sjekker hun konflikter en gang til (Sjekk 2). Dersom ok, kan hun applisere sin arbeidskopi til depotdokumentet som nå får en ny versjonsnummerering (DepoDok: versjon 9).



Figur 2, Arbeidsgang ved 2 brukere

Verktøyet gir støtte i å involvere flere i skriveprosessen. I dette legger vi at involveringen er mer å sammenligne med det å sitte ved siden av hverandre i et lokale mens skrivearbeidet foregår. XP som metode legger opp til at studentene skal samarbeide tett med hverandre, jamfør Weinbergs (1971) ide om "egoless programming" hvor produktet eies av et team som en helhet og at individet ikke holdes ansvarlig for problemer med programkoden. På norsk omtales dette ofte som et eierskap til et produkt/oppgave. For å komme nærmere "egoless programming" bruker Weinberg en metode som vi har beskrevet under kapitel Teori og som kalles parprogrammering. Vi hevder at disse prinsippene kan generaliseres til også å gjelde arbeidet med å produsere en tekst i et dokument, eller å jobbe på datafiler som har et filformat som støttes av denne type verktøy.

Dette at det er enklere å involvere flere i skrivingen kan motvirke den segregering som studentene ofte gjør i prosjektarbeid ved at studenten i en prosjektgruppe jobber med ulike elementer, for siden å sette disse sammen. Denne delingen av oppgaver kan være et hinder for at alle studentene i gruppen når læringsmålene for kurset.

Siden alle deltagere til en hver tid har tilgang til hele dokumentet så kan det bidra til at studentene føler et eierskap til dokumentet. Denne synligheten bidrar også til at mengden produksjon blir mer tydelig enn i de prosjektarbeidsformene som tradisjonelt brukes blant mange studenter. Dette er også omtalt i forrige kapittel: Trac.



## **Forsøk**

Virksomheten i prosjektet startet høsten 07. Metodikken er i ferd med å etableres som mer eller mindre permanent ved en rekke kurs ved avdelingen. Her rapporteres virksomheten i studieåret 07/08.

### ***Kurs og veiledning***

Høsten 2007 ble metoden anvendt på to ordinære kurs:

- Programmering for web (3.klasse informatikk). Evalueringen av kurset er knyttet til en stor gjennomgående programmeringsoppgave. XP ble brukt gjennom hele semesteret som obligatorisk metodikk for denne oppgaven.
- Innføring i programmering (1. klasse). XP ble brukt på en obligatorisk oppgave i siste del av kurset.

Våren 2008 ble metoden i litt ulike varianter anvendt på følgende:

- Kurset .Net (2.klasse). Obligatorisk arbeidsform for gjennomgående prosjekt.
- Kurset Software Engineering (2. Klasse) som er et prosjektkurs med obligatorisk arbeidsform.
- En del hovedprosjekter til bachelorstudiets 3.år.
- Noen masterprosjekter

Høsten 2008 ble metoden anvendt i kursene

- Dokumenter og web. Obligatorisk arbeidsform for gjennomgående prosjekt.
- Kurset Ledelse av mediaprojekt (3.klasse) som er et prosjektkurs med obligatorisk arbeidsform.

### ***Premisser***

Prosjektene i de aktuelle kursene er svært forskjellige hva angår nivå og omfang.

Når det gjelder høsten 2007 veiledet alle fire, artikkelforfatterne, på begge kursene. To av oss er faglærere på hvert vårt kurs. Faglærerne holder forelesninger på vanlig måte, 4 timer pr uke.

Vi har valgt å anbefale metoden og verktøyene overfor studentene med to begrunnelser:

- Produktet som studentene lager blir bedre

- Studentene lærer en aktuell arbeidsmetodikk

Vi har ikke stilt krav ut over den aller enkleste dokumentasjon av arbeidsoppgaver til neste møte. Det vil si at versjonering, parprogrammering, systematisk testing, er sterkt anbefalt, men står ikke som absolutte krav. Vi har heller ikke blandet oss inn i hvordan gruppene har organisert seg selv.

Gruppene består av 2-4 studenter, og er satt sammen av studentene selv. Vi - faglærerne - har fordelt gruppene mellom oss og har rollen som oppdragsgivere/kunder, mens studentene har rollen som programutviklingsgrupper.

Vi har videre sagt at prosessen i henhold til metoden ikke blir evaluert som en del av eksamensbesvarelsen. Deltakelse i prosjektprosessen er derimot formulert som et arbeidskrav i begge kursene. Det er ganske åpenbart at planleggingssystemet inneholder en god dokumentasjon av hva som er gjort i løpet av prosjektperioden, hvem som har gjort hva, når og hvordan. Det er et interessant spørsmål om denne informasjonen kan eller bør brukes som grunnlag for evaluering av studentene. På den ene siden vil det bidra til et bedre evalueringsgrunnlag, en mulighet for individuell karaktersetning og en forsikring mot plagiering. På den annen side er det problematisk i forhold til rollefordelingen og ansvarsfordelingen i prosjektarbeidet. Det ødelegger litt av rollen som oppdragsgiver og det kan lett føre til at den kollektive arbeidsformen og delingsfilosofien som tilstrebes blir underminert. Uansett hva som gjøres må det annonseres klart hvilke premisser arbeidet foregår på.

Hver gruppe har fått opprettet et Trac-prosjekt for sitt arbeid. Oppdragsgiver er i utgangspunktet administrator for gruppas Trac. Møter mellom oppdragsgiver og gruppe avtales fra gang til gang og holdes med 1-2 ukers mellomrom. Hvert møte er knyttet til en milepæl i Trac. Et typisk møte kan se omtrent slik ut:

1. Vi tar opp milepælen med oppgavene som skal være gjort til dette møtet, og diskuterer løsningene gruppa er kommet fram til. Fullførte oppgaver markeres som ferdige i Trac. Eventuelle uløste oppgaver overføres til neste milepæl.
2. Vi setter i fellesskap opp en ny milepæl med nye arbeidsoppgaver. Arbeidsoppgavene beskrives som forventninger som oppdragsgiver har til prosjektet. I XP-metoden omtales dette som "user-stories".
3. Studentene går selv inn i Trac, aksepterer oppgavene de har påtatt seg og fordeler dem mellom seg, og forplikter seg dermed til å fullføre disse oppgavene før neste møte. Deloppgavene tilsvarer trac-oppgaver.
4. Ofte har gruppa en del konkrete programmeringstekniske spørsmål som tas til slutt.



Metoden ble lansert på de offisielle kurssidene og introdusert på forelesninger. Metoden som sådan ble introdusert som obligatorisk, men med åpning for en betydelig fleksibilitet i alle detaljer. Typisk ble den programtestingsstrategien som XP foreskriver og parprogrammering sterkt anbefalt men ikke påbudt. Anbefalingene ble fulgt opp i veiledningsmøtene.

Verktøyene Trac og Subversion ble kort omtalt og demonstrert i forelesninger i de respektive kursene og bruken av dem ble fulgt opp i de enkelte prosjektgruppene. Bruken av Trac var obligatorisk, mens subversion ble sterkt anbefalt.



# Undersøkelsen

## ***Metodikk***

Innføring av XP i programmeringsundervisningen er gjennomført som et pedagogisk designeksperiment (Brown, 1992). Et designeksperiment er en forskningsstudie hvor forskere og deltagere (i første rekke underviserne) samarbeider om design, implementering og analyse av læringsforløpet (Greeno , 1998). I samarbeid utformer de eksperimentet og diskuterer forskningsspørsmål og teoretiske tilnærminger. Prosessen er både progressiv intervenserende (Collins , 2004). Designet kan bli endret under eksperimentet hvis det ikke fungerer tilfredsstillende, og resultatet brukes til å utvikle nye teorier og designe nye læringsforløp.

I denne undersøkelsen har både forskere og fagansvarlige deltatt som oppdragsgivere for prosjektgrupper. Etter hvert møte med prosjektgruppen har oppdragsgiver skrevet ned sine refleksjoner i en felles wiki. Disse refleksjonene har vært utgangspunkt for diskusjoner i forskningsgruppen og for endring av praksis.

I slutten av semesteret ble det gjennomført en større undersøkelse blant studentene hvor fire grupper fra hvert kurs ble intervjuet. Intervjuet ble gjennomført som et kvalitativt forskningsintervju (Kvale, 1996) hvor studentene gruppevis ble bedt om fortelle hvordan de hadde arbeidet sammen i prosjektet. En PC med studentenes Trac-omgivelse ble brukt som hjelpemiddel for å kontekstualisere intervjuet, og for å gi en bedre retrospektiv beskrivelse av prosessen. Resultatene fra denne undersøkelsen vil danne utgangspunkt for et redesign av den pedagogiske metoden. Samtidig ønsker vi å få en større teoretisk innsikt i hvordan XP kan brukes som en pedagogisk metode i prosjektarbeid.

## ***Intervjuer med studenter og veiledere***

Høsten 2007 intervjuet vi 4 grupper fra hvert av kursene som var med i forsøket. I tillegg intervjuet vi hverandre. Intervjuene ble gjennomført med utgangspunkt i en felles intervjuguide.

Våren 2008 intervjuet vi 2 grupper (hovedprosjekter)

Intervjuene ble transkribert og kategorisert.



# Analyse

## Gruppe1

Gruppe 1 bestod av fire gutter som fulgte kurset webdesign i femte semester. De kjente hverandre godt fra før og hadde også tidligere samarbeidet i kurs og prosjekter. Oppdragsgiver for prosjektet var en fagperson ved avdelingen, men vedkommende var ikke involvert i undervisningen av kurset. Målet med prosjektet var å lage et online bookingsystem for hytteutleie.

### Strukturert veiledning

Grappa hadde jevnlig møter med sin oppdragsgiver hver uke. Møtene varte vanligvis en time. I disse møtene presenterte studentene hva de hadde laget siden sist. Deretter formulerte de i fellesskap med oppdragsgiver et delmål som beskrev hva som skulle være ferdig til neste møte, uken etter. Det skjedde gjennom en gjensidig dialog med studentene. De forklarte at:

*S1: Vi sitter bare og snakker sammen og så skriver vi mer og mer på om hva som skal til.*

*S2: Helt til vi er enige.*

Etter møtet med oppdragsgiver tok studentene for seg delmålet og delte det opp i mindre og mer konkrete arbeidsoppgaver (trac-oppgaver) som skulle løses til neste møte. Både delmål (user-story) og trac-oppgaver ble lagret og dokumentert i Trac, og de var synlig for både oppdragsgiver og studentene.

Grappa trivdes med denne formen for oppfølging av en oppdragsgiver fordi det skapte struktur og tempo.

*S1: Spesielt det ukentlige møte, egentlig.*

*S2: Fordi at da får man en veldig god struktur, at man da må gjøre noe til neste gang.*

*S3: Med prosjekter så er det ofte at det er mest å gjøre på slutten ... Nå har vi jobba ganske jevnt med det her.*

Deres erfaringer fra tidligere prosjekter var at det er vanskelig å komme i gang. Det er lett å utsette arbeidet til man må gjøre det fordi ingen setter krav til dem underveis og innleveringen ligger langt frem i tid. Også i dette prosjektet var arbeidsinnsatsen svak i oppstarten. Studentene forklarte det slik:

*S1: Når vi først kom i gang så gikk det bra, da.*

*S2: Første uka gikk det litt tregt, men det tok seg opp. Da ga han*

*(oppdragsgiver) oss masse å gjøre, mer eller mindre konkrete oppgaver så da ble vi sparka i gang.*

De ukentlige mål og oppgaver i prosjektet ligger lagret i Trac. De utgjør en felles hukommelse for både oppdragsgiver og studenter, og det gir oppdragsgiver et redskap for å gripe inn hvis studentene ikke produserer eller har en forventet fremdrift. Hos denne gruppen måtte oppdragsgiver gripe inn og gi studentene mer konkrete oppgaver å arbeide med fordi de hadde problemer med å komme i gang.

### **Forpliktelse**

Det ligger en forpliktelse i at delmålene ble utformet i fellesskap med oppdragsgiver og at de selv måtte definere hvilke konkrete arbeidskrav som skulle løses til neste møte. Det var en skrevet kontrakt mellom oppdragsgiver og studenter og mellom studentene selv. Målet var at studentene både skulle føle et eierskap til prosjektet og en gjensidig forpliktelse til å løse arbeidskravene. En av studentene beskriver det slik: Det at vi som sagt må møtes med oppdragsgiver hver uke gir jo på en måte et krav. At vi må vise noe hver gang. På den måten vil man uansett komme litt fremover i prosjektet. Jeg kunne i alle fall ikke personlig bare møtt opp hver uke og ikke gjort noe som helst, og sagt pfft! vi tar det en annen gang, og så møter vi neste gang..." De fortalte videre at de hadde jobbet veldig mye mer i dette prosjektet, mer enn hva de hva de vanligvis pleide å gjøre. Forklaringen til det var at det ble stilt eksplisitte, synlige krav til dem hele tiden. Krav som var nedfelt i Trac og som de selv hadde vært med til å utforme. Dette gir jo veldig klart bilde av hva som skal gjøres til neste gang, kommenterte en av studentene.

### **Rollen som oppdragsgiver**

Studentene oppfattet det som positivt å ha en veileder som hadde rollen som oppdragsgiver og som ikke diskuterte konkrete programtekniske spørsmål.

Det legger seg et preg av realisme, at oppdragsgiver mer eller mindre later som han ikke skjønner den tekniske delen av det her. Så han sier bare han vil at du skal kunne gjøre med det og så får vi heller finne ut om hvordan vi skal gjøre det. Det synes jeg egentlig er fint.

Problemer som de ikke klarte å løse tok de opp med kursansvarlig på forelesning eller i egne veiledningsmøter.

### **Genuine samarbeidsprosesser?**

Medlemmene i gruppe 1 hadde lang erfaring med å arbeide sammen fra tidligere kurs og prosjekter. De kjente hverandre godt, både styrker og svakheter. De satt på høgskolen når de arbeidet med prosjektet. De satt i samme grupperom og programmerte som regel to og to. De beskrev det slik:

*S1: Nesten alt vi gjør av programmering og sånt noe foregår på skolen.*

*S2: Så da klarer vi å koordinere det når vi sitter ved siden av hverandre i alle fall.*

*S3: Vi jobber stort sett to og to.*

Det var de samme parene som jobbet sammen gjennom hele prosjektet. De følte at de var samkjørte og mente det vil bli for mye rot å skifte på det. De mente også at parprogrammering var en hjelp til å skrive kode fordi det hele tiden var to til å finne feil og komme med løsninger.

*S1: Når man jobber i par så funker det greit.*

*S2: Det er veldig all right, egentlig, da får jeg kommentarer på alt det gjerne jeg skriver med en gang.*

Når de programmerer i par så pleier de å skifte på hvem som sitter ved tastaturet. De fortalte:

*S1: Vi pleier å bytte litt innimellom.*

*S2: Når frustrasjonen blir for høy.*

*S3: Det er vel mest når han som skriver blir lei av å ta alle klager.*

De var komfortable med å sitte sammen når de skulle arbeide med prosjektet. Det gjorde arbeidet med koordinering enkelt. Alle visste hvor langt de var kommet og hvilke oppgaver som skulle løses. De kunne spørre hverandre om hjelp, diskutere løsninger og støtte hverandre.

*S1: Det er jo også veldig bra at vi jobber sammen hele tiden, så vi kan løpe og spørre hverandre og sjekke. Man må ikke vente til dagen etter, hvis det ikke funker....så... vi passer jo på hva alle gjør, i alle fall sånn littegrann innimellom, når vi har mulighet.*

*S2: Ja. S3: Det må man jo.*

## **Gruppe2**

Gruppe 2 bestod av tre gutter som tok førstesemesterkurset Innføring i programmering. De kjente hverandre ikke fra tidligere og hadde ingen erfaring med å arbeide sammen. Dette var det første prosjektet de skulle levere ved informatikkstudiet, og deres første kurs i programmering. Målet for prosjektet var å lage en applikasjon som generer rapporter i html basert på en database om

geografiske data. Oppdragsgiver for prosjektet var også kursansvarlig, og samme person som holdt alle forelesninger i kurset.

## Utskifting av gruppemedlemmer

Et spesielt trekk ved denne gruppen var en stor utskifting av gruppemedlemmer i starten. Av den opprinnelige gruppa var det bare en person tilbake mens to nye hadde kommet til underveis. Det var ikke fordi at gruppen hadde fungert dårlig i utgangspunktet, men fordi det var et innføringskurs på første semester, og det er vanlig med et stort frafall av IT-studenter tidlig i utdannelsen. Gruppa opplevde det som ganske frustrerende. Likevel mente de at løsningen av prosjektet ikke hadde blitt berørt av det. Den strukturerte arbeidsmåten med ukentlige møter og arbeidsoppgaver sørget for en jevn produksjon og fremdrift selv om medlemmene skiftet.

## Veiledning

Gruppa hadde ukentlige møter med sin oppdragsgiver. Et typisk møte startet med at de gikk gjennom arbeidsmålene fra forrige uke som var lagt inn i Trac som trac-oppgaver. Deretter definerte de nye arbeidsmål for neste uke. Studentene forklarte:

*S2: Vi setter jo mål for hver forskjellige sånn milepæl, og så setter vi sånne forskjellige tickets til det. Så, det første vi gjør er å gå gjennom de. ... Vi ser hvordan det har gått og om vi har gjort det vi skulle og sånn. Og så setter vi eventuelt nye mål for videre arbeid.*

*S1: i samråd med oppdragsgiver.*

Det var oppdragsgiver som skrev ned mål og arbeidsoppgaver for neste uke, men det ble gjort i dialog med studentene. Deretter fordelte gruppens medlemmer oppgavene seg i mellom. Det ble gjort direkte etter møtet med oppdragsgiver, og oppgavene ble fordelt parvis på studentene. Det vil si at to og to studenter tok ansvar for en oppgave og arbeidet videre med denne.

Studentene var veldig fornøyd med strukturen som opplegget ga dem, og at de hadde en oppdragsgiver som kunne planlegge det sammen med dem. Det la opp til en jevn progresjon som forpliktet til arbeid. De forklarte det slik:

*S1: Jeg synes det har vært veldig greit å ha de delmålene, de milepælene og de ticketene å forholde seg til fra gang til gang.*

*S2: Ja, for det gir jo en veldig bra pekepinn på hva vi burde gjøre. Ellers er det jo litt vanskelig å vite, hva skal vi gjøre for noe nå liksom.*

*S1: Ja.*

*S2: Det er veldig kjekt å gjøre det sammen med veilederen også.*



*S1: Det gjør det veldig mye mer oversiktlig også.*

*S2: Så har man det liggende der liksom. Det var et ganske fint verktøy det.*

Deres erfaringer med tidligere prosjektarbeid var at det ofte er vanskelig å komme i gang med arbeidet, og så utsetter man prosjekter til de siste ukene før innlevering. En student beskrev det slik:

*S1: Man surrer litt i begynnelsen og så blir alt av arbeid gjort mot slutten, kanskje to-tre uker før innlevering. ... Nå føler vi at vi er mer i rute ja, Vi har satt oss klare mål fra begynnelsen av og stort sett oppfylt dem.*

I dette prosjektet hadde de arbeidet jevnt fordi de hele tiden satte seg mål for hva som skulle gjøres. De hadde en oversikt over hvilke problemer som skulle løses. De var beskrevet som konkrete arbeidsoppgaver i Trac, og ble de ikke ferdige med arbeidsoppgavene, så måtte de videreføre dem til neste milepæl. De forklarte det slik:

*S1: Det at du får arbeidet jevnt fordelt gjennom året, og at du har faste tidspunkt som ting skal være ferdig på. Samtidig, hvis man ikke skulle klare å få det ferdig i tide så kan det også eventuelt videreføres til neste milepæl.*

## **Parprogrammering**

Gruppen praktiserte parprogrammering når de skulle løse arbeidsoppgavene, slik de hadde blitt anbefalt av faglærer, og slik det legges opp til i XP. De fortalte at de samlet seg på et prosjektrom minimum en dag i uken hvor de satt parvis og programmerte hele dagen. De opplevde parprogrammering som en effektiv metode til å løse problemer på, samtidig som de hadde glede av å kunne løse problemer sammen. Studentene forklarte:

*S1: Vi føler at det har fungert forholdsvis greit. Det blir som regel til at en sitter og ...*

*S2: en sitter og skriver og så kommer de andre med sånne innspill, da.*

*S2: Vi fikk anbefaling om å gjøre det, egentlig. Så da bestemte vi oss at vi like gjerne kunne gjøre det.*

*S1: Ja, og som sagt, det var uvant, men jeg synes at det har fungert veldig bra, ... hvis jeg for eksempel sitter og skriver og sitter fast med noe. Så ser kanskje Even noe som ikke jeg har sett da, fordi jeg konsentrerer meg helt om koden min, og koden min er det fulle og det hele, og så kommer han med noen innspill. Joda, det fungerer. En ganske ålreit metode å jobbe etter egentlig.*

*S2: Ja, vi har fått løst en del problemer vi har kommet over. Det er jo morsomt å sitte sammen også, så funker det liksom.*

*S1: Ja, dele gleden over og sammen få et program til å fungere.*

Når gruppa mistet et medlem og de var tre stykker tilbake, ble det vanskelig å dele oppgavene parvis. I stedet valgte de å programmere sammen som en gruppe. En satt og programmerte mens de to andre kommenterte.

### **Oppdragsgivers rolle**

Oppdragsgiverrollen var ukjent for studentene i undervisningssammenheng. Likevel oppfattet de den som et positivt bidrag. Den skapte forventning om at oppdragsgiver hadde en interesse i prosjektet, og at han kunne ha synspunkter til hvordan studentene skulle løse prosjektet. Samtidig mente det at dette bidro til å gjøre prosjektet mer realistisk. De forklarte:

*S3: Det at man definerer han som en oppdragsgiver framfor en veileder. Da synes jeg at du får et helt annet syn på den personen. En oppdragsgiver er til syvende og sist den som har gitt oppdraget, og den som skal motta det endelige resultatet. En sånn person vil kanskje få lite granne mere å si på hvordan det endelige resultatet skal se ut, enn om det bare er en vanlig veileder.*

*S1: Ja, det var veldig nytt, og jeg synes det var en litt morsom måte å gjøre det på. Det blir litt mer rettet mot, kan du si, hvordan man kanskje vil gjøre det i yrkeslivet. Så det synes jeg var absolutt veldig positivt, at man tørr å gjøre noe sånt.*

### **Positive til samarbeid**

Medlemmene i gruppen uttrykte et positivt syn på det å samarbeide. De opplevde at det var både motiverende og forpliktende for arbeidet med prosjekter.

*S1: Jeg føler at det er veldig ålreit å jobbe i gruppe. Da har du en gruppe som forventer noe av deg, og hvis du ikke gjør noe så svikter du ikke bare deg selv, da svikter du gruppen din.*

*S2: Ja, det er ganske god motivasjon for mange det der altså. De var positive samarbeidsformen selv om det også oppstod problemer underveis, som at de mistet gruppemedlemmer og nye kom til.*

*S1: Klart det er jo en utfordring også, for som sagt man kan jo ikke få det helt som en vil gjennom hele prosjektet, en må jo gi og ta litt og bli enige innad i gruppen.*

*S3: Det er alltid fare for, sånn som har skjedd med oss, at vi mister litt personer og sånne ting, men det har stort sett vært et bra gruppearbeid, føler jeg.*

De måtte også forholde seg til at kunnskapsnivået blant medlemmene var forskjellig:

*S1: Men vi har ikke akkurat tatt skade av det. Det er vel bare vi som har kunnet litte granne mer som har måtte bremse ned et par hakk og tenke over ting. Og det har vært godt for oss og det. Da har vi kanskje måttet tenke igjennom både en og to ganger hvorfor vi gjør det vi gjør og. Ja, du lærer mye av å lære bort og, kan du si.*

### **Gruppe3**

Gruppe 3 bestod av fire studenter som fulgte kurset Programmering for web i 3. klasse. Tre av disse kjente hverandre godt fra før, hørte til samme studieretning og hadde arbeidet sammen i flere sammenhenger tidligere. En av de tre utpekte seg som en leder for gruppen. Den fjerde studenten tilhørte en annen studieretning og hadde problemer med å innordne seg i gruppen. Mens de tre stort sett møttes på skolen og jobbet sammen hele tiden, holdt fjerdemann seg på avstand og jobbet hjemmefra. I tillegg til faglige nivåforskjeller i gruppen ble etter hvert dette kilden til en konflikt. I intervjuet med gruppen var kun to av studentene tilstede.

### **Struktur**

Gruppen hadde møter med sin oppdragsgiver annenhver uke. I møtene viste studentene frem hva som var gjort siden sist, før de ble enige med sin oppdragsgiver om hva som skulle gjøres før neste møte. Arbeidsoppgavene ble i møtet formulert som trac-oppgaver. I flere av møtene måtte arbeidsoppgavene videreføres til neste milepæl fordi studentene ikke hadde prioritert arbeid med prosjektet siden sist møte. Det kom tydelig frem at studentene ikke var spesielt fornøyde hverken med denne måten å arbeide på eller med måten kurset er organisert på. De opplevde metoden som en tvangstrøye, og trengte etter egen oppfatning ikke strukturen som metoden gir. De begrunnet dette med at de er disiplinerte og greier å strukturere seg selv:

*I2: Har rytmen, strukturen vært viktig for dere eller ikke viktig i prosjektarbeidet. Det at man hver gang dere møtes deler opp i små oppgaver stiller krav til løsning til neste gang. Nye oppgaver. Har dere klart dere like greit uten?*

*S1: Jeg kan svare for min egen del har det ikke vært nødvendig, jeg føler at jeg klarer godt å ta ansvar for hva som bør være gjort i løpet av en periode. Jeg stiller ganske høye krav til meg selv over hva som skal leveres. Og så vet jeg at det er andre som ikke stiller krav i det hele tatt. Det siktes ikke til gruppen.*

Men til tross for at de etter egen oppfatning ikke trengte struktur, utsatte de prosjektarbeidet. Ved flere anledninger klarte de ikke å gjøre ferdig arbeidsoppgavene som lå lagret som trac-oppgaver fra ett møte til det neste.

Denne gruppen nådde ikke målsetningen om jevnere arbeidsinnsats, men studentene selv oppfattet ikke dette som en problemstilling. De arbeidet sammen i flere fag, og sa selv at dette prosjektet fikk lavest prioritet siden det hadde den siste innleveringsdatoen. Det var tydelig at de ønsket å arbeide på denne måten. På spørsmål om ikke oppdragsgiver burde ha stilt strengere krav underveis, svarte studentene:

*S1: Det kunne kanskje vært noe hardere underveis, på den annen side har jeg vært glad for at det IKKE har blitt stilt krav, for det hadde vært innimellom at jeg har måttet si at de her to ukene har jeg ikke mulighet for å jobbe med faget, men neste to ukene har jeg tid så da blir det bare det faget, den muligheten der er jeg glad for å ha på skolen. Hvis jeg ikke har hatt den muligheten hadde det blitt utrolig vanskelig å gjennomføre den høsten her.*

### **Praktiserer ikke parprogrammering**

Tre av gruppemedlemmene hadde arbeidet sammen i flere sammenhenger tidligere, og sa selv at de i liten grad hadde gjort noen endringer i arbeidsform i forhold til hvordan de hadde arbeidet sammen tidligere. Gruppen beskrev en arbeidsform hvor tre av medlemmene tilbrakte ganske mye tid sammen, men hvor mye av arbeidet likevel ble gjennomført individuelt. Den fjerde studenten utførte mesteparten av arbeidet hjemmefra.

*I1: Dere ville høyst sannsynlig ha fortsatt med samme arbeidsformen, dersom det ikke kom spesielle krav om at man skal bruke sånn eller sånn arbeidsform.*

*S1: Delvis i alle fall, det kan hende at hvis man følte man hadde mye bedre tid på skolen, så kanskje endret det der litte grann. Men for min del henger alt det sammen med at vi har hatt en høst med en arbeidsmengde som er enorm. Hvor vi har hatt et fag som har hatt en arbeidsmengde på 20 studiepoeng i stedet for 10. Da har det aldri vært noe energi å prøve nye ting, da har det vært å ta den arbeidsformen som en synes passer best.*

*I2: Og det er?*

*S1: Den løsningen som vi har beskrevet nå. Vi har delt opp litt oppgaver oss i mellom og i og for seg sittet på samme arbeidsrom og hjulpet hverandre hvis det skulle være et problem underveis, men en har egentlig programmert en for en.*

Gruppen praktiserte altså ikke parprogrammering. De begrunnet dette med de faglige nivåforskjellene i gruppen, og med at en at studentene jobbet hjemmefra og at det dermed ble vanskelig å danne grupper.

*S1: Det hadde ikke vært noe i veien for å prøve det. .... Gruppen har bestått av store forskjeller i programmeringskunnskaper. Det er en ting, så har man delt inn oppgaver etter hvilke kvaliteter man har der å, mens da en 4. mann har vært vekk så er det på en måte 3 stk som skal danne gruppe på 2 igjen, det har ikke løst seg så veldig, det har ikke vært naturlig i det hele tatt.*

*I2: Hvis det hadde vært krav om at det skulle være parprogrammering hadde det vært støy eller hadde det vært hjelp?*

*S2: Det hadde skapt mere vansker enn nytte, sånn som det er nå. Hvis jeg ser for meg to personer som har veldig forskjell på kvaliteter på programmering vil det bli at den ene totalt overkjører den andre, det er ganske naturlig.*

**Skjev arbeidsdeling** Oppgavene ble fordelt etter kompetanse. Dette førte til en skjev arbeidsdeling. En av studentene gikk inn i en lederrolle og påtok seg koordineringsansvaret.

*S2: (...) Alt av oppgaver som har blitt delt inn har blitt delt inn etter kvalitetene til dem som skal programmere dem. Det er ganske lett å se at det er enkelte som nesten ikke har kodet noe særlig. Mens andre har kodet mye. Da har oppgaven blitt delt inn sånn: det vanskelige havner hos den som kan kode mest, det andre havner hos den som ikke kan så mye kode. Og ellers er det hvem som tar ansvar for koordinering. Det meste av arbeidet går ut i fra personligheten til folk og hvem det er som har interesse ikke minst ovenfor at det skal bli gjort, og ikke bare kommer til daglig på skolen og ta det som det kommer.*

Utsagnet over – som kommer fra gruppens «leder» - gir inntrykk av en lite samkjørt gruppe hvor følelsen av lojalitet, samhold og gjensidige forpliktesler i forhold til de andre gruppemedlemmene ikke er særlig sterk. Enkelte av gruppemedlemmene har bidratt med lite, og beskrives i tillegg som uengasjerte. Den skjeve arbeidsdelingen førte til en «oppvask» i gruppen, som de selv beskrev på denne måten:

*S1: Jeg har ikke noe i mot å fortelle om det. Det er enkelt og greit at det ble følt at det var et lavere bidrag fra halvparten av gruppen. Hvor en av den andre halvparten gikk litt hardt til verks at enten så øker man innsatsen nå eller så splittes den her gruppen i to. Arbeidsinnsatsen ble øket eller så hadde vi ikke sittet sammen som en firemannsgruppe i dag.*

Gruppen økte arbeidsinnsatsen noe som en følge av «oppvasken», men gjorde ingenting med arbeidsformen. Resultatet til slutt ble at flere oppgaver ikke ble gjort og dessuten at de fikk store problemer når de skulle sette sammen koden som var skrevet av de forskjellige gruppemedlemmene:

*S1: Du kan si at resultatet er en fungerende kode, sånn som det er kodet nå, men det hadde aldri gjort noe og hatt hardere standarder på hvordan ting skal være, for når vi kommer nå og skal begynne og.... Jeg ser på i vertfall på en bit jeg har kodet at hvis jeg hadde gjort det litt annerledes så hadde det vært lettere for han som implementerer selve siden og å få det inn. Det er ganske enorm endring dersom jeg skal endre det nå så det blir neppe gjort.*

## **Negative til verktøybruk**

Studentene forklarte at de opplevde Trac som et innviklet verktøy det var vanskelig å sette seg inn i, at det ble gitt for liten opplæring i verktøyet og at de ikke ønsket å prioritere dette. De mente de fint kunne klare å håndtere strukturen i prosjektarbeidet uten Trac:

*S1: Helt ærlig så kan jeg si litt om det Trac greiene her, og det systemet så kan jeg si litt om hva jeg synes om det. Det er en forferdelig dårlig løsning, i forhold til kursingen du har i å bruke det. Det har stort sett bare vært bare rot. Det som står der inne har vi hatt bare for å følge det som en del av kurset. I og med at det har vært lagt opp til at vi skulle bruke det så har vi brukt det, for at vi må, ikke for at vi har funnet noe som helst nytte i det underveis.*

*I2: Fortell hvorfor du ikke har hatt nytte av det?*

*S1: For at kommunikasjonen i gruppen alltid har vært såpass bra uansett oss imellom så vi har alltid visst hva vi skulle gjøre, de få punktene kunne vi like gjerne skrevet på et papir og hengt opp på veggen og krysset ut hvem som har gjort det. (...)*

*I2: Og dere hadde ikke behov for trac for å gi den strukturen?*

*S2: Nei, den strukturen klarer vi fint uten trac.*

Når det gjelder Subversion, gjorde gruppa et forsøk på å sette opp systemet i begynnelsen av prosjektperioden, men terskelen med å sette seg inn i systemet ble for stor i forhold til hva de betraktet som nytteverdien.

*S1: Ja, jeg har fått med meg hensikten med subversion, virker i og for seg smart, men greia der er at vi, vi brukte litt tid for å sette oss inn i det, så fikk vi ikke taket på det med engang, også så vi kanskje ikke den store nytten for vår del i det i forhold til det vi skulle gjøre, så skrinla vi hele greiene.*

## **Oppdragsgivers rolle**

Gruppen hadde en ekstern oppdragsgiver, det vil si en annen faglærer enn kursansvarlig. De oppfattet oppdragsgiver som en kunde som skulle ha

produktet, og var positivt innstilte til å fokusere på mer på resultatene enn på implementasjonsdetaljer i møtene.

*S1: Jeg ser forskjellen på at når man har en oppdragsgiver så koder man kun etter resultatet, at man gjør det for å oppnå et resultat som tilfredsstiller oppdragsgiver. Mens hvis du hadde hatt en faglærer som var oppdragsgiver lagt mye mer vekt på programmeringsstandarder og sånn som ble gitt i kurset da.*

*I2: Føler dere at det hadde vært en fordel at programmeringslæreren hadde vært tydeligere til å stille krav til?*

*S1: Jeg synes det har vært litt spennende å ha oppdragsgiver som ikke har stilt så høye krav, men mer interessert i et resultat også, løst det på akkurat den måten vi villet bare det blir løst på en måte. Det har vært litt allright fordi engangs skyld noe annet enn hva en er vant med. Så jeg vil ikke si at det er noe dum løsning.*

**Kritiske til kursopplegg** Gruppen syntes at kurset var for omfattende i forhold til uttellingen (10 studiepoeng). På spørsmål om hva slags endringer de kunne tenkt seg i kursopplegget, svarte de at de ønsket mindre prosjekt, grupper med færre gruppemedlemmer og den ene studenten svarte dessuten at han ønsket en annen samarbeidspartner på samme faglige nivå som han selv:

*S2: Ta halve kurset, pirk bort litt av teknologiene, lag noen øvingsoppgaver, til den biten der og så ikke nevnt detaljer rundt prosjektene før en er halvveis. En starter ikke med prosjektene før en er halvveis, det synes jeg hadde vært en bedre løsning.*

*S1: Hvis jeg kunne fått det optimalt, hadde jeg kodet sammen med en person som har akkurat de samme interessene, eller har like høye ambisjoner og interesser av å få samme resultater som står i stil med ambisjonene og ha gode programmeringskunnskaper. Og så ville man hatt at man stort sett at man sitter slik som vi gjør nå sitter på et rom 4-5 timer minst sammen om dagen. Prater og jeg føler at kommunikasjonen går bra nok da.*

Gruppen opplevde videre at det var avvik mellom oppdragsgivers ønsker og progresjonen i prosjektet og progresjonen i kurset:

*I1: Åssen har kurs progresjonen/ forelesningsprogresjonen vært kontra oppdragsgivers ønsker og det dere følte at dere trengte underveis?*

*S1: Hva skal jeg si. Jeg er egentlig ikke så veldig fornøyd med forelesningene, For min del kunne halvparten vært kuttet ut. Det har de også blitt, ikke fra forelesers, men fra min side. For jeg føler at mye av det som blir forelest om har vært helt totalt bortkastet for meg ut fra det*

*problemet som jeg skal løse. Her har du igjen den greia at du har en ekstern oppdragsgiver og du vet hvilket resultat du skal frem til. Sånn som jeg har gjort det, isteden for å gå og høre om en forelesning om XML som jeg ikke kommer til å bruke, eksempelvis, så har jeg heller satt meg ned for å se på Javascript for meg selv, for det er det jeg vet at jeg skal bruke. Hvis jeg hadde gått og hørt på XML så hadde jeg kastet bort to timer på noe som jeg ikke får noe løsning på. Dette henger sammen med tidsbruk. Hvis jeg hadde hatt veldig god tid så hadde jeg gått på begge tinga.*

## **Gruppe4**

Gruppe 4 bestod av 1 jente og 3 gutter som fulgte kurset Programmering for Web i 3. klasse. Gruppen hadde en annen faglærer enn kursansvarlig som oppdragsgiver.

### **Struktur**

Gruppen hadde møter med oppdragsgiver omtrent hver 14. dag hele gjennom hele prosjektperioden. I møtene ble de enige med oppdragsgiver om hva som skulle gjøres frem til neste møte. Målsetningene ble formulert som milepæler i Trac. Dagen etter møtet med oppdragsgiver, pleide gruppen å ha et eget møte hvor de brøt hver milepæl ned i mindre deloppgaver, formulerte disse som trac-oppgaver og fordelte ansvar for disse.

*S3: Det vi har gjort er at på tirsdag har vi fordelt ticketene og begynt å jobbe med dem og hvis vi har sett at det har vært behov for det, så har vi tatt ekstra møter på fredag for å kartlegge hva som står igjen og hvis det hadde vært en stor oppgave, store tickets. Og så hvis vi ikke har følt at det har vært behov for et ekstra møte på fredagen så har vi bare hatt msn-kontakt i løpet av uken og så har vi vært klare til mandagen.*

Studentene forklarte at de i og for seg var fornøyde med å ha jevnlig møter med en oppdragsgiver fordi dette sikret progresjon i prosjektet, og at de gjorde det de kunne for å bli ferdige med avtalte oppgaver før neste møte. Studentene var imidlertid ikke fornøyde med form og innhold i møtene, og følte at formulering av milepæler og trac-oppgaver tok fokus vekk fra det kurset egentlig handlet om.

### **Kritiske til undervisningsopplegget - ønsker "oblig"-kurs**

Studentene etterlyste fastere struktur med krav til hva som skulle leveres og hvordan, samtidig som de ønsket at noen kunne verifisere eller godkjenne koden deres. Gruppen var dessuten misfornøyd med forholdet mellom progresjonen i



prosjekt og forelesningene, og ga uttrykk for at de savnet en litt grundigere innføring i pythonprogrammering før de startet med prosjektet. Ideelt sett hadde de ønsket seg et todelt kurs, hvor første del hadde bestått av forelesninger og en serie med obligatoriske innleveringer, og et mindre prosjekt i andre del:

*I2: Utbyttet av å jobbe i prosjekt, hvordan har dere det i forhold til, altså, nå tenkte jeg i veldig tradisjonelt opplegg med obliger og skriftlig eksamen, individuell tilslutt. Hva foretrekker dere?*

*S1: Jeg ville nok foretrekke å droppe prosjekt og hatt obliger. Eksamen blir jo litt meningsløst i programmering synes jeg, det er jo ikke en reell, når du sitter og programmerer så sitter du jo med nettet og kompilator og... Du sitter ikke og skriver på papir. Men det obligatoriske det synes jeg...*

*I1: Problemet med det obligatoriske er at da er man nesten nødt til å ha en eksamen da for at da blir oppgavene like. ---- Man lager like systemer. Sånn så det er liksom skal hvis man skal ha obliger så blir det eksamen, det blir nesten det, nesten nødt til det, ellers så må du ha prosjekt. For så blir det ikke noe, ellers så blir det alt for likt. Og da blir altså koden til studentene lik altså. Det kan jeg love deg at den blir.*

*S2: En kombinasjon synes jeg kanskje egentlig hadde vært det beste, og hatt obliger først så man lærte litt bedre hvordan man på en måte programmerer med Python, og så kunne man hatt et prosjekt til slutt som det kunne blitt tatt en bedømming av. Det synes jeg.*

## **Ønsker faglig veiledning**

Studentene oppfattet oppdragsgiver som en arbeidsgiver, og ikke en faglig veileder. Studentene uttrykte en negativ holdning til denne rollen, og ville heller hatt en faglig veileder.

*I1: Men hvordan oppfatter dere veiledningssituasjonen? Hvilken rolle har oppdragsgiver i denne situasjonen?*

*S3: Hvis jeg ser på han som en ren arbeidsgiver, så synes jeg det er greit. men hvis han i tillegg skal fungere som en veileder, så synes jeg ikke... Da synes jeg han på en måte har kommet litt til kort. For da har vi sittet mer og tenkt på layout og design enn vi faktisk har tenkt på funksjonalitet i systemet, sett på kode og diskutert litt den delen, da.*

*I1: Men er det ikke oppdragsgiver han skulle være i prosjektet?*

*S1: Jo, det er jo det, så feilen her ligger strengt tatt ikke hos oppdragsgiver, men jeg synes det er et veldig merkelig valg.*

*I1: Kan du fortelle hvorfor du synes det?*

*S1: Altså når du går ut og søker etter jobb, så ser ikke folk etter... altså de ser på resultatet av prosjektet... karakteren. De ser ikke på om hvor vidt tracen var ok utformulert.*

I møtene med oppdragsgiver syntes studentene det ble fokusert for mye på design, funksjonalitet og formulering av trac-oppgaver og for lite på programmering. Gruppen opplevde det som problematisk at oppdragsgiver la vekt på andre sider ved programsystemet enn hva faglærer formidlet som viktig ved evaluering. De forklarte videre at de følte de ble dyttet i feil retning på møtene, og at de måtte starte med en del av prosjektet de ikke følte de hadde forutseninger for å gjennomføre så tidlig i kurset. Det vil si at sett fra deres perspektiv var progresjonen i kurset ikke i samsvar med progresjonen i prosjektet. Gruppen hadde blant annet følgende å si om møtene med oppdragsgiver:

*S1: Jeg synes det burde vært mer fokus på programmeringsdelen i hvert fall. Altså når man møter med oppdragsgiver, har vi jo ikke sett på kode i det hele tatt, faktisk.*

*S3: Jeg føler det mer som et kurs i systemutvikling enn et programmering... det kurset faktisk heter.*

*I1: Okay? Kan du forklare litt.*

*S3: Det ble lagt mer vekt på design av systemet enn på funksjonalitet. Greit nok at vi hadde enkelte deler som skulle fungerer som vareregistrering og brukerregistrering osv, men det ble lagt veldig mye design på formulering av tickets og at de skulle være såpass tydelig formulert. Og det var...*

*S2: Fokuset på møtene har også vært mere... Hvordan ser systemet ut? Hvordan kan brukerne dra nytte av systemet og at vi måtte tenke veldig mye på hvordan ting så ut og sånn og etterpå, da... Så fikk vi høre det at utseendemessig, så var ikke det så viktig, det var koden som var viktig. Vil føler kanskje at vi har blitt dyttet litt i feil retning på møtene.*

*S3: Faglærer sier at layout ikke skal dømmes, men i gruppemøtene så sitter vi egentlig og bare designer.*

Gruppen uttrykte et tydelig ønske om en faglig veileder som kunne gi tilbakemelding på koden. Dette var et stadig tilbakevendende tema under intervjuet, og var tydelig noe gruppen var veldig opptatt av. Men til tross for dette sterke behovet for faglig veiledning, hadde de aldri benyttet seg av muligheten de hadde til å oppsøke faglærer for å få hjelp:

*I1: Dere ville heller hatt en faglig veiledning enn at man på en måte skiller litt sånn... altså fag og oppdragsgiverrollen på en måte?*

*S1: ja*

*S2: Altså istedenfor da og bare ha en oppdragsgiver, et møte med oppdragsgiver, tror jeg det hadde vært nyttig for gruppen å få ha kanskje annenhver uke et møte med en veileder som på en måte hjelper til med kode. For vi er ikke i en jobbsituasjon nå. Vi er jo i en læringsssituasjon.*

*I2: Oppsøker dere faglærer liksom sånn... innimellom også, eller er det på*

*en måte...?*

*S3: Vi har hentet ned eksempelkode fra sidene hans. Men jeg har aldri snakket med han. Jeg har funnet svar på det jeg lurte på sidene. Men det hadde vært veldig kjekt med en evt. annenhver uke med en veileder som kunne sett om vi lå greit nok an.*

*I2: Ellers så er faglærer litt frustrert, da for det er ingen som spør han.*

*S2: men sånn er det jo. Det er sjelden noen går på møte til lærerne sine.*

*I2: Det er faktisk en del som gjør det. Som går og møter opp på kontoret til lærerne og spør om ting.*

*S1: Jeg føler at det er sånn høytidelig. Da må det være et veldig vanskelig problem.*

*I1: Bare for å si.... Dere etterlyser veiledning og faglærer etterlyser studenter som veiledes. Det er derfor jeg spør.*

*S1: Men nå tenker jeg mer på veiledning. Det skjedde vel aldri at vi sto fast på et problem. At jeg ikke visste hvordan jeg skulle teknisk gjøre det. Men det blir mer sånn at han kom og kanskje pekte ut ting som jeg ikke hadde tenkt på... som ikke var direkte feil, men som...*

*S2: Kunne hevet prosjektet... eller kvaliteten på prosjektet.*

*S1: struktur og... Mer sånne overliggende ting.*

## **Individualister**

Oppgavene ble forsøkt fordelt på “en mest mulig rettferdig måte”. Studentene forklarte at det meste av arbeidet utenom møtene ble utført individuelt. Noen arbeidet hjemme, andre på skolen. Ved behov, hadde de kontakt på msn. Gruppen hadde ikke praktisert parprogrammering. De begrunnet dette med at det oppfattet parprogrammering som en ineffektiv arbeidsform. I tillegg syntes de at læringsutbyttet ble større ved å programmere alene enn sammen med en annen person.

*I1: Kunne dere tenke dere å... Hvis man hadde fått... Hvis det hadde vært krav i det kurset om at man skulle ha drevet med parprogrammering.*

*Hvordan hadde dere syns det hadde vært?*

*S1: Jeg foretrekker jo alene, da, men...*

*S3: Jeg ser på det på samme måte som Helene at det er mer effektivt at man sitter med hver sin del, da. I hvertfall når man skal generere koden i utgangspunktet. At to får gjort mer på samme tiden enn det en får. Men jeg ser jo problemstillingen at det blir mer sammensveiset med en gang hvis man sitter to og to.*

*I1: Men dere ville ikke umiddelbart sett det som en fordel hvis man hadde det som krav at dere skulle parprogrammere?*

*S2: Nei, jeg syns i grunn man kan velge den arbeidsform man syns passer seg selv best. Altså. Kanskje for noen egner parprogrammering seg*

*bedre mens for andre liker å jobbe alene. Jeg føler på en måte at hvis man blir sittende og programmere to og to. Så er det en som sitter og programmerer og så er det en som sitter og sover.*

*S1: Ja. Og man lærer det jo bedre selv også. Det føler jeg i hvertfall. Jeg lærer ting bedre hvis jeg må gjøre det selv. Hvis jeg må sitte og... Se på den andre gjøre det...*

## **Negative til verktøybruk**

Gruppen brukte Trac til formulering av milepæler og trac-oppgaver, men ga altså uttrykk for at de syntes det ble lagt for stor vekt på bruk av Trac i forhold til hva de betraktet som kurssets læringsmål.

*S1: Jeg syns Trac og Subversion tok litt over. Tok litt fokus vekk fra det kurset faktisk handler om: Programmering. Det var altfor lite fokus på programmering. Hva er god programmering og hva er dårlig programmering. Hvordan skal vi løse dette her? Det var mer "hvordan skal vi formulere neste ticket". Det er jo ikke hele poenget.*

*S1: Jeg syns det kunne vært mer veiledning mot det som ble vektlagt i evalueringen av prosjektet. For sånn vi holdt på nå, så har liksom 80 % av tiden gått på trac-systemet og det skal ikke sensor se på engang.*

*I1: 80 %? Har dere skrevet så mye?*

*S1: Altså vi har sittet og diskutert hvor godt formulert er den enkelte ticket og.. ja!*

*S2: Vi har brukt mye tid på det, ja.*

*S3: jeg vet ikke hvor mye tickets de andre gruppene har lagt, jeg, men jeg følte på en måte at vi har lagd mer tickets enn vi har fokusert på selve systemet.*

*I2: Så det du sier at dere på en måte bruker mer tid på ticketene enn dere gjør på implementasjonen?*

*S1: Under veiledningsmøtene, så gjør vi det... Så klart at når vi sitter hver enkelt og jobber, så...*

*I1: Veiledningsmøtet? Jeg trodde dere brukte hele uken på å skrive tickets.*

*S1: Nei, nå snakker jeg om under veiledningen.*

De forklarte videre at de hadde hatt problemer med å sette seg inn i Subversion, og at de derfor ikke hadde brukt versjonskontrollsystemet på riktig måte:

*S3: Jeg hadde faktisk mer problem med å forstå Subversion ordentlig. For der hadde vi ikke noe ordentlig introduksjon. Tracen er jo såpass selvforklarende. Når du har sittet og fiklet litt med den. Det med Subversion vet jeg ikke om vi brukte riktig en gang. Jeg vet ikke om vi skulle forket ut en ny subversion fra hvert møte eller... Det fikk vi ikke noe*

*beskjed om heller. (...)*

*I2: Har dere hatt noe nytte av å bruke Subversion?*

*S1: Vi har i grunn ikke det.*

*I2: Ser du noe poeng i det i det hele tatt?*

*S1: jeg ser poenget hvis du bruker den rett.*

*I2: Hvis dere hadde fått litt mere opplæring i det. Eller hadde blitt brukt litt mere tid på forelesning, eller?*

*S1: Jeg ser helt klart poenget hvis det hadde blitt brukt rett, men sånn som vi brukte den nå, så er det ikke noe poeng bortsett fra du har en ren versjon av koden i tilfelle ett eller annet skulle skje.*

*I1: Det er jo å holde orden på forskjellige versjoner når man sitter og programmerer hver for seg.*

*S1: Sånn vi har delt opp oppgaven hvor en person har ansvar for en modul, så er ikke det noe problemstilling.*

*I1: Det har aldri vært noen sånne konflikter hvor dere må inn og...*

*S3: Det har jo vært tilfelle hvor vi har... Jeg har sittet og rettet på en del i samme filen som f.eks. Pål... Da har jeg han på msn i tillegg og spør: "hvorfor fungerer ikke dette?" eller "gjør du noe her nå?" Det hadde vi kanskje sluppet med subversion.*

## **Gruppe 5**

Gruppe 5 bestod av 4 gutter som fulgte kurset Innføring i Programmering i første klasse. De kjente ikke hverandre fra tidligere. Ingen av dem hadde programmeringserfaring fra før, men faglige nivåforskjeller utviklet seg likevel til et problem for denne gruppen. Gruppen har hatt ukentlige møter med veileder, men har hatt lite kontakt mellom møtene. De er enige om at de ville jobbet individuelt hvis de kunne valgt selv.

### **Selvutnevnt leder**

Gruppen gir et generelt inntrykk av å være dominert av en leder som også er faglig førende. Denne studenten er også den klart mest aktive under intervjuet.

*I1: Har dere noe sånn eksplisitt arbeidsdeling internt? At dere har noen prosjektleder eller noe sånt?*

*S2: S1 har vel blitt litt selvutnevnt.*

*I1: så dere gjør bare som han sier?  
latter\**

*I1: ja okay. Men dere har ikke hatt noen sånn formell diskusjon om det? Dere har liksom ikke bestemt dere for at du tar ansvaret for å lede og.*

*S1: Det skjedde vel bare helt i starten...*

### **Praktiserer ikke parprogrammering**

Spørsmål rundt parprogrammering ledet til en samtale om nivåforskjeller, og parprogrammering som pedagogisk virkemiddel. Studentene bekrefter at de ser at parprogrammering kan være en måte å bringe folk opp på nivå, men har ikke selv fått til dette. De sier at de har vært dårlige til å håndtere nivåforskjeller. Forskjellene har forsterket seg underveis. Diskusjonen forsterker inntrykket av at lederen tok over hele prosjektet.

*II: Nå vet dere... er sikkert orientert om at den metodikken som ligger til bunnen her kalles for extreme programming. Og en av delmetodene i XP er parprogrammering. At man alltid er to stk som skriver. Har dere praktisert det?*

*S1: Nei. Ikke jeg hvertfall.*

*II: Dere?(andre)*

*S2: nei*

*II: Er det noen som kunne sett at det kunne vært nyttig?*

*S1: Problemet er vel at folk kan ligge på forskjellig nivå. At det blir veldig. At en kanskje ikke klarer å følge med.*

*II: På en annen side er jo det en måte å bringe hele gruppen opp på.*

*S1: sant*

*II: åssen forholder dere til sånne forskjeller?*

*S1: Det har vel ikke gått så bra egentlig. Det har egentlig gått veldig dårlig.*

*II: hva tenker du på da?*

*S1: Det har blitt sånn at noen sklir ut og så blir de hengende etter. Og når vi kommer med nye sånne milestones og nye tickets i leveranser blir man hengende bak og så blir det bare verre og verre.*

*II: Nå er jo tross alt XP ikke ment som noen pedagogisk verktøy, men det er jo en av de sideeffektene man kunne tenkt seg å se på. At man faktisk greide å dra med seg flere i programutvikling.*

*S1: Problemet er vel det praktiske med å få til. Å faktisk være to og to sammen. Programmering er jo ikke noe syv til fire økt akkurat. Jeg for min del sitter mye om kvelder og netter.*

## **Gruppe 6**

Gruppe 6 bestod av 4 gutter som fulgte faget Innføring i programmering første semester. De arbeidet sammen også i et annet kurs.

## **Arbeidsform**

Ulik bakgrunn gjorde at gruppemedlemmene valgte å fordele kursene mellom seg, istedenfor at alle arbeidet med begge kursene. Dette ble etter hvert et problem, som de selv beskriver på denne måten:

*I: Ble det et problem?*

*S: Ja, det ble et stort problem. Det ER et stort problem ennå.*

*S: Det er et generelt problem trur jeg, at det er noen som er dyktige, det er noen som kan Java fra før så da konsentrere dem seg om å ta javadelen. Så får ikke dem - vi - som ikke kan det fra før, vi henger ikke med, for da går vi mer med samfunnsinformatikken, og så blir vi sittende i bakleksa, når vi da kommer og ser på hva dem har gjort, så er det veldig vanskelig å sette seg inn i hva som egentlig har skjedd, og kunne jo ikke gjøre noe med det.*

På grunn at stor arbeidsmengde og ulike innleveringsfrister på prosjektene i de to fagene, ble det ikke til at de prøvde ut parprogrammering som arbeidsform. I etterkant ser de at dette kunne ha vært hensiktsmessig med tanke på fordeling av kunnskap og kompetanse. De mener det ville fungert bedre med strengere føringer på arbeidsformen:

*I: Men du kan si da, det har ikke bidratt til å fordele på en måte kunnskapen og kompetansen hos dere?*

*S: Vi har ikke brukt det sånn, i alle fall.*

*S: Nei*

*S: Vi skulle nok kanskje ha gjort det, egentlig.*

*S: Har vel vært mer fokusert på å bli ferdig, enn å lære hverandre, og lære av hverandre, for å komme i mål.*

*I: Hvordan tror dere det hadde vært om vi hadde sagt at det skal dere gjøre, det skal stå at dere har gjort det?*

*S: Det hadde vært bedre*

*S: Det hadde vært bedre, det ja, for da hadde vi hatt det å forholde oss til måten det skulle gjøres på.*

## **Gruppe 7**

Gruppe 7 bestod av fire studenter, tre gutter og en jente, som fulgte kurset webdesign i femte semester. De kjente hverandre godt fra før og hadde også tidligere samarbeidet i kurs og prosjekter. Dette var den eneste gruppen som hadde faglærer som sin oppdragsgiver. Målet med prosjektet var å lage et spill. Tre studenter, to gutter og en jente, deltok på intervjuet.

Det som var spesielt med denne gruppen var at de i samme semester samarbeidet i flere kurs med prosjekter og innleveringer, og de valgte å dele arbeidsoppgavene på tvers av disse. Mens noen tok ansvaret for programmeringen i et prosjekt, tok andre ansvar for rapportskrivning i et annet.

De hadde som mål å være effektive, lage gode produkter og oppnå et best mulig resultat. Denne pragmatiske tilnærmingen til gruppearbeidet preget veiledningssituasjonen og samarbeidet i gruppa, og den påvirket den enkeltes læringsutbytte i kursene de fulgte.

### **Veiledning og rytme**

Gruppe 7 hadde jevnlige møter med oppdragsgiver. På disse møtene skrev vanligvis oppdragsgiver en til to delmål (user stories). Disse ble diskutert og skrevet i samarbeid med studentene. Studentene beskrev det slik:

*Milepælene er det oppdragsgiver som har lagt ut. Og så har han lagt ut en hovedticket, og så har vi lagt til tickets ettersom vi har oppdelt oppgavene.*

I etterkant av dette møtet, delte studentene arbeidsoppgavene opp i mindre oppgaver som de beskrev i nye trac-oppgaver.

*Vanligvis har vi diskutert hovedoppgavene som går på disse ticketene som vi har fått av faglærer. Så har vi koordinert dem på en måte, og laget egne tickets. Så har vi lagt de ut på Trac, og så har vi assignet de til en av oss.*

Studentene vedsatte denne måten som prosjektet ble drevet fremover av en oppdragsgiver. De forklarte:

*Det er veldig allright. Veldig bra, med møtene og alt. Var veldig fint, for da fikk vi deloppgaver hele tiden så vi slapp å... ja, vi fikk liksom vist litt veien vi skulle gå.*

*Får gjort litt hver gang, så du har et klart mål. Sånn at du slipper å tenke på alt med en gang. Slipper designersyndromet, med å designe hele systemet før du i det hele tatt kan begynne å kode. Vedig greit å begynne, sånn inkrementelt, slik at vi hele tiden bygger på. Starter med det mest essensielle. Oppdragsgiver har vært veldig allright sånn.*

*Hadde vi fått mer frie tøylar, sånn som vi ofte får, så ville vi mest sannsynlig jobba litt mer på slutten og litt mindre i starten tror jeg. Det er jo ikke så uvanlig det.*



## Samarbeid og fordeling av oppgaver

Når studentene skulle fordele arbeidsoppgavene i prosjektet, ble det gjort etter hvilke ferdigheter den enkelte hadde og basert på en samlet vurdering av alle innleveringer de hadde dette semesteret. Selv om det ikke var en planlagt strategi, gav de inntrykk av at det var en naturlig konsekvens av gruppens sammensetting og arbeidsbelastning. Det var heller ikke slik at det var noen som lurte seg unna og ikke gjorde noe.

De møttes omtrent hver dag, men på grunn av arbeidsfordelingen var det to av gruppemedlemmene som stod for det meste av programmeringen, mens de andre to bidro med blant annet dokumentasjon og brukerveiledning samt innleveringer i andre kurs.

Årsaken til at arbeidsdelingen ble slik, var etter eget utsagn at to av medlemmene er veldig flinke til å programmere, og at de to andre ble hengende etter fra starten av. Dette opplevdes som problematisk for de som ble hengende etter. De forklarte det slik:

*S1: Meg og "Per" har vært kode-folk. Hovedsaklig."*

*S2: Det har egentlig vært litt problematisk akkurat det der. De har hatt så veldig mye å gjøre med andre ting, så de har koda veldig mye hjemme.*

*"Jon" og jeg liker også å programmere, men de er milevis over oss når det gjelder erfaring og kunnskap. Så da ble det slik at de har delt koding mellom seg, og vi har koda veldig lite.*

*S2: I og med at vi jobber sammen i mange fag så har det jo vært mye sånn at.. ok.. når vi jobber med agentsystemer så jobber de med det. Det er jo ikke helt bra, for man skal jo jobbe like mye i hvert prosjekt, ikke sant, men når man jobber sammen om alt så er det sånn at, ok, jeg har mer å gjøre der, så ser jeg på det og så gjør dere noe annet."*

Et problem knyttet til dette var at læringsutbytte fra prosjektet var dårlig for to av studentene. En av studentene fortalte at for å kompensere for den manglende deltagelse i programmeringen måtte vedkommende kjøre sitt eget parallelle løp. Studenten fortalte:

*Oppdragsgiver sa at han ikke hadde lyst til å stoppe det (den raske utviklingen av prosjektet), og det skjønner jeg jo. Men for min del så følte jeg at det her bare suste over hodet mitt. Jeg satt meg ned og prøvde å bruke helgene på å lese meg opp ...*

## Parprogrammering og deling av kunnskap

De var ikke spesielt fornøyd med samarbeidsprosessene selv. De som hadde gjort mesteparten av programmeringen, sa at de foretrakk å arbeide alene. De hadde heller intet ønske om å involvere andre i sin kode. De forklarte

*Det har jo blitt mye sånn at vi har hatt ansvarsområder ... det har ikke vært så lett å koordinere, få noen andre inn på den koden på kort varsel. Fordi de tenker jo på så mye annet i tillegg. Så det har jo blitt litt sånn at jeg eier den ene halvdel av python-koden og "Per" eier den andre. I hvert fall er det vi som har laget den. Men det er jo sånn at vi fikser på hverandres kode likevel. Så det er ikke sånn at "den må du ikke røre" liksom.*

Det var også slik at de som hadde programmert lite, kviet seg for å se på programkoden til de andre. En sa det slik:

*Men jeg må jo si det at, jeg har ikke lyst til å gå inn og kludre i "Ole" sin kode. Eller "Per" sin, uten at de er tilstede. Men det er litt sånn, ja.. redd for å ødelegge noe.*

De reflekterte selv over at parprogrammering kunne være en mulighet for å løse dette problemet.

*Men en ide hadde kanskje vært å ... sette sammen folk som ikke er fullt så erfarne med folk som er erfarne med å programmere. Parprogrammering, for eksempel.  
... kanskje oppdragsgiver ser litt på hvem som er flinke og så prøver å oppfordre til å jobbe sammen med de som er gode for eksempel i dette tilfelle, at ikke "Ole" og "Per" skulle drive parprogrammering, men at jeg og "Ole" og "Per" og henne, for eksempel. Styre det litt.*

Men en student som hadde programmert mye responderte på dette med å si:

*Jeg trives jo egentlig best med å jobbe alene.*

Til dette utsagnet fikk han bred støtte fra de som ikke hadde programmert så mye også. Følgende kommentarer kom fra gruppen:

*Det gjør jo egentlig vi alle.  
Det er fordi vi er på IT, ikke sant.  
Det er alltid mest kos å sitte å kode alene.  
Jeg hadde trivdes mye bedre hvis jeg bare fikk være alene og lage min egen oppgave.*

## **Oppdragsgiver-rollen**

Gruppe 7 hadde som den eneste gruppen på dette kurset, kursets faglærer som sin oppdragsgiver. De uttrykte flere ganger at de var meget godt fornøyd med sin oppdragsgiver, som de oppfatter som både oppdragsgiver og faglig veileder.

*Jeg føler at han både er oppdragsgiver og veileder. Det er ikke noe problem å gå til han og spørre om noe man lurer på. Han kommer med egne forslag, og...  
Som han definerer seg selv, en teknisk kyndig oppdragsgiver.*

Selv om oppdragsgiver også var faglærer brukte de ikke møtene til å diskutere konkrete kodeproblemer. Diskusjonene gikk rundt produktet, mulige løsninger og tekniske løsninger på et mer abstrakt plan.

De ønsket ikke å legge skylden på oppdragsgiver for hvordan prosjektet var blitt organisert og programmeringsoppgavene var blitt fordelt. Som en sa det:

*Vi er jo voksne mennesker, så for min egen del så legger jeg jo skyld på meg selv også ....*

Likevel mente de at det hadde vært en fordel om oppdragsgiver oppfordret dem til å arbeide annerledes

*Det hadde vært bedre for oss, hatt det litt strengere og pusha litt på det. For da hadde vi fått mer deltakelse i programmeringa. For man lærer absolutt mest når man sitter og skriver koden selv. Det er noe helt annet å lese den etterpå. .. Så jeg tror nok det hadde vært bedre og hatt litt sterkere krav om XP.*

Også studenten som hadde programmert mye mente at det ikke hadde skadet prosjektet om de hadde blitt bremsset litt og ikke tatt styringa over programmeringen med en gang.

### **Forpliktelse i forhold til oppdragsgiver og medstudenter**

Grappa uttrykker at de føler en forpliktelse overfor oppdragsgiver til å bli ferdige med oppgavene de avtaler fra gang til gang.

*Sånn sett var det veldig greit at vi har faste møter. Da føler du at du skal gjøre det ferdig til den datoen. Det er ikke sånn at du utsetter det, liksom. Det er en forpliktelse. Blir ikke sånn som i agentsystemer. Nei... det skrev vi faktisk i evalueringen, at vi ønsket oss flere faste møter med faglærer.*

Det har også vært en gjensidig forpliktelse innenfor gruppa om å få ting ferdig. Men denne forpliktelsen har vært drevet av andre faktorer enn et gjensidige engasjement i et programmeringsprosjekt. Den har vært drevet av en gjensidige

forståelse av å produsere best mulig innleveringer i flere kurs. Gode resultater og produkter har vært viktig.

*(...) I og med at vi jobber sammen i mange fag så har det jo vært mye sånn at.. ok.. når vi jobber med agentsystemer så jobber de med det. Det er jo ikke helt bra, for man skal jo jobbe like mye i hvert prosjekt, ikke sant, men når man jobber sammen om alt så er det sånn at, ok, jeg har mer å gjøre der, så ser jeg på det og så gjør dere noe annet.*

### **Bruk av verktøy**

Grappa brukte både Trac og SVN. De brukte Trac i forbindelse med møtene med oppdragsgiver, men så ikke helt poenget med systemet. En student sa:

*Når det gjelder Trac, har jeg et litt rart forhold til det. Synes det ble litt sånn ...ja..vi brukte det kanskje ikke nok. Fikk ikke helt se nytte av det. Så det ble mer slik at vi gikk inn på Trac og oppdaterte det fordi vi måtte liksom. Det er det jeg føler.*

De mente ante prosjektstyringa hadde fungert like bra med mer uformelle verktøy

*Vi fikk jo oppgavene når vi var i møtet. Så jeg vet ikke... hvis vi bare hadde fått det skriftlig eller muntlig på møte og skrevet det ned, vet ikke om det hadde vært så veldig mye verre. Jeg tror kanskje det hadde fungert like bra."*

Derimot var de begeistret for SVN og de mulighetene som dette systemet gir. Se fortalte:

*Men SVN fungerer jo helt utmerket."  
Det gjør at vi kan sitte å jobbe på forskjellige ting samtidig. Og så er du alltid sikker på at du har siste utgaven, slipper å sitte å knote med å sende mailer hit og dit. Og oppdatere og slette. Kan være sikker på at det synkroniseres ordentlig, sånn at vi ikke har forskjellige versjoner av den ene filen fordi noen har glemt å gi den og sånn.*

### **Gruppe 8**

Gruppe 8 bestod av tre studenter i første semester som fulgte kurset Innføring i programmering. Denne gruppen skiller seg fra de andre studentgruppene i kurset, da den består av godt voksne studenter som samtidig med de ordinære studiene følger et kvalifiseringsløp for å kunne studere; Tress.

De synes generelt at det er mye å gjøre. De sliter med å henge med faglig og synes det er mange tekniske utfordringer. I tillegg til dette kurset samarbeider de også i andre fag.

## **Samarbeid**

Studentene i Gruppe 8 møtes stort sett hver dag på et grupperom for å jobbe sammen på prosjekter, innleveringer og for hjelpe hverandre. Dette gjelder ikke bare for programmeringskurset, men for alle fag de har på Høgskolen.

De bruker minnebrikker for å utveksle notater. Når de sitter hjemme, kommuniserer de mye med hverandre over MSN og e-post.

Arbeidsoppgaver som skal løses i prosjektgruppa fordeles etter hvilke ferdigheter de ulike gruppemedlemmene har. Det har medført at det bare er en person som har programmert på prosjektet. De andre har arbeidet med dokumentasjon eller innleveringer i andre fag. Som en student uttrykte det:

*Jeg har skrevet forsiden i HTML og det er mitt verk, men jeg har virkelig ikke noen oversikt over programkoden.*

## **Track og Svn**

De bruker Track på møtene med oppdragsgiver ellers minst mulig. Svn har de ikke vurdert å bruke. De synes slike systemer er tungvinte og overflødige, og ønsker ikke å sette seg inn i noe nytt.

*Vi har andre metoder som er lettere tilgjengelig enn det å sette seg inn i noe nytt, det er veldig greit verktøy det vi holder på med.*

De føler seg også usikre på å måtte lære et nytt system. De har problemer nok med å forholde seg til vanlige applikasjoner som msn og e-post.

*Så skal du da bruke masse ressurser på å lære deg et helt nytt system på toppen av alle de andre tingene som går, det er nesten uoverkommelig på så kort tid. Jeg må ha mye mere tid på det.*

De mener heller ikke at behovet er så stort siden de møtes hver dag. De har skrevet noen arbeidsmål i Track i form av trac-oppgave, men føler ikke at de har hatt bruk for det. De forteller:

*Vel, vi har lagt de ticketene der.. vi har egentlig ikke hatt noen bruk for det i det hele tatt.*

*Vi har gjort det veldig enkelt, vi da, ... Altså, når vi er ferdig med en ticket, så er den closet, og har vi nye, så setter vi på nye.*

De vil likevel ikke forkaste et slikt system helt og holdent, og ser anvendelsen som nyttig i større prosjekter. Men de mener de må få mer tid til å lære det

*Jeg ser stor nytte av et sånt system. Jeg har drevet mitt eget firma i mange år, og ser klart nytten av å ha et sånt. Men det er for mitt vedkommende, enda en sak jeg må forstå hvordan virker. Jeg må ha mer tid på meg, rett og slett, før jeg kan bruke det. Men jeg ville veldig gjerne brukt et sånn et, vi har vel et tilsvarende mønster, som da er ikke på datamaskin, men som er på den gammeldagse måten.*

I stedet har de laget seg egne strukturer og oversikter over arbeidsoppgaver som skal løses.

### **Veilederrollen**

De oppfatter veileder både som oppdragsgiver og veileder. De har behov for hjelp til å løse konkrete programmeringsproblemer, og da er ikke oppdragsgiverrollen alene dekkende.

De har møte med oppdragsgiver stort sett hver uke og oppsøker han også innimellom når det er noe de lurer på. På møtene definerer de nye arbeidsmål sammen med oppdragsgiver for neste uke. Det er oppdragsgiver skriver ned disse arbeidsmålene i en trac-oppgave.

## ***Veileder 1***

### **1. Oppdragsgiver, veileder, faglærerrollen.**

Det er i hovedsak to modeller som beskriver rollene. En har en veileder som er mer en oppdragsgiver som bestemmer veien videre og stiller krav underveis. Den andre har en veileder som lar gruppen i større grad slippe til med sine synspunkter og vilje. Flere av gruppene startet med en tilsynelatende god produksjon, men det ble senere avdekket at dette ikke stemte eller at det sklei ut. Veilederne uttrykker et samstemt ønske om at oppfølgingen i større grad må være basert på stramhet og ansvar til den enkelte deltaker. Flere av problemene var uavhengig av klasse trinn. Det ser ut som at det er viktig å avklare rollen til veilederfunksjonen slik at alle grupper får en nokså lik opplevelse av rollene i gruppemøtene og at deltageren får litt lik faglig støtte. Vår erfaring er at veileder i de aller fleste tilfellene er viktig for prosessen til gruppene. Det kan være interessant å se nærmere på forskjellige rollemodeller: Veileder, oppdragsgiver, faglig kyndig kontraktør, faglærer, studass, bedriftsleder, sjef, bytting av roller under møte og sensor, relatert til hvor i studiet kurset holdes (1-3 kl).

**2. Rytme og tempo (Fremdrift hos studentene).** De fleste gruppene startet med tilsynelatende god fremdrift, men flere av gruppene fikk problemer med å opprettholde produksjonen etter en tid. Vi registrerte i flere grupper en tydelig skjevfordeling av produksjonen. Veilederne er tydelige på at en tettere oppfølging med ansvar til den enkelte vil bidra til en forbedring. Brukt en tydelig arbeidsgiverrolle: ”gjort som en arbeidsgiver ville gjort, ringt han og tatt han inn og forlangt at vedkommende deltok i møtene”. Det er viktig å tydeliggjøre spillereglene for studentene.

**3. Forpliktelser i forhold til oppdragsgiver og medstudenter.** Premisser som ikke er helt tydelig for studentene medfører svakere forpliktelser ovenfor oppdragsgiver og de andre gruppedeltakerne. Det er derfor viktig å bruke en del ressurser på å gå gjennom spillereglene for prosjektgjennomføringen. Eksempel på spilleregler er: parprogrammering, testing, verktøybruk, arbeidsfordeling og ansvar. Her kan de forskjellige rollene til veileder være et bidrag til å styrke oppfølgingen av forpliktelsene som er inngått. Fagpersonalet uttrykte stor enighet om at tekst som beskriver mål, oppgaver og ansvar må være mer tydelig og ordrik enn det var i prosjektgruppene. Dette for å redusere missforståelser.

**4. Bruk av verktøy.** Gruppene tok i bruk Trac mens Subversion var det flere som ikke brukte. Noen brukte verktøyet som en ”minimumsløsning” det vil si at de gjorde en del, men lite aktivt bruk som arbeidsverktøy. En gruppe brukte verktøyene helt ut som et arbeidsverktøy i prosjektet. Et poeng er at verktøyet bør brukes for å lære studentene opp til å delta i en organisert kreativ prosess. Vi

erfarte at ordrike tekster og presisjon i tekst kan være viktig for å redusere misforståelser og hjelpe på progresjonen i produksjonen. En samstemt gruppe hevder: brukt aktivt og med tett og hyppige møter vil verktøyene bidra til et positivt læringsbidrag og være med å sørge for en jevnere arbeidsinnsats gjennom semesteret.

**5. Parprogrammering.** Det var svært få som brukte parprogrammering. Studentene er veldig bevist på at de tar bare i bruk elementer som de føler gir en direkte avkastning eller er knyttet opp mot evalueringen. Selv om enkelte i ettertid uttrykker at de tror det kan være brukbart å bruke. En del grupper brukte metoden ubevist i det de satt i samme rom og ved problemer så jobbet de på samme maskin. (De som prøvde det var positiv). Det er vanskelig å motivere studenter som er kommet langt i studiet (3. Kl) til å ta i bruk parprogrammering. De bruker de samme metodene som de har brukt før. Fagpersonalet hevder at for å få studentene til å bruke metoden må det være en mye tøffere styring med krav. Det vil blant annet si at ved ansvarstildeling får to ansvar for en arbeidsoppgave og at de skal sitte sammen å jobbe med den. Det er en utfordring at racerne for med seg noen og at de som er svake i programmering får med seg noen.

**6. Refleksjon.** Det er viktig at gruppedeltakerne får et eierskap til prosjektet og oppgaven, det vil si at de får en: ”mitt” og ”vårt” følelse. Hvis vi skal få den kreative delen av slike prosesser til å fungere så må vi legge forholdene til rette for å få folk til å være kreative sammen. Det kan se ut som studentene ikke er vant til denne å drive slike prosesser. For å få verktøyet til å fungere som en hjelp i prosjektprosessen så må tekstene som beskriver mål og arbeidsoppgaver være utfyllende og presise, det vil blant annet si med avgrensninger og ansvar. Det ser også ut som om at det bør brukes en del tid på å introdusere verktøyet og at krav til obligatorisk bruk er nødvendig for å nå læringsmålene i forhold til selve prosessen. Tett samarbeid og versjonskontroll er viktig for effektiviteten. En gruppe leverte en ”mildest talt fillete” kode, blant annet så testet de feil versjon i over en time. Veileder fremførte: ”Bruk av versjonskontroll hadde disse tjent voldsomt på”.

I gruppene var det stort sett slik at de som er gode til å programmere blir bedre mens de som ikke er så gode blir ikke så mye bedre. Vi tror at bedre styring av arbeidsfordeling å jobbe sammen i par kunne gitt større læringsutbytte for de som ikke var så gode. Gruppedeltakerne er klar over forskjellene og dette åpner for å ha en åpen diskusjon i gruppen for å bedre forholdene for at flere skal nå læringsmålene.

Det er en stor arbeidsbelastning med veiledning av flere grupper. Det kan derfor være aktuelt at det i større grad brukes bachelor og master studenter som veiledere, og at faglærer mer fungerer som en mentor for veilederne. Det kan



være aktuelt at faglærer har et par grupper for en mer direkte erfaring av faglige problem. Verktøy vi brukte er en støtte for veileder for å vite hva en holder på med i gruppen samtidig som det er dokumentasjon som støtter en evalueringsprosess.

Formalia som vi bør være strenger på er: Arbeidsfordeling, ansvar, bruk av Trac og Subversion, parprogrammering og testing.

**7. Samarbeid og kommunikasjon i gruppen.** Samarbeid og kommunikasjon i gruppen er stort sett avhengig av det faglige nivået til den enkelte innenfor kurstemaet, og viljen til å bidra til fellesskapet med å møte opp. Noe som kan gi et bidrag til forbedring av dette, er bruk av verktøyet og at mål og arbeidsoppgaver er fyldige samt tydelig i teksten. Bruk av verktøyet kan også bidra til at en tidligere kan gripe fatt i problemer og få dratt personer med før de eventuelt forsvinner ut av kurset. Parprogrammering kan bidra til å fremme samarbeid i gruppen.

**8. Oppdeling av oppgaver.** Det å la gruppene stå for oppgave beskrivelser og ansvarstildeling gir for stor frihet til ansvarsfraskrivelse. Vi så enkelte grupper hvor en "tar" all arbeidsoppgavene og i praksis hindre at de andre bidrar i prosjektet. Fagpersonalet hevder at mer transparent prosess sammen med at den enkelte arbeidsoppgave og ansvar tildeles i møte, vil bidra til bedre læring for flere av gruppene. Det er viktig at møtene holdes hyppige fordi dette bidrar til en jevnere produksjon i gruppen samt at arbeidsfordelingen kan justeres.

**9. Deling og arbeid med å bygge kunnskap.** I flere av gruppene jobbet enkeltpersoner eller deler av gruppen for seg selv og var lite villig til å bidra til kunnskapsutviklingen til de andre gruppedeltakerne. Her må vi legge til rette for systemer som kan utvikle samarbeid og bedre kommunikasjon i gruppene. Et par bidrag til dette kan være bruk av verktøyene og parprogrammering. Det som er vanskelig her er at vi ikke skal bidra til at enkelte fører at vi legger for mange begrensninger på deres mulighet for å lære og produsere.

**10. Krav/opplegg i kurset og kodestandard.** Ingen oppsummering.

**11. Tekst om skolestart intro.** Denne oppsummeringen er basert på hele teksten i intervjuet. Det å starte tidlig med å erfare mekanismer og fordeler/ulempes i prosjektgjennomføring er en fordel for det videre studiet. Det er mer snakk om rekkefølgen og tempo på temaene. Det er viktig at alle studenter har en felles plattform på disse temaene, dette fordi de senere tar kurs på tvers av studie et de er tatt opp på.

**12. Veiledning.** Fagpersonalet hevder at de skal ha en tettere styring av oppgave og ansvarsfordelingen ved å være tydeligere i de neste prosjektene de deltar i.

Det bør være en gjennomgang før kursstart med en tydeliggjøring av oppgave og veilederrollen slik at det er en mest mulig omforent fremstilling av denne. Det bør sies nærmere på de forskjellige rollene som en veileder har i slike prosesser: Veileder, oppdragsgiver, faglig kyndig kontraktør, faglærer, studentassistenter, bedriftsleder, sjef, bytting av roller under møte og sensor, relatert til hvor i studiet kurset holdes (1-3 kl). Det å bruke bachelor og masterstudenter som veileder samtidig som faglærer mer har en mentor funksjon ovenfor disse.

**13. Emne plassering i studiet.** Introduksjon til prosjektarbeid, utveksling av kunnskap i grupper, bør introduseres tidlig i studiet. Dette fordi studentene legger til seg vaner i studiet og vår erfaring er at de er lite villig til å endre de først vanene dersom de opplever at de fungerer for ”meg”. Dessuten er det vel slik at de gjør ikke mer arbeid enn de får uttelling for, og da er det vanskelig å få gehør for at det blir bedre i senere prosjekt.

## ***Veileder 2***

**1. Oppdragsgiver, veileder, faglærerrollen.** Det uttrykkes klart et behov for at veilederrollen må inneholde flere aspekter. I det ligger det at rollene og de aktuelle aspekter må være avklart mellom veilederne på det enkelte kurs og at dette tydeliggjøres ovenfor studentene. Et poeng her er å veilede for et bra produkt som gir god evaluering og trekke en klar minimumsgrense for å stå. I en gruppe brukte veilederen et perspektiv som arbeidsgiver som stilte krav til kunnskapsbygging ved å lage en arbeidsoppgave på at deltakerne skulle gå på forelesningene i faget. Flere av gruppene utfordret rollen som oppdragsgiver med krav til produktet, de prøvde å lage det de selv ville lage. Viktig som veileder at hele gruppen er med og lager klare begrensede arbeidsoppgaver og ansvar i de ukentlige møtene. Da dette bidrar til struktur og tempo. Det didaktiske opplegget kan bidra til at gruppen på et tidlig tidspunkt kan få hjelp dersom deler av gruppen ikke deltar aktivt i gruppen, fordi de blir overkjørt av ”overambisiøse” personer. Det kommer frem at sammenhengen mellom delmål (user story), milepæler og trac-oppgaver brukes forskjellig av veilederne.

Kommentar: Det bør kanskje avklares hvordan milepæler skal brukes. De kan være innleveringer underveis, men med en god produksjonsoppfølging i prosjektet kan det bety mindre behov for innlevering underveis.

Gruppene har liten trening i å bruke risikostyring og prototyper som kastes, i prosjekter de jobber med. De vil heller starte med det enkle og så bygge ut dette, selv om det lett medfører dårlige strukturer og svak kvalitet i kildekoden med hensyn på gjenbruk og vedlikehold. Veileder kan veilede to grupper som samarbeider om en løsning.

**2. Rytme og tempo (fremdrift hos studentene).** Vi registrerte at uten konkrete krav knyttet til personer skjer det ofte lite. Kravene må være tydelig og møtefrekvens på en uke fungerer bedre enn mer sjeldne møter.

**3. Forpliktelser i forhold til oppdragsgiver og medstudenter.** Flere av gruppene viser at de prioriterer å lage et prosjektresultat som er ” greit ” eller nok til å stå fremfor hva oppdragsgiver ønsker. I dette ligger det at de legger press for å unngå å forbedre kvaliteten i produktet, de lager minimumsløsninger på hver del fremfor et helhetlig kvalitetsprodukt og vektlegging av læringsmålene i kurset. Enkelte utøver en større forpliktelse ovenfor at produktet skal bli bra og sine egne ambisjoner enn ovenfor at resten av gruppen skal bli delaktig i produktet og nå læringsmålene. Det kan være vanskelig uten tett oppfølging å avdekke problemet tidlig slik at en kan ta stilling til om gruppen skal splittes eller andre tiltak settes i verk. Mange av gruppene uttrykker seg positivt til at strammere oppfølging har gitt en jevnere produksjon enn de har vært vandt med fra før. Flere av gruppene fordeler ansvar ut i fra kompetanse mer enn ut i fra et behov for læring.

For de som er kommet et stykke i studiet er det indikasjoner på at enkelte har en større forpliktelse ovenfor vaner hos den enkelte enn oppdragsgiver og læringsmålene for kurset.

Det er viktig at rollene beskrives klart for alle veiledere og for studentene. Likeså at utforming av arbeidsoppgaver og ansvar hele tiden er klar for alle.

**4. Bruk av verktøy.** Når grupper får krav for det endelige målet, men lite konkret om hva som skal lages til nese gang skjer det lite. Allerede i starten må de kortsiktige mål for den enkelte være konkrete og tydelige for at gruppen skal komme godt i gang. Dette vil hjelpe studentene i gang med å sette seg inn i situasjonen og retningen på prosjektet samt at dette ville ha bidratt til et tydeligere eierskapsforhold. En lite styrende veileder i starten kan gjøre at gruppen blir usikker og opplever at Trac og Subversion kan bli for mye i tillegg til prosjektoppgaveproblematikken. Det uttrykkes tro på at dette kan håndteres bedre med å være betydelig styrende slik at studentene tidligere tar situasjonen. Med styrt menes her fordeling av oppgaver i fellesskap. En uklar presisjon på trac-oppgaver kan åpne for at studentene drar prosjektet i en retning som ikke helt er ønsket ut fra læringsmålene for kurset. Flere grupper uttrykker seg positivt til at de tvinges til å levere noe til neste møte og at det er dokumentert ved hjelp av verktøyet. Det kan være en konflikt mellom studentene og veileder i hvorledes verktøyet skal brukes og da særlig mengden og presisjonen på det som skrives i verktøyet. Det kan være en ide og la en gruppe bestå av subgrupper på to og to og så tildele trac-oppgaver til den enkelte subgruppe. Det uttrykkes enighet om at verktøy som Trac og Subversion kanskje bør innføres i tilknytning til prosjekter som ikke har for mange kompliserende element og læringsmessig

mange nye element som eksempel på det siste kan være innføring i programmering.

**5. Parprogrammering.** Det å tildele trac-oppgaver til to personer kan bidra til at studentene bruker parprogrammering i større grad. Det å innføre parprogrammering tidlig i første semester kan bidra til at studentene får en positiv opplevelse av det og i større grad brukere det senere i studiet.

**6. Refleksjon.** Dersom de kortsiktige krav (neste uke) er uklare blir ofte produksjonen av mindre mengde og kvalitet enn om kravene har tydeligere og klarere mål med hensyn på arbeidsoppgave, mengde, tidsforbruk og personlig ansvar. Det er viktig at studentene får en viss kontroll over hvordan ting er ellers kan de miste eierskapet over prosessen. En måte å gjøre det på: At man i møtet laget et felles delmål, hvorpå studentene umiddelbart satte seg ned og delte opp i konkrete arbeidsoppgaver (trac-oppgaver) og tildelte ansvarlig person(er). For å tvinge de til parprogrammering kan den enkelte oppgave tildeles til to navn. Vi hevder at det er en klar indikasjon på å komme i gang tidlig i studiet med Trac og Subversion da studentene vil ha nytte av det senere i studiet. XP som metode støtter å utvikle eksperimentelt, da det å ha hyppige møter gir god mulighet og kontroll på rytme og tempo i prosjektet. Verktøyene er også et godt hjelpemiddel i denne prosessen. Det ble påpekt at det å lage prototyper bare for å sjekke ut teknologien, utseende eller lignende var vanskelig for enkelte grupper. Det ble påpekt at det kan være en konflikt mellom XPs eksperimentelle metode med prøv og kast som lett blir til at det beholdes og en mer helhets løsning med gode konstruksjoner i produktet. Det vil lette prosessen i de svake gruppene å ha klart for seg hva nedre grense for å stå er. Du har en student som gir veldig mye og som vil at det skal bli så bra som mulig. Han orker liksom ikke å involvere de andre. Han lar hensyn til at produktet skal bli bra gå foran hensynet til de andre i gruppa. Er det riktig å tvinge en sann gruppe til å fungere, eller ville det vært bedre hvis man oppdaget det på et tidligere tidspunkt å splitte den opp?

**7. Samarbeid og kommunikasjon i gruppen.** Det er viktig å være tydelig og klar på arbeidsoppgavene til den enkelte til neste møte. Samtidig er det viktig at gruppen har et eierskap til prosjektet, dette betyr at de må føle at de er involvert i arbeidet med å spesifisere arbeidsoppgavene. Det var stor variasjon på gruppene når det gjaldt aktivitet på møtene. Vi opplevde at veileder ble lurt til å tro at alle var aktiv mens virkeligheten var at det var en som laget kode. En gruppe hadde en som vil at det skal bli så bra som mulig og produserte på egen hånd og neglisjerte produksjonen til de andre. Veiledningen bør klare å handtere interessekonflikt mellom studenter og veiler med oppdragsgiverrollen. Enkelte grupper jobber mye i felleskap, mens andre kommuniserer hovedsakelig vi nettet. Trac kan bidra til en tydeligere og mer transparent kommunikasjon i

intern i gruppen og med veileder, men det fordrer at trac-oppgaver utformes til å beskrive konkrete arbeidsoppgaver og at det følges opp på neste møte.

**8. Oppdeling av oppgaver.** Det er fra veiledernes syn viktig at det deles i klare og distinkte oppgaver som knyttes til den enkelte deltager. Dette skal så følges opp på neste møte. Det er viktig at deltakerne er involvert i arbeidet for å få et eierskapsforhold til prosjektet. Det er et spørsmål om veileder skal lage et delmål først og så lager studentene arbeidsoppgaver ut i fra denne eller skal alt lages i fellesskap. Vi opplevde at i en gruppe skjedde det at en ikke delte på oppgavene, men derimot ville ha så godt som mulig resultat at han neglisjerte de andre i gruppen. En gruppe ville lage et absolutt minimum og formet trac-oppgave slik at det ikke ble store utfordringen. En gruppe hadde en som sto for den virkelige produksjonen mens de andre lot som de deltok. Dette lurte veileder til å tro at gruppens deltagere var flinke og arbeidsomme. Det er et spørsmål om en skal bruke risikostyring eller noe som virker, som retningslinje for styringsprosessen i prosjekter, eller skal begge brukes i samme prosjekt og med en tydeliggjøring ovenfor studentene av valget mellom de to.

**9. Deling og arbeid med å bygge kunnskap.** De fleste deler oppgave ut i fra kompetanse mer enn ut i fra å nå læringsmål. Noen grupper prøver å gjøre mest mulig i fellesskap ut i fra behov for læring.

**10. Krav/opplegg i kurset og kodenstandard.** Kvaliteten på kildekoden var meget variert, noen hadde kode som var strukturert og god å lese. Studentene dannet seg tidlig en oppfatning av opplegget, mens det senere ble klart at det i flere av gruppene var divergerende oppfattelse av kursets mål og opplegg relativt veiledere og kursledelse sin oppfattelse. Det kan være et problem med XP som metode at vi kan gå i ring i utviklingen. Hvorledes skal dette håndteres i et undervisningsopplegg som skal lede til en karakterevaluering i et kurs? Det kan være vanskelig å få gruppene til å jobbe kollektivt kontra individualister som i etterkant setter sammen biter. Flere av gruppene har vanskelig for å ta i bruk eksperimentelle metoder som de ikke er vant med å bruke, eksempel på dette er risikostyring. XP metoden og verktøyet gjør det greit å la grupper samarbeide og dele et felles prosjekt. Dette ville gjort det enklere og mindre sårbart ved frafall i gruppene, en kunne da enklere slått sammen grupper.

**11. Tekst om skolestart intro.** Trac og Subversion bør ikke innføres i det første programmeringskurset. Men heller i et kurs som ikke har så mange nye temaer. Men vi anså det som viktig at studentene ble kjent med den type arbeidsmetoder som støttes av slike verktøy.

**12. Veiledning.** Mange grupper baserer prosjekt oppgaver på lite arbeid i starten og mye arbeid på mot slutten. Det er derfor viktig at veileder er klar og tydelig i sin styring av prosjektet for å få fremdrift. Dette bidrar til at flere når

læringsmålene i kurset. Et tema er at studentene kan argumentere for at de er villedet til en dårligere karakter, dette kan påvirke veiledningssituasjonen og lede til konflikt mellom veileder og gruppa. Det er viktig at alle i gruppa får et eierskap til prosjekt og derfor må de involveres i prosjektet, vi må være oppmerksom på at dette kan være i konflikt med oppdragsgiver rollen. Det er viktig at veiledningen følger rutinene med å legge konkrete planer til neste møte og sjekke ut hva som er produsert siden forrige møte.

**13. Emne plassering i studiet.** Det er riktig å plassere emnet så tidlig i studiet idet det kan hjelpe til med struktur og fremdrift i studiearbeid og før de får lagt til seg vaner som kan være vanskelig å endre. Parprogrammering som arbeidsteknikk bør de ha tidligere i studiet idet det kan være vanskelig å innføre det i 2. og 3. år.

Kommentar: For enkelte grupper kan det være vanskelig å endre arbeidsvaner som gir studenten stor frihet til å skyve på arbeidet noe de opplever har fungert godt tidligere i studiet. De er derfor lite villig til å legge om til metoder som støtter mer struktur og oppfølging av produksjonen i gruppen.

# Diskusjon

## Oppdragsgiverrollen

Før oppstart av prosjektet hadde vi flere diskusjoner rundt hvordan vi skulle definere oppdragsgiverrollen. En kunde har på den ene siden synspunkter når det gjelder design, funksjonalitet og bruk av ressurser. Det andre ytterpunktet er den faglige veilederen som vektlegger detaljer rundt implementasjon. En kunde har vanligvis ikke kjennskap til slike detaljer. To av oss fungerte som både faglærere og oppdragsgivere i samme kurs. Vi opplevde det som kunstig ikke å kunne svare på faglige spørsmål som kommer opp i møte med studentene. Samtidig ønsket vi at møtene med oppdragsgiver skulle simulere en reell jobbsituasjon. Rollen som faglig kyndig kontraktør ble et slags kompromiss mellom kunderollen og veilederrollen.

Fra intervjuene fra høsten 2007 går det frem at 7 av de 8 gruppene opplevde det som positivt å ha en oppdragsgiver å forholde seg til. Uavhengig av om de hadde en faglærer i kurset å forholde seg til eller en ekstern oppdragsgiver, uttrykker de en forståelse for ulike roller. På spørsmål om de synes det er kunstig ikke å diskutere programmeringsproblemer med oppdragsgiveren, svarer en av gruppene med ekstern oppdragsgiver:

*"Synes egentlig det hadde vært litt kunstig hvis vi skulle ha gjort det, i hvert fall hvis det var et sånt forhold til arbeidslivet, så behøver jo ikke kunden å ha noen kunnskap om kildekode, så det er en ganske naturlig situasjon, egentlig. Å ha andre å henvende seg til når det er spørsmål om kildekode og programmering...For det kunden er interessert i er åssen programmet kan brukes og åssen det ser ut. Hva som skal implementeres, ikke hvordan det gjøres.."*

En annen gruppe som hadde kursets faglærer som oppdragsgiver, har følgende svar på hvordan de oppfatter oppdragsgivers rolle:

*"Bare det at man definerer han som en oppdragsgiver framfor en veileder, synes jeg, hva skal jeg si, da synes jeg at du får et helt annet syn på den personen. At det er, oppdragsgiver synes jeg på en måte er, til syvende og sist den som skal motta, altså den som har gitt oppdraget er den som skal motta det endelige resultatet. At den kanskje, en sånn person vil kanskje få litt mere å si på hvordan det endelige resultatet skal se ut enn om det bare er en vanlig veileder."*

*"Det var veldig nytt, og jeg synes det var en litt morsom måte å gjøre det på. Det blir litt mer sånn, blir rettet mot, kan du si hvordan man kanskje vil*

*gjøre noe tilsvarende i yrkeslivet. Så det synes jeg var absolutt veldig positivt, at man tør å gjøre noe sånt."*

I tillegg å gjøre opplevelsen av prosjektet mer reell, legger studentene vekt på at det er positivt med flere personer å forholde seg til og at oppdragsgiver og faglærer fokuserer på forskjellige sider ved oppgaven de skal løse. Vi tolker svarene til studentene dit hen at det er positivt med ekstern oppdragsgiver forutsatt at det er mulig å få hjelp med faglige problemstillinger på annen måte.

Kun en av gruppene vi intervjuet gir uttrykk for at de ville foretrukket en veileder fremfor en oppdragsgiver. Denne gruppa opplevde det slik at møtene med oppdragsgiver hadde feil fokus, og at det ble brukt for mye tid på formulering av oppgaver og for lite tid på det de oppfattet som de egentlige problemstillingene.

*"Altså, istedenfor bare å ha en oppdragsgiver, et møte med oppdragsgiver, tror jeg det hadde vært nyttig for gruppen å få ha kanskje annenhver uke et møte med en veileder som på en måte hjelper til med kode. For vi er ikke i en jobbsituasjon nå. Vi er jo i en læringsituasjon."*

*"...Jeg mente vel bare at du har større utbytte av å ha en koder som så på koden min og kanskje ga tips og mulige forbedringer enn den der teaterarbeidslederen."*

De følte også at de var blitt dyttet i feil retning i møtene med oppdragsgiver, og at det hadde vært vanskelig å komme i gang med prosjektet i forhold til hva de hadde vært igjennom på forelesningene.

I motsetning til flere av de andre gruppene var denne gruppa passiv i den forstand at de ikke benyttet seg av muligheten til å oppsøke faglærer for å få hjelp. Vi kan derfor si at de i alle fall delvis selv må ta ansvaret for at de ikke får den faglige veiledningen de etterspør. På en annen side så peker de på et reelt problem når de sier det avvik mellom progresjonen i kurset og retningen oppdragsgiver trekker prosjektet i.

### **Rytme , tempo og struktur.**

En av de mest merkbare endringene, er rytmen og tempoet som metoden tilfører prosjektarbeidet. I løpet av en periode har studentene hatt møte med oppdragsgiver hvor de i fellesskap blir enige om målsetningene for neste periode, egne møter hvor de diskuterer målsetningene og bryter dem ned mindre arbeidsoppgaver som deretter fordeles, og en påfølgende arbeidsperiode. Totalen av alle disse aktivitetene er rytme og tempo.



Flere av gruppene fremhever det at de "må" produsere noe før neste møte med oppdragsgiver, og peker på at de har unngått forskyvning av arbeidet frem til slutten av undervisningsperioden.

**Intervjuer:** *Ok, dette med måten å organisere prosjektene fra vår side, ikke sant, vi bruker trac og har jevnlig møter med oppdragsgiver, i alle fall foreslår en arbeidsform for deres del, altså legger opp til at dere skal jobbe i par og så videre, at man definerer en struktur for prosjektet. Hvordan har det fungert for dere? Har det vært til noe hjelp, eller på en måte, jobber dere egentlig som dere alltid har gjort likevel?*

*"Ja, jeg synes det funker veldig mye bedre nå enn hvis vi for eksempel skulle hatt individuelle prosjekter."*

*"Spesielt det ukentlige møte, egentlig."*

*"Fordi at da får man en veldig god struktur, at man da må gjøre noe til neste gang."*

*"Med prosjekter så er det ofte at det er mest å gjøre på slutten og det blir det jo forså vidt med oss også i det her. Nå har vi jobba ganske jevnt med det her."*

*"Med såpass godt fundament så burde det ikke være så stort problem å få det ferdig. Tror jeg, hehe."*

**Intervjuer:** *Så det vil si at dere opplever selv at dere har jobba mer nå enn dere ville gjort hvis Børre bare delt ut prosjektet og sagt at kom og spør hvis det er noe?*

*"Ja, veldig mye mer."*

Noen fremhever også hjelpen de har fått med å koordinere arbeidet i møtene med oppdragsgiver, og peker på at bruken av verktøy bidrar til å tydeliggjøre strukturen i prosjektarbeidet.

*"jeg synes det har vært veldig greit å ha de delmålene, de milepælene og de ticketene under dem milepælene å forholde seg til fra gang til gang da."*

*"ja for det gir jo en veldig bra pekepinn på hva vi burde gjøre liksom. Ellers er det jo litt vanskelig, hva skal vi gjøre for noe nå liksom."*

*"det er veldig kjekt å gjøre det sammen med veilederen da også."*

Seks av de åtte gruppene vi intervjuet høsten 2007 gir klart uttrykk for at rytmen har resultert i jevnere innsats. En av gruppene gir uttrykk for at de forholder seg nærmest likegyldig til arbeidsformen, og mener den for deres del ikke har vært nødvendig og at de på egen hånd ville greid å organisere arbeidet og levere det som trengtes. Den siste gruppa etterlyser tydeligere struktur og strengere krav til hva som skal gjøres fra gang til gang. Gruppa har arbeidet sammen i flere kurs, og har fordelt arbeidet seg imellom på kurs istedenfor innenfor kursene. Noen av gruppemedlemmene kom som en følge av dette aldri inn i prosjektet. Gruppa opplevde etter hvert dette som et stort problem, og ble hengende etter med prosjektarbeidet. Den er i så måte et godt eksempel på en gruppe der det kunne vært hensiktsmessig å gå inn i oppdragsgiverrollen som en "arbeidsgiver". I utgangspunktet lot vi studentene selv fordele arbeidsoppgavene innad i gruppa. I etterkant ser vi at det i forhold til utviklingen i enkelte grupper kunne vært hensiktsmessig å definere rollen som en arbeidsgiver med ansvar for fordeling av arbeidsoppgaver og oppfølging av gruppemedlemmene. Arbeidsgiver er blant annet ansvarlig for den enkeltes kompetanseutvikling, og må videre sørge for at oppgaven løses på en kostnadseffektiv måte. Metoden bidrar til innsyn i arbeidsdelingen som vi vanligvis ikke har. Brukt på en bevisst måte kan enkeltstudenter hindres fra å falle ut på et tidlig tidspunkt i prosjektet ved å bli gitt konkrete arbeidsoppgaver som konkretiserer hva de må gjøre for å komme faglig a jour.

## Referanser

1. Astels, Miller and Novak (2002): *A Practical Guide to eXtreme Programming*. Prentice Hall. Isbn:0-13-067482-6
2. Brown, A.L. (1992) *Design experiments: Theoretical and methodical challenges in creating complex interventions*. Journal of the learning Sciences, 2(2), 141-178.
3. Collins, A.; Joseph, D. & Bielaczyc, K. (2004). *Design research: Theoretical and methodical issues*. Journal of the learning Sciences, 13(1), 15-42
4. Edgewall Software (2003-2009) *Trac. Integrated SCM & Project Management*. <http://trac.edgewall.org/> [03.12.09]
5. Greeno, J.G. (1998). *The situativity of knowing, learning and research*. American Psychologist, 53(1), 5-16.
6. Jeffries, R. (2001). *What is Extreme Programming?* XProgramming.com. <http://www.xprogramming.com/> [03.12.09]
7. Kvale, S. (1996) *Interviews: An introduction to qualitative research interviewing*. Thousand Oaks, Calif.:Sage publications.
8. Niemeyer, P. (2003). *Extreme programming with Ant. Building and Deploying Java Applications JSP, EJB, XSLT, XDoclet and JUnit*. Sams Publishing. Isbn:0-672-32562-4
9. Sharma, P. (2004). *An Introduction to Extreme Programming*. <http://my.advisor.com/doc/13571> [03.12.09]
10. Steinberg and Palmer (2004) *Extreme Programming. A Hands-On Approach*. Pearson Education/Prentice Hall. Isbn:0-13-047381-2
11. Tigris.org (2001-2008) *Subversion*. <http://subversion.tigris.org/> [03.12.09]
12. Weinberg, G. (1971) *The Psychology of Computer Programming*. New York: Van Nostrand. (Chs. 17,25)
13. Wells, D. (1999-2009) *Extreme Programming: A gentle introduction*. <http://www.extremeprogramming.org/> [03.12.09]