

Ways of Thinking in Informatics
Søknad om status som merittert underviser

Tor-Morten Grønli

October 30, 2019

Innholdsfortegnelse

1 Innledning	3
2 Undervisning og fagutvikling	3
2.1 Undervisningshistorie	3
2.2 Fagutvikling	6
3 Min pedagogikk	9
3.1 Undervisningsfilosofi	9
3.2 Studentveiledning	11
3.3 Evaluering og vurdering	12
3.4 Eksempel på studentevaluering	13
4 Forskning og formidling	15
4.1 Forskning på egen praksis	15
4.2 Roboten Pepper	16
4.3 Formidling	18
5 Ledelse	18
5.1 Utdanningsledelse	18
5.2 Forskningsledelse	19
5.3 Administrativ ledelse	20
6 Internasjonalisering	21
7 Avslutning	22
8 Oversikt over vedlegg	23

1 Innledning

Jeg vil med dette søke status som merittert underviser ved Høyskolen Kristiania, og forankring fra instituttleder finnes i vedlegg 14. Dette dokumentet og tilhørende vedlegg redegjør for min praksis og filosofi. Jeg verdsetter ikke bare verdien i kunnskap generert fra informatikk alene, men ser at kunnskap generert i samspillet mellom kollegaer, bransje og studenter, og forankret i tverrfaglighet og anvendbarhet, er det som skaper størst verdi. Nøkkelen til suksess i samfunnet vårt og for å fremme vårt fagfelt er å bygge kunnskap innen tverrfaglige disipliner, trekke på ulike forskningsmetoder og i gjennom det kunne fjerne grenser. Jeg vil videre utdype hvordan jeg mener prinsippene for tverrfaglighet, samarbeid, medvirkning og aktivisering kan brukes til å generere ny kunnskap og utdanne de neste generasjonene innen fagfeltet. Jeg vil i gjennom dokumentet fremheve hvordan jeg har lært, jobber og utvikler meg som pedagog, foreleser, veileder og fagperson i spennet mellom *kunnskap - næringsliv - og studenter*. Det hele starter med min undervisningshistorie.

2 Undervisning og fagutvikling

Jeg vil i denne delen beskrive min undervisningshistorie, mitt arbeid med undervisningsmateriale, og nyutvikling, revisjon og vedlikehold av emner og studieprogrammer.

2.1 Undervisningshistorie

De tidlige årene 2004 til 2008

Min undervisningshistorie begynner i 2004 som uteksaminert kandidat fra Bachelor i Informasjonsteknologi fra Norges Informasjonsteknologiske høyskole (NITH). Jeg gikk fra statusen som student over til en stilling som først stipendiat og deretter til høyskolelektor (2007) ved samme institusjon. Perioden 2004 til 2007 var for institusjonen preget av rask utvikling og høy innovasjonstakt hva gjaldt studieprogrammer og emne-portefølje, som en reaksjon på Dotcom boblen som sprakk. Jeg fikk en unik mulighet som nyutdannet teknolog til å sette preg på prioriteringene, utforske mulighetsrommet ny teknologi gav og å være en aktiv bidragsyter i diskusjonene rundt hvilken retning emnene på institusjonen skulle ta. Perioden var preget av at jeg utforsket og demonstrerte hvordan ny teknologi kunne integreres som en del av undervisningsopplegget. Eksempelvis var nyutvikling av to emner knyttet til

programvareutvikling innen for Microsoft .Net teknologi (PG320 og PG420 i 2006/07) med på å bringe til nå ukjente teknologiområder til studieprogrammene i systemutvikling ved NITH. I samme periode, 2004 til 2007, tok jeg initiativ til og fikk videreutvikle studentassistent / studentveileder-rollen ved NITH til et systematisert og organisert veilederkorps ledet av en Hovedveileder. Denne funksjonen / rollen har siden da fortsatt å eksistere i vår organisasjon og går fortsatt i 2019 under navnet *Hovedveileder*. Det er en av våre viktigste tiltak for å støtte opp under undervisningsopplegget, gi tillit og ansvar til dyktige studenter og er et helt sentralt element i hvordan studentene veiledes til å nå læringsutbyttene sine.

Når jeg tenker tilbake på de tidlige årene i min undervisningskarriere ser jeg de er preget av stort pågangsmot omkring teknologi, ved å kontinuerlig være oppdatert på hva siste og nyeste trender som bransjen benyttet til programvareutvikling var og bringe dette rett inn i klasserommet. Et viktig prinsipp da var hvordan man kunne kombinere demonstrasjoner i klasserommet, såkalt "live-koding", med teoretisk forankring og forklaringer av underliggende konsepter. Nøkkelen til dette er å utnytte nysgjerrighet, vitebegjær og pågangsmot kombinert med kunnskapsdeling. NITH gav tilgang til å aktivt pleie et bransjenettverk og for meg betydde dette at jeg kunne bringe næringslivet inn som formidlere og ha de med i undervisningsopplegg. Ved læring er det viktig for meg å bidra med erfaringsdeling tilbake, og et eksempel på dette var da jeg i 2008 holdt foredrag for industrien / bransjen om nettopp kunnskapsdeling på bransjekonferansen *Javazone* under tittelen *Effektiv kunnskapsdeling - kanskje den viktigste nøkkelen til suksess*.

Mellomårene 2008 til 2014

I 2008 startet jeg på min doktorgrad ved Brunel University, England. Samtidig med dette, ble jeg fast ansatt som høyskolelektor ved NITH og fikk tidlig i min karriere mulighet til å ha studiespesialiseringsansvar ved å lede Bachelor Informasjonsteknologi med spesialisering i Programmering fra 2009. Jeg benyttet også muligheten i perioden 2007 til 2010 til å være konsulent i næringslivet for systemutviklingsbedriften Mesan. Dermed kunne jeg ikke bare formidle til studentene egen undervisning, men bringe praksisen fra bransje til klasserom, og tilbake til bransjen. Rett påfølgende, etter å kunne vise til gode resultater som spesialiseringsansvarlig, fikk jeg tillit som studieprogramleder for hele Bachelor Informasjonsteknologi fra 2011 (med 6-8 underliggende spesialiseringer). Jeg hadde i denne perioden ansvar for flere aktiviteter slik som opprettelse av studieprogramlederforum for å koordinere, kvalitetssikre og strategisk forvalte porteføljen av Bachelor programmer. Jeg

var også aktivt ute i media med blant annet innlegg på bransjekonferanser (Javazone 2008 og Smidig 2011) samt innlegg i dags- og fagpresse f.eks Computerworld i 2011 (Vedlegg 7). I denne perioden hadde jeg stor undervisningsbelastning og var aktivt involvert i emneutvikling innen en rekke studieprogrammer, og nyutviklet /moderniserte emner slik som databaser, distribuert programmering, mobil programmering, systemutvikling, prosjekt software engineering. NITH hadde på denne tiden også flere campuser og jeg fikk i min rolle utfordret meg på koordinering av undervisning, formidling av intensjoner og filosofi knyttet til emner, summativ vurdering, studieprogramvedlikehold, bransjekontakt og forskningsforankring på tvers av geografiske regioner. I gjennom kontakten med bransjen så jeg også behov for endring i flere av emnene vi foreleste og brakte dette inn som en del av revisjonsarbeidet. Et eksempel på dette er et tverrfaglig emne i 2. studieår på Bachelor informasjonsteknologi, Prosjekt Software Engineering, som har til formål å forene all kunnskapen de får i sitt andre studieår på tvers av 6-8 spesialiseringer i tverrfaglige team. Dette emnet hadde i en årrekke blitt forelest ved hjelp av tradisjonelle systemutviklingsmetoder. I perioden etter 2005 stod bransjen ovenfor en endring og dreining til smidige metoder og jeg brakte dette inn i undervisningen ved å endre/modernisere kursopplegg, ta bransjesertifisering innen rammeverket (SCRUM) og lede opplæringsworkshopper for egen fagstab. Denne perioden modnet og utviklet meg som akademiker og la på mange måter grunnlaget for prinsippene i min senere akademiske karriere.

Nyere tid 2014 -

Symptomatisk for perioden fra 2014 og fremover er at etter Master Applied Computer Science og senere Master Human-Computer Interaction ble godkjent og startet som studieprogrammer (se seksjon 2.2 Fagutvikling) dreide perioden fra 2014 og frem til i dag over til høyere grads undervisning og studieprogramutvikling. Etter modningen og utviklingen i mellomårene, opplever jeg at jeg i denne perioden har en annen selvtillit og evnen til å prøve ut, evaluere og reflektere over undervisningsopplegg øker. I tillegg til at mastergradsnivå fører til et økt krav om faglig dybde, forskningsforankring og utvikling av selvstendige, reflekterende og kritisk evaluerende studenter finner endringer i samfunnet også veien til undervisningen. Smarttelefoner, Youtube og bevisste studenter fører med seg et økt press på edutainment, varierte undervisningsformer og høy kvalitet i undervisningen. Jeg tror at grunnlaget lagt i de tidlige årene tilknyttet uredd eksperimentering med ny teknologi, erfaringen i innarbeidelse av aktive undervisningsformer tilegnet

i mellomårene og modenheten igjennom professoropprykk, internasjonal undervisningserfaring og undervisning på alle nivåer fra bachelor til doktorgrad har gjort at undervisningen i nyere tid stadig blir bedre. Bevisstheten det har tilført gjør at pedagogiske modeller og prinsipper som velges er ikke lenger et resultat av kopiering av andres praksis, men heller basert på kunnskap om fagområdet og innsikt i, og kritisk blikk på, egen praksis. Av høydepunkter fra nyere tid vil jeg trekke frem internasjonal undervisningserfaring (detaljert i seksjon 6), tyngden ved tverrinstitusjonell erfaring fra undervisningen min i deltidsstilling ved OsloMet (Vedlegg 4) og oppdragene på erfaringsmasteren på Institutt for Informatikk ved Universitet i Oslo. I tillegg har jeg via OsloMet fått muligheten til å erfaringsoverføre til kollegaer via deres mentorprogram for opprykkskandidater til toppstillinger.

Slik som jeg har forsøkt å belyse kjennetegnes min historie av en ikke helt A4 vei igjennom bachelor, master, phd / høyskolelektor, førsteamanuensis og professor løp, men heller en vei som utfordret grenser, preget av tidlig og stort ansvar og et utviklingsløp i takt med en organisasjon under utvikling og fornyelse som nettopp gav et unik mulighetsrom for utvikling og ansvar langt utenfor hva som typisk kjennetegner de tradisjonelle akademiske stillingsnivåene. Artig nok tok det også mange år før jeg til slutt formaliserte min pedagogiske kompetanse ved å gjennomføre kurs i universitets og høyskolepedagogikk i 2018 (Vedlegg 3). Gjennomsnittlig undervisningserfaring for deltakerne på kurset var riktignok 11 år, og dette forteller meg at modenheten som skal til for å forstå hvordan pedagogikk, undervisning og studentlæring går hånd i hånd - det kan sees som ulike speilbilder av det samme, som det tar tid å utvikle.

I vedlegg 1 er komplett CV med oversikt over mine pedagogisk relaterte stillinger og oppgaver, og i vedlegg 2 er det oversikt over emner jeg har utviklet / forelest og engasjement innen veiledning. Spesifikt det som gjelder internasjonal erfaring knyttet til undervisningsledelse og undervisningsoppdrag er omtalt i seksjon 6

2.2 Fagutvikling

Min undervisningsfilosofi er drevet av troen på at studentene trenger anvendte ferdigheter med en dyp teoretisk forståelse av *whys og hows*. Jeg tar sikte på å utdanne dyktige utøvere og reflekterte håndverkere for bransjen, og nysgjerrige, kritisk tenkende, reflekterte teoretikere for academia. Jeg har erfaring med undervisning på PhD-nivå, masternivå og bachelornivå. Dette

utgjør mer enn 400 studiepoeng undervist og utviklet på bachelor innen ulike deler av informatikk, slik som: grunnleggende og avansert programmering, mobil programmering for Google Android og Apple iOS, databaser, systemutvikling for web, prosjektledelse for programvareutvikling, smidige metoder, bachelorprosjekt, for å nevne noen områder, fullstendig liste finnes i vedlegg 2. Videre har jeg undervist og utviklet mer enn 200 studiepoeng på masternivå i sentrale områder innen informatikk slik som systemutvikling, arkitektur, mobil systemutvikling, visualisering, smidige prosjektledelse og muliggjørende teknologier for å nevne noen. Jeg har på masternivå også erfaring med undervisning ved flere ulike institusjoner nasjonalt og internasjonalt. Jeg har hatt eneansvar for full kurs ved Høyskolen Kristiania (og dets infusjonerte enheter som Westerdals Oslo ACT og NITH), Universitetet i Oslo (Norge), International Hellenic University (Hellas), Brunel University (England) og Copenhagen Business School (Danmark). På doktorgradsnivå har jeg utviklet, forelest og evaluert kurs i systemutvikling for web og arkitektur ved Addis Ababa University (Etiopia).

Utvikling og vedlikehold av studieprogrammer

På bachelornivå har jeg utviklet, og vært studieprogramleder for, følgende spesialiseringer innen bachelor informasjonsteknologi: *Programmering* (180 studiepoeng), *Frontend- og mobilutvikling* (180 studiepoeng) og *Intelligente systemer* (180 studiepoeng), alle ved Høyskolen Kristiania (sin infusjonerte skole NITH). Utvikling og vedlikehold av bachelorprogram er en svært givende aktivitet ettersom den gir muligheten til å kombinere forskning, med kunnskap om sentrale emner innen informatikk, samtidig som man i dialog med næringslivet kan lage den beste studieprogramprofilen for å utdanne fremtidens IT kandidater. I 2018 og 2019 har jeg vært initiativtaker til, og ledet prosessen med å lage, høyskolen Kristiania sitt første engelskspråklige bachelorprogram - Bachelor in Data Science (Vedlegg 11). I denne prosessen har jeg kunnet kombinere mye av min tidligere kunnskap om og erfaring fra studieprogramutvikling og spilt på internasjonalt samarbeid med våre kontakter i staben på Copenhagen Business School for å kunne arbeide frem en ny, unik, grad som dekker et fagområde tradisjonelt sett på som å være på mastergradsnivå. Arbeidet med å lage denne graden er også et godt eksempel på hvordan jeg oppfatter at vellykket studieprogramutvikling kan foregå. Her ble suksesskriteriene å kombinere lagarbeid og innsikt fra egen fagstab ved instituttet, diskusjon og forankring med studentrepresentanter, innspill og diskusjon med internasjonale samarbeidspartnere og høringsrunder med næringslivet for å vite at kandidatenes profil er avstemt med behovene.

Videre på masternivå deltok jeg i arbeidsgruppen som utviklet masterstudiet i informasjonssystemer (120 studiepoeng) ved NITH. Min viktigste rolle var å sikre en teknologisk forankring av et studieprogram for informasjonssystemer og hjelpe til med å identifisere hvilke emner innen informasjonssystemer som ville gi riktig forankring og speile trender og samfunnsutvikling det var viktig å ta hensyn til i studieprogrammet. Etter denne gjennomføringen, ledet jeg etterpå arbeidsgruppen som utviklet et masterprogram i Applied Computer Science (120 studiepoeng). I denne runden hadde jeg ansvaret for søknadsprosessen inkludert planlegging, prosjektledelse, utvikling av studieprogrammet, internasjonale relasjoner og ikke minst sikre bred involvering av studenter og ansatte ved instituttet. Fra arbeidet med dette masterstudiet fikk jeg verdifull kunnskap ved å ta et masterprogram fra ide til konkret realisering. Denne prosessen innebar et betydelig samarbeid, og var avhengig av utviklingen av gjensidig forståelse og kunnskapsdeling med alle parter i arbeidsgruppen, fra student og faglig ansatt til internasjonale samarbeidspartnere og bransjekontakter.

Utvikling av læringsmateriell

For alle emner som jeg har undervist, har det vært naturlig å utvikle komplett kursmateriell i tillegg. Fra de tidlige årene i min undervisning har dette tradisjonelt handlet om å utvikle lysbilledsett, øvinger, tilhørende tekniske demonstrasjoner og tilgjengelig gjøre supplerende ressurser for studentene. Etter hvert som erfaringen har vokst og man har blitt mer rutinert på forelesning, har forelesninger og kursmateriell blitt en arena for å eksperimentere og utforske alternativer slik at variasjonen blir større. Den første opplevelsen som ga meg innsikt i utnyttelsen av undervisningsmateriell til å utfordre, variere og styrke egen praksis var kravet fra UHR om at all undervisning på bachelornivå skal være forskningsforankret. For meg gjorde dette det mulig å basere og inkludere egen forskning i undervisningsmaterialet, og i gjennom dette engasjere studentene, gjøre de nysgjerrige og sammen med de kunne vise hvordan forskningsprosjekter kan være en naturlig del av forelesningsmaterialet. Spesielt i emnene knyttet til temaer i hurtig endring, slik som mobil teknologi, har det gitt veldig god respons fra studentene å ikke bare kunne gjengi etablert praksis, men ta studentene med som deltakere i å utvikle ny kunnskap. Et eksempel på dette er at eksamensoppgaver blir gitt på pågående forskningsprosjekter, slik som deloppgave to ved Android eksamen våren 2019 som skulle knyttes til pågående forskning om IoT (Vedlegg 12).

På kursene som er undervist på masternivå, er det for alle et adskil-

lig tyngre innslag av å være forskning. Fokuset på studentmedvirkning her inkluderer blant annet muligheten for studentene til å gjøre egne prosjekter om til forskningsoppgaver, og jeg kan være med å tilrettelegge for at de kan bidra til ny kunnskapsutvikling ved å publisere i nasjonale og internasjonale tidsskrifter eller konferanser. Eksempler på at jeg har hjulpet studenter til å foredle sine innlevering fra kurs og masteroppgaver til internasjonale publikasjoner er blant annet arbeider fra Lorentzen om smarte klær [1], Falsen Hiis om sensorer i treningsutstyr [2], Lundar om evaluering av web-arkitekturer [3] og Mikkelsen om nyteknisk rundt etablerte arkitekturpraksiser [4]. Felles for disse er at de både ga studentene og meg innsikt i ny kunnskap, og målet om potensiell publisering motiverte disse dyktige studentene til å yte ekstra, og det ga de muligheten til å ta resultatene ut tilbake til egen praksis i næringslivet; slik som tilfellet var for Lundar og Mikkelsen.

3 Min pedagogikk

Jeg vil først introdusere og reflektere rundt min undervisningsfilosofi og så videre til evalueringsfilosofien og samvirke med studentene.

3.1 Undervisningsfilosofi

Mitt syn på undervisning er preget av informatikk som akademisk fagområde. Vi utdanner praktikere som skal ha med seg konkrete ferdigheter og kunnskaper ut i bransjen, og sågar omtales kandidatene tidvis som håndverkere. Nettopp derfor er det svært viktig at disse studentene har en sterk teoretisk forankring for å kunne forstå hva de gjør, hvorfor de gjør det og ta velbegrunnede valg i arbeidssituasjoner. Det betyr videre at undervisningen disse studentene møter skal være en kombinasjon av praktiske ferdigheter og teorikunnskap. I takt med at verktøy, programvare og teknologier oppdateres må studentene se forbi teknologien og bli bevisst på hvordan teknologi i seg selv ikke er målet, men heller et virkemiddel på vei mot et ferdig produkt. Jeg følger aktivt Biggs Constructive Alignment teori [5, 6] om sammenhengen mellom alle aktivitetene, og forsøker å se helheten som en prosess fra læringsutbyttebeskrivelsene, til valgte undervisnings- og veiledningsformer, igjennom formativt vurdering og til summativ vurdering i eksamen.

Informatikk som fagområde er relativt ungt sammenlignet med andre fagområder. Det er et område preget av høy innovasjonstakt og hurtig endring i hva som er korrekt arbeidsmåte, verktøyportefølje eller teknologiske rammeverk. Det er dermed et viktig prinsipp å utdanne til nysgjerrighet.

Jeg har en viktig oppgave i å ikke bare holde meg selv faglig ajour, for derigjennom å levere oppdatert og relevant undervisning, men også å utdanne reflekterte og nysgjerrige profesjonsmennesker. I gjennom min undervisning må jeg legge til rette for at de får de riktige referanserammene og evnene til selv å ta ansvar for læringen sin videre. Slik som Erstad [7] beskriver må de tilegne seg ferdigheter i navigering, analyse og kunnskapsbygging for å bli komplette digitale samfunnsborgere. Det er så rik tilgang på et stort utvalg av digitale medier som video, blogg, tekst, programvare, bilder etc. at det ikke lenger holder å kunne anvende det, men studentene må vite hva de skal velge å bygge sin læring på. Nettopp derfor er det akademiske fundamentet jeg gir igjennom studiene et viktig grunnlag for at de videre kan bygge sin faglige karriere og profesjonsutvikling.

For å kunne oppnå målsettingene om hva som kjennetegner den reflekterte, nysgjerrige og faglig kompetente avgangsstudenten må varierte undervisningsformer benyttes. Alle studenter er forskjellige, og de lærer på forskjellige måter. Det betyr at de igjennom et undervisningsløp må møte en rekke læringsformer som gir de muligheten til å lære på egne premisser. Auditorieundervisning er fra noen hold en presset undervisningsform, men som Skodvin [8] poengterer kan dette fortsatt være en meget velegnet undervisningsform gitt riktige rammer, planlegging og aktivisering. Jeg slutter meg til denne betraktningen og slår personlig et slag for auditoriebasert undervisning. Forelesning som form bruker jeg aktivt for å gi studentene oversikt over et fagområde og sette temaer i perspektiv. Ved å supplere ren formidling med aktive læringsformer [9] kan forelesningen bli en dynamisk arena for kunnskapsutvikling. Jeg bruker aktivt dialog, spørsmål, diskusjoner og oppgaveløsning som studentaktiviserende elementer slik at læring kan oppnås både igjennom erfaring, refleksjon og situasjonsbestemt selvkompetanse og relasjonskompetanse. I gjennom mye undervisning av programmeringsfag har jeg aktivt jobbet med å ha live programmering som et element også i auditoriebaserte forelesninger. Dette går ut på at man foran, eller sammen med, studentene skriver applikasjonskode underveis som en del av forelesningen [10]. Jeg synes dette er en krevende og givende undervisningsform som gjør at man underveis igjennom samspillet med studentene kan tilrettelegge (justere) vanskelighetsnivå, demonstrere, forklare og engasjere.

Det er viktig for meg at undervisning blir variert slik at alle studenter kan lære. Ulike personer lærer på ulike måter, og det er mitt ansvar å variere hvilke metoder og teknikker de møter i takt med temaene som formidles. Derfor mener jeg man også må gå videre, og lengre enn bare å aktivisere studentene, men faktisk gi de ansvar for å lære. Ved å tørre å stille krav til studentene kan man oppnå at de i større grad blir bevisst på

hvordan de lærer, og hva de lærer. Aktiv bruk av læringsutbytte sammen med vurderte vurderingsformer [11] er for meg viktige elementer for å sikre at studentene kan bli selvstendige, kritiske og reflekterende. Dette oppleves fort som krevende for en del studenter, særlig de svake og deler av gruppen med middels flinke. Derfor bruker jeg aktivt dialog med studentene i timene, samtaler med tillitsvalgte og fagevalueringer for kontinuerlig forbedre og tilpasse opplegget på en slik måte at flest mulig lærer mest mulig.

Oppsummert er min undervisningsfilosofi og undervisningsstil preget av troen på at studentene trenger gode praktiske ferdigheter de kan anvende direkte, sammen med en dyp teoretisk forståelse av hva, hvorfor og hvordan. Min målsetting er å utdanne reflekterte og dyktige håndverkere til IT bransjen, og samtidig legge til rette for at de også er aktuelle som nysgjerrige, kritisk tenkende, teoretisk forankrede praktikere til en akademisk karriere. Et læringssyn forankret i helheten fra læringsutbyttebeskrivelser til gjennomføring og summativ vurdering [6].

3.2 Studentveiledning

Jeg har siden 2005 veiledet bachelorstudenter i gradens avsluttende år, og totalt veiledet mer enn 80 bacheloroppgavegrupper (hver gruppe fra 2-5 studenter) til vellykket gjennomføring. Jeg har veiledet totalt 30+ masterstudenter til vellykket gjennomføring i avhandlingen siden 2008. For både bachelor- og masterveiledning har prosjektene vært relatert til anvendt informatikk. Veiledning er en givende aktivitet og gir muligheten til å engasjere studenter i forskningsrelaterte aktiviteter og sammen arbeide med å nå prosjektmålet. Det er for meg en viktig aktivitet som kan gi studentene interessante perspektiver, fremme god refleksjon og gi dem praktiske ferdigheter.

Høyskolen Kristiania har ikke et eget doktorgradsprogram som gjør veiledning av egne doktorgradskandidater umulig, men jeg har derimot i gjennom mitt akademiske nettverk vært engasjert i doktorgradsveiledning, og for to av studentene er Høyskolen Kristiania arbeidsgiver. Jeg er hovedveileder for K. Hansen som arbeider seg frem til en doktorgrad fra Copenhagen Business School, Danmark, innen helseinformatikk. Dette muliggjør et tverrfaglig internasjonalt samarbeid med Center for Business Data Analytics i København, Danmark. Videre er jeg biveileder for A. Biørn-Hansen som studerer til en doktorgrad ved Brunel University, England, innen området mobil kryssplattformutvikling. Ph.d.-veiledning legger til rette for å bygge miljø, overføring av kunnskap og prosjektsamarbeider og for meg er dette sentrert i min Mobile Technology Lab.

Forskning er alltid en samarbeidsinnsats, og veiledning på PhD-nivå har

brakt nye internasjonale samarbeidspartnere til lab'n så vel som nye muligheter å engasjere bachelor og masterstudenter tilknyttet forskningsprosjektene ved deres grader. Jeg har også tidligere vært biveileder for G. Mesfin ved Addis Ababa university, Etiopia, frem til hans forsvar i 2016. I gjennom den prosessen kunne også jeg utvikle meg og mine veiledningsferdigheter i samhandling med hovedveilederen fra England, og lokal koordinator ved universitet i Addis Ababa. Dette gav verdifull innsikt i hva det vil si å veilede også på tvers av geografiske og kulturelle grenser, f.eks hvordan akademiske titler og rang-ordninger er mer dominante i noen kulturer enn hva vi normalt er vant til i Skandinavia.

3.3 Evaluering og vurdering

Her er det naturlig å se litt nærmere på min anvendelse av undervisvurdering, formativ vurdering, og sluttvurderinger, summativ vurdering - selv om disse glir litt over i hverandre, blant annet fordi det er vanskelig å gi en formativ tilbakemelding uten først å ha gjort en oppsummering av status. Spørsmålet blir derfor ofte kontekst, vurderer/evaluerer man for å bringe en prosess videre, for å forme en utvikling, eller primært for å oppsummere et stykke arbeid. En effekt av dette, er at man kan benytte de samme verktøyene både summativt og formativt, men dersom de benyttes formativt kommer det et element av veiledning, feedback, refleksjon og lignende inn i tillegg til den informasjon vurderingen/evalueringen gir alene. Summativ vurdering (vurdering av studentene sin avsluttende prestasjon) handler for meg mye om varierte vurderingsformer. I min rolle som studieprogramleder er jeg opptatt av å forankre at man igjennom et studieprogram blir møtt med og blir prøvet på en rekke måter, slik at alle får en sjanse til å prestere. I min rolle som emneansvarlig og foreleser er det viktig at sluttvurdering gjøres på en hensiktsmessig måte som gjør at studentene får vist sitt nivå i henhold til læringsutbyttet.

Deretter kommer skillet mellom vurdering og evaluering, der vurdering, og som beskrevet i avsnittet over, er vurdering av faglige prestasjoner av fagkompetanse, mens vi med evaluering gjerne tenker på vurdering av i hvilken grad undervisning medfører læring. For å belyse dette vil jeg trekke frem et eksempel fra egen praksis: Uavhengig om klassene man foreleser er store eller små, er formativ vurdering, veiledning og tilbakemelding, noe som tar mye tid, men samtidig er det en sentral del av læringsutbyttet. For eksempel der studentene leverer en mappe, med en prosess med innlevering, tilbakemelding og revidert innlevering for så (forhåpentligvis) ende i godkjenning er anvendt teknikk her. Jeg ser klare paralleller til bruken av Blooms

taksonomi som gjengitt av Prøitz [11], i forhold til at dette gir studentene en mulighet for å vise at de kan bevege seg opp fra nivået med å gjengi kunnskap til å kunne reflektere, analysere og vurdere. Det er viktig å legge til rette for at studentene får sin tilbakemelding så fort som mulig, slik at de kan ta lærdom tett opp mot utføringen av arbeidet. For å sørge for god struktur og enhetlig tilbakemelding forankres tilbakemeldingen i sjekkliste jeg har utarbeidet slik at det f.eks alltid pekes på noe positivt først. Deretter kan man trekke frem forbedringspunkter, men ikke kritisere. Denne fremgangsmåten har også forankring i Nicol and Macfarlane-Dick [12] sitt femte prinsipp om at positiv forsterkning og positiv tilbakemelding er med på å styrke studentens mulighet til å lykkes og til å selv regulere sitt arbeid, dvs. øke / holde / minke intensitet.

Min filosofi og praksis knyttet til evaluering av eget arbeid er nært knyttet opp til min hovedfilosofi for undervisning, forskning og veiledning - den skal være åpen, lyttende og forskningsforankret. Jeg synes det er svært lærerikt og givende å få tilbakemeldinger fra studentene, og passer på å etablere god kontakt med klassens tillitsvalgt enten om det er i mitt virke som foreleser, emneansvarlig eller studieprogramleder. Evaluering har som formål å øke kvaliteten på gjennomføringen av undervisningen, materialet og formidlingen. I tillegg skal det være med på å være justerende for hvordan studentene best kan oppnå sitt læringsutbytte. I alle emner jeg har forelest, nasjonalt og internasjonalt, har det alltid vært foretatt underveis evaluering i form av samtale med studentene individuelt eller i gruppe. Tilbakemeldingene herfra er justerende for andre halvdel av kurset. Sluttevalueringer er typisk avholdt via spørreskjemaer som er anonyme, og tilbakemeldingene herfra er justerende for neste gjennomføring av emnet, eller andre relaterte emner (Eksempel på internasjonal sluttevaluering i vedlegg 6). Det er viktig for meg å lukke informasjonsflyten tilbake til studentene, så jeg åpner alltid første time i et nytt kurs med å avklare deres og mine forventninger, samt gå igjennom hva hovedtrekkene fra forrige sluttevaluering var. For å belyse hvordan jeg jobber med tilbakemeldinger, har jeg valgt å trekke frem to aspekter i dette avsnittet. Første en episode fra studentevaluering og deretter en fra kollegaveiledning.

3.4 Eksempel på studentevaluering

Denne situasjonen er hentet fra undervisning i 4. semester for Bachelorstudenter i informasjonsteknologi. I emnet Android programmering, er det ca. 60 studenter som deltar og de kommer fra tre forskjellige tekniske spesialiseringer innen sin bachelorgrad. Alle studentene har erfaring med program-

mering fra tidligere semestre i studiet, men dette er første gang hvor de møter et mobilt system som utviklingsplattform. Undervisningen her gjennomføres i et auditorium og det er en kombinasjon av demonstrasjon, øving, diskusjon og "live koding". Konsepter knyttet til programmering av mobile enheter kan tidvis være komplisert da det krever en annen tilnærming og en annet tankegang enn ordinær programmering de har bedrevet til nå.

Etter at jeg hadde gjennomført ca. 1/3 av emnet, ble det gjort en oppsummeringsrunde med klassens tillitsvalgt. Jeg fikk tilbakemelding om at studentene syntes temaene var vanskelige, at tempoet gikk for fort og at det var dermed vanskelig å holde motivasjonen oppe. For å imøtekomme dette planla jeg neste undervisnings-økt til å ha utstrakt fokus på studentmedvirkning og aktivisering. Jeg åpnet timen med en quiz over temaene i første tredjedel av semesteret, og igjennom dette fikk jeg et tydeligere innblikk i hva de syntes var mest utfordrende. Deretter gikk jeg over til delen med nytt stoff i undervisningen, men jeg valgte ikke mitt tradisjonelle mønster. I stedet for å starte med å forelese, gikk vi i gang med å kode sammen et prosjekt fra grunnen av. Her la jeg vekt på ekstra gjennomgang av prinsippene som var knyttet til det studentene hadde svart var vanskelig. Kodingen ble ikke gjennomført av meg alene, men jeg fikk studentene til å fortelle hvert kommende steg. Etter dette gikk over til å ha en del med klassisk forelesnings, før vi igjen gikk tilbake og gjorde en runde til av koding sammen etterfulgt av forelesning.

Jeg synes dette var spesielt vellykket fordi denne gjennomføringen adresserte utfordringer studentene tydelig hadde. Den innledende delen med en selvtest gav et godt bilde av situasjonen studentene stod i den dagen. Jeg ser at det er interessante perspektiver som bringes opp av Rohrer og Pashler [13] i forbindelse med bruk av tester som læringsstrategier. Videre gjennomføringer med emnet kan lett anvende tester i ulike deler for å påvirke ulike deler av studentene sin læring i alt fra selvbevissthet på kunnskapsnivå, til vurdering av læring fra en økt til teknikk for innarbeidelse av nytt fagstoff. Det ble igjennom å snu om på opplegget mulig å ta direkte inn i undervisningen tilbakemelding fra studentene. Den oppsummerende delen av dette hvor jeg foreleste, etter å ha kodet, fungerte dermed som en retrospektiv teknikk i forhold til hva vi hadde gjort. I tråd med Skodvin [8] ble denne delen av forelesningen en økt som bidro til overblikk, orientering og informasjon om mulighetene. I etterkant ser jeg at det som ble gjort følger prinsipper for aktiv læring som gitt av Freeman mfl. [9], slik som kombinasjon av diskusjon, problemløsning, workshop og involvering. Det er interessant å følge deres diskusjon rundt STEM fagene og studentengasjement [14], og likeledes se hvordan denne typen emner kan ha behov for andre og varierte undervis-

ningsformer. Fra egen praksis stemmer dette overens med erfaringene fra å utdanne praktikere samtidig som de skal få et teoretisk fundament og bli utdannet til å ta bevisste, begrunnende valg.

Dette episoden har ikke ført til at jeg har lagt om undervisningsopp-
leggene mine til å følge denne modellen som ble brukt i dette tilfellet her, men
det har i etterkant ført til at jeg har hatt et aktivt fokus på at det i alle timer
må være en aktivitet som engasjerer og involverer studentene. Jeg tror vari-
erte vurderings- og undervisningsformer bare er to av elementene for å støtte
opp under studenter sin læring og at "vellykket undervisning" er en personlig
oppfattelse fra foreleser, og ikke nødvendigvis et korrekt mål på om studen-
tene hadde "vellykket læring".

4 Forskning og formidling

Forskning på egen praksis er et sentralt element for å forstå hvorfor noe virker, mens andre fremgangsmåter ikke virker. Ved å basere seg på et faktabasert syn kan man være klar over når elementene man studerer faktisk også er den begrensende faktoren for videre utvikling av undervisningen [15].

4.1 Forskning på egen praksis

Jeg har igjennom årene formalisert forskning på egen praksis, og trekker her frem tre hovedbidrag:

- Prosjektbasert læring versus problembasert læring i gjennom "stilasbygging" (Vedlegg 8)
- Innovasjon i studiemodeller for levering av Bachelor Informasjonsteknologi - Industribachelor (Vedlegg 9)
- Forståelse av hvordan egen undervisningspraksis påvirker studentenes læring - et studie av "live koding" i klasserommet (Vedlegg 10)

I det første eksemplet om prosjektbasert læring versus problembasert læring i gjennom "stilasbygging" (scaffolding) ble det studert et tverrfaglig prosjekt i 2. klasse på bachelor informasjonsteknologi. Etter noen år med å ha fulgt en strategi om å gi studentene utfordringer og "kaste de på dypt vann", ble fokuset med innføringen av denne metoden at man i mye større grad skulle tilrettelegge, veilede og gi maler eller eksempler. Gradvis etter som studentene ble kjent med problemområdet og prosjektoppgave begynte

vi gradvis å fjerne støttene og åpne opp for mer frihet. Fra publikasjonens utgivelse og i årene som fulgte ble denne metodikken adoptert i stor grad av fagstab, spesielt i prosjektemner. I kontrast til tidligere opplevde vi at studentene sin grad av suksess ble mer målrettet og det var svært interessant å studere deres egen refleksjon (totalt over 5 år fra ca 1400 deltakende studenter) over læringsprosessen som en del av emnet og datagrunnlaget for artikkelen.

I det andre eksemplet var fokus på innovasjon i studiemodeller. Artikkelen tar for seg prosessen frem til og våre erfaringer med å kjøre en såkalt *Industribachelor* som vil si at studentene bruker fire år i stedet for tre på sin bachelorgrad, men samtidig opparbeider seg et og et halv års arbeidserfaring ute i næringslivet som en del av graden sin. Jeg utarbeidet denne gjennomføringsmodellen i 2009/2010 etter dialog med næringslivet og hovedformålet med denne artikkelen var å sette innovasjon av studietilbudet i en pedagogisk kontekst. Ved å gjøre dette og presentere funnene på den årlige norske informatikkonferansen oppnådde vi å gjøre erfaringsdeling med vårt akademiske fagmiljø. I gjennom erfaringene som ble delt, og diskusjonene som fulgte, håper vi å kunne inspirere flere til å gjøre lignende ordninger som legger til rette for å kombinere teori og praksis, tenke nytt i forhold til oppnåelse av læringsutbytte og koble bransjen tettere på studentene i gjennom studieløpet.

Det tredje eksemplet jeg valgte å trekke frem her handler om et prosjekt vi gjennomførte for å forstå implikasjonene rundt egen undervisning. Innen for programmeringsemner er det vanlig praksis å kode live i klasserommet som en del av undervisningsopplegget. Vi fant ikke noen tidligere forskning som hadde sett på hvilket effekt dette hadde på eller hvordan *studentene faktisk opplevde* denne formen for formidling. Det var mye tidligere forskning på foreleser sin erfaring og rolle, men vi snudde rundt på spørsmålet. Resultatene baserer seg på intervjuer gjort av studenter som jeg foreleste i sitt første studieår og mer viderekommende studenter i sitt fjerde semester. Innsikten som kom ut av artikkelen gjaldt både fremgangsmåte, tempo og tematikk og har i etterkant hatt direkte effekt på hvordan undervisningen i programmering har blitt formidlet til studentene.

4.2 Roboten Pepper

Høsten 2018 søkte jeg og S. Fagernes fra institutt for teknologi om og fikk tildelt NOK 150 000 (se vedlegg 5) i interne pedagogiske midler til innkjøp og utvikling av pedagogisk opplegg til en humanoid robot, *Pepper*. Vi ville

i prosjektet bevisstgjøre studentene på utfordringene og mulighetene som ligger i interaksjon med og oppbygningen av intelligens i roboter slik som Pepper fra Softbank Robotics. Vi vil igjennom utvikling av pedagogisk materiale og undervisningsopplegg utdanne våre studenter til å stå helt i front hva gjelder teknologisk kompetanse, forståelse for bruk og interaksjon med, samt grunnleggende kunnskaper om oppbygning av intelligens i, den nære fremtids kanskje viktigste innovasjon: roboter.

Etter anskaffelse av roboten Pepper har vi arbeidet tverrfaglig mellom Master Human-Computer Interaction og Master Applied Computer Science, hvor man forsker på denne typen interaksjonsformer, denne typen teknologi, og ikke minst de utfordringene som er knyttet til menneske-maskin kommunikasjon og selve teknologien.

Etter å ha blitt bedre kjent med Pepper og dens muligheter (og begrensninger) har vi integrert den i undervisningen og laget pedagogisk opplegg rundt roboten. Pepper ble våren 2019 benyttet som eksamensoppgave i masteremnet *Multimodal Interaction* hvor studentene igjennom aktive læringsformer, workshop og lab ble eksponert for, opplært i og tilegnet seg konkret erfaring med roboten. Igjennom undervisnings- og eksamensopplegget fikk studentene selvstendig ansvaret for Pepper og implementerte under veiledning to eksamensoppgaver om hvordan Pepper kunne adressere problemstillingen ensomhet blant eldre. En gruppe implementerte den som en fysisk trener for eldre mennesker, og en annen gruppe implementerte en løsning hvor man spilte brettspill med roboten.

I begge tilfeller førte Pepper til en unik innsikt for studentene i å se hvordan moderne og kommende teknologiske løsninger kan benyttes i praksis. De fikk førstehåndserfaring i de tekniske og interaksjonsmessige utfordringene ved at de måtte implementere løsninger og ikke bare lese om det i litteratur. Arbeidsøktene så ut til å være svært intense og ikke minst krevende og vi vurderer at studentene hadde stort læringsutbytte av dette. Vi opplever også at studentene syntes det var veldig spennende å få hands-on erfaring med denne typen teknologi.

I etterkant av emnet har studentenes eksamensoppgave blitt til en forskningsartikkel til sporet for *Multimodal Interaction and User Experience ved International Conference on Human-Computer Interaction*. I tillegg har en av studentene fra dette emnet valgt å jobbe videre med problematikk innen kommunikasjon med roboter, basert på erfaringene fra undervisningen.

Eksemplet med Pepper viser hvordan nyteknisk rundt teknologi, undervisningsopplegg og studentaktive læringsformer kan skape innovasjon. I tillegg til å være en suksess i klasserommet har Pepper prosjektet også ført til oppslag i avisen *Natt og Dag*, et besøk hos *Kultur- og likestillingsminister*

Trine Skei Grande og prosjektet har blitt valgt ut som eksempel i utdanningstrategien til Høyskolen Kristiania.

4.3 Formidling

Jeg har gjennom årene av forskningen min formidlet forskningsresultater offentlig gjennom forskjellige kanaler. Jeg har skrevet kronikker (Eksempel i vedlegg 13) og bidratt til artikler i aviser om hvordan teknologi og forskning kan bygge bro mellom industri og akademia, og forbedre og / eller løse organisatoriske utfordringer og brukernes behov. Jeg har blitt brukt som ekspert i intervju om mobilapplikasjoner / smarttelefoner i flere aviser og videre har jeg ved forskjellige anledninger uttalt meg om aktuelle problemstillinger knyttet til studieprogrammene og hvordan / hvorfor studiene uteksaminerer viktige kandidater for bransjen (Eksempelvis vedlegg 7). Jeg har også deltatt i TV-intervjuer relatert til mobilapplikasjoner og jeg har vært jurymedlem i den norske finalen i den verdensomspennende konkurransen AppCircus. Jeg har holdt innlegg på bransjekonferanser slik som for systemutviklere, Java-Zone, og den største norske Agile prosjektlederkonferansen, Smidig. Disse formidlingsaktivitetene har trent og økt ferdighetene mine i å kommunisere mitt felt til en ikke-teknisk gruppe og hjulpet meg med å fokusere på å forklare den viktige rollen teknologien spiller i dagens samfunn.

5 Ledelse

5.1 Utdanningsledelse

Jeg har siden 2009 kontinuerlig ledet en eller flere bachelor- og / eller master fordypninger innen IT og har gjennom dette fått lang erfaring med å lede studieprogrammer både på bachelor- og masternivå. For begge nivåer har dette ansvaret inkludert utvikling og vedlikehold av emneporteføljen, administrativt ansvar for studiet, ansvaret for kvaliteten på undervisningen og tilrettelegging for at studentene kan oppnå læringsutbyttet som er satt. Dette er et givende arbeid og bidratt til å gi meg kunnskap om, og ferdigheter i, personalutvikling og motivasjon i de forskjellige studieprogrammene og innsikt i hvordan fagstab, studenter og bransje sammen bygger kvalitet. Gjennom et stort antall diskusjoner med bransjerepresentanter og vitenskapelig ansatte, har jeg kontinuerlig utviklet emneporteføljene og også identifisert utfordringer det har vært behov for å løse knyttet til undervisningsopplegg eller temaer, eller knyttet til pedagogisk praksis og formidling. Videre har dette ansvaret også ført til dialog og samarbeid med regional

bransje for å forankre prosjekter og diskutere kursinnhold med det formålet å knytte studentene tettere på og sikre relevans. De daglige utfordringene i studieprogramledelse inkluderer å utfordre det teoretiske grunnlaget, bygge en sterkere forskningsforankring og utfordre studentenes evne til å reflektere over hvorfor og hvordan en gitt teknikk, metode, rammeverk eller tilnærming er best. Videre på master- og phd-nivå er det i enda større grad evnen til å koble forskning og undervisning som utfordres, sammen med det å kunne skape gode muligheter for studenter til å delta i forskningsrelaterte aktiviteter som en del av sin studiehverdag.

Fra 2011 til 2016 og fra 2018 til pågående har jeg hatt ansvaret for studieprogramleder-forumet på tvers av alle spesialiseringene innen bachelor informasjonsteknologi. Dette ansvaret omfattet oppfølging av den ansvarlige for hver spesialisering, og sikring av at alle spesialiseringer ble vedlikeholdt og kjørt etter gjeldende nasjonale kvalitetskriterier. I praksis kjøres dette som månedlige møter hvor vi følger opp kvalitet, løser utfordringer i fellesskap og trekker på hverandres kunnskap og tidligere erfaringer i samarbeid. Med en sammensatt gruppe av sterke individer skaper dette et godt miljø for faglige diskusjoner, og givende utfordringer i å kunne oppnå felles enighet samtidig som det er rom for forskjellige meninger. En viktig del av rollen har vært å utvikle målemetoder og prosedyrer for å følge opp underveis- og sluttevalueringer av kurs, studentprogresjon, og tilhørende målinger som knytter seg nært til høyskolen sin strategi og målstyring av de enkelte studieprogrammene. Rollen ga meg også muligheten til å være involvert som medlem av ledergruppen ved høyskolen. Dette åpnet nye arenaer for min personlige utvikling, og gjorde meg i stand til å jobbe med strategisk porteføljeforvaltning og ga verdifull innsikt i budsjettering, utvikling av studieprogram, personalledelse, rekruttering av studenter og å arbeide det overordnede, langsiktige, strategiske perspektivet for høyskolen.

5.2 Forskningsledelse

Jeg har gjennom årene vært i stand til å få bred erfaring innen forskningsledelse inkludert forskningsgrupper, lab, søknader, veiledning og ledelse av forskjellige arrangementer. I 2011 initierte og grunnla jeg forskningsgruppen for informasjonssystemer sammen med professor Bygstad. Gjennom dette var jeg med på å initiere en strukturert tilnærming til forskning i organisasjonen og var i gjennom innlegg på semestersamlinger en pådriver for at forskning og undervisning kunne bli sett på som en tett integrert virksomhet og ikke adskilte disipliner. Etter hvert som høyskolen vokste i størrelse skapte det en åpning for to forskningsgrupper og i 2014 planla og

opprettet jeg den andre forskningsgruppen, for anvendt informatikk. Med forankring fra ledelsen bygget og ledet jeg gruppen fra 2014 til 2019. Mitt hovedansvar er å legge til rette for forskningsprosjekter, organisere månedlige forskningsmøter og promotere anvendt informatikk for de ansatte, studentene og næringslivet. Suksessen til en forskningsgruppe kommer fra de individuelle forskernes bidrag og engasjement. Læring fra den første forskningsgruppen var viktig for meg å skape et miljø der nye så vel som erfarne forskere kunne møtes for å dele ideer, diskutere prosjekter og engasjere til tverrfaglighet. Gjennom utviklingen av denne forskningsgruppen sikret jeg meg midler fra fakultetet for to doktorgradsstudenter, og i 2016 hadde gruppen vokst til 16 medlemmer, inkludert ph.d.-kandidater, heltidsansatte og tilknyttede deltidsansatte.

Erfaringene fra disse to gruppene gjorde at jeg i 2015 grunnla *Mobile Technology Lab (MoTeL)*. Jeg sikret en innledende finansiering på en million norske kroner for å bygge en fysisk infrastruktur og møteplass med tilhørende teknologi og utstyr. Arbeidet inkluderte design av et forelesnings- / lab, et utstillingsrom for prosjekter og et eye-tracking lab. Planlegging, design, møblering og etablering av lab'n ga verdifull innsikt nye områder og det var viktig for meg at dette ikke bare skulle være en forskningslab, men et møtepunkt det var naturlig å integrere i undervisning på bachelor og master, så vel som bli en arena for studentprosjekter. Etter et års drift og deretter flytting til ny campus i 2016, ble den fysiske lab'n omdisponert til å bli en ressurs for hele instituttet. Mobile Technology Lab som konsept brakte jeg videre som en "virtuell" lab slik at den fortsatt er en overordnet organisasjonen for forskning- og undervisningsprosjekter, doktorgradsstudenter og masterstudenter. I de siste 3 årene har vi årlig tilbudt prosjekter og hatt studenter som aktive drivkrefter for organiseringen. I 2019 opprettet vi det første tverr-institusjonelle prosjektet som innebefatter to institutter, bachelor og master studenter fra begge instituttene og eksternt samarbeid med forskningsmiljøet på Simula, hvor vi gjør arbeid knyttet til måling og evaluering av effekten teknologi har på livsstilsendringer.

5.3 Administrativ ledelse

Fra 2016 til 2017 vikarierte jeg som dekan ved fakultet for teknologi ved Westerdals Oslo ACT, med ansvar for budsjett, rekruttering og daglig drift av et fakultet med en stab på 31 heltids- og deltidsansatte. Gjennom å ha ansvaret for hele fakultetet fikk jeg kjenne på nye utfordringer slik som utvikling og daglig ledelse, prioritering av prosjekter, tildeling av midler og planlegging av aktiviteter. Samtidig med å være utfordrende var det også givende og arena

for personlig utvikling. Dette var også en arena som ga innsikt i nye former for kommunikasjon med tillitsvalgt-apparatet blant studentene, innsikt i å kunne identifisere og løse utfordringer knyttet til studenttilsyn og undervisningsaktiviteter. Denne stillingen ga et nytt perspektiv og innsikt i nettopp hvor teamorientert god undervisning og godt miljø er. Dette fikk meg til å reflektere over at det gir et bedre faglig samarbeid på alle nivåer når det handler om å få mest mulig ut av hver enkelt student og ansatt, og at vi kun gjennom samarbeid kan oppnå varige resultater. Sammen kan vi skyve grenser i undervisning, forskning og personlig utvikling for å komme over hindringer, fjerne barrierer og skape ny kunnskap.

6 Internasjonalisering

Internasjonale nettverk er viktig erfaringsgrunnlag også for å bygge profil og forstå menneskelig læring og utvikling [16]. Jeg har bygget og undervist PhD kurs innen programvarearkitektur på Addis Abeba University, Ethiopia og jeg har forelest i flere runder for greske studenter om webteknologier ved International Hellenic University (Vedlegg 6). Jeg har også i tre runder forelest ved handelshøyskolen i København om apputvikling for mobil for økonomistudenter. Denne varierte erfaringen har vært en svært solid plattform å bygge videre på i møte med internasjonale studenter i egne kull i Norge. I tillegg har erfaringen gitt stor innsikt i hvordan ulike kulturer har ulike tradisjoner for undervisning og for hva som er god pedagogikk. Alt fra overholdelse av oppmøtetider til engasjement i undervisning og hvorvidt det er normalt å diskutere med foreleser eller ei er tematikk som berøres. Alt i alt vil jeg si at dette har vært en stor styrke for min utvikling, og også en test på om egen pedagogisk modell holder vann utenfor Norges grenser.

Jeg har utviklet et aktivt forhold til mange kolleger ved både nasjonale og internasjonale institusjoner. Jeg publiserte regelmessig artikler med forskere fra Universitetet i Agder (Norge), Oslo Akershus høgskole (Norge), Copenhagen Business School (Danmark), University of Copenhagen (Danmark), Brunel University (UK), Addis Ababa University (Ethiopia), Oxford Brookes University (UK), Greenwich University (UK) og Yeditepe University (Turkey). Med alle disse samarbeidspartnerne har jeg en aktiv kontakt, og vi driver med årlige aktiviteter som spenner fra forskning til eksamener og gjesteforelesninger. Gjennom mitt Mobile Technology Lab (MoTeL) er jeg offisielt anerkjent samarbeidspartner med Center for Business Data Analytics (cbsBDA) ved Copenhagen Business School, Danmark. Det strategiske

samarbeidet med CBS inkluderer både gjensidig lab-prosjekter så vel som undervisningsaktiviteter ved begge institusjonene. Siste eksempel på dette er utviklingen av Bachelor i Data Science hvor vi gjensidig har sparret om innhold og studieprogramprofil. Sammen med University of Leuven, Fakultet for Odisee, Brussel, Belgia tok jeg initiativ til og ledet samarbeidet i de to første rundene av PingFin-workshopen for andreårs bachelorstudenter, med 4 europeiske universiteter som deltakere. Dette var en fire dager lang workshop som hadde som mål å generere økonomisk og teknologisk forståelse, samt å eksponere studentene for samarbeid på tvers av land og kulturer.

7 Avslutning

Veien videre begynner der undervisningskarrieren startet - med en lyttende nysgjerrighet. Pedagogisk arbeid har i disse årene lært meg at det er alltid noe som kan bli bedre, og at de som står meg nærmest i form av kollegaer og studenter er de som har størst innsikt i hvordan jeg kan bli bedre. Vi bedriver på samme måte som studentene livslang læring og vi må ivareta det privilegiet vi har ved å arbeide i kunnskapsindustrien. Takk studenter - vi sees i morgen :)

8 Oversikt over vedlegg

1. CV
2. Emneoversikt
3. Kursbevis og vitnemål fra UH per kurs
4. Uttalelse fra OsloMet
5. Roboten Pepper
6. International Hellenic University engasjement og studentevaluering
7. Formidlingsartikkel fra Computerworld
8. Artikkel om Scaffolding av studentprosjekter
9. Artikkel om Industribachelor
10. Artikkel om livecoding
11. Studieprogrambeskrivelse Bachelor Data Science
12. Android eksamensoppgave
13. Kronikker fra Morgenbladet
14. Anbefaling fra instituttleder

References

- [1] K. Lorentzen, T.-M. Grønli, G. Ghinea, M. Younas, and M. Satpathy, “Sensors in your clothes: Design and development of a prototype,” in *International Conference on Mobile Web and Information Systems*, pp. 302–312, Springer, 2016.
- [2] T. F. Hiis, T.-M. Grønli, and G. Ghinea, “The smartpt: Wearble sensor integration and tracking in training equipment,” in *2016 IEEE International Conference on Computer and Information Technology (CIT)*, pp. 96–99, IEEE, 2016.
- [3] J. A. Lundar, T.-M. Grønli, and G. Ghinea, “Performance evaluation of a modern web architecture,” *International Journal of Information Technology and Web Engineering (IJITWE)*, vol. 8, no. 1, pp. 36–50, 2013.
- [4] A. Mikkelsen, T.-M. Grønli, and R. Kazman, “Immutable infrastructure calls for immutable architecture,” in *Proceedings of the 52nd Hawaii International Conference on System Sciences*, 2019.
- [5] J. Biggs, “Enhancing teaching through constructive alignment,” *Higher education*, vol. 32, no. 3, pp. 347–364, 1996.
- [6] J. Biggs, “Aligning teaching for constructing learning,” *Higher Education Academy*, vol. 1, no. 4, 2003.
- [7] O. Erstad, “Educating the digital generation-exploring media literacy for the 21st century,” *Nordic Journal of Digital Literacy*, vol. 10, no. Jubileumsnummer, pp. 85–102, 2015.
- [8] A. Skodvin, “Mellom kateter og kaos. forelesning i forskjellige varianter. i h. strømsø, kh lycke & p. lauvås (red.),” *Når læring er det viktigste. Undervisning i høyere utdanning*, pp. 125–139, 2006.
- [9] S. Freeman, S. L. Eddy, M. McDonough, M. K. Smith, N. Okoroafor, H. Jordt, and M. P. Wenderoth, “Active learning increases student performance in science, engineering, and mathematics,” *Proceedings of the National Academy of Sciences*, vol. 111, no. 23, pp. 8410–8415, 2014.
- [10] J. Paxton, “Live programming as a lecture technique,” *Journal of Computing Sciences in Colleges*, vol. 18, no. 2, pp. 51–56, 2002.
- [11] T. S. Prøitz, *Læringsutbytte*. Universitetsforl., 2015.

- [12] D. J. Nicol and D. Macfarlane-Dick, “Formative assessment and self-regulated learning: A model and seven principles of good feedback practice,” *Studies in higher education*, vol. 31, no. 2, pp. 199–218, 2006.
- [13] D. Rohrer and H. Pashler, “Recent research on human learning challenges conventional instructional strategies,” *Educational Researcher*, vol. 39, no. 5, pp. 406–412, 2010.
- [14] T. Kennedy and M. Odell, “Engaging students in stem education.,” *Science Education International*, vol. 25, no. 3, pp. 246–258, 2014.
- [15] G. Biesta, “Why what works wont work: Evidence-based practice and the democratic deficit in educational research,” *Educational theory*, vol. 57, no. 1, pp. 1–22, 2007.
- [16] R. Säljö, *Läring i praxis: et sosiokulturelt perspektiv*. Cappelen akademisk, 2001.

Vedlegg 1: CV

Pedagogisk CV – Tor-Morten Grønli

Professor, Ph.D. i Informatikk

Adresse: Institutt for Teknologi, Høyskolen Kristiania,
Prinsens gate 7-9, Oslo, Norway.

Telefon: + 47 48156476

E-post: tor-morten.gronli@kristiania.no

Nåværende hovedansvar

- ❖ Professor ved Institutt for Teknologi, Høyskolen Kristiania
- ❖ Studieprogramleder for Master Applied Computer Science, Institutt for Teknologi, Høyskolen Kristiania
- ❖ Studieprogramleder Bachelor Informasjonsteknologi, Institutt for Teknologi, Høyskolen Kristiania
- ❖ Leder av Mobile Technology Lab, Institutt for Teknologi, Høyskolen Kristiania
- ❖ Visiting Scholar ved Copenhagen Business School, København, Danmark
- ❖ Professor II ved OsloMet, Institutt for Teknologi, Kunst og Design

Utdanning

2012	PhD i informatikk	Brunel University, England
2007	Master of Technology (Distinction)	Brunel University, England
2004	Bachelor i Informasjonsteknologi	Norges IT høyskole

Ansettelser

2017 - forts	Professor	Institutt for Teknologi, Høyskolen Kristiania
2015 - forts	Leder av Mobile Technology Lab	Institutt for Teknologi, Høyskolen Kristiania
2014 - forts	“Visiting Scholar”	Copenhagen Business School, København, Danmark
2014 - forts	Professor II	OsloMet, Institutt for Teknologi, Kunst og Design
2016 - 2017	Dekan (vikar)	Institutt for Teknologi, Westerdals Oslo Act
2010 - 2017	Førstemanuensis	Institutt for Teknologi, Westerdals Oslo Act
2007 - 2010	Høyskolelektor	Norges IT høyskole
2004 - 2010	Høyskolelærer / stipendiat	Norge IT høyskole
2007 - 2010	Systemkonsulent	Mesan (nå NetCompany)

Akademiske ansvarsområder

2011 - 2016 & 2018 - forts	Studieprogramleder Bachelor Informasjonsteknologi	Institutt for Teknologi, Høyskolen Kristiania
2014 - forts	Studieprogramleder for Master Applied Computer Science	Institutt for Teknologi, Høyskolen Kristiania
2017 - forts	Spesialiseringsansvarlig for Bachelor IT Intelligente systemer	Institutt for Teknologi, Høyskolen Kristiania
2014 - 2016	Spesialiseringsansvarlig for Bachelor IT Frontend- og mobilutvikling	Institutt for Teknologi, Westerdals Oslo Act
2012 - 2016	Spesialiseringsansvarlig for Bachelor IT Frontend- og mobilutvikling	Institutt for Teknologi, Westerdals Oslo Act
2009 - 2015	Spesialiseringsansvarlig for Bachelor IT Programmering	Institutt for Teknologi, Westerdals Oslo Act
2015 - forts.	Grunnlegger og leder av Mobile Technology Lab	Institutt for Teknologi, Høyskolen Kristiania
2014 - forts.	Grunnlegger og leder av forskningsgruppe for anvendt informatikk	Institutt for Teknologi, Westerdals Oslo Act
2011	Med-grunnlegger av forskningsgruppe for informasjonssystemer	Norges IT høyskole

Studieprogramutvikling

2018 - 2019	Ansvarlig for utvikling av Bachelor in Data Science	Institutt for Teknologi, Høyskolen Kristiania
2016 / 2017	Deltaker i gruppen for utvikling av Master in Human Computer Interaction	Institutt for Teknologi, Westerdals Oslo Act
2013 / 2014	Ansvarlig for utviklingen av Master in Applied Computer Science	Norges IT høyskole
2017	Ansvarlig for utviklingen av for Bachelor IT spesialisering Intelligente systemer	Institutt for Teknologi, Westerdals Oslo Act
2014	Ansvarlig for utviklingen av for Bachelor IT spesialisering Frontend- og mobilutvikling	Norges IT høyskole
2009 - 2015	Spesialiseringsansvarlig for Bachelor IT Programmering	Institutt for Teknologi, Westerdals Oslo Act
2015 - forts.	Grunnlegger og leder av Mobile Technology Lab	Institutt for Teknologi, Høyskolen Kristiania
2014 - forts.	Grunnlegger og leder av forskningsgruppe for anvendt informatikk	Institutt for Teknologi, Westerdals Oslo Act
2011	Med-grunnlegger av forskningsgruppe for informasjonssystemer	Norges IT høyskole

Internasjonale stilinger

2014 - forts	“Visiting Scholar”	Copenhagen Business School, København, Danmark
2010 - forts	“Research collaborator”	Brunel University, England
2015	Ekstern foreleser	Addis Ababa University, Etiopia

Internasjonal undervisningserfaring

2014 - 2016	Masterkurs i “Introduction to programming, Smartphone app development for E-business”	Copenhagen Business School, København, Danmark
2014 & 2015	Masterkurs i “Web programming”	International Hellenic University, Hellas
2015	PhD level course I Introduction to software engineering and project management	Addis Ababa University, Etiopia
2010	Masterkurs i “Software Design and Architecture”	Brunel University, England

Nasjonal undervisningserfaring

2005 - forts.	Bachelornivå undervisning	Institutt for Teknologi, Høyskolen Kristiania
2010 - forts.	Masternivå undervisning	Institutt for Teknologi, Høyskolen Kristiania
2014 - forts	Bachelornivå undervisning	OsloMet, Institutt for Teknologi, Kunst og Design
2018 & 2019	Masterkurs i erfaringsledelse	Universitet i Oslo

En oversikt over alle kurs som er forelest finnes i vedlegg 2.

Akademiske komiteer

2018	Deltaker i ansettelseskomite for førsteamanuensis	Norges Arktiske Universitet
2018	Deltaker i komité for utarbeidelse av særavtale for fagstab	Høyskolen Kristiania
2017	Leder av ansettelseskomite for førsteamanuensis	NTNU Gjøvik
2016	Leder av ansettelseskomite for førsteamanuensis	Høyskolen i Østfold
2015 - 2016	Leder av komiteen for fusjon av forskningsmiljøene ved Westerdals, NISS og NITH. Utarbeidelse av ny forskningsstrategi	
2013	Deltaker i ansettelseskomite for førsteamanuensis	Høyskolen i Oslo & Akershus
2012	Deltaker i ansettelseskomite for førsteamanuensis	Universitet i Oslo
2010	Ekstern deltaker i vurderingskomite av kvalitet ved masterprogram	Brunel University, England

Sakkyndig komité

2019	Vurdering av søknader til Oslofjorfondet for Forskningsrådet
2017	Vurdering av Noroff Bachelor i "Applied Data Science" (2. innsending)
2016	Vurdering av Noroff Bachelor i "Applied Data Science" (1. Innsending)
2013	Vurdering av Noroff fagskoletilbud for "Utvikling av apps"

Veiledning

2019	Mentor i professoroppyrkk-programmet ved OsloMet for C. Dalland
2005 - forts.	80+ Bacheloroppgaver / grupper innen informatikk
2014 - forts	30+ Masteroppgaver innen informatikk
2015 - forts	PhD veileder for K. Hansen innen temaområdet helse, sosiale medier og informatikk
2016 - forts	PhD veileder for A. Biørn-Hansen innen temaområdet kryssplattformutvikling
2015/2016	Bi-veileder for A. Mesfin fra Addis Ababa University, Etiopia innen hans prosjekt på mobil teknologi

Inviterte foredrag / kurs

2010 - 2012	Kurs i agil prosjektledelse	Tekna
2014	Kurs i agil prosjektledelse	Emergence School of Leadership
2018	Workshop om forskningsbasert undervisning	Høyskolen Kristiania

Verv og styremedlemskap

2015 - forts	Styremedlem i Norsk informatikk konferanse
2015 - forts	Styremedlem i fagrådet for Naturvitenskaplig, matematiske og teknologiske fag (NMT)

Fagfellevurdering i pedagogiske utgivelsessteder

2017 - forts	Fagfelle-vurderer for ACM Special Interest Group of Computer Science Education (SIGSCE)
2016 - forts	Fagfelle-vurderer og styremedlem i Norsk Konferanse for undervisning og didaktikk i IT-fagene (UDIT)
2018 - forts	Medlem i forskningsgruppen for fagdidaktikk ved institutt for teknolog, Høyskolen Kristiania

Formidling

2019	Innlegg på forskningsformidlingsdagene 2019 om offentlig kommunikasjon av Fou resultater
2016	Jurymedlem i den Norske finalen til mobilutviklingskonkurransen AppCircus
2015	Deltagelse i TV2 hjelper deg i forbindelse med test av mobiltelefonapper
2018	3 kronikker i morgenbladet om "Fagspråk i respirator"
2011	Innlegg i bransjebladet ComputerWorld
2011	Innlegg på bransjekonferansen Smidig
2008	Innlegg på bransjekonferansen JavaZone
	Årlige gjesteforedrag hos bedrifter om mobile løsninger og tingenes internett. F.eks System, Netcompany og Accenture.

Innsamlede midler

2019	Søknad om pedagogisk fyrtårnsprosjekt ved høyskolen Kristiania (videre til 2. intervjurunde)	NOK 1 000 000
2018	Event tracking prosjekt med institutt for ledelse og organisasjon	NOK 300 000
2018	Tildelt interne midler for prosjektet Physical Health Electronically Assessed Data	NOK 491 000
2018	Tildelt interne pedagogiske midler for prosjekt Roboten Pepper	NOK 150 000
2018	Tildelt interne midler for vedlikehold og oppgradering av Mobile Technology Lab	NOK 150 000
2016	Tildelt interne midler for oppbygging av Eye-tracking lab	NOK 500 000
2015	Tildelt interne satsingsprosjekt midler for opprettelse av Mobile Technology Lab	NOK 1 000 000
	Deltatt i søknad om EU midler in internasjonale konsortier i 2012, 2015 og 2017.	Ingen tildeling

Publikasjoner

Følgende publikasjoner er resultat av forskning på egen undervisningspraksis (listen er alfabetisk):

B. Bygstad, B. R. Krogstie, and T.-M. Grønli, "Learning from achievement: Scaffolding student projects in software engineering," *International Journal of Networking and Virtual Organisations*, vol. 6, no. 2, pp. 109–122, 2009.

B. Bygstad, B. R. Krogstie, and T.-M. Grønli, "Scaffolding Project Based Learning with the Rational Unified Process. Experience from 5 years of Student Projects in Software Engineering," *Norwegian Conference on ICT*, 2006

R. Gonzalez, W. Presthus, and T.-M. Grønli, "Industry bachelor programme: An experience report," in *Norwegian Conference on ICT*, 2015

T.-M. Grønli, S. Fagernes "The Live Programming Lecturing Technique. A Study of the Student Experience in Introductory and Advanced Programming Courses", *in review for ACM SIG Computer Science Education*, 2019

Følgende publikasjoner er resultat av videre arbeid med student-innleveringer (listen er alfabetisk):

K. Eksen, T. Serif, G. Ghinea, and T.-M. Grønli, "inloc: Location-aware emergency evacuation assistant," in *2016 IEEE International Conference on Computer and Information Technology*, pp.50–56, IEEE, 2016.

T.-M. Grønli, B. Flesch, R. Mukkamala, R. Vatrupu, S. Klavestad, and H. Bergner, "Internet of things big data analytics: The case of noise level measurements at the roskilde music festival," in *2018 IEEE International Conference on Big Data (Big Data)*, pp. 5153–5158, IEEE, 2018.

T. F. Hiis, T.-M. Grønli, and G. Ghinea, "The smartpt: Wearble sensor integration and tracking in training equipment," in *2016 IEEE International Conference on Computer and Information Technology*, pp. 96–99, IEEE, 2016.

S. Ladwa, T.-M. Grønli, and G. Ghinea, "Towards encouraging a healthier lifestyle and increased physical activity—an app incorporating persuasive design principles," in *International Conference on Human-Computer Interaction*, pp. 158–172, Springer, 2018.

M. Lehmann, A. Biørn-Hansen, G. Ghinea, T.-M. Grønli, and M. Younas, "Data analysis as a service: an infrastructure for storing and analyzing the internet of things," in *International Conference on Mobile Web and Information Systems*, pp. 161–169, Springer, 2015.

K. Lorentzen, T.-M. Grønli, G. Ghinea, M. Younas, and M. Satpathy, "Sensors in your clothes: Design and development of a prototype," in *International Conference on Mobile Web and Information Systems*, pp. 302–312, Springer, 2016.

K. Lorentzen, S. Fagernes and T.-M. Grønli, "Adapted Sensor Data Visualization on a Smartphone: The Case of a Fire Emergency Rescue Application " in *Norwegian Conference on ICT*, 2017

J. A. Lundar, T.-M. Grønli, and G. Ghinea, "Performance evaluation of a modern web architecture," *International Journal of Information Technology and Web Engineering*, vol. 8, no. 1, pp. 36–50, 2013.

G. Mesfin, G. Ghinea, T.-M. Grønli, and W.Y. Hwang, "Enhanced agility of e-learning adoption in high schools," *Journal of Educational Technology & Society*, vol. 21, no. 4, pp. 157–170, 2018.

A. Mikkelsen, S. Honningsøy, T.-M. Grønli, and G. Ghinea, "Exploring Microsoft HoloLens for Interactive Visualization of UML Diagrams," in *Proceedings of the 9th International Conference on Management of Digital EcoSystems*, pp. 121–127, ACM, 2017.

A. Mikkelsen, T.-M. Grønli, and R. Kazman, "Immutable Infrastructure Calls for Immutable Architecture," in *Proceedings of the 52nd Hawaii International Conference on System Sciences*, 2019.

Vedlegg 2: Emneoversikt

Oversikt over emner som er forelest

Bachelor:

2019

- PGR202 - Android programmering (7.5 ECTS)

2018

- PGR202 - Android programmering (7.5 ECTS)
- IKT200 - Introduksjon til Intelligente systemer (7.5 ECTS)

2017

- PGR201 - Android programmering (7.5 ECTS)
- IKT200 - Introduksjon til Intelligente systemer (7.5 ECTS)
- PGR100 - Objektorientert programmering 1 (7.5 ECTS)

2016

- PJ6000 – Bachelorprosjekt (15 ECTS)
- PG4600 - Mobil programmering (7.5 ECTS)

2015

- PG4600 - Mobil programmering (7.5 ECTS)
- PJ6000 – Bachelorprosjekt (15 ECTS)

2014

- PG5100 – Enterprise programmering 1 (7.5 ECTS)
- PG6100 – Enterprise programmering 2 (7.5 ECTS)
- PJ3100 – Project Software Engineering (15 ECTS)
- PJ6000 – Bachelorprosjekt (15 ECTS)

2013

- PG560 – Enterprise programmering 1 (10 ECTS)
- PG660 – Enterprise programmering 2 (10 ECTS)
- PJ3100 – Project Software Engineering (15 ECTS)

2012

- PG3200 - Android programmering (5 ECTS)
- PG560 – Enterprise programmering 1 (10 ECTS)
- PG660 – Enterprise programmering 2 (10 ECTS)
- PJ312 – Project Software Engineering (10 ECTS)

2011

- PG560 – Enterprise programmering 1 (10 ECTS)
- PG660 – Enterprise programmering 2 (10 ECTS)
- PJ312 – Project Software Engineering (10 ECTS)

2010

- PJ312 – Project Software Engineering (10 ECTS)
- PG660 – Enterprise programmering 2 (10 ECTS)

2009

- PG560 – Enterprise programmering 1 (10 ECTS)
- PG660 – Enterprise programmering 2 (10 ECTS)
- PJ312 – Project Software Engineering (10 ECTS)

2008

- PG560 – Enterprise programmering 1 (10 ECTS)
- PG660 – Enterprise programmering 2 (10 ECTS)
- PJ311 – Project Software Engineering (10 ECTS)
- PG520 – .Net Mobil programmering (5 ECTS)
- PG420 – .Net programmering 2 (10 ECTS)

2007

- PG560 – Enterprise programmering 1 (10 ECTS)
- PG660 – Enterprise programmering 2 (10 ECTS)
- PJ310 – Project Software Engineering (10 ECTS)
- PG420 – .Net programmering 2 (10 ECTS)

2006

- PG501 – Enterprise programmering (10 ECTS)
- PG320 – .Net programmering 1 (5 ECTS)
- PJ301 – Project Software Engineering (10 ECTS)

2005

- DB102 – Databaser (5 ECTS)
- PJ301 – Project Software Engineering (10 ECTS)
- PJ400 – Undersøkellesmetoder (5 ECTS)

For alle bachelor emnene så består gjennomføringen typisk av 24 timer undervisning i form av seminar, forelesning, workshop og lignende. Og 24 timer øving under veiledning hvor studentene jobber med øvingsoppgaver.

Master (Alt foregår på engelsk)

2019

- Systems Development (7.5 ECTS)
- Mobile Computing (7.5 ECTS)
- Research Methods in Computer Science (7.5 ECTS)
- UI Programming and Architectures (7.5 ECTS)
- Master Thesis (52,5 ECTS)

2018

- Systems Development (7.5 ECTS)
- Mobile Computing and IoT (7.5 ECTS)
- Research Methods in Computer Science (7.5 ECTS)
- UI Programming and Architectures (7.5 ECTS)
- Master Thesis (52,5 ECTS)

2017

- Systems Development (7.5 ECTS)
- Mobile Computing (7.5 ECTS)
- Architecting the Internet of Things (7.5 ECTS)

2016

- Architecting the Internet of Things (7.5 ECTS)
- Systems Development (7.5 ECTS)
- Programming and API for Interaction (5 ECTS)

2015

- Systems Development (7.5 ECTS)
- Integration Oriented Architecture (7.5 ECTS)
- Architecting the Internet of Things (7.5 ECTS)
- Programming and API for Interaction (5 ECTS)

2014

- Agile Project Management (7.5 ECTS)
- Systems Development (7.5 ECTS)
- Integration Oriented Architecture (7.5 ECTS)
- Programming and API for Interaction (5 ECTS)

2013

- Agile Project Management (7.5 ECTS)

2012

- Agile Project Management (7.5 ECTS)

2011

- Software Design and Architecture (15 ECTS)

2010

- Enterprise System Architecture (15 ECTS)

For alle masteremnene så er gjennomføringsformen sammensatt av en kombinasjon av workshop, forelesninger, gruppearbeid, seminar og andre undervisningsformer.

Andre nasjonale - OsloMet

2019

- Visualisering (10 ECTS), Bachelor
- Programming and API for Interaction (10 ECTS), Master

2018

- Visualisering (10 ECTS), Bachelor
- Programming and API for Interaction (10 ECTS), Master

2017

- Programming and API for Interaction (10 ECTS), Master

2016

- Programming and API for Interaction (10 ECTS), Master

2015

- Programming and API for Interaction (10 ECTS), Master

2014

- Programming and API for Interaction (10 ECTS), Master

Internasjonalt

2016

- Copenhagen Business School, Denmark - E-business Smartphone App Development (5 ECTS) - Bachelor

2015

- International Hellenic University, Greece - Web Information Systems (10 ECTS) - Master
- Addis Ababa University, Ethiopia - Software Design and Management (10 ECTS) - PhD
- Copenhagen Business School, Denmark - Smartphone App Dev. (5 ECTS) - Bachelor

2014

- International Hellenic University, Greece - Web Information Systems (10 ECTS) - Master level
- Copenhagen Business School, Denmark - Smartphone App Dev (5 ECTS) - Bachelor

2010

- Brunel University, UK - Software Design and Architecture (10 ECTS) - Master level

Vedlegg 3: Kursbevis og vitnemål fra UH ped kurs

KURSBEVIS

Universitets- og høskolepedagogisk basiskompetanse Våren 2018

Tor-Morten Grønli

gjennomførte basiskurs i universitets- og høskolepedagogikk, UHPED6000, ved Senter for profesjonsstudier, OsloMet – storbyuniversitetet.

Kurset er et obligatorisk kvalifiseringstilbud som fyller minstekravet til pedagogisk basiskompetanse for faglig tilsatte ved universiteter og høyskoler.

Kurset har et omfang tilsvarende 15 studiepoeng (sp), og er organisert som en serie med fire todagers samlinger og en siste, femte, endagssamling. Det er lagt vekt på aktive arbeidsmåter og en utforskende tilnærming til universitets- og høskolepedagogikk.

Emnet har fire, brede innholdskomponenter:

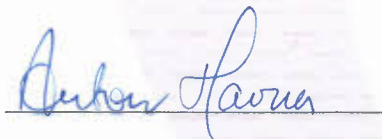
- Kunnskap om profesjon, profesjonsutdanning, profesjonalisering. Høyere utdanning. Akademiske standarder.
- Undervisning i profesjonsutdanningene. Læreplaner. Lærerrolle. FoU-basert undervisning. Ulike undervisningssituasjoner. Fagforskjeller og – likheter.
- Studenters læring, perspektiver og tradisjoner. Undervisning, feedback og veiledning. Gruppeundervisning og gruppearbeid.
- Vurdering, studentevaluering av utdanning, styrking av egen undervisning.

Fullført opplegg forutsetter minimum 75 % tilstedeværelse på kurset, samt ha dokumentert pedagogisk basiskompetanse gjennom arbeidskrav, deltakelse i samlingene, kollegaveiledning og et selvvalgt prosjektarbeid som leder til en avsluttende prosjekt rapport.

Oslo, juni 2018



Oddgeir Osland
senterleder



Anton Havnes
Professor/kursansvarlig

Karakterutskrift

Tor-Morten Grønli

Pilestredet Park 21
0176 OSLO

Navn: **Grønli, Tor-Morten**

Fnr: 310381 42383

Studenten har oppnådd følgende resultater ved OsloMet - storbyuniversitetet:

Emne	Termin	Studie- poeng	Karakter	
UHPED6000	Universitets- og høgscolepedagogisk basiskompetanse	2018 vår	15	Bestått

Sum: 15,0

Oslo, 29. august 2018

Michgona Artykova

underskrift

Vedlegg 4: Uttalelse fra OsloMet

Oslo, den 24.10.2019

Referanse for Tor-Morten Grønli

Jeg har vært Tor-Morten nærmeste leder ved OsloMet siden 2014 da han ble ansatt i 20% stilling i tillegg til sin hovedstilling ved Høyskolen Kristiania.

Selv om Tor-Morten kun har jobbet deltid ved institutt for informasjonsteknologi har hans bidrag til utdanningene vært betydelig. Han har vært aktivt involvert i både undervisning og veiledning, både på bachelornivå og på masternivå. Han er en ettertraktet veileder, som hvert år tiltrekker seg flere studenter og studentgrupper enn det som er opprinnelig planlagt, og han har vist en stor fleksibilitet i arbeidsoppgavene og i temaer han er villig til å veilede studenter i.

Studentevalueringer for emnene som Tor-Morten leder eller er involvert i er gjennomgående gode. Studentene fremhever spesielt at hans undervisning holder et høyt faglig nivå, både akademisk og teknisk, og at de lærer mye fra hans innsikt og tilbakemeldinger. Han har også klart å involvere masterstudenter i forskningsprosjekter, og noen av dem har resultert i publikasjoner.

Tor-Morten har vært en viktig bidragsyter i flere utviklingsprosesser ved institutt for informasjonsteknologi. Han har blant annet vist en tydelig evne til å tenke helhetlig om utdanninger og å kommunisere effektivt med andre i kollegiet, for eksempel da han har identifisert behov for oppdatering av emner. Tor-Morten har god oversikt over metoder og innhold i utdanninger både i Norge og i utlandet og er alltid villig til å dele sine erfaringer og kunnskaper.

Tor-Morten har også deltatt i mentorprogrammet for professorkandidater, der erfarne professorer følger en adept gjennom et helt år og støtter utviklingen av deres opprykksøknad til professor. Det kan nevnes at det er vanskelig å finne nok mentorer ved OsloMet, fordi dette er en tidskrevende oppgave, selv om man har hovedstilling ved OsloMet. Det er derfor ekstra imponerende at Tor-Morten, som kun jobber 20% ved OsloMet har klart å dedikere tid til denne viktige oppgaven.

Jeg kan uten tvil anbefale at Tor-Morten får status som merittert underviser.

Hvis dere har behov for mer informasjon eller har andre spørsmål er dere velkomne til å ta kontakt.

Mvh



Laurence Habib
Instituttleder
Institutt for informasjonsteknologi
OsloMet – storbyuniversitet
Epost: laurence@oslomet.no
Telefon: +47 41612157

Vedlegg 5: Roboten Pepper

Til prosjektleder
Tor-Morten Grønli

12.10.2018

Søknad om pedagogiske utviklingsmidler

Høgskolen Kristiania har lyst ut midler til å stimulere til økt undervisningskvalitet. Utlysningen er forankret i Kvalitetsmeldingen (St meld 16 Kultur for kvalitet) som vektlegger forskning på egen undervisning, studentaktivisering, kontakt med arbeidslivet, nye vurderingsformer, og bruk av nye læremidler (digitalisering). Det ble oppfordret til å søke om to typer av midler:

Fyrtårnprosjekter har som mål *direkte* å forbedre undervisningen gjennom å være nyskapende i retning av fremdragende undervisningskvalitet. Prosjektene skal ha en langsiktig målsetting om senere å søke status som Senter for fremdragende utdanning (SFU). Prosjektperioden har to års varighet. Tildelingsbeløp 500.000 – 1.000.000.

Utviklings søknader der formålet er å forbedre kvaliteten på studieprogrammer og emner gjennom å *videreutvikle* sentrale deler av undervisningsaktiviteter, veiledning og vurdering. Prosjektperioden har ett års varighet. Tildelingsbeløp 10.000 – 150.000.

Det er kommet inn 25 søknader, hhv 2 fyrtårnprosjekter og 23 utviklingsprosjekter. Til sammen er det søkt om 4.236.559 kroner; 3.136.559 i utviklingsprosjekter og 1.100.000 kroner i fyrtårnprosjekter. Den totale rammen for bevilgninger er på 2.000.000 kroner. Samtlige institutter og flere administrative støttefunksjoner (Karrieresenteret, Avd. for studentoppfølging, biblioteket) er representert blant søkerne. Dette er svært positivt og indikerer høyt trykk på undervisningskvalitet på tvers av instituttene og avdelingene ved høgskolen.

Viser til innsendt søknad om pedagogiske utviklingsmidler til prosjekt *Interaktiv IT-utdanning med roboten Pepper*. Søknaden er vurdert av en komite under ledelse av professor i pedagogikk, Ingeborg Krange. De andre medlemmene i komiteen har vært, en ekstern pedagog, leder av senter for læringsteknologi, direktør avdeling for nettstudier, læringsteknologi og pedagogisk utvikling, leder av senter for pedagogisk utvikling og en masterstudent. En begrunnelse for søknadsbedømmingen følger vedlagt.

På grunnlag av komiteens vurdering, har avdeling for nettstudier, læringsteknologi og pedagogisk utvikling under ledelse av Kristin Arnesen, gjort vedtak om tildeling:

Prosjektet *Interaktiv IT-utdanning med roboten Pepper* tildeles kr. 150 000,-.

Bruk av midlene

Det er opprettet et prosjektnummer i Agresso for ditt prosjekt: 11030

Når du bestiller varer eller tjenester oppgir du følgende til leverandøren:

- Høyskolen Kristianas e-faktura-adresse: leverandor@kristiania.no
- Ditt prosjektnummer
- Høyskolen Kristianas org.nr. 954831604

Refusjon av utlegg leveres i [Unit4-portalen](#). Be om at ditt prosjektnummer belastes.

Kontaktpersoner for spørsmål:

- Refusjoner; regnskapsmedarbeider [Trude Nordal](#)
- Refusjon av reiseutlegg; lønnskonsulent [Arne Skovdahl](#)
- Avlønning av forskningsassistenter; lønns- og personalkonsulent [Eva Halonen](#)
- Prosjektregnskap; controller [Magnus Mølsted Andersen](#)

Prosjektperiode

Prosjektet forventes å følge oppgitt fremdriftsplan, og ferdigstille selve prosjektfasen senest ved utgangen av 2019.

Dersom prosjektet blir forsinket, eller det oppstår ulike former for avvik iht oppsatte planer, så ber vi prosjektleder ta kontakt med Senter for pedagogisk utvikling ved benedicte.blom@kristiania.no for felles å komme frem til en løsning.

Formidling av resultater

Prosjektet forventes å formidle sine resultater i henhold til søknadens formidlingsplan. Søker må forvente å dele erfaringer og resultater fra sitt prosjekt på en heldagskonferanse med erfaringsutveksling som mål mot slutten av 2019.

Personvern og meldeplikt

Hvis prosjektet skal behandle [personopplysninger](#), og/eller [sensitive personopplysninger](#), så skal det [meldes til NSD](#). Med innføringen av ny personvernforordning har konsesjonsplikten for behandling av sensitive personopplysninger falt bort, men sensitive personopplysninger er også personopplysninger som krever at prosjektet meldes til NSD.

Viktig: Hvis prosjektet er meldepliktig til NSD og/eller REK, så skal prosjektleder gi beskjed til forskadm@kristiania.no når prosjektet er meldt inn. Er prosjektleder i tvil, ber vi han/hun ta kontakt med forskningsadministrasjonen som vil hjelpe med å avklare om prosjektet er meldepliktig.

Medisinsk og helsefaglig forskning

Hvis prosjektet skal behandle [medisinske og helsefaglige data](#), så skal dette godkjennes på forhånd av *Regionale komiteer for medisinsk og helsefaglig forskning*.

Hvis prosjektet krever godkjenning av REK; husk å [registrere din Cristin person-ID i din søknadsprofil hos REK](#). Så snart prosjektet er godkjent, vil det da bli automatisk overført fra REK via SPREK og til prosjektkatalogen i Cristin. Slik kan REK følge opp godkjente prosjekter.

Oppskrifter:

- [Finn din Cristin person-ID og lagre den på ditt personkort i REK](#)
- Hvordan du [redigerer helseprosjekt fra REK i Cristin](#)

Databehandleravtale

Eventuelle eksterne prosjektdeltakere/studenter som skal behandle persondata på vegne av prosjektet, må inngå databehandleravtale med Høyskolen Kristiania. Ta kontakt med forskningsrådgiverne dersom du trenger mal for databehandleravtale.

Sikker lagring av data

Vi minner om hva som er sikre lagringssteder for persondata:

Innsamlingsfase	Analysefase	Langtidslagring
<ul style="list-style-type: none">• Videoopptaker/Lydropptaker (som ikke er online)• Spørreskjema/Intervjunotater på papir• H:/• Mobile diskette/Minnepinne (krypterte)	<ul style="list-style-type: none">• H:/• Mobile diskette/Minnepinne (krypterte)	<ul style="list-style-type: none">• NSD• Mobile diskette/Minnepinne (krypterte)

Lykke til!

Vennlig hilsen,



Sander Sværi
Prorektor utdanning



Kristin Arnesen
Direktør avd. nettstudier, læringsteknologi
og pedagogisk utvikling

Vedlegg: Vurdering av søknaden

Vedlegg - Vurdering av søknaden

Søknaden er vurdert av en komite under ledelse av professor i pedagogikk, Ingeborg Krange. De andre medlemmene i komiteen har vært, en ekstern pedagog, leder av senter for læringsteknologi, direktør avdeling for nettstudier, læringsteknologi og pedagogisk utvikling, leder av senter for pedagogisk utvikling og en masterstudent.

Sterke sider ved prosjektet og søknaden

Komiteen anser dette som et nyskapende prosjekt (roboten Pepper) som er godt forankret i det pedagogiske utviklingsarbeidet ved instituttets ulike studieprogrammer og emner.

Studentrepresentanten sier dette om prosjektet: Jeg syntes dette prosjektet i stor grad er pedagogisk forankret ved at studentene får oppleve den teknologiske utviklingen «hands on», og ikke kun lese og lære om det teoretisk.

Vi oppfordrer til å gå i partnerskap med læringsforskere som kan følge bruken av Pepper i undervisningen – dette kan gjerne være en PhD kandidat.

Svake sider ved prosjektet og søknaden

Komiteen har ingenting å påpeke her.

Søknadsskjema for utviklingsmidler

Søknadsskjema for prosjektmidler 2018. Søknadsfrist 1.september 2018

Søker og prosjekt

Om prosjektet:

Prosjekttittel	Interaktiv IT-utdanning med roboten Pepper
Prosjektleder	Tor-Morten Grønli
Søknadsbeløp	148 000
Institutt(er)	Institutt for teknologi
Studieprogram/emne/annet	Master Applied Computer Science (Emner: 7.5stp Mobile Computing & Internet of Things, 7.5stp Interactive Technologies), Master Human-Computer Interaction (Emner: 7.5stp Multimodal Interaction), Bachelor IT Intelligente systemer (Emner: 7.5stp Android programmering, 7.5stp Maskinlæring)
Annet (feks evt antall studiepoeng, antall studenter mv)	Totalt vil 5 emner på hver 7.5 stp bli berørt, samt opplegg for masteroppgaver og forskningsprosjekter knyttet til Mobile Technology Lab bli muliggjort. Basert på nåværende registrerte studenter estimeres det at ca 100 studenter vil bli eksponert for dette i løpet av ett års gjennomføring.

Prosjektbeskrivelse

Bakgrunn:

Teknologisk utvikling og nyskaping er ledende faktorer for innovasjon i de aller fleste industrier og bransjeområder i dag. Utviklingen har over de siste 10 årene gått i et enormt tempo og vi har blitt introdusert for, adoptert og dagliggjort en rekke teknologiske innovasjoner. 2017 markerte ti-års jubileum for smarttelefonen, en komponent vi i dag ser på som en naturlig integrert del av hverdagen vår. Vi har gått fra å bruke en enkel enhet for hvert gjøremål til å ha integrerte løsninger. Vi oppfatter i dag synkronisering av bilder, dokumenter, musikk, bøker og andre personlige preferanser på tvers av alle enhetene vi bruker som naturlig og forventet. Løsningene vi bruker blir nå bygget basert på skybaserte plattformer igjennom distribuerte arkitekturer som har heterogene sammensetninger som standard og ikke unntaket.

I denne samme perioden har automatiserte løsninger også begynt å vokse frem. Talestyring igjennom virtuelle assistenter, slik som Apple sin Siri, Amazon sin Alexa og Google Assistant, har gjort dette nærmest hverdagslig og leder oss i en retning av automatisering for personlig styring. Eksempelvis har forskningsprosjekter sett på hvordan Alexa brukes og oppfattes i hjemmet (Purinton et al., «Alexa is my new BFF», Proceedings of Computer-Human Interaction 2017). Semi-automatiserte selvkjørende biler basert på teknologisk innovasjon gjennom bilmodeller hos blant annet Tesla, Volvo og BMW. Smarthus etablerer seg som preferert løsning ved nybygging eller restaurering av hjem, og de samme teknologiene tilrettelegger for styring av hverdagsgjenstander i hjemmet, effektiviserer sikkerheten i alarmsystemer og brukes som støttesystemer for å kunne la eldre bo i sitt eget hjem lenger

(Telenor smarte hjem <https://www.telenor.no/privat/smarte-hjem/> og Innovasjon Norge Smarte byer <https://www.innovasjon Norge.no/no/mulighetsomrader/smarte-samfunn-og-byer/>).

Søknadsskjema for utviklingsmidler

Søknadsskjema for prosjektmidler 2018. Søknadsfrist 1.september 2018

Alle disse perspektivene er kun noen utvalgte eksempler på hvordan den teknologiske utviklingen utfordrer oss, støtter oss og gir oss muligheter. Algoritmene som ligger bak er veletablerte, men anvendes nå i nye situasjoner og kontekster. I front av denne utviklingen ligger nå roboter som teknologisk spydspiss. Det er eksempler på prosjekter hvordan man igjennom bruk av roboter kan løse oppgaver som veiledning av passasjerer på flyplasser, hjelpe hotellgjester i resepsjoner (Pepper i Disneyland <https://www.softbankrobotics.com/us/solutions/pepper-marriott>) eller guide på kjøpesentre (Westfield <https://www.softbankrobotics.com/us/solutions/pepper-westfield>). Teknologien er nyskapende og innovativ, men introduseres i stadig flere bransjer.

For at våre utdanninger skal være relevante, spesielt på masternivå, må våre studenter få hands-on erfaring i å oppleve og påvirke interaksjon og teknologiske utfordringer og være rustet til å være bransjeledende premissgivere innen sine respektive felt for informatikk og interaksjon.

Om prosjektet:

Vi vil i dette prosjektet bevisstgjøre studentene på utfordringene og mulighetene som ligger i interaksjon med og oppbygningen av intelligens i roboter slik som Pepper fra Softbank Robotics (<https://www.softbankrobotics.com/emea/en/robots/pepper>). Vi vil igjennom utvikling av pedagogisk materiale og undervisningsopplegg utdanne våre studenter til å stå helt i front hva gjelder teknologisk kompetanse, forståelse for bruk og interaksjon med, samt grunnleggende kunnskaper om oppbygning av intelligens i, den nære fremtids kanskje viktigste innovasjon: roboter.

Vi vil i dette prosjekt gå til innkjøp av én stykk 6. generasjons robot av typen Pepper. Pepper er en robot som basert på en Linux plattform muliggjør implementasjon av applikasjoner i dens eget proprietære språk, samt i Python og Android. Roboten har en rekke sensorer innebygget for berøring, datainnsamling fra miljøet, bevegelse, kommunikasjon og interaksjon, og har en levetid på 10 til 12 timer effektiv bruk mellom hver opplading. Roboten kommer med både engelsk og norsk talestyring og to års garanti og brukerstøtte fra leverandøren. Dette prosjektet vil utvikle pedagogisk materiale og undervisningsopplegg gjennom forelesninger, øvingsopplegg, prosjektoppgaver, masteroppgaver knyttet til tre studieprogrammer og vi vil videre beskrive forankringen til de konkrete emnene i de respektive studieprogrammene.

Master Applied Computer Science:

7.5stp Mobile Computing & Internet of Things:

Fokus her er utvikling og forskning på mobile løsninger og sensorbaserte systemer. Pepper vil her bli brukt som plattform for utvikling, og integrasjon med sensorene på enheten vil være sentralt.

7.5stp Interactive Technologies:

Fokus her er på utvikling av løsninger som setter teknologien i kontekst med et samfunnsproblem, bransjeutfordring eller en akademisk problemstilling. Pepper vil her bli brukt som caset gruppene skal realisere sin løsning på våren 2019.

Masteroppgave:

Pepper som teknologisk konsept vil være gjenstand for to masteroppgaver våren 2019. Den ene oppgaven vil ha fokus på teknologisk implementasjon, og den andre vil ha fokus på interaksjon og brukeropplevelse. Oppgavene sin problemstilling vil utformes i samarbeid med oss, som henholdsvis ansvarlige for Applied Computer Science og Human-Computer Interaction.

Søknadsskjema for utviklingsmidler

Søknadsskjema for prosjektmidler 2018. Søknadsfrist 1.september 2018

Master Human-Computer Interaction:

7.5stp Multimodal interaction:

I dette faget skal studentene lære om hvordan interagere med systemer basert på varierte grensesnitt og modaliteter. Eksempelvis vil kommunikasjon gjennom sensorer, berøring, tale og lukt være aktuelle interaksjonsformer. Pepper vil her bli brukt som case og plattform for caset som studentene realiserer våren 2019.

Masteroppgave:

Pepper som teknologisk konsept vil være gjenstand for to masteroppgaver våren 2019. Den ene oppgaven vil ha fokus på teknologisk implementasjon, og den andre vil ha fokus på interaksjon og brukeropplevelse. Oppgavene sin problemstilling vil utformes i samarbeid med oss, som henholdsvis ansvarlige Applied Computer Science og Human-Computer Interaction.

Bachelor IT Intelligente systemer:

7.5stp Android programmering

I dette emnet handler det om programmering av mobile løsninger på Android plattformen. Pepper sin plattform er også Android, noe som betyr at studentene vil kunne bli eksponert for programmering av robot-grensesnitt allerede i sitt andre semester.

7.5stp Maskinlæring

Maskinlæring handler om strukturene, algoritmene og fundamentet for å bygge «intelligens», dvs logikk, inn i en maskinkomponent. Pr nå foreleses det både teoretisk og praktisk igjennom programmering. Pepper sitt bidrag vil være at studentene får muligheten til ikke bare å programmere en enkelt algoritme, men også teste den ut i kontekst av dens naturlige bruksområde innen kunstig intelligens nettopp i en robot.

Prosjektets relevans

Fra introduksjonsdagene for Master ACS og Master HCI, hvor Evry brakte med seg sin Pepper til oss, viste de igjennom sin gjesteforelesning hvordan roboter og automatisering av tjenester igjennom slike grensesnitt er et av de store satsningsområdene deres fremover. De fremholdt nytteverdien og viktigheten av at kandidatene som uteksamineres har erfaring med og ikke bare kjennskap til de teknologiske nyskapningene og trendene som preger samfunnsutviklingen. Vi vet fra vår egen forskning på mobile løsninger, internet of things og interaktive løsninger at dette feltet utvikles hurtig og at det trengs ytterligere bidrag for å videreutvikle interaktive løsninger, teknisk implementasjon og ikke minst nytteverdi for sluttbrukere.

Er prosjektet nyskapende?

Prosjektet vil være nyskapende ved at det tar tak i en av tidens største trender og innovative teknologier. Ved å ha tilgang til slikt utstyr på instituttet og bringe det inn i undervisningen gjør av vi kan være med å ligge i forkant og prege utviklingen og bruk. Det eksisterer i dag få utdanningsinstitusjoner i Norge som bruker slikt utstyr i undervisningen og vi vil med dette kunne være med på å eksemplifisere nye pedagogiske opplegg (Det er oss bekjent at OsloMet har kjøpt inn én stykk Pepper, men primært til bruk innen forskning).

Søknadsskjema for utviklingsmidler

Søknadsskjema for prosjektmidler 2018. Søknadsfrist 1.september 2018

Hva er forankringen i praksisfeltet?

Roboten Pepper er allerede i dag utbredt i praksisfeltet i en rekke bransjer fra handel og service til sikkerhet og helse. Den har i flere sammenhenger blitt brukt på sykehus som for eksempel populære tilskudd på barneavdelinger og til aktivisering av demente pasienter. Hotellresepsjoner og kjøpesentre har mange eksempler på bruk av den for å svare på spørsmål knyttet til navigering og informasjon om butikker. Det er og et eksempel fra 2016 om hvordan flyplassen i Brussel har anvendt den til å skape forutsigbarhet i sikkerhetskontrollen for sine passasjerer. Videre viser Evry sin egen stilling som konseptutvikler for Pepper at også IT bransjen nå ser strukturert på hvordan nye typer grensesnitt blir en del av IT porteføljen som deres kunder vil etterspørre når det lages moderne IT løsninger.

Hvordan bidrar det til økt læringsutbytte / bedre læringsopplevelse for studentene?

Studentene vil igjennom dette kunne få jobbe fysisk og konkret med teknologi for å visualisere problemstillinger og se effekten av sine løsninger. Ved å jobbe hands-on vil de få viktig førstehånds kunnskap om hvordan en løsning oppleves fra brukeren sitt perspektiv og hvordan interaksjon med nye, moderne grensesnitt kan gjøres. De vil ved å jobbe hands-on kunne virkeliggjøre løsningene og det fører til en synlig effekt umiddelbart i klasserommet. De vil med dette arbeide mindre abstrakt, og i stedet konkret løse utfordringer knyttet til interaksjon og kunstig intelligens som normalt er vanskelige å se effekten av for praksisfeltet.

I hvilken grad involverer det studentene?

Studentene vil være i sentrum for hele prosjektet og ikke mulig å gjennomføre uten deres bidrag. Kjernen i prosjektet er utvikling av pedagogisk og undervisningsopplegg, samt gjennomføringen av dette. Dette betyr at studenter på både bachelor og master vil møte Pepper i ulike emner (som spesifisert over) og derigjennom møte og løse utfordringer knyttet til programmering, design, algoritmer og interaksjon. Ved å eksponere masterstudentene for bruk av Pepper i undervisning, vil vi der gjennom sette det i en forskningskontekst. Totalt vil vi dermed kunne gjøre bruk av Pepper i undervisning på bachelor og master, forskningsprosjekter på master og ikke minst sammen med studentene kunne gjøre offentlig disseminering av resultatene i presse, populærvitenskapelig presse og akademiske artikler.

Faglig, pedagogisk og/eller strategisk forankring

Den faglige nytteverdien er ovenfor beskrevet for de respektive emnene vi vil anvende Pepper i. I tillegg vil slik bruk av teknologi støtte opp under målene til alle tre studieprogrammer, Master Applied Computer Science, Master Human-Computer Interaction og Bachelor IT Intelligente Systemer om å være praksisnære, nyskapende og innovative. Noen få eksempler fra overordnede læringsmål slik som «*has specialised knowledge within user-centred design and interaction design*» (Master HCI), «*practical skills in analysing complex individual- and organizational problems, research issues and technology innovation opportunities*» (Master ACS) eller «*kunne designe og implementere løsninger for intelligente systemer fra forretningskonsept til ferdig løsning* » (Bsc Intelligente systemer) er med å på vise at dette er forankret i viktige læringsmål for alle tre studieprogrammer.

Pedagogisk og strategisk forankring er knyttet til høyskolen og instituttets ambisjoner om å være ledende på pedagogisk modell, lede utviklingen av fagområdet og ha en praksis-nær tilnærming til bruk av teknologi forankring i arbeidslivets behov.

Søknadsskjema for utviklingsmidler

Søknadsskjema for prosjektmidler 2018. Søknadsfrist 1.september 2018

Gjennomførbarhet

Gjennomførbarheten vurderes til å være meget høy, da vi begge to allerede forsker og foreleser på den grunnleggende teknologien. Alle emnene som det planlegges brukt i eksisterer som en del av godkjente studieprogrammer og det er studenter i løpet på alle tre programmer. I tillegg forsker vi begge to både på interaksjonen, programmering og har flere publikasjoner direkte knyttet til fagfeltet. Vi har begge to også over 15 års erfaring i undervisning, undervisningsplanlegging og utvikling av undervisningsopplegg, og er således sikre på at vi på en god pedagogisk forankret måte kan anvende denne teknologien i undervisning, f.eks. med pedagogiske opplegg forankret i *active learning*.

Samarbeid, evaluering og deling

Alle disse emnene blir årlig vurdert igjennom fagevalueringer og forelesers egevaluering. Dette vil være tilfelle også i de kommende gjennomføringene, og vi vil i så måte knytte til noen ekstra spørsmål knyttet direkte til Pepper for å få konkret tilbakemelding. Det pedagogiske opplegget og undervisningsmaterialet som lages vil bli gjenbrukt i neste gjennomføring av emnet.

For kommunisering og deling vil det være sentralt å bruke dette til å promotere studieprogrammene i markedsføringsammenheng, og som historier for utad å konkretisere og eksemplifisere hva man jobber med og hvordan man jobber ved instituttet og studieprogrammene. Resultatene fra masteroppgavene vil bli brukt til å lage akademiske artikler for å formidle ny kunnskap rundt slike løsninger. Erfaringene vi gjør oss fra bruken av Pepper i undervisning ser vi også for oss at er naturlig skrives opp i en akademisk artikkel som sendes til Norsk konferanse for undervisning og didaktikk i IT-fagene (UDIT) i 2019.

I tilknytning til samarbeid viser vi til kommende punkt som beskriver samarbeidet med vår eksterne partner Evry, som vi mener vil være med å styrke studentenes bransjekontakt, gi virkelighetsnære case og eksponere de for arbeidslivet i en undervisningssituasjon.

Forankring av prosjektet

På institutt for teknologi er en av våre viktigste målsettinger å utdanne morgendagens teknologer. Dette gjør vi i gjennom å gi de praktisk kunnskap med teoretisk forankring, godt fundamentert på akademiske resultater og oppdatert innsikt fra vårt bransjenettverk. Det forventes at vi ikke bare holder tritt med den teknologiske utviklingen i sektoren, men er med på å drive denne utviklingen videre. Vi har flere eksempler på at vi igjennom moderne teknologiutdanning er nettopp med og påvirker. Vi har hospitanter fra videregående skole, en veletablert industribachelor-ordning, vi har aktivt bruk av gjesteforelesere fra næringsliv, en høy akademisk kvalitet og aktivitet, samt tre forskningsgrupper. I tråd med høyskolens strategi om å være *praksisnær og forskningsbasert*, samt *utdanne studentene til livet etter eksamen*, så mener vi dette prosjektet vil være et godt eksempel på å kombinere nyskapende teknologi med undervisning, alt knyttet opp mot trender og innovasjoner studentene må forholde seg til som kommende premissgivere i bransjen.

Søknadsskjema for utviklingsmidler

Søknadsskjema for prosjektmidler 2018. Søknadsfrist 1.september 2018

Fremdriftsplan:

Prosjektfasen - dvs. prosjektaktivitetene - forventes å være fullført innen 1 år fra tildeling. Evaluering og formidling av prosjektresultatene kan imidlertid skje etter at prosjektet er gjennomført. Fyll inn hovedaktiviteter/milepæler i prosjektperioden (år og måned).				
Hovedaktivitet/milepæl	Fra		Til	
	År	Måned	År	Måned
Utvikling av undervisningsopplegg for Maskinlæring og Android Programmering	2018	10	2018	12
Gjennomføring av Maskinlæring Android programmering	2019	1	2019	5
Utvikling av undervisningsopplegg for Mobile Computing and IoT	2018	11	2019	2
Gjennomføring av Mobile Computing and IoT	2019	2	2019	2
Utvikling av undervisningsopplegg for Interactive Technologies	2019	2	2019	3
Gjennomføring av Interactive Technologies	2019	5	2019	5
Utvikling av undervisningsopplegg for Multimodal Interaction	2019	2	2019	3
Gjennomføring av Multimodal Interaction	2019	5	2019	5
Definering av masteroppgaver	2019	1	2019	2
Gjennomføring av masteroppgaver	2019	1	2019	6
Publisering av resultater	2019	6	2019	12

Om prosjektleder:

Prosjektleder, Tor-Morten Grønli, er professor ved institutt for teknologi. Undervisningserfaringen består av ca 15 år med undervisning, fagutvikling, studieprogramutvikling og pedagogisk utviklingsarbeid. Totalt sett har jeg undervisningserfaring fra PhD nivå, master nivå siden 2010 og bachelor siden 2003. Totalen blir over 400 studiepoeng med undervisning på bachelor og over 100 studiepoeng på master, innen fagfeltene informatikk, programmering, prosjektarbeid, mobile løsninger, visualisering med mer. Jeg har undervist nasjonalt og internasjonalt og er gjesteprofessor ved Copenhagen Business School. Jeg har i gjennom årenes løp utviklet en lang rekke studieprogrammer, emner og undervisningsmateriale. Jeg har veiledet over 50 bachelor grupper, over 20 masterstudenter og har i dag to PhD stipendiater. Eksempel på forskningsaktivitet er at jeg har grunnlagt, og leder, forskningsgruppen for anvendt informatikk, samt lab-gruppen Mobile Technology Lab. Jeg har deltatt internasjonalt i flere EU søknader og vært med å etablere samarbeidet med Centre for Business Data Analytics ved Copenhagen Business School. Totalt har jeg over 70 fagfelleverderte publikasjoner.

Søknadsskjema for utviklingsmidler

Søknadsskjema for prosjektmidler 2018. Søknadsfrist 1.september 2018

Evt prosjektdeltakere:eefew

Navn	Stilling	Institutt
Siri Fagernes	Høyskolelektor	Teknologi
Tor-Morten Grønli	Professor	Teknologi

I tillegg til prosjektleder, er Siri Fagernes med i denne prosjektgruppen. Siri har tilsvarende lang erfaring i undervisning, fagutvikling, studieprogramutvikling og pedagogisk utviklingsarbeid fra Telenor FoU, OsloMet og vår høyskole. Siri har ansvaret for studieprogrammene Bachelor IT interaktivt design, og Master Human-Computer Interaction. Siri har sin PhD innen distribuerte systemer og forsker aktivt på universell utforming, optimering og human-computer interaction. Prosjekt-teamet er sammen ansvarlige for Mobile Technology Lab, og i forrige semester veiledet vi, sammen med våre to PhD stipendiater, 7 masterstudenter til fullførelse – hvorav én også vant «Best poster award» på den internasjonalt anerkjente konferansen Social Media and Society. Begge har flere pågående forskningsprosjekter sammen, samt samarbeider allerede aktivt med undervisning i flere emner og utarbeider pedagogiske undervisningsopplegg sammen. Begge medlemmene av prosjekt-teamet har godkjent UHPED fra OsloMet.

Samarbeid

Prosjektet legger opp til samarbeid med konsultentselskapet Evry og spesifikt vår kontakt der Kristine S. Lorentzen som leder deres initiativ for AI/maskinlæring/robotikk knyttet til roboten Pepper. Kristine er tidligere student fra Kristiania (WOACT) og jobbet i sin masteroppgave på et av våre forskningsprosjekter. Siden uteksaminering har vi regelmessig kontakt, og sist under oppstartsdagene for Master ACS og Master HCI, ledet Kristine en kreativ workshop med nettopp roboten Pepper.

Eventuelt eksterne samarbeidspartnere:

Navn	Stilling	Arbeidssted
Kristine S. Lorentzen	Konsulent / konseptutvikler for Pepper	Evry, Oslo

Kostnadsoverslag og finansiering

Budsjett:

Formål	Sum
Innkjøp av Pepper (akademisk pris) 14400 EUR + 300 EUR frakt (avrundet til hele tusen) Se vedlegg for prisbekreftelse.	143 000
Materiell (kasse) for oppbevaring av Pepper	5 000
Total søknadssum	148 000

Søknadsskjema for utviklingsmidler

Søknadsskjema for prosjektmidler 2018. Søknadsfrist 1.september 2018

Eventuelt annen finansiering

Søkes det om annen finansiering? (ja/nei)	Nei
Hvis ja, hvor søkes det midler?	-
Søknadsbeløp	-
Er den andre søknaden innvilget (ja/nei)	-
Hvis ja, med hvilket beløp?	-

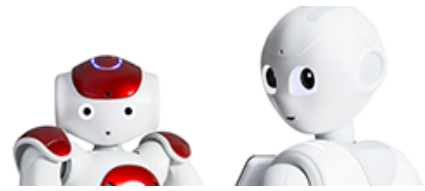
Finansieringsplan:

Utgifter

Aktivitet/formål	Når mnd/år-mnd/år	Kostnader
Innkjøp av Pepper	11/2018	143 000
Innkjøp materiell	11/2018	5 000
Totalsum		148 000

Inntekter

Finansieringskilder	Inntekter kr	Bekreftet ja/nei
Interne utviklingsmidler	148 000	nei
Totalsum		148 000



Quote

Date : 31/08/2018
 Quote Number : 00032830
 Aldebaran ID : NO00207333

SoftBank Robotics Europe SAS

43 rue du Colonel Pierre Avia
 75015, Paris
 France
 Your contact : **Jonathan BOIRIA**
 jboiria@softbankrobotics.com

Billing Details :

Kristiania University College
 Tor-Morten Grønli

Norway

Shipping Details :

Kristiania University College
 Tor-Morten Grønli

Norway

1 Pepper unit, Academics Edition, including 2 years of warranty

Products

Product Reference	Description	Unit Price	Quantity	Total Price
PEP18AEUGD-B2A-2Y	Pepper For Academics Edition including: - 1 Pepper robot 1.8a version - 2 years warranty - Pack of 2 languages (English + 1 of your choice)	EUR 14.400,00	1	EUR 14.400,00
- PEPPER18AWHGDGD-ACAD	Pepper 1.8a robot for Academics - JP/US/EU power cords - 2 years warranty		1	
- SW-ADLG	Pack of 2 languages for your robot (English + one of your choice)		1	

Total

Shipping and other fees : EUR 250,00

Total Excl. Taxes : EUR 14.650,00

Tax : 0%

Grand Total : EUR 14.650,00

Signature

By signing this quote, you expressly approved SoftBank Robotics Europe SAS General Terms and Conditions, which are at the back of this quote. Our General Terms and Conditions will govern and prevail on any clients documents and general and/or specific terms and conditions.

Date and Customer Signature : (Please return this quotation with the mention "signed as agreed" and fully authorized signature and visa, together with the corresponding purchase order.)

Expiration Date : 30/09/2018

Incoterms: DAP - Delivered at place

Payment Terms : Cash in advance

Vedlegg 6: International Hellenic University engasjement og studentevaluering

HELLENIC REPUBLIC



INTERNATIONAL
HELLENIC
UNIVERSITY

SCHOOL OF SCIENCE & TECHNOLOGY

14th km THESSALONIKI – N. MOUDANIA
THERMI- THESSALONIKI
570 01 GREECE
Tel.: +30 2310 807 524
Fax: +30 2310 474 520
E-mail: infotech@ihu.edu.gr
www.tech.ihu.edu.gr

Thessaloniki, 20/11/2015
Ref. No: SST 218

Dear **Prof. Groenli**,

It is with great pleasure that we shall welcome you to the International Hellenic University as visiting professor for the ICT-related MSc programmes.

The duration of your contract will be from **16 October 2015 to 11 April 2016**. During this period you will be responsible for the preparation, teaching (15 hrs) and assessment of the course "**Web Programming**". Teaching will take place on the university premises on campus (14th km Thessaloniki – Nea Moudania). You will also be responsible for the supervision and/or evaluation of a maximum of three (3) student dissertations. If necessary, you will be responsible for setting re-sits or new assignments. You may be required to attend an Appeal Board in case such a process is initiated by any of the students with respect to your course.

Apart from teaching obligations, we ask you to undertake to support, communicate and advise IHU students throughout the academic term via the electronic notice board & e-mail.

In addition, you will be asked to complete a Teaching Staff Data Form and a Course Specification Form. Please note that Greek legislation and the Hellenic Quality Assurance and Accreditation Agency (HQA) require that all Greek higher education institutions prepare documented Annual Internal Reports and Internal Evaluation Reports. To this purpose, the HQA asks all higher education institutions to ensure the systematic collection of the respective data from all their teaching staff through the completion of the aforementioned data forms. This process is fundamental to the quality assurance of the educational work of our University and we would sincerely appreciate your cooperation in this regard. Further information about the quality assurance framework implemented at the IHU can be found on the Quality Assurance Unit web site: <http://qau.ihu.edu.gr>.

The course office will provide you with full information about the particular arrangements of the timetable and teaching hours for your course, as well as with guidance regarding completion of the Teaching Staff Data Form and Course Specification Form in due course.

Please send our Programme Manager, Dr Christos Koidis (c.koidis@ihu.edu.gr), your course syllabus and any relevant material needed for your unit (reading lists, case studies, exercises) in due time, according to the schedule.

The International Hellenic University undertakes to pay you a fee of **1.893,75 €**, which payment will be effected within one month of the completion of your obligations, as set out above, under the following condition.

In accordance with international agreements on the avoidance of double taxation of income and capital, if such agreement has been ratified between your country and Greece, you will need to submit to the International Hellenic University a form invoking its implementation (valid until the end of the year during which the income became due). Please keep in mind that no contract can be issued or signed before we receive the above form from you. If your country has not signed this agreement with Greece, an automatic deduction of 20% income tax will be applied on the agreed amount (Law 4172/2013, article 64). For information on countries that have signed this agreement with Greece, please visit http://www.gsis.gr/gsis/info/gsis_site/ddos/ choosing 'B. Conventions for the Avoidance of Double Taxation'. For the relevant forms, please, choose 'C. Forms for Double Taxation Conventions.

The University will cover the cost of your flights (economy class), which can be arranged either by yourself or by the School Office (p.seremetis@ihu.edu.gr / tel. +30 2310 807521). It is vital to note that whether booked through the University or privately, your flight costs can only be covered by the University on receipt of the booking confirmation (with prices) and your boarding card stubs. The IHU will also cover 50 Euro per night towards your accommodation for "days of teaching" nights. The above amount corresponds to the bed and breakfast rate (double room) at the Avalon Hotel, which lies within easy walking distance from the university. If you are happy to stay there, we can organise both the booking and direct payment, upon request. Alternatively, we shall refund the equivalent upon submission of your hotel bill. Please let the School Office know your preferences.

We hope the above meets with your agreement and would ask for your written confirmation of the same. If you have any queries, please, do not hesitate to contact Dr Christos Koidis (c.koidis@ihu.edu.gr / tel. +30 2310 807501).

We trust that our students will gain from your expertise and you will enjoy working for IHU. We look forward to welcoming you to the IHU.

Yours sincerely,

The Dean of the School



Prof. Ioannis Vlahavas

Evaluation Report 1st Term 2015-16 | International Hellenic University

Christos Koidis <c.koidis@ihu.edu.gr>

Thu 24/03/2016 17:24

To: tmg@westerdals.no <tmg@westerdals.no>

Dear Prof. Groenli,

I hope my email finds you well.

As part of our policy, students proceed to the evaluation of the teaching staff every term.

Please, find below your personal evaluation for the 1st term 2015-16 as well as overall average.

The information provided is strictly confidential.

Please, let me know should you require any clarification.

Kind regards,



Dr Christos Koidis | Programme Manager
International Hellenic University
T+30-2310-807501 | M+30-6972017040
c.koidis@ihu.edu.gr | www.tech.ihu.edu.gr



Think before you print. 14th km Thessaloniki-N.Moudania | 57001 Thermi, Greece | www.ihu.edu.gr

Course:	Web Programming						
Term:	1st 2015-2016						
Instructor:	Prof. Groenli						
No	Question	Responses					Grade (0-5)
		very poor	poor	adequate	good	excellent	
1	The content of the course was	1	0	7	11	7	3,88
2	The course content was intellectually stimulating	strongly disagree	disagree	neutral	agree	strongly agree	3,88
		1	2	5	9	9	
3	The recommended readings and exercises were appropriate and useful	strongly disagree	disagree	neutral	agree	strongly agree	3,69
		1	2	6	12	5	
4	The course has developed my problem solving, analytical, professional or strategic leadership skills	strongly disagree	disagree	neutral	agree	strongly agree	3,69
		1	2	8	8	7	
	Course materials (notes, handouts,	strongly disagree	disagree	neutral	agree	strongly agree	

5	case studies, online resources etc) were clear, relevant and timely	1	4	6	8	7	3,62
6	The professor was, overall	very poor	poor	adequate	good	excellent	4,23
		1	0	2	12	11	
7	The professor was good at explaining the material and helping me learn	strongly disagree	disagree	neutral	agree	strongly agree	4,12
		1	0	6	7	12	
8	The professor encourages questions and discussion	strongly disagree	disagree	neutral	agree	strongly agree	4,38
		1	1	3	3	18	
9	The professor was contactable (in person and electronically) and helpful	strongly disagree	disagree	neutral	agree	strongly agree	4,23
		1	0	4	8	13	
Overall Performance:							3,97

Professors' Overall Average 1st Semester 2015-2016 Evaluation	4,34
--	-------------

CONFIDENTIALITY WARNING - DISCLAIMER This message and any attachments (the "message") are confidential and intended solely for the addressees. If you receive this message without being the intended addressee, please delete it (together with any attachment) without copying, forwarding or opening it and immediately notify the sender. Any unauthorized use or dissemination or disclosure, either whole or partial, is prohibited. E-mails are susceptible to alteration. The International Hellenic University shall not be liable for the message if altered, changed or falsified, nor shall it bear any responsibility for any loss, disruption or damage to your data or computer system that may occur from the use of data contained in, or transmitted with, this message.

ΓΝΩΣΤΟΠΟΙΗΣΗ-ΔΗΛΩΣΗ ΠΡΟΣ ΤΟΥΣ ΠΑΡΑΛΗΠΤΕΣ Αυτό το μήνυμα και τυχόν συνημμένα σε αυτό αρχεία (το μήνυμα) είναι εμπιστευτικά και απόρρητα και απευθύνονται μόνο στους σκοπούμενους παραλήπτες. Σε περίπτωση που λάβετε αυτό το μήνυμα χωρίς να είστε ο σκοπούμενος παραλήπτης, παρακαλούμε να το διαγράψετε (μαζί με τα συνημμένα σε αυτό αρχεία), χωρίς να το αντιγράψετε, να το προωθήσετε σε τρίτους ή να το ανοίξετε και να ειδοποιήσετε άμεσα τον αποστολέα. Απαγορεύεται η μη εξουσιοδοτημένη χρήση ή διάδοση ή κοινοποίηση σε τρίτους του συνόλου ή μέρους του μηνύματος αυτού. Τα μηνύματα ηλεκτρονικού ταχυδρομείου (e-mails) είναι δεκτικά αλλοίωσης. Το Διεθνές Πανεπιστήμιο της Ελλάδας δεν θα ευθύνεται για αυτό το μήνυμα στην περίπτωση αλλοίωσης, μεταβολής ή παραποίησης του, ούτε θα υπέχει ευθύνη για οποιαδήποτε απώλεια ή αλλοίωση ή άλλη ζημία στα δεδομένα ή στο σύστημά σας, οφειλόμενη σε χρήση δεδομένων που είτε περιέχονται στο μήνυμα αυτό είτε μεταδόθηκαν με αυτό.

Evaluation Report 1st Term 2015-16 | International Hellenic University

Christos Koidis <c.koidis@ihu.edu.gr>

Thu 24/03/2016 17:24

To: tmg@westerdals.no <tmg@westerdals.no>

Dear Prof. Groenli,

I hope my email finds you well.

As part of our policy, students proceed to the evaluation of the teaching staff every term.

Please, find below your personal evaluation for the 1st term 2015-16 as well as overall average.

The information provided is strictly confidential.

Please, let me know should you require any clarification.

Kind regards,



Dr Christos Koidis | Programme Manager
International Hellenic University
T+30-2310-807501 | M+30-6972017040
c.koidis@ihu.edu.gr | www.tech.ihu.edu.gr



Think before you print. 14th km Thessaloniki-N.Moudania | 57001 Thermi, Greece | www.ihu.edu.gr

Course:	Web Programming						
Term:	1st 2015-2016						
Instructor:	Prof. Groenli						
No	Question	Responses					Grade (0-5)
		very poor	poor	adequate	good	excellent	
1	The content of the course was	1	0	7	11	7	3,88
2	The course content was intellectually stimulating	strongly disagree	disagree	neutral	agree	strongly agree	3,88
		1	2	5	9	9	
3	The recommended readings and exercises were appropriate and useful	strongly disagree	disagree	neutral	agree	strongly agree	3,69
		1	2	6	12	5	
4	The course has developed my problem solving, analytical, professional or strategic leadership skills	strongly disagree	disagree	neutral	agree	strongly agree	3,69
		1	2	8	8	7	
	Course materials (notes, handouts,	strongly disagree	disagree	neutral	agree	strongly agree	

5	case studies, online resources etc) were clear, relevant and timely	1	4	6	8	7	3,62
6	The professor was, overall	very poor	poor	adequate	good	excellent	4,23
		1	0	2	12	11	
7	The professor was good at explaining the material and helping me learn	strongly disagree	disagree	neutral	agree	strongly agree	4,12
		1	0	6	7	12	
8	The professor encourages questions and discussion	strongly disagree	disagree	neutral	agree	strongly agree	4,38
		1	1	3	3	18	
9	The professor was contactable (in person and electronically) and helpful	strongly disagree	disagree	neutral	agree	strongly agree	4,23
		1	0	4	8	13	
Overall Performance:							3,97

Professors' Overall Average 1st Semester 2015-2016 Evaluation	4,34
--	-------------

CONFIDENTIALITY WARNING - DISCLAIMER This message and any attachments (the "message") are confidential and intended solely for the addressees. If you receive this message without being the intended addressee, please delete it (together with any attachment) without copying, forwarding or opening it and immediately notify the sender. Any unauthorized use or dissemination or disclosure, either whole or partial, is prohibited. E-mails are susceptible to alteration. The International Hellenic University shall not be liable for the message if altered, changed or falsified, nor shall it bear any responsibility for any loss, disruption or damage to your data or computer system that may occur from the use of data contained in, or transmitted with, this message.

ΓΝΩΣΤΟΠΟΙΗΣΗ-ΔΗΛΩΣΗ ΠΡΟΣ ΤΟΥΣ ΠΑΡΑΛΗΠΤΕΣ Αυτό το μήνυμα και τυχόν συνημμένα σε αυτό αρχεία (το μήνυμα) είναι εμπιστευτικά και απόρρητα και απευθύνονται μόνο στους σκοπούμενους παραλήπτες. Σε περίπτωση που λάβετε αυτό το μήνυμα χωρίς να είστε ο σκοπούμενος παραλήπτης, παρακαλούμε να το διαγράψετε (μαζί με τα συνημμένα σε αυτό αρχεία), χωρίς να το αντιγράψετε, να το προωθήσετε σε τρίτους ή να το ανοίξετε και να ειδοποιήσετε άμεσα τον αποστολέα. Απαγορεύεται η μη εξουσιοδοτημένη χρήση ή διάδοση ή κοινοποίηση σε τρίτους του συνόλου ή μέρους του μηνύματος αυτού. Τα μηνύματα ηλεκτρονικού ταχυδρομείου (e-mails) είναι δεκτικά αλλοίωσης. Το Διεθνές Πανεπιστήμιο της Ελλάδας δεν θα ευθύνεται για αυτό το μήνυμα στην περίπτωση αλλοίωσης, μεταβολής ή παραποίησης του, ούτε θα υπέχει ευθύνη για οποιαδήποτε απώλεια ή αλλοίωση ή άλλη ζημία στα δεδομένα ή στο σύστημά σας, οφειλόμενη σε χρήση δεδομένων που είτε περιέχονται στο μήνυμα αυτό είτε μεταδόθηκαν με αυτό.

Vedlegg 7: Formidlingsartikkel fra Computerworld

COMPUTERWORLD

E-helse Olje/energi Bygg/anlegg Offentlig it Fintech



COMPUTERWORLD | MACWORLD | IT-BRANSJEN | TELECOM REVY | EVENT | WHITEPAPER |
STILLING LEDIG | E-AVIS

STILLING LEDIG

[Alle stillinger](#)

 **Systek** Smidig prosjektleder
Systek AS

 **BKK** Driftssjef
BKK Digitek AS



Lærer alt om Java på tre år

Studenter ved NITH kan bytte til .Net når de er ferdig utdannet.

MARR SØREHUS

NAKU SUNREURS

Publisert: onsdag 06. april 2011, kl. 07:05

Fra første semester på NITH får studentene innføring i objektorientert programmering og utvikling av algoritmer.

- Programmering er en av de sju linjene vi har på NITH. Alle studenter følger det samme løpet det første året, og kan etter dette ta fordypning på 2.- og 3.-året, sier Tor-Morten Grønli, førstelektor og ansvarlig for studieprogrammet Programmering.

Programmering står ved siden av linjer som spillprogrammering, spilldesign, 3D-grafikk, interaktivt design og digital markedsføring. Det er mellom 25 og 30 studenter som hvert år velger å spesialisere seg innen programmering. På nettsidene sine skriver høyskolen at "gjennom studiet utdannes du til å konstruere arkitekturer fra bunnen, designe systemer og utvikle attraktive applikasjoner - alt gjennom en dyp opplæring i programmeringsspråket Java".

- Det er en god og bred linje. Den har grunnleggende it med mye programmering, databaser, prosjektarbeid, datateknikk og operativsystemer. Det skaper en bred kunnskap, et generelt dannelsesfundament innen it, forteller Grønli.

Dekker grunnleggende Java

Programmeringsspråket Java er grunnpilaren i programmet gjennom hele studietiden. Hvert semester består 10 av 30 studiepoeng av Java, altså en

ANNONSE ▼

COMPUTERWORLD
EVENT PUBLICWORLD 11. DESEMBER
Meet Ullevaal

Slik setter vi fart
på digitaliseringen
av offentlig sektor

Steffen Sutorius, direktør i Difi

tredjedel av hele studiet.

STILLING LEDIG

Alle stillinger _



**Anskaffelsesleder
IKT**

Politiets Fellestjenester

Java
er

gjennomgående, man får en dyp innføring, sier Grønli.

Det dekker ifølge førstelektoren alle nivåer fra grunnleggende Java og objektorientering, til programmering på desktop, distribuerte systemer og design av arkitekturen til systemer.

- Vi går veldig langt i spesialiseringen, og dekker alle deler av Java som Oracle har.

Ved siden av dette lærer man å jobbe i prosjekter, xml, operativsystemer og Ruby on rails, som er et rammeverk for å lage dynamiske web-applikasjoner.

- Studenter må lære å jobbe med dynamiske skriptspråk. Vi følger med på nye trender og ny arkitektur, sier Grønli.

Lærer å teste

I løpet av det andre året lærer studentene blant annet å bruke unntakshåndtering på en fornuftig måte, lage trådbaserte applikasjoner, gjøre seg kjent med ulike typer datastrukturer og bruke ulike typer Collections og Maps med tilhørende algoritmer. I tillegg lærer de å lage klient/tjener-applikasjoner, sette opp en Apache Tomcat-server og utplassere web-applikasjoner på denne og lage programmer som kommuniserer med databaser. De lærer også å teste programmer ved bruk av ulike testverktøy.

- Ja, det meste går på Java. Av dette får vi gode programmerere og gode arkitekter. Jeg mener det er viktig å ha dypdekompetanse. Etter det kan man godt skifte til .Net eller Ruby eller andre programmeringsspråk. Har man lært et språk helt ned til bunns, kan man bytte dialekt etterpå.

Signerer ved juletider

Uansett er Java-programmerere veldig ettertraktet. Ferdig som programmerere er det et bredt spekter av arbeidsoppgaver man kan ta fatt på, som konsulent, som utvikler, som selvstendig næringsdrivende eller som del av en it-avdeling i privat eller offentlig sektor. Flere får jobb før studiene er ferdig.

- De flinkeste programmeringsstudentene blir signert ved juletider. Et halvt år før de er ferdig utdannet.

Programmering på NITH satser også på mye på mobil utvikling basert på Android. Kunnskap om mobil utvikling er med på å aktualisere utdanningen til studentene og vil fremover bli en naturlig del av en systemutviklingsleveranse, mener Grønli.

I tillegg er metodologi viktig, og i det andre året er det mye fokus på smidige metoder, blant annet gjennom praktisk bruk av Scrum.

Men det er ikke bare teknologi det dreier seg om, selv om man er programmerer.

- Det er viktig å skape et kreativt rom, å

tenke løsninger. Den menneskelige biten, det å forstå problemet, er utrolig sentralt. Det er viktig å kunne kommunisere med kunder, å forstå behovet og kunne lage tekniske løsninger ut fra det, sier Grønli.

LES OM:

Utvikling Og Metode Utvikling

ANNONSØRINNHold



Storkommunen Asker skal bruke Workplace

Gjennom kommunikasjonsplattformen «Workplace by Facebook» skal ansatte i Asker, Hurum og Røyken jobbe sammen



LES OGSÅ:
Top 16 RPA
tools today



LES OGSÅ:
Top 16 RPA
tools today

Utvikling Og Metode

Vedlegg 8: Artikkel om Scaffolding av studentprosjekter

Learning from achievement: scaffolding student projects in software engineering

Bendik Bygstad*

Norwegian School of IT, Norway
E-mail: bendik.bygstad@nith.no
*Corresponding author

Birgit R. Krogstie

Norwegian University of Science and Technology, NTNU
Norwegian School of IT
Tor-Morten Grønli, Norway
E-mail: birgitkr@idi.ntnu.no

Tor-Morten Grønli

Norwegian School of IT, Norway
E-mail: grntor@nith.no

Please reduce
abstract (maximum
of 150 words).

Abstract: It has become almost a truism that students learn more from working on projects than from lectures. Various pedagogical approaches, such as Problem-based Learning, Project-based Learning (PBL) and Work-based Learning, have contended that students learn more from solving real-world problems in a setting where the teacher no longer instructs, but acts more like a coach.

There are, however, some inherent problems in the project-learning approach that are underrated in the literature. The most important one may be described this way: If the project task is trivial, the students will not learn much. If the project task is nontrivial and complex, the students will often not succeed in solving it. We suggest in this paper that the solution lies in providing appropriate scaffolding. In broad terms, scaffolding means to provide the necessary support for the learner to gradually master what is needed to complete a task.

The empirical background for the study is a two-semester course in Software Engineering (SE) at the Norwegian School of IT, with data collected over a period of five years with 1400 Bachelor programme students. We conclude that PBL in this setting may be successfully scaffolded by a formal SE method. In particular, the iterative and incremental structure of modern SE may serve as a basis for effective project learning. As our main contribution, we point to six types of scaffolding, each addressing an essential aspect of SE project work. Combined, they provide the necessary scaffolding for PBL in software engineering.

Keywords: scaffolding; student projects; software engineering education.

Reference to this paper should be made as follows: Bygstad, B., Krogstie, B.R. and Grønli, T-M. (xxxx) 'Learning from achievement: scaffolding student projects in software engineering', *Int. J. Networking and Virtual Organisations*, Vol. X, No. Y, pp.000–000.

Biographical notes: Bendik Bygstad holds a PhD in Computer Science from Aalborg University and a Master of Sociology from the University of Oslo. He is currently an Associate Professor at the Norwegian School of Information Technology. His main research interest is the relationship between SW development processes and innovation, and qualitative research methods. He has published articles in such journals as *Information Systems Journal*, *International Journal of Project Management*, *International Journal of Technology and Human Interaction*, *Electronic Government* and *Software Process*.

Birgit Rognebakke Krogstie holds an MSc in Computer Science from the Norwegian University of Science and Technology (NTNU) and a Master of Educational Theory from the University of Oslo. She is a PhD candidate at the NTNU in the area of computer supported cooperative work with a research focus on work and learning in software development project teams.

Tor-Morten Grønli received a Master's degree in Distributed Information Systems and Computing from Brunel University, Uxbridge, UK. He is currently a PhD candidate at the School of Information Systems, Computing and Mathematics at Brunel University, Uxbridge, UK, and a Lecturer at the Norwegian School of IT. His current research interests include software engineering, pervasive and ubiquitous computing, and context-aware systems.

An earlier version of this paper was published at the *NOKOBIT Conference* in Molde, Norway, 20–22 November 2006.

1 Introduction

It has become almost a truism that students learn more from working on projects than from lectures. Various pedagogical approaches, such as Problem-based Learning (Helle *et al.*, 2006), Project-based Learning (PBL) (Markham, 2003) and Work-based Learning have contended that students learn more from solving real-world problems in a setting where the teacher no longer instructs, but acts more like a coach.

We subscribe, in principle, to this thinking. There are, however, some inherent problems in the project-learning approach that are underrated in the literature. The most important one may be described this way: If the project task is trivial, the students will not learn much. If the project task is nontrivial and complex, the students will often not succeed in solving it, and learning from failures is both expensive and discouraging.

We suggest in this paper that an effective strategy for solving this challenge is the careful application of scaffolding, with reference to Bruner's original concept (Wood *et al.*, 1976). In broad terms, scaffolding means to provide the necessary support for the learner to gradually master what is needed to complete a task. Used correctly, scaffolding increases the likelihood that the project groups will succeed in accomplishing a complex task, such as Software Engineering (SE). Our research is motivated by a pedagogic interest and relates to empirical data on actual courses collected over several years. Rather than testing a hypothesis, we apply an interpretive research approach to gain an understanding of a particular practice within SE education, and thereby contribute to the development of this pedagogical field. Our research question, then, is *How can scaffolding be applied successfully in SE student projects?*

The rest of the paper is organised as follows: In Section 2 we present the main challenges of teaching and learning SE and our key concept of scaffolding. In Section 3, we briefly outline our research methodology and present our case. Then in Section 4, we present our scaffolding approach in detail and discuss our results. Section 5 concludes the paper and points to further research.

2 Research review

In this section we first discuss major challenges of SE education. Then we define and discuss the scaffolding concept.

2.1 Learning software engineering

SE is an arena well suited to studying scaffolding, because it presents both the students and the lecturer with a number of challenges. SE is one of the most complex tasks man has ever set out to perform. Simplified, it may be described by two separate challenges. The first is to translate a real-world problem into a design for a software solution; the next is to actually build, test and deploy the solution into the real world (Brooks, 1987). Since the discipline was born in the 1960s, it has been well documented that building large information systems is a high-risk undertaking, demanding both technical and social skills. Even after 50 years of experience, the failure rates of software projects are still surprisingly high (Sommerville, 2001; Standish Group, 2001).

The challenges of teaching SE reflect this situation. From a pedagogical point of view, the main objectives of a typical SE course are:

- to integrate the knowledge and skills from various computer science modules
- to teach the students the process steps and artefacts of SE.

Both objectives represent real pedagogical challenges, because the first one is analytically quite demanding for the students, and the second one is – if taught as theory – excruciatingly boring.

This has led to a number of innovative approaches. Most of them build on pedagogical principles from constructivist thinking (Vygotsky, 1978; Blumenfeld *et al.*, 1991; Dewey, 1997); the students should ‘learn by doing’ (not by being lectured), they should construct a concrete artefact, they should work collaboratively (not individually) and the role of the teacher should be as a facilitator (not authoritarian).

In SE courses, PBL has been reported to work successfully and is widely applied. For example, the annual Conference on Software Engineering Education and Training (CSEE&T) usually includes a track for SE projects. The PBL approaches share a number of attributes: they aim to present larger cases that emulate real-world problems, emphasise collaborative learning, aim at integrating knowledge from different parts of the computer science curricula, and also explore new relationships and roles between students and lecturers (Morrison, 2004; Gulbahar and Tinmaz, 2006). For example, Sindre *et al.* (2003) reports from a cross-course project that fully integrates four other disciplines at the NTNU in Norway, thus achieving a scaling effect to make the case

more realistic. At the same time, pitfalls of collaborative systems development that may affect SE student projects have been identified, *e.g.*, free riders, configuration management and time management (Morrison, 2004).

2.2 *Scaffolding*

A pedagogical challenge for PBL in SE is that the task in itself is quite demanding, leading to a high risk of failure for the student groups. Reflection on the reasons for failure may generally be a source of learning, if there is an opportunity to try out new and better ways of doing things in the next learning cycle (Dewey, 1997). In an SE project, there will normally be phases and/or iterations giving ample opportunities to reflect, learn and improve. Complete failure in the project is, however, likely to be detrimental to students' experience harming motivation for further work and learning (Bandura, 1994). Further, learning in a project relates to a set of skills necessary to successfully complete the project. If serious problems prevent students from having *any* relevant experience with certain central SE tasks (*e.g.*, because of dependencies with other, noncompleted tasks), the learning objectives of practising various skills might not be reached. We accordingly believe that the pedagogical design of an SE project course should have mechanisms to ensure a reasonably successful completion of the essential tasks involved in the SE project. What are the effective strategies for achieving this within the frames of PBL?

Learning can be regarded as happening in a *Zone of Proximate Development* (ZPD). The zone describes the 'space' between what learners can achieve without help and what they can achieve with the help of a tutor or supervisor (Vygotsky, 1978). The skills and knowledge defined by the learning objectives of a PBL design may be found at the outer limit of the students' ZPD as the project starts. As students learn throughout the project, with course staff providing support for them to reach beyond what they are able to do alone, the students' skills develop and the outer limit of the ZPD expands. When a project course is completed, the successful completion of a (structurally) similar project may thus be attainable to the students without much staff supervision.

Providing support for the learner to gradually master what is needed to complete a task has been denoted *scaffolding* by Bruner (Wood *et al.*, 1976), who builds on Vygotsky's work. Scaffolding was initially used as a metaphor for the process where an adult assists a child to carry out a task that is outside the child's current capabilities. This allows the child to concentrate on the elements within reach, gradually increasing his skills. "At best, this process continues until he becomes acquainted with and skilled in all aspects of task activity, to the point where he can initiate and control his own behavior in the absence of an instructor" (Stone, 1993).

In a higher education context, scaffolding includes activities such as providing directions, keeping the student on track, offering assessments, increasing efficiency and reducing risk. Scaffolding typically implies the use of artefacts like computers and documents. On one hand, such learning materials limit the students' options in the learning process. On the other hand, the learning materials *support* the learning process by reducing sociocognitive load and implementing a learning agenda (Suthers, 2006).

SE project courses vary in respect of requirements for students to follow a certain process model and, if a specific model is required, how many templates and guidelines are provided, and how strictly they should be adhered to. Iterative software development approaches represent current practice within SE (Jacobson *et al.*, 1999; Larman, 2004),

and thus many SE courses are based on iterative process models in order to provide students with sought-after competence. Requiring strict adherence to a process model within PBL might, however, raise questions about the necessary room for challenge and for developing individual approaches and solutions.

Anticipating our conclusion, we argue, based on empirical data, that relatively structured iterations in student projects at a certain stage of SE education provide opportunities for scaffolding that greatly benefit learning results. Gradual mastery of the basic steps in conducting project tasks and producing project artefacts in the early phases of a project enables students to explore other equally challenging dimensions of project work. A structured process allows students to come up with (sufficiently) creative and unique solutions with little risk of getting lost in the basics of project artefacts, and with less risk of the whole project failing.

3 Method

In this section we present our method and our case.

3.1 Method

The empirical part of this research is a case study of an SE module in the Norwegian School of IT (NITH) from 2002 to 2006. Two of the authors were lecturers in the course. The course has been under continuous improvement. The experiences from each year were discussed with involved staff and external sensors at the end of the year. These views, supplemented with student evaluation results, were then fed back in the form of improvements for the next year's running of the module.

In 2006 we took a longitudinal view of the module, analysing the trends and patterns (Miles and Huberman, 1994). Data collection was performed with a number of sources:

- Student and marking statistics and student satisfaction data constitute a quantitative resource.
- The projects were very well documented by the student groups. Each project group wrote six iteration reports, a final project report, and also individual and group reflection notes. In particular, the reflection notes were a valuable source of information about the projects, because they described the learning process in detail, as experienced by the group and by each individual. The individual reflection notes were treated confidentially in the projects, and were quite open in describing the interaction between students. In the citations from this material, the quotes were translated from Norwegian.
- Since two of the authors were also lecturers, participative observations of literally hundreds of tutoring sessions with project groups constitute an important part of our empirical material. We acknowledge the limitations in this type of 'data', and use these observations only as background.

First, we analysed the student marks and student feedback data in the years 2002–2006. In addition, we analysed the comments given by students in their evaluation feedback sheets. This gave us a longitudinal view of the results.

Then we reviewed in detail the reflection notes for 2005 and 2006, both group and individual notes. Each reflection note was usually between one and two pages, enabling us to do a detailed qualitative analysis of the issues identified as relevant for the topic of scaffolding. These included support for the main structure and microstructure of the project, the degree of perceived integration with other courses, challenges associated with product and document integrity, and collaboration within the groups.

3.2 *The SE course*

The NITH is a private university college in Oslo. Specialising in IT education, the college works closely with the industry to develop relevant courses, offering Bachelor programmes in IT and in SE.

The PJ300 course was developed in 2001, with the aim of integrating the different courses of the second year of the IT Bachelor's programme. These modules are Programming (Java), Databases (Oracle), Systems Analysis and Design (UML), Technology (Data communications) and Technology and Organisation. The structure of the Bachelor programme is shown (somewhat simplified) in Table 1.

Table 1 Structure of Bachelor of IT programme

3rd year	Programming or Systems analysis	IT in organisations	IT security	Work-based project	
2nd year	Programming and databases	Technology and organisation	Technology	Systems analysis	Project SE engineering
1st year	Programming	Databases	Technology	Systems analysis and projects	

The main objectives of the PJ300 module were:

- to provide a learning space for the students to integrate the knowledge and skills from the other five modules
- to teach the students the basic skills of SE.

As described in the introduction, both objectives represent real pedagogical challenges.

Our strategy to meet these challenges was to provide the students with a large software development case, large enough to last a year and including knowledge and skills from the other parallel modules. The case needed to be constructed in such a way that students at all levels of skill would be comprehensively challenged. In our case, this could be achieved by requiring the project groups to apply UML and patterns (Larman, 2001) in analysis and design, to programme the software solution in Java and Oracle, and run it in a complex environment of PCs, a database and web servers. At the start of the second Bachelor year, this challenge is way above the skills of the students, and the project was designed so that teaching in other parallel modules provided the students with the necessary skills just in time for the project.

To structure this development process, we chose the Rational Unified Process (RUP). There were two reasons for this choice. First, RUP is a modern and widely used SE framework in the IT industry, thus supporting NITH's aim of working closely with industry. Second, RUP provided a quite extensive set of guidelines and templates for the development process (Jacobson *et al.*, 1999).

The number of students (decreasing annually, because of the general decline in the number of IT students in Norway) and the specifics of the case during the five years of operation are described in Table 2.

Table 2 Number of students and project case

<i>Year</i>	<i>Number of students</i>	<i>Case</i>	<i>New technology introduced</i>
2002	632	Airline tickets	Java
2003	379	Hotel and activities booking	Java servlet
2004	255	Airline tickets	Web services
2005	102	Web-based exam test system	WAP application
2006	71	Sushi restaurant order system	J2ME application
<i>Sum</i>	<i>1439</i>		

The structure of the project year is detailed and mandatory. The case is given by the course leader, and is quite detailed, describing the system in five modules. The groups are free to choose how many of the system modules they intend to develop, the modules being increasingly demanding. The more modules that are successfully developed, the better the achievable grade on the result.

The structure of the project year is shown in Table 3. In the first week, the students fill in a form addressing personal motivation and technical skills. Based on these data, the course staff form relatively homogeneous student groups of four. A key feature of PJ300 is that it follows an iterative structure. The project is structured by six iterations, in accordance with RUP. Each iteration is a 'mini project', starting with objectives and requirements, proceeding with design, programming and testing, and ending with a formal meeting with the 'customer', *i.e.*, the tutor. Each iteration is done in a one-week block, where the students work during the full week with the project.

Table 3 Structure of PJ300 during the year

	<i>Week</i>	<i>Iteration</i>	<i>Artefact objective</i>
Autumn semester	34	1	Project plan and GUI prototype
	41	2	Draft SW architecture, application module 1
	50	3	SW architecture, application modules 1+2
Spring semester	3	4	Application modules 1+2+3
	9	5	Application modules 1+2+3+4 (+5)
	16	6	Finished application
Exam	20		Product demo

Please provide heading for Column 1.

Each iteration is documented in a short report, UML models, source code and iteration plans. The iteration result is deployed at a website maintained by the group during the project. At the end of the spring semester the students post their final report and reflection notes, and then the website is closed and made available for the two censors. In the exam the students present their work orally and demonstrate the completed application. The censors pose questions on both process and technology, and give final feedback and grades.

4 Findings and discussion

This section presents the findings from our case study. Our main finding is that various types of scaffolding have been proven effective for learning in this project. We will systematically present and discuss these types.

4.1 *The results of the course*

The results of the PJ300 have been quite good, as shown in Table 4.

Table 4 Results of PJ300

<i>Year</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>
2002/2003*						
2004	36%	35%	22%	5%	2%	1%
2005	38%	34%	14%	12%	2%	
2006	25%	45%	15%	14%		

Note: * Norway changed its grading system in 2004, so grades from 2002 and 2003 are not commensurable. The profile, however, is the same as for the following years.

The quite satisfying results shown in Table 4 are even more impressive when we take into consideration that the actual students of NITH have medium to low grades from high school, and also much poorer results in the programming modules than in the project course. We should, however, add that projects generally achieve better marks than other courses partly due to the ‘free rider’ phenomenon.

The results from the students’ feedback evaluation sheets give the same picture. Students have consistently given quite high satisfaction scores. The maximum score is 6 (very satisfied) and the lowest is 1 (quite unsatisfied). For 2005 and 2006 the scores are as shown in Table 5.

Table 5 Student evaluation results

<i>Year*</i>	<i>Content and relevance</i>	<i>Lecturer</i>	<i>Exercises</i>	<i>My own learning</i>	<i>Overall assessment</i>
2005	5,2	5,1	3,9	5,2	4,8
2006	5,8	4,3	4,3	5,8	5,4
Average NITH	4,5	4,4	3,7	4,0	4,0

Note: * The student feedback was not measured the same way before 2005.

It is notable that while the scores for *content/relevance*, *my own learning* and *overall assessment* are significantly higher than the NITH average, the scores for *lecturer* and *exercises* do not show better results.

4.2 *Our scaffolding explained in detail*

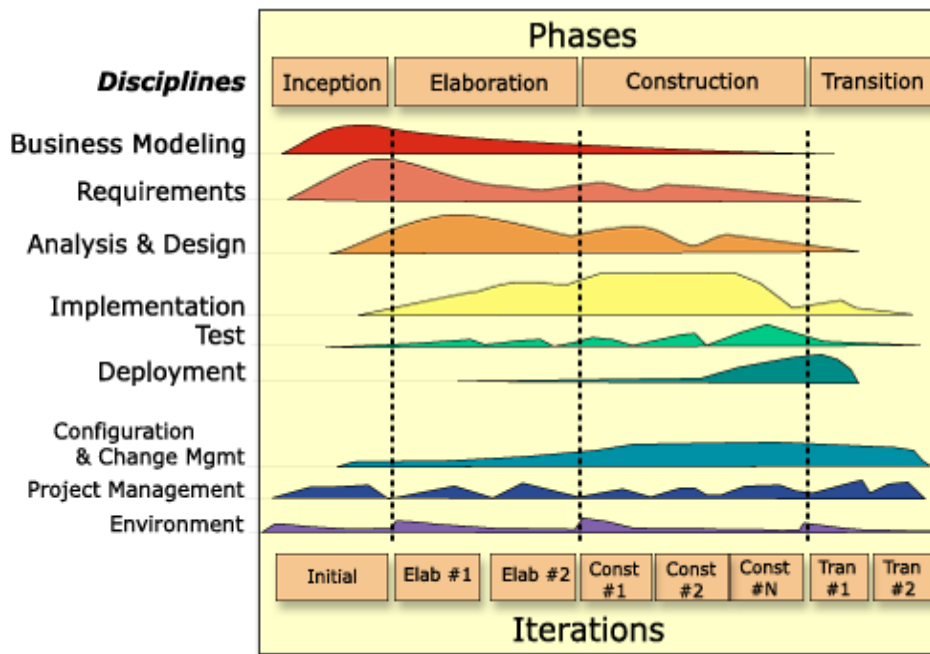
We attribute the good grades and high degree of student satisfaction to various types of scaffolding, listed below:

- The main structure (phases and iterations) is scaffolded by RUP.
- The microstructure (activities) is scaffolded by RUP.
- Technical development is scaffolded by other courses, as shown in Table 1.
- Product integrity is scaffolded by predefined product modules.
- Document quality is scaffolded by predefined document structure and web-based hand-in and assessment.
- Group cooperation is scaffolded by homogeneous groups and role definitions.
- Lecturers and supervisors are providing and timing the scaffolding activities.

4.2.1 Main structure (phases and iterations) is scaffolded by RUP

Scaffolding may be supported both at a macro (structure) and micro (activities) level (Gutzdial, 1994). At a macro level the PJ300 is supported by the iterative structure of RUP, shown in Figure 1. RUP is structured in four phases, each consisting of one or several *iterations*, mini-projects with a clear objective.

Figure 1 Structure of the Rational Unified Process



Source: IBM Corp (2002)

As described in the previous section, a key feature of PJ300 is that it follows an iterative structure. The structure of each iteration is relatively uniform through the six iterations. The students usually struggle with the first two iterations, but gradually the iteration structure is routinised and managed. One student group wrote in their reflection notes:

“We learned most by working with incremental parts which were developed in iterations. Using RUP helped us to go from theory to practice in a large project that was rather similar to an industrial project.”

We propose that the repetitive character of iterations gradually increases the students’ confidence, in accordance with the principles of scaffolding described in Section 2. It is also congruent with Bruner’s notion of a ‘spiral curriculum’, where structural elements are repeated (Bruner, 1960). The SE method has in fact been utilised in other pedagogical settings as an iterative framework for learning. For example, in Spain, a course using the iterative and incremental structure of XP was used as a pedagogical technique, as well as a subject for the students (Alfonso and Botia, 2005).

4.2.2 Microstructure (activities) is scaffolded by RUP

A software process such as RUP includes a main structure for a project (iterative and incremental), a set of artefacts to be produced (documents, models, components) and a sequencing of detailed activities. In professional SE this structure is established to achieve *control* over the development process.

When implemented in a teaching context, does this impose restraints on the creative and exploratory learning process? Our evidence indicates no.

In the first two years (2002–2003), we left the students free to choose how they wanted to use RUP, with options ranging from using only the main structure and improving the rest, to following RUP in every detail, using the textbook and RUP online as resources. The result was disappointing; the groups spent the first two iterations doing research and discussing RUP, and were not able to plan and manage productively. The groups’ discussions of RUP were generally poor, because they did not have any real experience using it.

One student wrote:

“What has not worked well was the start of the iterations. A lot of ideas came up, and we discussed a lot, which led to a late start.”

We solved this problem through (from 2004) a strong structuring of the iteration plans and the definitions of artefacts. Instead of trying to make their own interpretation of RUP, the groups based their project planning on prespecified lists of activities and required artefacts. The result was quite successful, and the groups could work productively from the start. We also saw that less skilful students were able to be productive enough to deliver the minimum requirements and achieve a passing mark (or better) in the module. It is noteworthy that this did not stop the groups from reflecting on the weaknesses of RUP and criticising the framework at the end of the project. An example is the following quote from a student:

“Although I do not have any experience with lightweight methods such as XP, I think they are more my style than RUP. I like to attack problems directly by programming. It is OK to draw models on paper and discuss it with others, but I manage fine without doing this.”

4.2.3 Technical development is scaffolded by other taught courses

The technical challenges of PJ300 are quite demanding for a second-year student. The students are asked to design and program a Java (web) application using a set of software design patterns. To succeed, the group needs skills in Java, web, database and data communications. In addition, the most ambitious groups develop a module for mobile technology, enabling them to communicate with the system using a mobile phone.

Here are two typical comments:

“The design patterns we are learning in systems design were quite useful. Initially, I thought these patterns were just crap, but in fact there were a number of useful points there.”

“In the beginning, the assignment seemed to be very large and complex, but throughout the year we have had lectures in programming and design which made it possible to solve the project case. These lectures were well-timed for using the knowledge in the following project weeks.”

The modules that teach these skills are delivered in parallel with the project. In the period when the students need skills in Java servlets, the Java course is scheduled to teach this subject. Thus, when meeting a problem, they may also ask the lecturer in another module for guidance.

4.2.4 Product integrity is scaffolded by predefined product modules

Before the project is launched, a predesigned solution is built by the staff to act as a basis, the initial scaffold. The solution addresses the learning outcomes in the module and represents one among many possible ways of solving the case. The solution should possess a quality that makes it useful as a pedagogical tool for the staff, whether in direct supervision of the projects or in teaching-related courses. We ensure this quality by making the ‘layers pattern’ (Larman, 2001) as a foundation for the architecture. One comment from an individual reflection note:

“I have reached a deeper understanding for how different technologies together constitute a whole product. Throughout the project new challenges have emerged which had to be solved. This has greatly increased my learning.”

Students aiming for the most advanced deliverables will not be left trying and failing. Rather, the product integrity in the advanced parts is scaffolded by the provision of a product skeleton that can be built upon and that directs a path for the students to follow.

4.2.5 Document quality is scaffolded by predefined document structure and web-based hand-in and assessment

All artefacts are documented on the group’s website from the start of the project, and at the end of the project the result is given a grade. The structure of the website is defined from the start of the project.

As noted by Gulbahar and Tinmaz (2006), the website has several benefits. First, it gives both the project member and supervisor easy access to the materials, even from outside locations. This is important during a long project, during which some students may be ill or travelling abroad. This also eases the collaboration and sharing of information within the group by enabling easy, location-independent access to all

documentation. Second, the students can concentrate on the system development with its necessary documentation – and not on writing reports for the teacher. Third, a portfolio-based assessment emulates a real-world situation, thus supporting a central objective of PBL (Gulbahar and Tinmaz, 2006).

Our experience is that the students are quite satisfied with web-based assessment. However, some lecturers and censors had objections to reading so much documentation on-screen, and asked for printouts.

4.2.6 Group cooperation is scaffolded by homogeneous groups and role definitions

The composition of the group is important. We have experimented with different approaches, starting with heterogeneous teams (students with great differences in skill profiles), but concluding with homogenous teams.

Ideally, a good project team is a heterogeneous one (Belbin, 1993), enabling the project members to contribute individually. However, in PJ300 this created a lot of tension during the first years, because the strong students could not accept that weaker students were given credit for work to which they did not contribute. A basic problem with PBL is free riders (Morrison, 2004), which is considered unfair and unacceptable also by Norwegian students. When such problems occasionally arise, there is a standard procedure with the possible outcome of exclusion. We reduced the free rider problem by making the groups as homogeneous as possible regarding both technical skills and personal motivation. In our experience, the homogeneously composed groups encourage both skilled and less skilful students to perform at the boundary of their competence, *e.g.*, within their ZPD. One group wrote:

“All members were able to contribute to the project, therefore there were no problems with losing the respect for project members because of ‘free riders’ who are in the group but do not contribute.”

This, however, was not always the case. One group reported:

“One member was almost impossible to contact, which created a lot of tension. Deadlines were not held, and neither chatting nor e-mail helped. In the end the student was excluded from the project.”

RUP includes a set of role definitions: project manager, architect, systems analyst, programmer and tester. The roles are described in detail, and are mapped to specific activities. The students sometimes choose to redistribute roles for each iteration.

4.2.7 The role of tutors and supervisors

A basic tenet in PBL is that the tutor should act in a supporting, not instructive, manner. However, successful scaffolding requires that advice provided by tutors and supervisors should relate to explicit learning objectives for the module.

Adequate scaffolding thus requires that tutors and supervisors responsible for the learning process be fully aware of what is good practice (*e.g.*, for a project process) and what is good product quality. It also requires an understanding of the students’ level of competence at the beginning of the project and during its course, and an understanding of

adequate pedagogical steps to get from the current to the desired level of competence. Providing good advice to students on their project work remains a situated endeavour, however meticulously preplanned the learning process and its scaffolding may be.

The provision of appropriate scaffolding to project groups accordingly puts strong demands on supervisors who normally have a limited amount of time allocated for the supervision of each group. Well-structured support material (plans, guidelines, predesigned solutions, *etc.*) related to the types of scaffolding discussed in the previous section is likely to make it easier for course staff to prepare for and engage in the day-to-day issues surfacing in the supervision of project teams.

5 Conclusion

This article reports on an SE project at the Norwegian School of IT, with 1400 students over five years, using the approach of PBL. As the projects are designed to have a certain size and difficulty to provide a real challenge, the risk of failure is considerable.

Building on Bruner's concept of *scaffolding*, we argue that PBL in an SE course may be successfully facilitated by teachers and supervisors.

This allows the students to learn from achievement instead of failures. Supporting PBL with the RUP includes a number of types of scaffolding:

- The main project structure is scaffolded by RUP's phases and iterations.
- The microstructure is scaffolded by RUP activities.
- Technical development is scaffolded by other taught courses, such as programming.
- Product integrity is scaffolded by predefined product modules.
- Document quality is scaffolded by predefined document structure and web-based hand-in and assessment.
- Group cooperation is scaffolded by homogeneous groups and RUP role definitions.

Limitations of the study presented in this article relate to the fact that the empirical data are from only one case, albeit a large and longitudinal one.

We suggest that further research address scaffolding mechanisms used in similar and alternative approaches to SE education. Also, the types of scaffolding identified in our work might be used as a basis for the design of similar pedagogical approaches in other educational settings.

References

- Alfonso, M.I. and Botia, A. (2005) 'An iterative and agile process model for teaching software engineering', *18th Conference on Software Engineering Education & Training (CSEET'05)*.
- Bandura, A. (1994) 'Self-efficacy', in V.S. Ramachaudran (Ed.) *Encyclopaedia of Human Behaviour*, New York: Academic Press, Vol. 4, pp.71–81.
- Belbin, M. (1993) *Team Roles at Work*, Butterworth Heinemann.
- Blumenfeld, P.C., Soloway, E., Marx, R.W., Krajcik, J.S., Guzdial, M. and Palincsar, A. (1991) 'Motivating project-based learning: sustaining the doing, supporting the learning', *Educational Psychologist*, Vol. 26, pp.369–398.

- Brooks, F.P. (1987) 'No silver bullet – essence and accident in software engineering', *Computer*, Vol. 20, No. 4, pp.10–19.
- Bruner, J. (1960) *The Process of Education*, Cambridge, MA: MIT Press.
- Dewey, J. (1997) *Democracy and Education – An Introduction to the Philosophy of Education*, New York: The Free Press.
- Gulbahar, Y. and Tinmaz, H. (2006) 'Implementing project-based learning and E-portfolio assessment in an undergraduate course', *Journal of Research on Technology in Education*, Vol. 38, No. 3, pp.309–327.
- Gutzdial, M. (1994) 'Software-realized scaffolding to facilitate programming for science learning', *Interactive Learning Environments*, Vol. 4, No. 1, pp.1–44.
- Helle, L., Tynjala, P. and Olkinoura, E. (2006) 'Project based learning in post-secondary education – theory, practice and rubber sling shots', *Higher Education*, Vol. 51, pp.287–314.
- IBM Corp (2002) 'Rational unified process', *RUP On-line*.
- Jacobson, I., Booch, G. and Rumbaugh, R. (1999) *The Unified Software Development Process*, Reading: Addison Wesley.
- Larman, C. (2001) *Applying UML and Patterns: An Introduction to Object-oriented Analysis and Design and the Unified Process*, Prentice Hall.
- Larman, C. (2004) *Agile and Iterative Development: A Manager's Guide*, Addison-Wesley.
- Markham, T. (2003) *Project Based Learning Handbook*, Buck Institute for Education.
- Miles, M.B. and Huberman, A.M. (1994) *Qualitative Data Analysis*, Thousand Oaks: Sage Publications.
- Morrison, J. (2004) 'Facilitating collaborative learning within programming projects', *Issues in Information Systems*, Vol. 5, No. 1, pp.221–225.
- Sindre, G., Stålhane, T., Brataas, G. and Conradi, R. (2003) 'The cross-course software engineering project at the NTNU: 4 years of experience', *16th International Conference in Software Engineering Education and Training (CSEET'03)*, Madrid, Spain.
- Sommerville, I. (2001) *Software Engineering*, Harlow: Pearson Education.
- Standish Group (2001) *Extreme Chaos*.
- Stone, C.A. (1993) 'What is missing in the metaphor of scaffolding?', in M.E. Forman and C. Stone (Eds.) *Contexts for Learning: Sociocultural Dynamics in Children's Development*, New York: Oxford University Press, pp.169–183.
- Suthers, D.D. (2006) 'Technology affordances for intersubjective meaning making: a research agenda for CSCL', *Computer-supported Collaborative Learning*, Vol. 1, pp.315–337.
- Vygotsky, L.S. (1978) *Mind in Society. The Development of Higher Psychological Processes*, Cambridge, MA: Harvard University Press.
- Wood, D., Bruner, J. and Ross, G. (1976) 'The role of tutoring in problem solving', *Journal of Child Psychology and Psychiatry*, Vol. 17, No. 2, pp.89–100.

Vedlegg 9: Artikkel om Industribachelor

INDUSTRY BACHELOR PROGRAMME: AN EXPERIENCE REPORT

Rolando Gonzalez, Westerdals Oslo ACT – Faculty of Technology, gonrol@westerdals.no
Wanda Presthus, Westerdals Oslo ACT – Faculty of Technology, prewan@westerdals.no
Tor-Morten Grønli, Westerdals Oslo ACT – Faculty of Technology, grntor@westerdals.no

ABSTRACT

Students within many disciplines, information technology being one, face challenges of getting their first job, such as lack of work experience and outsourcing. This paper describes a variant of a Bachelor education degree, which is called the Industry Bachelor Programme. The motivation behind the Industry Bachelor Programme is to introduce the students to working in a company as consultants so that they may enter directly into a work environment when they finish their studies. Guided by Action Research as the main approach, we report from three years of collected data from interviews, grade reports and documentation. Our contribution is twofold. First, we contribute to academic literature by extending an existing framework for university and company collaboration. Second, we offer a roadmap with detailed tasks, responsibilities and timespan for universities that want to start an Industry Bachelor Programme.

Key words: Bachelor Students, Study Programme, Industry, Internship, Education, Action Research

1. INTRODUCTION

“You cannot get employment without work experience and you cannot get work experience without employment” says the proverb. According to Norton recent challenges in universities are to prepare the students for employment and to be entrepreneurial on a global scale (Norton 2009). In today's competitive market, especially within information technology, students face challenges of outsourcing from low-cost countries, which makes it important for them to have hands-on experience in their field of study that makes them able to enter a work place and deliver from day one. The motivation behind the Industry Bachelor Programme is to introduce the students to working in a real company as real consultants so that they may enter directly into a work environment when they finish their studies. This means they will learn how the techniques they learn at school are used in practice, and apply both theory and skills in live projects.

While research is abundant within pedagogy with the aim on how to motivate the student and tailor teaching, as well as the benefits of internships in the industry, we found less research on how to plan, conduct and evaluate an extended student internship over several years. We aim to contribute to closing this gap by sharing our experiences from an ongoing programme called Industry Bachelor at a Norwegian university college. The Industry Bachelor Programme is a continuous collaboration between the university college and two large consulting companies, where bachelor students are employed as consultants during parts of their studies. They receive all benefits as regular employees, including salary. Students may apply for Industry Bachelor in their third semester. The applicants are selected by the companies based on interviews and grades achieved up to that point. If the applicant is accepted, the student signs a contract and starts working in the chosen company in their fourth semester. Due to this workload, Industry Bachelor is expanded to four years instead of the normal three years bachelor.

Our research question reads: *what can we learn from our experience when introducing an Industry Bachelor Programme in higher education?* The rest of the paper has the following elements: first, we review related literature on internships and similar arrangements between universities and the industry. Then we present our overall method and describe our case. We discuss our findings before presenting our contribution to theory as well as a roadmap for universities and companies who may want to engage in an Industry Bachelor Programme. Finally, we point to limitations of the study and present our conclusion.

2. RELATED LITERATURE

In this section, we describe related research. We start by presenting the employers' requirements, followed by the benefits of work placements or internships.

What do employers want?

Employers want to employ, ideally, people that have both hard and soft skills and already have had a certain amount of work experience (Franz 2008; Hagan 2004; Robin 2011). This puts pressure on the universities in higher education in order to improve the employability of the students by developing the students' hard and soft skills that are applicable to the industry. A survey done by Hagan (2004, p. 3), where employers of ICT graduates were asked how universities should prepare their graduates, showed that 30% of respondents thought that the universities should provide students with more work experience, industry awareness, generic skills, business skills and technical skills. Further Hagan refers to a study (ACNielsen 2000, p. 32) where students were found to be most deficient in problem solving skills, oral business communication skills and interpersonal skills with other staff, i.e. mostly soft skills (Franz 2008).

What is employability?

In relation to the deficiencies identified above amongst other in different soft skills, the concept of employability is important. Employability is referred to by (Franz 2008, p. 2, 3) first in the general sense *"to a person's ability to gain employment"* then gives an account to how the government and industry refers to it being *"the ability to get the most out of employment both for the employee as well as the employer"*. In addition, an important part of employability skills lies in the ability to progress within an organisation to *"achieve one's potential and contribute successfully to enterprise strategic directions"* which may be dependent on both hard and soft skills. Franz (2008) finds that several reports and papers differentiate between employability skills, as being the soft skills, and technical skills, also called discipline specific skills.

Work placement arrangements at universities

How does one solve the challenge of providing the students with the skills required by their employers? The literature tells us that several universities give their students some work practice, in order to improve their employability. This is done through for example Cooperative Education, Sandwich courses (Coll and Eames 2000; Pauling and Komisarczuk 2007), encouraging internships (Tvedt et al. 2001) practicum and other variants of work experience, but this varies from university to university and from course to course (Franz 2008).

An example of a university that has work placement is Victoria University of Wellington, presented by Pauling and Komisarczuk (2007). The university has an arrangement that requires the student to work minimum 150 hours at various employers, with a 300 hours minimum at one employer, and the goal is to reach 800 hours in total. The work placement may be conducted as part time or full-time. In the case of full-time, it is advised to include the summer trimester since the student may take longer than normal to complete the degree. The work placement is set as a core requirement for all students so it covers a wide breadth of areas and must provide for wide variance in students levels of maturity since knowledge, experience and confidence of students vary greatly.

To enable the work placement at Victoria University of Wellington a structure of four main actors that all are part of making the work placement are mentioned: Student, Industry, Careers Service Enablers/Educators and Department/School Administrators, Enablers & Educators, see figure 1 by (Pauling and Komisarczuk 2007, p. 126). The cooperation between the university and the Careers Service is paramount in this case to act as enablers for students and employers. The Careers Services has a key role in developing the student skills in finding employment through amongst other CV writing workshops, interviews and assessment skills workshops and information about employers. In addition, the Careers Service arranges events where companies present themselves for the students.

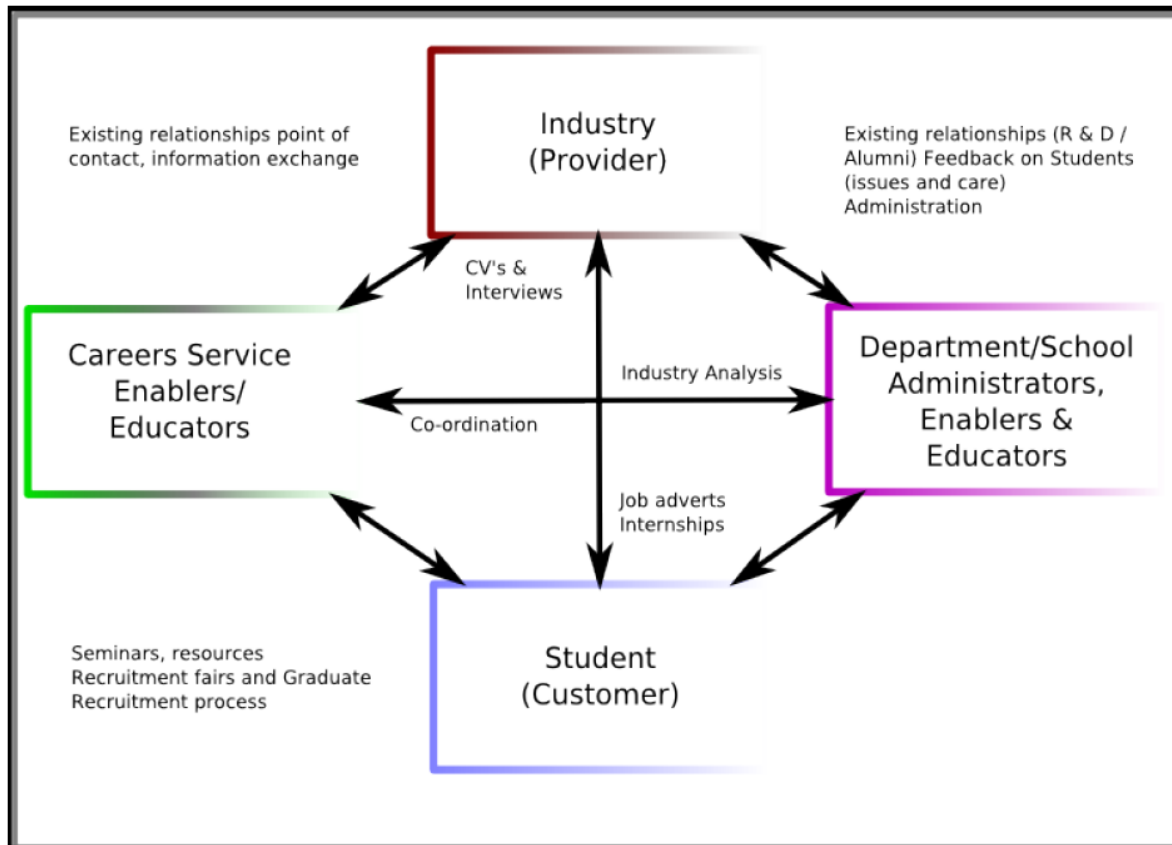


Figure 1: Relationships, Information Exchange and Process (Pauling and Komisarczuk 2007, p. 126).

Benefits of work placement for employer and students

There are many benefits for the employer. Examples include that applicants have relevant experience from working life, and for the employer this is a form of try before hire (Pauling and Komisarczuk 2007). These benefits could also be obtained during summer internships, but as the authors point out, internships over the summer is usually not a peak time in productivity, and due to staff holidays fewer supervisors and other resources are available to the student.

The result of work placement is that the student are “*more likely to than others to have reflected positively on their university experience and to have achieved employment within their chosen field*” (Orrell 2004, p. 1). In addition, work placement provides students “*with an appreciation of the IT work place, enhances their skills with a range of industry experience and tools and provides a step towards professional accreditation*” (Pauling and Komisarczuk 2007, p. 125).

Summing up, we found several articles on programmes similar to our Industry Bachelor. Most of these articles are evaluations on the various benefits for the students and employers, but we found little research that provides a recipe for universities on how to initiate, conduct and evaluate an Industry Bachelor Programme. Inspired by the concept of *Technology Roadmap* (Garcia and Bray 1998) we will create an Industry Bachelor roadmap which can help universities to initiate, conduct and evaluate a similar programme. According to Garcia and Bray, a roadmap mainly helps formalizing processes within businesses. A roadmap is an effective tool for planning and coordination, which fits within a broader set of planning activities. It also helps develop consensus among the actors involved.

3. METHOD

We have used Action Research as our main approach for this research. Action Research is commonly used in Information Systems (Baskerville 1999; Baskerville and Myers 2004), but also in teaching and learning in universities regardless of the discipline (Norton 2009). According to Baskerville, Action Research is suitable when four criteria are fulfilled: (1) *an action and change orientation*, (2) *a problem focus*, (3) *an*

“organic” process involving systematic and sometimes iterative stages, and (4) collaboration among participants (Baskerville 1999, p. 9). We describe how each criterion is met in our study below.

An action and change orientation

The university college strives to be in close contact with the industry, and this model was initiated in order to strengthen the cooperation with the industry.

A problem focus

The university college wanted to provide the students with even better preconditions to get employment right after graduation.

An “organic” process involving systematic and sometimes iterative stages

An academic year is per nature cyclic, and the Industry Bachelor Programme spans several semesters in an iterative nature.

Collaboration among participants

The Industry Bachelor Programme involves and requires collaboration between four main participants (which we will refer to as actors): students, lecturers, administration, and companies.

Action Research is not without challenges (Baskerville 1999) and it is easy to fall into a consultant trap, meaning that the study mainly contributes to practice and less to theory. Norton also note that often teachers are forced to take a stance whether they want to be a “*teacher who researches*’ or a *‘researcher who teaches*’” (Norton 2009, p. 5). Being aware of these challenges, we carefully identified and planned a dual contribution to both theory (mainly based on the interviews with the students, lecturers and administrative staff) and a roadmap for practice. We were inspired by Baskerville’s model as framework for analysing the data (see figure 2).

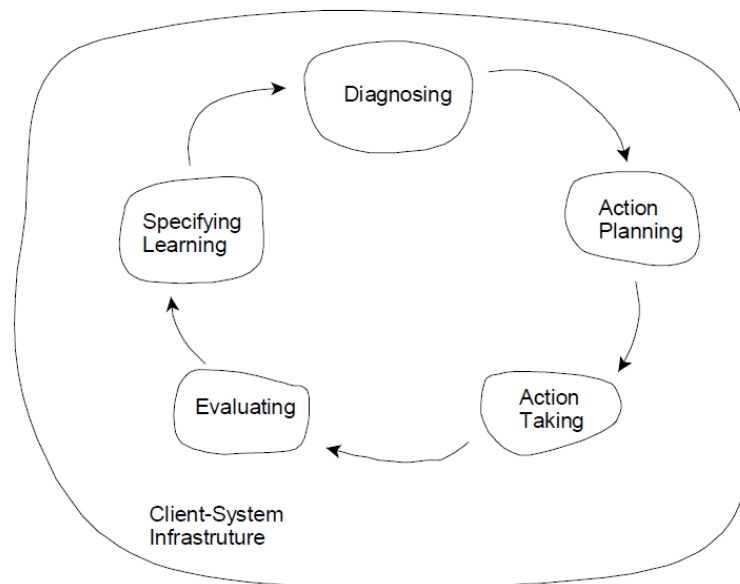


Figure 2: The Action Research Cycle (Baskerville 1999, p. 14)

3.1 Data collection

Our main data material consists of semi-structured interviews with students (Industry Bachelor and regular) and lecturers. The questions for the Industry Bachelor students are found in appendix 1. In addition, we conducted a small survey with regular Bachelor students and lecturers. Finally, we had access to documentation of the study model and e-mail correspondence, typically with the companies.

In order to promote quality assurance, we started to interview Industry Bachelor students in 2012. Each interview was conducted face-to-face and lasted about 30 minutes each. The interviews were recorded and

transcribed afterwards. The students were informed that they would be anonymous and that the results should be used in a research context only, meaning, for example, that the employer would not see the transcripts. In addition, we conducted a survey with six regular bachelor students and four lecturers. Table 1 below sums up the data sources.

<i>Data source</i>	<i>Actor</i>	<i>Collected (year)</i>
20 interviews, face-to-face	Industry Bachelor Students	2012: 6 students 2013: 4 students 2014: 10 students
2 interviews	Dean and senior administrator	2015
1 survey	Lecturers and regular students	2013: 10 respondents in total
Transcripts of grades	Regular and Industry Bachelor students	2015
E-mail correspondence	Companies	2015

Table 1: Overview of collected data for this study

3.2 Data analysis

We have made six and a half iterations in the Action Research Cycle, one iteration per calendar year (January – December). We use calendar year as opposed to an academic year because the students started their Industry Bachelor in their fourth semester, which begins in January. According to Baskerville, Action Research is typically cyclical, and “*Action failures (in terms of the immediate problem situation) are as important as, perhaps more important than, action successes. Action should continue until the immediate problem situation is relieved*” (p. 19). Based on Baskerville’s article we based our analysis in accordance with the respective phases. Since we had a rich data material with mainly text, we also drew on techniques by Miles and Huberman, especially identifying themes and trends from the collected data.

4. THE BACKGROUND STORY (CASE DESCRIPTION)

This action research case takes place in a university college – faculty of technology, that specialises in computer science and information systems. Students typically specialise within programming, E-business, and design. In 2008, the university college was contacted by a large consulting company, who were familiar with the students through hiring several of them. The company’s motivation for contacting the school was to increase the students’ workplace experience during the bachelor education.

The student’s tasks at the company varies but they are all relevant to their studies within Information Technology. Usually, all students have an introduction period where they learn about the routines and project development methods (for example the agile method called SCRUM) of the company. They also perform internal tasks such as testing, documentation, and prototyping. After the introduction period that may last between two and three months the students are usually placed on consultant assignments as regular project members. The types of assignments have varied greatly, ranging from working with mobile applications, developing web pages/applications to working with new experimental technology. The types vary because the company receives many different assignments from their clients.

In regards to communication between the school, work place and the students, it is the responsibility of the Industry Bachelor coordinator to ensure that the communication flows well. The Industry Bachelor coordinator communicates with the students throughout the semester by, for example, checking that they have been registered correctly on courses and have received the information that they need.

Usually the communication between the school and work place is limited to the planning of information meetings. The exception is for example when a student has decided to drop out of the Industry Bachelor Programme.

First iteration: 2009

Initial and informal conversations between the company and the university college took place during the spring semester of 2009. The discussions included feasibility issues of such a collaboration project, which eventually resulted in “Yes”. In February 2009, a formal group was established at the university college to discuss various programme models. Three models were suggested: one where the company was responsible for lecturing while the student was working, the second model placed the lecturing responsibility solemnly

at the university college. The third was a hybrid of the first two models. The second model was chosen and refined, making all lecturing to be the school's responsibility. During six months, from March to August, the university college's faculty, administration, and company worked together in finalising the study model, administrative decisions, and marketing material. From August, promotion of the Industry Bachelor Programme began. Several information meetings were held, culminating in an application deadline on November 1st 2009. The company conducted interviews with applicant students. One student (*Student 1*) was enrolled. The study model spans four years and is structured as follows:

- First, second and third semester: at university
- Fourth semester: work at company, self-study for two courses at university
- Fifth semester: at university
- Sixth semester: work at company, self-study for two courses at university
- Seventh semester: work at company, no courses
- Eight semester: at university

Second iteration: 2010

The first student started in January 2010, and the Industry Bachelor Programme was officially initiated. From January to June, *Student 1* worked at the company. At the same time, the university college evaluated and improved upon the initial version of the Industry Bachelor Programme. For example, it turned out the students were not registered for exams, and this practical issue was resolved. From August to December, the second round of recruiting students to the Industry Bachelor Programme was conducted. This included new information meetings, along with advertisement and application deadline of Nov 1st 2010. As last year, the company interviewed students during December. Four students were enrolled.

Third iteration: 2011

A new evaluation was made, which resulted in two actions. First, continuance of the following actions for each time period:

- January – June: the previous cohort started, assessments of previous year
- August – November: Advertisements and information meetings for potential students
- December: Interview and recruitment organised by the company

Second, because the university college now had three classes in various stages of the Industry Bachelor Programme, a need for more control emerged. This resulted in one person being assigned as coordinator for the Industry Bachelor Programme. The coordinator's role was to follow-up the students, and to be a bridge between the students and the administration, and between the student and the company. Another responsibility was to examine and assess the Industry Bachelor Programme, such as how it affected the university college, the students, the lecturers and fellow students who were following the regular bachelor programme of three years. This evaluation was conducted through interviews and surveys. The role also included arranging the information meetings given by the university, and meetings where the companies visited the university and presented themselves to the students. The program coordinator reported to the dean.

Fourth iteration: 2012

As a part of the assessment, we now conducted interviews with the students, including *Student 1* who had started in 2010. In addition, a second company joined the Industry Bachelor Programme.

Fifth iteration: 2013

By company request, E-business students were eligible for applying to the programme.

Sixth iteration 2014 and half of the seventh (2015):

Recruitment continued as previous years. Table 2 sums up the cohorts and the current number of enrolled students.

<i>Cohort</i>	<i>Start of Bachelor (August)</i>	<i>Start of Industry Bachelor (January)</i>	<i>Graduating Bachelor (June)</i>	<i>Number of students enrolled</i>
1	2008	2010	2012	1
2	2009	2011	2013	4
3	2010	2012	2014	2
4	2011	2013	2015	4
5	2012	2014	2016 (Planned)	7
6	2013	2015	2017 (Planned)	5
7	2014	2016 (Planned)	2018 (Planned)	N/A

Table 2: The various cohorts and corresponding number of students

5. FINDINGS

In this chapter, we present our findings from analysing the various sources from table 1. First, we summarise (shown in table 3) and describe the findings from the interviews with the students. The themes and trends are based on the semi-structured interview as found in appendix 1. Then we present the findings from documentation, e-mail correspondence and surveys.

<i>Themes</i>	<i>2012</i>	<i>2013</i>	<i>2014</i>	<i>Trends (2012-2014)</i>	<i>Comment</i>
Career counsellor satisfaction	Satisfied	Satisfied	Career counsellor came late but once appointed they were satisfied	Overall satisfied	Slightly decrease
Relevant task	Satisfied. May take time to get relevant tasks	Same as previous year	Same as previous year	Satisfied. Some students find it takes time to get relevant tasks.	Stable
Workload	8 hours	8 hours	8 hours	The students are content with workload	Stable
Influence on studies	Learn how to be structured. The students miss being in the lectures	Same as previous year	Same as previous year	The students learn how to be structured. The students miss being in the lectures	Stable
Learning process in company	Learn a lot, steep learning curve	Same as previous year	Same as previous year	Same as previous year	Stable
Work satisfaction	Very satisfied	Very satisfied	Very satisfied and emotionally attached	Slight increase in satisfaction	Slight increase
Suggested improvements from the university college	Practical problems	Practical problems	Students miss contact with fellow students and teachers	The university college solves the problems as they come along	Slight decrease in problems
Miscellaneous	Students do not regret joining the programme	Same as previous year	Same as previous year	Students do not regret choice of joining the programme	Stable

Table 3: Summing up themes and trends from the interviews with the students

Themes 2012

We had our first interviews in spring of 2012 where we interviewed six Industry Bachelor students that were in three different academic years of their Industry Bachelor. We found the students to be overall satisfied with their work environment although there were some administrative challenges, in regards to the university college, which they gave negative feedback on. However, the students had in the end a high degree of job satisfaction and they appreciated being regarded as regular employees, i.e. they did not get different treatment for being students.

The students were also very satisfied with how much they were learning although the learning curve was very steep in the beginning. Workload varied, sometimes they had to work overtime, but this was OK for them as long as the projects they were working on were relevant to what they expected to work with after finishing their studies. Some of the students said that they felt a responsibility in notifying the company both when they had too much or too little to do; in other words, they acknowledged that it was their own responsibility to make sure they had the right amount of tasks.

Some of the students thought that not all of the assignments were relevant, one example being different types of testing when the students were rather expecting more programming related tasks. However, when the students expressed to the company that they would like a different type of tasks the company was very flexible in assigning the students new tasks or projects.

Regarding learning it was clear that the students felt they learned both soft and hard skills in the company, amongst other how to participate in projects and handling different technologies. The students had to learn the project methods and routines of the company. Although they had learned about the same project methods at the university college, they now acted out these in real life. In addition, the students said that they gained knowledge beyond the curriculum, for example, how to work in a more structured way and managing their time.

An important part of the Industry Bachelor Programme is the counselling the students receive at the company. The students get a career counsellor who gives advice on their progression at work. The students were satisfied with the career counsellor that they all received. They also appreciated that this counsellor did not monitor them, but if they asked for help, they would get this quickly. We identified through the interviews that the career counsellor's other roles varied, for example, some of the students had a career counsellor that also was their project leader.

In regards to the administrative challenges mentioned above in the first paragraph, there was a clear discontent. The reasons was amongst other (i) due to errors in the registration of students to courses that they were taking besides the Industry Bachelor, and (ii) problems with registration to exams and some errors concerning the financial assistance due to how much the students were earning and how many study points/credits (ECTS) they were obtaining each semester.

In addition, the students were somewhat dissatisfied with the amount of communication with the university college. They requested the university college to follow-up the Industry Bachelor students more frequently. A few of the students also were somewhat worried about falling out of the system, such as not being able to find co-students to work with in projects when they returned to the lectures. A couple of students went as far as saying that the marketing of Industry Bachelor was a bit misleading in the sense that they had gotten the impression that they would earn much money. Our data material does not reveal the salary that students were receiving, nor did we have access to the employment contract, therefore we could not pursue this finding further.

Themes 2013

In spring of 2013, we did the second round of interviews and spoke with four students. They were highly content both in the sense of being included in social arrangements, and working in projects which included communicating, planning, and conducting research on new and unfamiliar technologies. The students felt that they were equal to the regular employees.

Like the previous year, the students informed that they experienced a steep learning curve when beginning in the company. However, they were overall content with the amount of hours they worked although how much they worked varied from student to student. Regardless of the workload being high or low, the students realized that it was their responsibility to notify the company if they had too many or too few tasks.

Regarding their learning process, they learned to be more structured and disciplined and have more than one thought in the head at the time. The fact that the students were working on something which had consequences and an actual impact made an impression on the students. In addition, they received good counselling from both their career counsellor and their co project members, i.e. staff that they worked with.

On the negative side, the students remarked that it took some time to get a relevant assignment. They also, as last year, mentioned that they fell somewhat out of the academic environment. They required more information from the university college about what to do with both their courses and regarding the financial aid system.

Themes 2014

In the third round of interviews in spring 2014, we interviewed ten Industry Bachelor students. Again, the students reported that they were very satisfied with both the work environment and the social arrangements. They seemed to be emotionally attached to their colleagues, who they saw as very competent people that they could ask for help at any time. Their career counsellor functioned as mentor but also they had many other people in the company that could serve the same function. They reported that they were given a career counsellor a bit late this year; however, this was less problematic since they had so many other people to ask for guidance.

The learning curve was still steep in the beginning, but it eventually stabilized. They were able to handle the workload (which they saw as moderate) and, at the same time, they felt they learned very much. In addition, they were very content with flexible work hours. Their workload at the company seemed to have either none or slightly negative effect on their academic work, although it had a positive effect on their student projects. Some of the students informed about receiving non-relevant tasks, and once again testing was mentioned, however they understood that they could express their concerns and that the company would listen. Overall, the majority perceived the tasks assigned to them to be relevant.

Trends 2012-2014

From table 3 we see that most of the trends remain stable. We find that the student's satisfaction with the career counsellor, appointed by the company, is overall good, with a slight dip in 2014 because many students had to wait before appointed one. Regarding getting relevant tasks, most students are also satisfied, however they also experience that it takes some time. The workload remains stable over the three years (eight hours) of which students find acceptable. Related to this theme and trend it is important to note that the students have to read and prepare for a reduced number of subjects in addition to their workload.

What we find (positively) surprising is the work satisfaction. The majority of the students reply "*very satisfied*" during all three years, and in 2014, some students even added that they "*feel like a family*" in regards to their colleagues. As such, we sense a slight increase in the students' work satisfaction.

The students were asked to suggest improvements from the university college's part. In 2012, some students complained about practical problems such as not being registered for the regular subjects that they had to study for on their own. This problem was mainly due to the administration's IT system, which did not have a category for Industry Bachelor. However, these practical problems were solved before the next year and there is a slight decrease in such problems.

When it comes to how the Industry Bachelor Programme influences their studies (remember that the students have to follow some subjects in parallel and that they come back as full time students after half a year) the majority of the students inform that they learn how to be more structured, both when they work and when they study. However, they state that they miss being present in the lectures. This trend remains stable, just as their learning process in the company. The learning curve remains equally steep.

Finally, when we asked the students if they had any additional comments, the reoccurring answer was that they did not regret having joined the Industry Bachelor Programme, and this trend remains stable.

Findings from documentation, e-mail correspondence and survey with lecturers and regular students

We also analysed the transcripts of grades because we were curious whether the Industry Bachelor were obtaining better grades better than the regular students were. We compared the Industry Bachelor students' grades with the regular bachelor students in two subjects: programming (first and second semester) and advanced programming (third and fourth semester). Our findings were that the Industry Bachelor students

did indeed have better grades, typically A or B, in these two courses. However, we believe this is because good grades (A or B) are crucial for getting a contract in the first place. In this regard, we drew on e-mail correspondence with the companies, who informed that they did not perceive grade C to be good enough when some students applied. From the interviews, we noted that only one student said that joining the programme had lowered her/his average grade by one whole grade. The other students did not think it influenced their grades. We also sent a survey with three simple questions (see appendix 2) to the lectures of programming and the regular students in the classroom. The response rate was rather poor, only six regular students and four lecturers chose to answer. The findings from the survey was that all of the six students had heard of the Industry Bachelor Programme, but only half of the lecturers (2 of 4) even knew it was an option. The answers to whether the Industry Bachelor students excelled or contributed to learning in the classroom or during exercises was a unison negative.

6. DISCUSSION

One of the main goals of introducing the Industry Bachelor was to give students the possibility of getting practical experience in their fields of study through working in a company. The expected outcome was to improve the student's employability and to provide a closer link between academia and industry. The interviews with the students show an overall high degree of satisfaction in regards to what they learn through working on real live projects in a company. In particular, the majority is satisfied with the assigned tasks, workload, and social environment. The overall impression is that they do not regret choosing Industry Bachelor, rather they see it as a place where they have acquired knowledge outside the curriculum, and that it prepares them for working life.

In comparison to the infrastructure presented and Pauling and Komisarczuk (2007, p. 126) we have a similar structure with the student, company and schools administration and actors. Where we differ is on the fourth actor where they have identified Careers Service that consists of resources to help develop student skills in finding employment. In our case, the school already has an agreement with a company. Instead, we have another role that is the Industry Bachelor coordinator, which has as his primary role to ensure the information between the student and school administration and between the school and company. The Industry Bachelor coordinator is in that sense the glue between the other actors. Moreover, it is important that this role remains dedicated to ensure that Industry Bachelor runs smoothly.

The company with which we have the Industry Bachelor agreement is large with many resources. We believe this contributes to good routines in regards to both training and how they follow-up the newly employed students. Pauling and Komisarczuk (2006, p. 132) write in their article that roughly half of the students indicated a preference to work within a single large organization, although the other half indicated a preference to work in a number of smaller companies during the internship period. While the majority of the students informed that they "*feel like a family*", a few indicated that the large size of the company made the workplace feel impersonal.

The career counsellor is one of the roles that help the students, and the regular employees, to discuss and plan what they want to work with and what courses may be relevant to take to improve different skills. It is interesting to compare the satisfaction of the Industry Bachelor students to the students' responses in the research of Pauling and Komisarczuk (2007) where there was an expressed dissatisfaction amongst a number of students on how they felt that the employer organization was unprepared. This was especially true in terms of training and work structures. The students in Pauling and Komisarczuk's study had to find companies themselves, which we suspect is the reason for the dissatisfaction.

In our study, we have identified another important cause of satisfaction, which is the highly competent staff the students work with in a team. The students also see them as counsellors, at least until they are able to work more independently. When the student faces a problem that he or she does not solve on their own, the staff will provide tutoring to overcome the challenge.

We found it interesting that many of the students did not perceive some of the assignments, especially testing, as relevant. Many of the students seem to expect to program from day one, which they seem to think is the most relevant task. This is an important discovery of the interviews since testing is actually an

essential part of software development. Consequently, the school should focus more on teaching the students the importance of testing. This is an example of how we can obtain closer ties between academia and industry. For example, this experience report from the Industry Bachelor Programme may influence how and when the school teaches the students the different topics such as testing.

7. CONTRIBUTIONS AND SUGGESTED FURTHER RESEARCH

Our contribution is twofold. First, we try to narrow the gap of missing research from the literature review section.

7.1 Contribution to research

Existing literature focus on employability and our data indicates that the students who embark on the Industry Bachelor Programme are likely to be offered a permanent position in the company. Literature also showed that work practice is beneficial to both students and employers, which our study confirms. However, we found less literature which put the main focus on the student, and in this study we have included the students view as one of the main sources. The same literature also points to the fact that many students only work during summertime, which can be unfortunate due to slower production and staff holiday. Our study proposes a new programme which we call the Industry Bachelor, and, in particular, we have described our experiences based on data from three years in this paper.

7.2 Contribution to practice

Our second contribution is an Industry Bachelor Programme Roadmap, which can be seen as an extension of the framework by Pauling and Komisarczuk (2007), shown in figure 1. The key roles from figure 1 are visualized in figure 3 (the roadmap) and key responsibilities and tasks are indicated. The roadmap can be useful to all parties involved in Industry Bachelor Programme or similar, as it visualises their responsibilities and time-span.

Below follows our proposed roadmap with a set of pre-actions. When contemplating the roadmap we note that the presence of each actor (Industry Bachelor coordinator (“IB-coordinator”), Student, Administration, and Organisation) is almost equal during the school year or one iteration, with a somewhat stronger presence of the Industry Bachelor coordinator as discussed in Chapter 6. Moreover, the roadmap excludes aspects or actors such as financial issues (for example, in Norway there is a state educational loan fund called *Lånekassen*). We chose to exclude this actor because not all students have loans from this loan fund, and because financial issues will differ from country to country. It should be noted that the roadmap illustrates one normal iteration, meaning that the roadmap does not cover unplanned events such as student drop-out, or the company being forced to dismiss students for various reasons.

The roadmap in figure 3 below illustrates the four main actors with boxes, and the arrows between them show the connections and sequence. For example, in September, the Industry Bachelor coordinator is the one who initiates contact with the company. If a school wants to engage in a programme similar to the Industry Bachelor Programme this roadmap hopefully provides the main activities and actors involved, however the roadmap should be customised for each school and incremented after each iteration.

Pre-actions:

- ✓ The university has decided to implement Industry Bachelor as part of study programmes
- ✓ A dedicated coordinator has been appointed
- ✓ Agreement with at least one organisation
- ✓ Industry Bachelor as study programme must be implemented in the university's IT systems

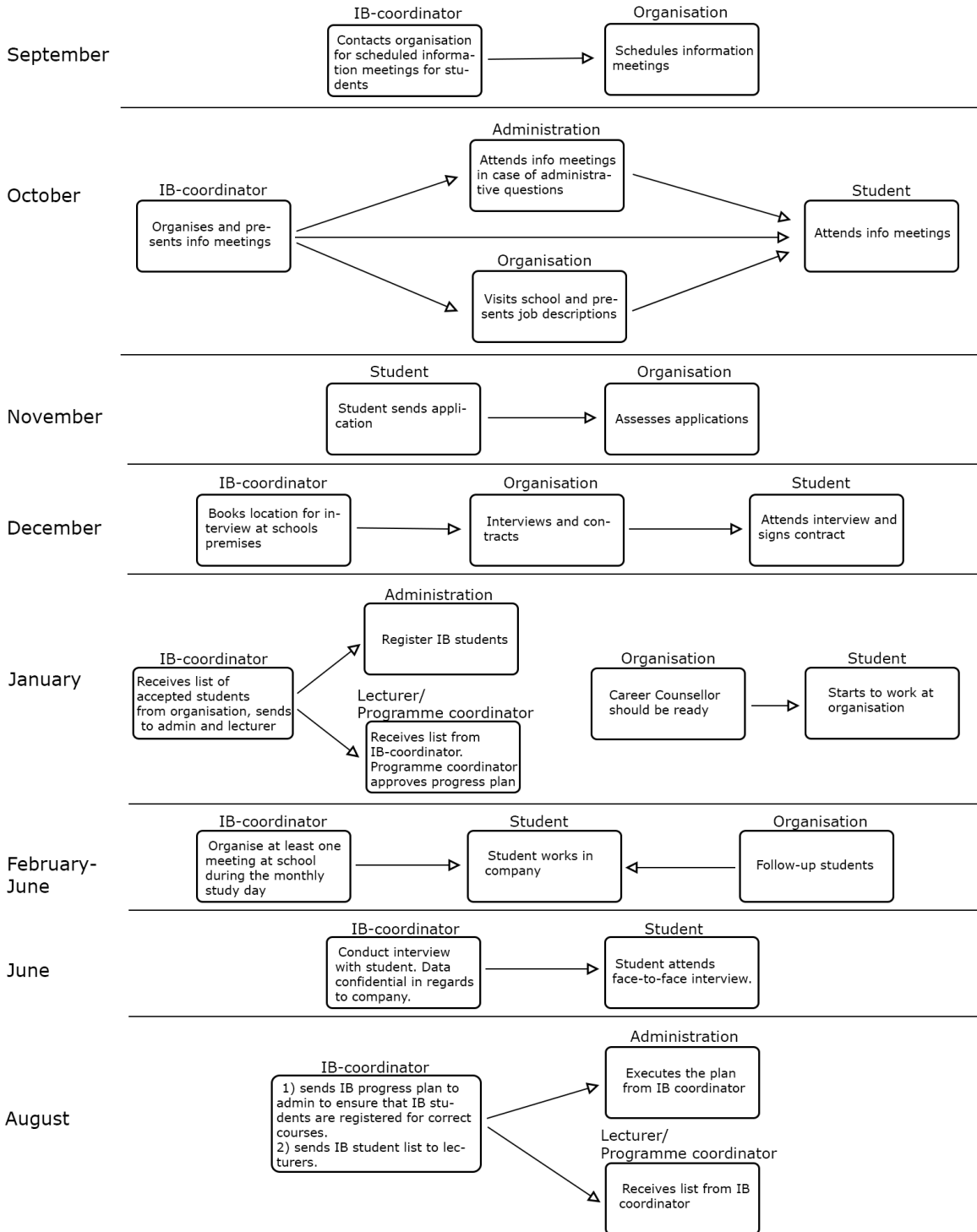


Figure 3: Our proposed roadmap for the Industry Bachelor Programme

7. 3 Limitations and suggested further research

We acknowledge that our study comes with limitations. While we would have liked to know more about the companies' experiences and assessments of the students, we do not have enough data material to investigate this. Further research can include interviews with this actor (the companies). However, since the companies continue to engage students we can assume that they are satisfied. Our data material does not include insights in the student's work conditions. We would have liked to know more about the contract, salary, types of technologies used in the companies, and whether or not the students get benefits such as certifications. Finally, we feel obligated to point to the fact that the companies are allowed to choose the students they want, mainly based on the student's grades. The companies may even refrain from engaging any new students; they are not obligated to accept a given number of students at any time. This means that mainly the A- and B-students have the possibility to join the Industry Bachelor Programme. We hope that future research can investigate this in more detail.

8. CONCLUSION

This paper has investigated: *what can we learn from our experience when introducing an Industry Bachelor Programme in higher education?* Based on data collected over three years we propose a roadmap that builds on existing research. In particular, we find that our study confirms the study by Pauling and Komisarczuk (2007) with four actors and a mutual cooperation and dependency between them, especially between the industry partner and academic educators. Further to this we hope that this study can be useful to universities and companies who are interested in starting a similar programme in order to better prepare the students for working life.

REFERENCES

- ACNielsen. (2000) *Employer Satisfaction with Graduate Skills*. Commonwealth of Australia, Department of Education, Training and Youth Affairs, pp. 1-61
- Baskerville, R. (1999) "Investigating Information Systems with Action Research." *Communications of The Association for Information Systems*, 19(Article 2), pp. 1-32
- Baskerville, R., and Myers, M. D. (2004) "Special issue of Action Research in Information Systems: Making IS Research Relevant to Practice – Foreword." *MIS Quarterly*, 28(3), 329-335
- Coll, R. K., and Eames, C. (2000) "The Role of The Placement Coordinator: An Alternative Model." *Asia-Pacific Journal of Cooperative Education*, pp. 9-14
- Franz, J. M. (2008) "Pedagogical Model of Higher Education/Industry Engagement for Enhancing Employability and Professional Practice." *Presented at Proceedings Work Integrated Learning (WIL): Transforming Futures, Practice...Pedagogy...Partnerships*, Manly, Sidney, Australia, pp. 164-169
- Garcia, M. L., and Bray, O. H. (1998) "Fundamentals of Technology Roadmapping." *Sandia Nat. Labs., Albuquerque, NM, SAND97-0665, Mar. 1998*, pp. 1-33
- Hagan, D. (2004) "Employer Satisfaction with ICT Graduates." *Presented at Australasian Computing Education Conference*, pp. 119-123
- Norton, L. S. (2009) *Action Research in Teaching and Learning: A Practical Guide to Conducting Pedagogical Research in Universities*: Routledge.
- Orrell, J. (2004) "Work-integrated Learning Programmes: Management and Educational Quality." *Presented at Proceedings of the Australian Universities Quality Forum*, pp. 1-5
- Pauling, J. W., and Komisarczuk, P. (2007) "Review of Work Experience in a Bachelor of Information Technology." *Presented at Ninth Australasian Computing Education Conference (ACE2007)*, pp. 125-132
- Robin, G. J. (2011) "Do Companies Look for Education, Certifications or Experience: A Quantitative Analysis." *Presented at SIGMIS-CPR'11*, pp. 1-5
- Tvedt, J. D., Tesoriero, R., and Gary, K. A. (2001) "The Software Factory: Combining Undergraduate Computer Science and Software Engineering Education." *IEEE*, pp. 633-642

APPENDIX 1

Semi-structured interview for the Industry Bachelor students

1) Supervision at <Company Name>

What do you think of the follow-up from your career counsellor at <Company Name>?

2) Tasks

Do you think that your assignments are relevant to your studies and curriculum?

3) Workload

How do you perceive your workload (too large/too small/adequate)?

4) Studies

How do you think being an Industry Bachelor student affects your studies?

5) Learning

What do you think about your learning process as an Industry Bachelor student – do you learn a lot (which is relevant)?

6) Job satisfaction (welfare)

How do you like it as an Industry Bachelor student when you are at Company 1/2?

7) Improvements

Do you think of anything that the university college could have been organised better to improve the Industry Bachelor students?

8) Miscellaneous

Do you want to add anything?

APPENDIX 2

Survey for lecturers (of the Industry Bachelor students), and regular students

1) Are you aware of the Industry Bachelor Programme? (Yes/No)

2) Have you noticed whom (some of) the Industry Bachelor students are? (Yes/No)

3) Do you think that the Industry Bachelor students have contributed in the lecture, exercises, or in other ways, with the knowledge that they have obtained in the organisation? (Yes/No + optional comments)

Vedlegg 10: Artikkel om livekoding

The Live Programming Lecturing Technique

A Study of the Student Experience in Introductory and Advanced Programming Courses

Author name
Affiliation
City, State
country
author@university.com

Author name
Affiliation
City, State
country
author@university.com

ABSTRACT

This paper investigates the topic of teaching programming in higher education. The teaching method often referred to as *live programming* has become a widely applied lecturing strategy for teaching programming subjects in an interactive fashion. Lectures based on live programming normally involve live demonstrations, explanations and interaction with the students. Although this technique seems to be very popular amongst students and instructors, we hypothesise that it also involves potential challenges. In this paper, we investigate the perceived difficulty and promise of following such an approach from a student perspective. We present results from interviews with 1st and 2nd year information technology Bachelor students about their experience with live programming. Our results indicate that students' engagement and desire to learn through active learning techniques still are very much valid also in introductory and advanced programming courses. Furthermore, we also interpret from our findings a suggested model of a repeated cycle of lecture, demo and exercise as highly beneficial to the student learning process.

CCS CONCEPTS

• **Social and professional topics** → **Computing education**; *Information technology education*; • **Applied computing** → **Education**; *Interactive learning environments*;

KEYWORDS

Live programming, Computer Science education, introductory course, advanced course, Bachelor

ACM Reference Format:

Author name and Author name. 2018. The Live Programming Lecturing Technique: A Study of the Student Experience in Introductory and Advanced Programming Courses. In *Proceedings of ACM Special Interest Group on Computer Science Education (SIGCSE'19)*. ACM, New York, NY, USA, Article 4, 6 pages. https://doi.org/10.475/123_4

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SIGCSE'19, Feb-Mar 2019, Minneapolis, Minnesota USA

© 2018 Copyright held by the owner/author(s).

ACM ISBN 123-4567-24-567/08/06.

https://doi.org/10.475/123_4

1 INTRODUCTION

Programming courses constitute an important integral part of most Bachelor programmes in informatics, information technology, computer science or related programmes in higher education throughout the world. Despite an increasing focus on exposing kids and young adults to programming at an early age through initiatives such as the Norwegian project *Kidsa Koder* [8], or the related international initiatives *Computing at School* [1] and *Code Club* [7], programming is still primarily taught in higher education. The nature of programming when coming from a general, non-information technology, background is the concept of the domain-specific approach and vocabulary, put to use through advanced tools and they must become proficient within a short timeframe [11].

A combination of student active learning strategies, different pedagogical models, and combined efforts of lecturers and student assistants, usually encompass the team aimed at solving the task of educating the students. The majority of the students rarely have any large, structured knowledge base regarding programming and a minority of the cohorts are usually self-taught from Internet, YouTube and online courses such as Udemy¹, Udacity² or Lynda³. Similarly, a small portion of the students may have prior higher education or some industry experience which creates a small fundament. However, the lecturers task is to bring this majority of unexperienced, unexposed and heterogeneous group up to a level of basic understanding and mastering of the topic of programming.

A much widespread technique to achieve this goal for the lecturer is to use the concept of *live programming*. Live programming refers to the process of designing and implementing a coding project during a lecture [10]. This can be conducted in several ways. One technique is to create an example application from scratch during the lecture, often with frequent interaction with the students. Another common approach is to prepare the "skeleton" of an application in advance, and fill this template with content, either from copy-paste or from writing line by line throughout the lecture. A third approach is to use collaborative tools to activate the students in participating in developing an application, as showcased by Gonzalez and Lauvas [9].

Earlier studies such as [13, 15] investigated live programming used in introductory courses. However, their focus was primarily on the perspective of the lecturer, and *how* live programming can be used as a lecturing technique. We would like to shift the focus to the perspective of the student. Instead of measuring the learning outcome, we concentrate this research on their experience of this

¹<https://www.udemy.com>

²<https://www.udacity.com>

³<https://www.lynda.com>

widespread technique and try to understand how it is valued by the students. Additionally, we bring a new perspective into this topic by not only investigating introductory programming courses. To the best of our knowledge, there is a lack of research investigating this phenomena in *non introductory* courses, and we seek to ask the same questions to students of an advanced programming course.

Our research question in this article is *from the student's perspective, what are the experienced positive and negative effects on their learning process from live programming lectures?*

The rest of the paper is structured as follows. Firstly, we present selected literature in related work, and then we describe our methodological approach. To follow, our results are presented and we discuss their implications. Finally, we conclude the paper and indicate directions for future research.

2 RELATED WORK

As discussed early on by Bonwell et al. [4], and later in-depth investigated by Weltmann [16], active learning facilitates for larger involvement and increased activity from the students besides just listening to the lecturer. Reading, writing, discussing, solving problems, coding, tutorials, and participating in group exercises, are just a snippet of examples of this educational technique. Engagement in the class room and facilitation of active learning can very well be many-faceted and are broadly explored in literature, but still academics diverge rather than agree on a common definition. Following the thoughts of Prince [12] we find a core characteristic of active learning by seeing it as a meaningful, cognitive challenging task undertaken by students while a lecture is given - often in relation to a proposed task or question from the lecturer.

Active learning is not a replacement or substitute for traditional lectures, but rather a philosophy of conscious choice of strategies for teaching a topic to a group of students. The effect of active learning was investigated by Freeman et al. [5] and their results indicated an improved student performance, unrelated of class size, course type or level, across *science, technology, engineering and mathematics (STEM disciplines)* which were investigated. The overall goal is still to pursue a best possible learning process, and the taught of this achieved by more actively having the students engaging in the learning process is not new, already visited by Bloom [3] in his early works, but then in relation to his taxonomy of learning objectives.

Studying active learning strategies in relation to introductory computer programming are reported from several studies. An interesting perspective is brought up by Seeling and Eickhold [14] in their research investigating three varieties of active learning in an introductory programming course during three semesters. One with traditional lectures, the second with active learning, and the third with workshop- or lab-based sessions. They experienced that a shift from traditional lecturing to active learning produced inconclusive results. Overall exam performance increased, overall course performance where sustained, and course evaluation of mastery declined. Investigating their data, they conclude that the active learning enhanced significantly results of lower performing students, but otherwise were insignificant when compared across all result categories. Further studies will be conducted in a mix between lab-session and active learning lectures. Interestingly, they

do no investigate or ask the students, on how their experience on the subject matter is.

We do seem to find abundance of studies about different perspectives of live programming in introductory programming classes and their positive effect. One such study is from Rubin [13], which presented a study on evaluating the effectiveness of using live programming in programming classes. This research analysed the outcome based on two surveys, and the results indicate that students prefer live programming compared to only being shown static code examples.

Shannon and Summet [15] did a study investigating the impact of live programming in a student-active learning context. Their main motivation was to determine how effective live programming is for specific concepts within programming courses, like if-statements, loops, methods and arrays, and to explore whether this strategy has more impact for certain topics than others, and to see the short-term learning effect. Their concept of live programming referred to students working on programming exercises in class in a flipped-classroom context. To measure the effect of each session the authors used in-class quizzes during the course to determine the short-time learning effect. Based on their study, it is difficult to evaluate the effect of live programming, as this work is really about active learning techniques. Introducing the quizzes may also have had an effect on the learning outcome of the sessions.

Gaspar and Langevin [6] investigated the use of so-called student led live programming, showing the benefits of letting the students engage in active learning during the programming lectures. The authors emphasise the importance of students learning the process and strategies behind programming, and not just the syntax of a program. Their results showed that the student led approach, combined with assignments with more than one possible solution, was successful in the sense that the students were creative in applying a range of different techniques to solve the given problem.

3 METHODOLOGY

This paper builds upon the qualitative tradition, and data have been gathered through the use of semi-structured interviews [2]. An interview guide with open ended questions was developed, and through traversing back and forth in the prepared questions we investigated the conceptual phenomena. Semi-structured interviews were selected since we are studying concepts and trying to understand and identify relevant issues from the student perspective. A total of 10 participants were interviewed over a period of three weeks.

3.1 The Participants

The participants were selected among students in the first and second year of a Bachelor in Information Technology. The first year students were chosen on the basis that they are regularly exposed for live programming in their 7.5 ECTS course of *Object Oriented Programming*. Similarly, the second year students were recruited from a 7.5 ECTS course in *Android programming* that exposes them to live programming on a weekly basis in an advanced course on mobile development. Table 1 gives an overview of the year of study distribution over gender and age-span.

Table 1: Participant distribution

	Students		
	20-25	26-30	> 31
Bachelor 2. semester	-	1 male, 3 female	-
Bachelor 4. semester	2 male	3 male	1 male

3.2 The Interviews

All interviews lasted between 20 and 40 minutes and followed a semi-structured style. The participants were informed of the purpose of the study, were offered to receive the finished article when published, and were given the option to end the interview or withdraw at any time.

Our purpose with the interviews was to get an understanding of the students' individual learning experience related to the live programming in the lectures they had attended. We had prepared an interview guide of open-ended questions, and a conversational approach was followed in the interviews with the students. During the semi-structured interviews questions related to the following topics were asked:

- The amount of live programming used in the programming courses they had attended so far in their education, and in what manner the live programming had been carried out by the instructors.
- How the student chose to engage in the live programming sessions, for instance if they would try to code along with the instructor throughout the lecture, or if they would rather pay attention to the programming without coding themselves.
- Their personal opinion of how effective live programming is related to their learning, with a particular focus on what they regarded potential benefits using this teaching strategies.
- Any potential challenges or issues related to the live programming sessions.
- If the students had any suggestions to possible improvements that could be done to the way it has been carried out, based on their experience.

4 RESULTS

Based on the survey as detailed in section 3, we will further present the results in relation to the topics we used as starting point for our interviews.

4.1 Live programming teaching styles

The student responses indicate that live programming is being taught using several different approaches. While some instructors edit code in prepared code snippets, others perform live programming of smaller and larger applications from scratch. In some cases, the live programming is done for shorter intervals during a lecture containing other lecture components, while others choose to perform the entire lecture in a code-along manner. The difference in teaching approaches may be related to pedagogical strategies of the lecturer and his or her individual preference. Some students

commented on this as being an advantage as they were exposed for several different methods, whereas other found this a distracting factor.

The students reported that there is also a significant difference between lecturers with the respect to the programming speed during the live sessions, and this referred to both the coding and the switching between the files being edited in the lecture. It was also mentioned that there is a lot of variation in how much the instructor explains along the way. One student said that some instructors are good at explaining, while others that are very experienced themselves, seem to forget how many small details that need to be correct for the whole program to work properly.

Some students mentioned that they mostly have experienced that live programming involved small snippets of code, and when larger programs are displayed, it is normally pre-coded, and just modified in the live-session. Other students reported experiences of lecturers spending a large amount of time on live programming and to produce programs of larger size and increased complexity.

The students also explained how different kinds of tools are being used by different lecturers. The *Scrimba* tool [9], which allows for displaying code from the audience on the lecturer's screen, was mentioned as being very useful tool to be exposed for, although a bit scary at first.

4.2 Student activity during lectures

When we asked about how the students choose to participate during the live programming sessions, the students varied in their responses. Many of the students said that they would live code along with the lecturer in the beginning, when the code snippets were small and the programming speed sufficiently low for them to keep up. They also reported that they would often give up, and simply pay attention to what was going on when the programming speed increased, or that they got lost of other reasons.

One student described how he initially tried to code along with the lecturer, but how he ended up simply typing off the lecturer without also perceiving the explanations that were given along with the code bits. After a while the student chose to skip coding along, and instead paying attention to the live programming and the explanation. After the lecture, the student would code the examples from the session on his own.

Another student explained that he sometimes codes along with the lecturer, and other times not, and that this is strongly related to the student's energy level each particular day. If he is tired, he will normally just pay attention, and if the energy level is higher, he programs along with the instructor. The student specified how he felt the live coding is valuable independently of whether he codes along or not since he always has the opportunity to ask questions during the lecture and participate actively that way.

A couple of students chose to be active through the live coding by taking notes throughout the session, and explained how they felt that they got a better overview of what is going on this way. One of them would sometimes write down some of the code by hand, to get a better understanding.

4.3 Benefits

All of the participants reported that they enjoy and prefer live programming as part of the programming lectures. The participants were also quite uniform in their answers with respect to what they like about the live programming. Several mentioned how they appreciate experiencing the process of problem solving, and one specified how this is particularly when it is done from scratch. Further, when comparing to simply being exposed to pre-coded programs on a slide, several students specified how they prefer seeing the program being built, and not only reading text, *"just seeing the final code does not give you any hint of how it was built, or the sequence of how things are done"*.

Another element mentioned by several students was the perceived interactivity in the live programming sessions. Some referred to how it is easier to pay attention because something happens, and how the lecturer can respond to comments and questions, make changes to the code and explain better what is going on. It was also commented on how live programming slows the general lecture speed down, which makes it easier for them to follow and digest the material. One also reported how he enjoys observing an experienced programmer, and how he learns the most from how they perform the entire process from beginning to end.

4.4 Challenges

With respect to what is perceived as drawbacks or challenges with live programming during lectures, most of the students mentioned issues related to the speed of programming. *"If it is done too fast, it is easy to lose track, and difficult to get back in. Looking down at the wrong time can make you fall out of the coding completely."* It was also mentioned how it can be perceived as difficult to ask questions if you are stuck.

While a slow pace was referred to as a good thing by many, one student stated that too long periods of live programming without any disruptions can be very dry or boring. In that context it was advised to use a blend of live coding and other lecturing elements during the session.

Several students also mentioned how good explanations along the way are necessary. *"It is very difficult if one does not understand individual parts of the code, particularly if no explanations are given with respect to why the code is there and what it does. That can be very stressful."* It was also mentioned by students in the first year, that it is important to remember to explain elements that may be obvious to the more experienced programmers.

Context switches were also reported as a significant challenge, for instance when the lecturer moves between different windows and files during the programming. Also, it was mentioned by one student that complex applications are not suited for live programming, as this makes the student dividing the attention between paying attention to understand and typing the code.

Experiencing code that does not compile or work appropriately was also referred to as problematic and stressful. Some students also commented on how they relied on having up-to-date software and hardware to be able to keep up with the lecturer, due to technical problems with running the code otherwise. Some also mentioned how the instructor sometimes struggle with getting things to work properly, and how this could be very time-consuming.

4.5 How to improve the learning experience

When asked how the learning experience could be improved, several students named finding the appropriate programming speed to be the most important. Many of them also expressed a wish for better explanations, like choice of strategy along the way, and other issues related to the actual problem solving. A highlighted idea was to show the context of the live programming, for instance through a switch between diagrams or illustrations and the coding. By doing this, more explanations of trivial matters would be explained several times, and combined with lower speed, these detailed explanations were expected to improve the learning process. One student mentioned how important it was to allocate sufficient time, to avoid time pressure and stress towards the end of the lecture.

The students were a bit torn in regards of the environment for the lecture. Some liked the use of an auditorium, not a classroom, otherwise you could not see properly from the back rows. Another student explained learning was better in settings with fewer students, compared to sitting in a large auditorium. Another pointed towards time of day as important, to be able to concentrate and to be awake to keep up.

5 DISCUSSION

Interesting aspects to discuss and different student strategies during the lecture emerged both from surveying related work and were made concrete through the data collection. From the interviews we learned that students seem to try out a variety of techniques for engagement, collaboration and learning during the lecture. This is in line with the findings from Bonwell et al. [4] and Weltmann [16]. All the candidates interviewed reported they copied (or had tried to copy) the lecturers code simultaneously as it was written. One filled in that this was a fine technique in the introductory course, but as the topics became more advanced this stole the focus. Another technique that were used was to take notes of interesting elements heard or seen during the lecture for then after class returning to these an investigating them with sufficient time and concentration. A majority also mentioned that their primary recipe for success was to just pay attention to what was going on. By doing this it allowed for a bird's-eye view and by this giving an overview of the code sample, and it allowed them to focus their attention to what was told by the lecturer during the walkthrough and coding.

We interviewed 2nd and 4th semester students, but there where not observed any particular differences with the distribution. The natural assumption would be that live programming becomes less and less important as the students mature and typically the examples become more complex. In the first year, students are learning introductory techniques such as for-loops and if-sentences. In later years, the concepts become more important and it is harder to make complete examples becomes harder to programme live. Although the examples are larger, and the complexity increased in the second year, the students still report on similar experiences in both year 1 and 2.

Perceived benefits from live programming where eagerly described by the interviewed students. A common reported characteristic in our findings is that it is a beneficial way to learn. They mention it is easy to ask questions, and these can lead to new examples or demoed changes on the go, as well as interesting diversions.

Together with exemplification, thorough explanations are very well received and reported to be of great use. Further to this, observing also give the students the possibility to see the bigger picture and understand strategies applied to solve a problem. As promoted by Gaspar and Langevin [6], it is more focus on the process, rather than the end product, any with this they learn more about the programming process and utilised strategies. It is also mentioned that through this they pick up a variety of tips and tricks connected to the development tools, shortcuts and other related unrelated information.

Another interesting observation from the interviews is that the audience (students) perceive this form of lecture to be interactive and active. This do not only go for the periods while they code simultaneously, but also when purely observing. The students feel they are being activated. As supported by Freeman et al. [5], active learning strategies support increased learning outcomes. Their results found this to be supported unrelated of class size and course type. Our interviews are inconclusive towards this and suggests it must be further investigated. As presented, some students preferred small classes whereas others found 1:1 learning ideal.

The interviewed students highlight speed as one important challenge. If the lecturer goes too fast it is very difficult to be able to understand the topic, comprehend flow and understand the rationale behind decisions taken during the session. What determines "too fast" is a subjective opinion, but they commonly point out that they at some stage typically drop out. This can be due to a surprising shortcut keyboard command doing a magic trick or being lost between alternating files or unable to understand an explanation. Once lost, the interviewed students point out that it is close to impossible to regain attention and catch up. We do see speed as one of the major hurdles and this finding indicates it potentially a major obstacle. It could be natural for some of the lecturers to go with an approach where the students could control the speed, for instance an example would have been a video the students could pause. However, no such thing was mentioned as being tried out.

Unsatisfactory explanations are also a challenge reported by several. It is energy-consuming to be able to follow and to be able to understand and learn. The lecturer must be able to give a precise, accurate and well-reasoned comment or explanation to the written code. If this fails, then the entered code remains without a contextual connection, and seemingly lost and without any practical opportunity to be remembered. As researched by Gaspar and Langevin [6], the learning process of programming is about more than just syntax and detailed program code. Sometimes it is easy for students to get fixated on the details in the code and difficult to grasp the overall picture of what are doing. Still, it is necessary to see the larger picture to understand what one is doing in the smaller scale.

We also asked if they could share some thoughts on how this lecture technique could be further improved. Several of the interviewees immediately answered that better explanations would be the first thing that comes to mind. The reason for this being that since this is well known to the lectures s/he is not always stating the obvious. These, seemingly, unimportant details are essential to the understanding, particularly for the introductory classes. The second item on the list of improvements relates back to the results reported for challenges, namely speed. The lack of speed leads to the

students being bored and feeling it becomes to elementary, whereas to high speed loose the attention of the lecture room. They point out they have no solution to how to balance speed, but nevertheless it is felt like an active barrier when not matched to their liking.

6 CONCLUSION AND FUTURE WORK

We asked the research question *from the student's perspective, what are the experienced positive and negative effects on their learning process from live programming lectures?* and researched this through gathering data using semi-structured interviews with bachelor students.

Differentiating from previous works, we sat out to investigate this question purely from the students' perspective. Additionally, we added the dimension of studying an advanced programming course in addition to an introductory course. Some of our findings such as students' engagement and desire to learn through active learning techniques are supported by previous findings [5], and the importance of understanding both the syntax and the larger picture at an architectural scale [6]. Other findings such as the huge importance of speed, the re-explanations of already covered material and attention to details were brought forth as new aspects from the students. This indicates that continues shift between lecture, demo and exercise, repeated in small iterations throughout the whole lecture, might be an interesting angle to pursue and measure the effect of.

We sincerely think it is a need for more in depth results connected to the phenomena of live programming from the student perspective, and we especially think advanced courses must be researched to a greater extent. Measured achieved learning outcome and student performance are examples of two topics worthy of further pursuit.

6.1 Limitations

We acknowledge our research has some limitations. Firstly, we have a limited number of participants and they are all from the same institution. Although they have experienced the topic of live programming from multiple lecturers in multiple courses this might still bias their attitude. Secondly, being a small sample of 10 interviewed students, we cannot generalise from this study, but rather investigate, question and highlight topics for future research.

REFERENCES

- [1] Computing at School Initiative. 2018. Computing at School. <http://www.computingatschool.org.uk>. Accessed: 08.08.2018.
- [2] Lioness Ayres. 2008. Semi-structured interview. *The SAGE encyclopedia of qualitative research methods* (2008), 811–813.
- [3] Benjamin S Bloom. 1971. *Taxonomy of Educational Objectives: The Classification of Educational Goals: By a Committee of College and University Examiners*. David McKay.
- [4] Charles C Bonwell and James A Eison. 1991. *Active Learning: Creating Excitement in the Classroom*. 1991 ASHE-ERIC Higher Education Reports. ERIC.
- [5] Scott Freeman, Sarah L Eddy, Miles McDonough, Michelle K Smith, Nnadozie Okoroafor, Hannah Jordt, and Mary Pat Wenderoth. 2014. Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences* 111, 23 (2014), 8410–8415.
- [6] Alessio Gaspar and Sarah Langevin. 2007. Active learning in introductory programming courses through Student-led "live coding" and test-driven pair programming. In *International Conference on Education and Information Systems, Technologies and Applications, Orlando, FL*.
- [7] The Code Club Initiative. 2018. Code Club. <https://www.codeclub.org.uk/>. <https://www.codeclub.org.uk/> Accessed: 08.08.2018.

- [8] Kidsa Koder. 2018. Laer Kidsa Koding. <https://kidsakoder.no>. <https://kidsakoder.no> Accessed: 08.08.2018.
- [9] Per Lauvas and Rolando Gonzalez. 2017. An experience report using Scrimba: An interactive and cooperative web development tool in a blended learning setting. *Proceedings of Norsk Informatikkonferanse (2017)*.
- [10] John Paxton. 2002. Live programming as a lecture technique. *Journal of Computing Sciences in Colleges* 18, 2 (2002), 51–56.
- [11] Leo Porter, Mark Guzdial, Charlie McDowell, and Beth Simon. 2013. Success in introductory programming: what works? *Commun. ACM* 56, 8 (2013), 34–36.
- [12] Michael Prince. 2004. Does active learning work? A review of the research. *Journal of engineering education* 93, 3 (2004), 223–231.
- [13] Marc J Rubin. 2013. The effectiveness of live-coding to teach introductory programming. In *Proceeding of the 44th ACM technical symposium on Computer science education*. ACM, 651–656.
- [14] Patrick Seeling and Jesse Eickholt. 2017. Levels of active learning in programming skill acquisition: From lecture to active learning rooms. In *Frontiers in Education Conference (FIE)*. IEEE, 1–5.
- [15] Amy Shannon and Valerie Summet. 2015. Live coding in introductory computer science courses. *Journal of Computing Sciences in Colleges* 31, 2 (2015), 158–164.
- [16] David Weltman. 2007. *A comparison of traditional and active learning methods: An empirical investigation utilizing a linear mixed model*. The University of Texas at Arlington.

Vedlegg 11: Studieprogrambeskrivelse Bachelor Data Science

Programme description

Bachelor in Data Science

180 credits

2020-2023

The study programme was accredited: 00.00.00
The programme description has been approved by the Education Committee (UU): 00.00.00

Innhold

1. Introduction	3
1.1 Formal requirements	3
2. Learning outcomes	4
3. Programme structure	6
3.1 Academic progression	7
3.2 Modules 1st academic year	7
3.3 Modules 2nd academic year	9
3.4 Modules 3rd academic year	10
3.5 Elective modules	11
3.6 Bachelor's project	11
4. Internationalisation and international student exchange	12
4.1 Internationalisation schemes	12
4.2 Schemes for international student exchange	12
5. Teaching and assessment forms	14
5.1 Learning platform and teaching in practice	14
5.2 Examination and assessment methods	14

1. Introduction

The Bachelor in Data Science is a research-based full-time undergraduate level program covering the academic disciplines of computer science, information science, statistics, computational linguistics, and data ethics.

1.1 Formal requirements

Higher Education Entrance Qualification is required for admission. In addition, sufficient formal mathematic skills needs to be documented (equal to Norwegian Mathematics R1 or S1+S2).

For applicants with foreign education a satisfactory English language test is required.

2. Learning outcomes

All study programmes at Kristiania University College have adopted overarching learning outcomes that each student is expected to have achieved having completed the course. The learning outcomes describe what the student is expected to be able to do as a result of the learning acquired throughout the course. The academic outcome is divided into three categories: Knowledge, Skills and General competence.

Knowledge

The student has:

- broad knowledge of data science fundamentals such as linear algebra, probability & statistics, data structures, data science algorithms, architectures and infrastructures of data science, visual analytics, text analytics, predictive analytics, machine learning, deep learning, data regulation, data security, data privacy, and data ethics
- ability to understand knowledge on theories, frameworks, algorithms, methods, techniques, and tools to analyze, describe, and solve complex and interdisciplinary challenges within design, development, adoption, implementation, and exploitation of both internal and external data pipelines for organizations
- broad knowledge of analyses of various data pipelines (such as text, prediction and visual) with an analytical focus on deriving meaningful facts, actionable insights valuable outcomes, and sustainable impacts to support domain-specific processes and functions
- a broad understanding of and ability to aid technical aspects of data science applications in organizations and society
- knowledge about research, methods, techniques and tools to support data-driven organizational decision making
- knowledge of frameworks for integrating data-driven decision making into organizational practices
- a good understanding of global and local perspective on data pipelines and data science applications.

Skills

The student has:

- the ability to acquire, produce, apply and update new knowledge in data science using visual, text or predictive analytics and to apply results within new application domain areas
- deep insight into use of, comprehensive technical skills in Data Science, in addition to academic skills and the ability to reflect over own practice
- the ability to conceptualize, implement, evaluate and reflect over data-driven architectures and data-driven decision making

- ability and capacity to innovative and independently reflect, and take action using the taught methods, techniques and tools
- achieve writing skills for academic and technical documentation and oral communication and presentation skills

General competence

The student has:

- the ability to work independently and in teams including interdisciplinary groups, diverse professional and academic competences
- the ability to understand and reflect upon ethical considerations of the domain of data science in relation to both work and professional scenarios
- the ability to critically reflect upon cases from local, national, and international environments using written, oral and other related forms of expression
- ability to box-in complex problems and take forward actions in situations with uncertainty of outcome or data in-completeness in order to provide innovative solutions

3. Programme structure

The *Bachelor in Data Science* is a three-year study with a total 180 credits, of which 150 credits comprise compulsory courses, and 30 credits comprise optional (elective courses). The study programme is structured with four courses with 7.5 credits each per semester. The fifth semester there are elective courses and possible exchange visits, and in the sixth semester the student writes the Bachelor's thesis.

The study programme consists of three components: core modules, shared modules with Bachelor IT, elective courses visits and a compulsory Bachelor's thesis.

The study programme is implemented over six semesters, and the structure is as follows:

Bachelor in Data Science				
1st term (autumn)	Databases 1 7.5 credits	Data Ethics and Regulations 7.5 credits	Linear Algebra 7.5 credits	Python Programming 7.5 credits
2nd term (spring)	Information Risk and Security 7.5 credits	Big Data and Cloud Computing 7.5 credits	Probability and Statistics 7.5 credits	Visual Analytics 7.5 credits
3rd term (autumn)	Agile Project 15 credits	Machine Learning 7.5 credits	Data Structures and Algorithms 7.5 credits	Software Design 7.5 credits
4th term (spring)		Deep Learning 7.5 credits	Elective 7.5 credits	Elective 7.5 credits
5th term (autumn)	Text Analytics 7.5 credits	Predictive Analytics 7.5 credits	Elective 7.5 credits	Elective 7.5 credits
6th term (spring)	Research Methods 7.5 credits	Bachelor Thesis 22.5 credits		

Shared module with Bachelor IT	Elective modules	Core modules
-----------------------------------	------------------	-----------------

3.1 Academic progression

The Bachelor in Data Science curriculum consists of mandatory core curriculum courses and elective courses on emerging topics of academic interest and industry relevance. The courses are designed to engage students in a diversity of learning activities in the traditional classroom and blended learning format (lectures, in-class exercises, mandatory assignments, tool-training workshops, individual and group projects, internships, etc.). The program emphasizes socio-technical interactions with other students, real-world datasets, and cutting-edge research prototypes and best-of-breed commercial tools throughout the program.

A foundational first year introduces the underlying concepts of algebra and programming, as well as ethical aspects and important laws and regulations in relation to working with data analyses. Through the second year the students gain in depth knowledge of the first fundamentals of data science from a machine learning and software perspective. Through the first part of year three state of the art analysis paradigms are investigated and will allow the students to solve real world business problems and provide substantial decision support material. The electives in semester 4 and 5 will allow the students to influence the character of their degree, either focusing on breadth or depth. The study finishes with a bachelor project in a company, in which the students get to use all the competence acquired through the study in order to solve a problem or assignment for an external employer.

3.2 Modules 1st academic year

Subjects, each accounting for 7.5 credits	Description
Databases 1	Following the completion of Databases 1, you will be able to explain what a relational database is, what it can be used for and how it differs from other forms of persistent storage. You will be able to model and structure data for a domain. You will be able to create tables, enter different types of data into them, link them together and extract reports using SQL queries. You will also be able to explain and apply the principles of good design (normalisation, key usage).
Data Ethics and Regulations	The first half of the course introduces the fundamental principles of moral philosophy and ethics and their applications to data science processes and outcomes. The second half of the course discusses the various data regulations at the local, regional, national and international levels that students and provides frameworks for incorporating them into data science projects to evaluate ethical implications as well as to ensure legal compliance.
Linear Algebra	This course teaches Linear algebra and other mathematical/statistical foundations of data science. Linear algebra is the mathematical foundations of data science that deals with planes, vectors and vector spaces matrices and linear

	<p>transformation of vectors and matrices, which provides basic foundations for many supervised and unsupervised algorithms in machine learning.</p> <p>The course consists of lectures and hands-on training with open source libraries (such as numpy and sciPy) in Python. Finally, students will develop mathematical and statistical skills that are needed to understand the foundations of various algorithms in the data science.</p>
Python Programming	<p>This course teaches Python programming for data science on how to collect, transform, model, analyze and visualizes broad range of datasets. Students will learn on how to use the Python programming language to work with numerical, string, and more complex data formats, and to perform data analysis with basic data mining and machine algorithms using both supervised and unsupervised approaches.</p> <p>The course will focus on open source technologies and consists of lectures and hands-on training with open source libraries in Python for data mining, machine learning and data visualizations. Finally, students will develop practical programming skills in problem solving by working on real-world datasets as part of their course project.</p>
Information Risk and Security	<p>The aim of this course is to give the students knowledge of business models, ecosystems and technologies used in intelligent systems. Students will acquire knowledge about sensors, steering behavior, networks, infrastructure and applications. The course aims to give skills in evaluation and selection of appropriate technologies/sensors. The students will discuss ethical considerations within the field.</p>
Big Data and Cloud Computing	<p>The goals of this course are two-folds: First it provides knowledge of key concepts, methods, techniques, and tools of big data. Second, it provides an overview of cloud Computing, its enabling technologies, main building blocks and architectures and hands-on experience through projects utilizing public cloud infrastructures (Amazon Web Services (AWS) and Microsoft Azure). As part of these goals, the course will also introduce and cover the topics of cloud infrastructures, virtualization, software defined networks and storage, cloud storage, and distributed programming models such as map-reduce, parallel programming models like Dryad, dockers and containers and so on.</p> <p>The course will focus on open source technologies to the extent possible and consists of lectures and hands-on training with open source libraries and public cloud infrastructures such as AWS and Azure. Finally, students will develop practical programming skills in cloud storage systems and learn to develop different applications in several distributed programming paradigms.</p>

Probability and Statistics	This course introduces theoretical principles of probability and statistics with a focus on practical applications in data science. Topics include but are not limited to: permutations and combinations, frequentist vs. subjectivist probability, parametric vs. non-parametric statistics, probability distributions, Bayesian inference, null hypothesis significance testing, confidence intervals, effect sizes, point estimation, linear regression, multiple regression and logistic regression.
Visual Analytics	This course teaches theoretical principles of and computational techniques for visual analytics of large datasets. The course will enable students to design, develop, and evaluate information dashboards for applications in various domains. The students will be able to reflect upon the different models, theories, and frameworks for visual analytics from a data science perspective.

Table 1. Modules 1. academic year

3.3 Modules 2nd academic year

Subjects, each accounting for 7.5 credits	Description
Agile Project	The course aims to give students a deeper experience in handling a complete systems development project, with a focus on how to use an agile methodology, Scrum. The students are to carry out an extensive project case and will be experienced in using modern techniques and tools.
Machine Learning	The course provides knowledge of the key concepts, techniques and methods related to machine learning. Topics include an understanding of the mathematical basics of data mining and machine learning, linear models for regression such as maximum likelihood, sequential learning, regularized least squares and classification models such as probabilistic generative models, probabilistic discriminative models. Furthermore, the course provides the students with practical hands-on experience on machine learning using open source machine learning libraries such as scikit-learn in Python programming language. After completing the course, the students will be able to apply and use appropriate machine learning techniques in various data science domains.
Data Structures and Algorithms	This course aims to teach the mathematical foundations and computational applications of data structures and algorithms. The first half of the course will cover the topics in and aspects of data structures while the second half of the course will cover the design and analysis of algorithms. The course will focus on the predominant distributed and parallel computing algorithms and how their computing time and memory usage complexity will impact different data science use cases.
Software Design	The course will enable the students to design and further develop larger software systems in line with known techniques

	for modeling, testing and implementation.
Deep Learning	The course provides knowledge of the key concepts, techniques and methods related to deep learning. The candidate gains in depth knowledge of mathematical foundations of deep learning, neural networks and has advanced skills in applying the appropriate tools, techniques and development of these respective areas. Furthermore, the course provides the students with practical hands-on experience on deep learning using open source deep learning libraries in Python programming language. After completing the course, the students will be able to apply and use appropriate deep learning techniques in various data science domains.
Electives	Will be announced on KUC website and LMS.

Table 2. Modules 2. academic year

3.4 Modules 3rd academic year

Subjects, each accounting for 7.5 credits	Description
Text Analytics	The course provides knowledge of the key concepts, techniques and methods in natural language processing to text analytics. The students gain in depth knowledge of natural language processing, and will further apply this to practical scenarios with acquired skills in text classification methods. The course provides students with hands-on experience on text analytics using open source machine learning libraries such as scikit-learn, Natural Language Toolkit (NLTK) in Python programming language. After completing the course, students will be able to apply and use various NLP techniques such as sentiment/emotion analysis opinion mining etc. on text documents/ corpora.
Predictive Analytics	The course provides knowledge of the key concepts, techniques and methods in predictive analytics. This course will cover methods and tools for data pre-processing for forecasting tasks in data science, techniques for selecting well-suited models for analysis, model performance evaluation tools. The course provides students with hands-on experience on predictive analytics using open source statistics tools such as R. After completing the course, students will be able to apply and use various predictive analytics techniques such as regression, time series on numerical datasets.
Research Methods	The course aims to introduce research methods with a focus on methods that are especially relevant for the IT business. The course supports the bachelor's degree project.
Bachelor Thesis	The students will get practical, real-life experience by carrying out a project in a company, establish their own business, or participate in a research project. They will be able to demonstrate broad knowledge of central topics and theories and

	to show skills in using methods, tools and technology.
Electives	Will be announced on KUC website and LMS.

Table 3. Modules 3. academic year

3.5 Elective modules

For the study programme in Bachelor Data Science, students are presented with optional modules in the fourth and fifth term that total 30 credits. Up-to-date information on elective modules can be found on Kristiania University College's website and through the learning platform.

3.6 Bachelor's project

In this final assignment, students will demonstrate that they can immerse themselves in, and apply their knowledge to, key areas of the bachelor's degree, and lie within its field of study. Students will gain professional experience by conducting a project within a business, establishing their own company or participating in a research project. Students will demonstrate a broad knowledge of the discipline's key topics and theories, and skills in applying its methods, tools and technology.

4. Internationalisation and international student exchange

The course has schemes for internationalisation and international student exchanges, according to the Regulations on the Supervision and Control of the Quality of Norwegian Higher Education (Studietilsynsforordningen) of February 2017 (§ 2-2, sections 7 and 8)

The schemes for internationalisation are adapted to the level, scope and uniqueness of the course. The content of schemes for international student exchanges is academically relevant.

4.1 Internationalisation schemes

In this context, internationalisation means that the education offered has an international context and that the students are exposed to a diverse range of perspectives.

The education offered has an international context and exposes the student to a varied perspective on Service Management. This is achieved through the use of international literature and international cases during classes.

The whole program is conducted in English and the teaching staff is a combination of national and international staff. The many cases used in the courses during teaching and exercises are all based in an international context.

For specific internationalisation arrangements, please see the course descriptions for the study programme.

4.2 Schemes for international student exchange

With regard to schemes for international student exchanges, the university college offers the following mobility programmes;

- Nordplus in the Nordic and Baltic countries
- ERASMUS+ in Europa
- 'Study Abroad', for students within and outside Europe

Exchanges are arranged in the fifth term for the Bachelor in Data Science.

Exchange modules from current partner institutions are approved by academic course leaders, for entry into the applicable bachelor's degrees, with a scope of *30 credits*. Information on places of study and exchange modules abroad, for the current course offer and year group, is published on the University College's website and learning platform.

For nomination to a student exchange, certain grades and an application of motivation are required. Other requirements for documentation of creative work/portfolios may be demanded and Kristiania University College may conduct interviews of the applicants for exchanges.

Kristiania University College aims to send well qualified and motivated students to reputable foreign institutions.

Kristiania University College wishes to have a small number of active agreements within prioritised academic and research areas. The International Office is responsible for organising Kristiania University College's exchange programmes.

Exchange schemes are open to students who are studying a degree-awarding course and have completed a minimum of 60 credits at Kristiania University College.

5. Teaching and assessment forms

5.1 Learning platform and teaching in practice

The program is characterised by the use of lectures, exercises and active- and problem-based learning. Problem-based learning means that students are faced with practical and professional challenges. Students must, on their own initiative, acquire necessary information and discuss this together with fellow students and supervisors. The use of lectures and exercises will help provide students with individual professional expertise in all modules. Project-organised teaching means that all terms have interdisciplinary projects. Through the project work, students are also provided with more individual, professional expertise experience in collaboration with others, project management and project organisation. Through the course of study, active learning is also employed as an important part of the didactics model and put emphasis on an environment which blends student activity, with self-study, workshop and instructor led exercises.

5.2 Examination and assessment methods

Throughout the course, several different exam and assessment forms are utilised. The assessment forms are adapted to the learning outcomes of each individual module, whilst simultaneously striving for an appropriate distribution of different exam forms through each term of the course. The assessment forms reflect the desire to promote accountability amongst students. Therefore, several possible assessment forms exist that are adapted to the uniqueness of each module and provide students with different forms of challenges both individually and in groups. Throughout the study program hand ins, written exams, oral exams and group based projects are just examples of what the students will meet.

Some modules may involve mandatory work requirements. Work requirements are requirements the student must fulfil in order to be allowed to take the exam. The right to take the exam requires approved work requirements. The scope and plan for work requirements are specified in the module descriptions. (The assessment expression for work requirements is Approved/Not approved).

For more information about the exam, see Kristiania University College's website.

Research Anchoring for Bachelor in Data Science

At Department of Technology, the R&D activity is organised into three research groups; *Information Systems*, *Applied Computer Science* and *Subject Didactics*. There are also three research labs: *Software Testing*, *Behaviour Lab* and *Mobile Technology Lab*. Our general principle for research activity in the department is to actively support the study programs, and to ensure that the study programs are research-based in their academic anchoring. At the department, three different levels of research-based teaching are practiced:

- The teaching materials used in courses and study programs are based on research articles and textbooks;
- Active research projects and research results are actively communicated as part of the teaching and syllabus;
- An active researcher lectures on own research as part of the lectures and study program.

The Bachelor program in Data Science is a research-based program that covers the academic disciplines of computer science, information science, computational linguistics and data ethics. This links back to the three research groups mentioned above, which actively covers these topics in ongoing research projects.

Examples of relevant research is currently conducted in the Applied Computer Science research group and the Mobile Technology Research Lab. In a joint project with department of health sciences, PHEAD, personal health and electronically assessed data are collected, analysed, presented and interpreted using techniques taught in the bachelor program. A second example that we would like to highlight is an ongoing project between *Center for Business data analytics* at Copenhagen Business School, Denmark, and the Mobile Technology Lab at the department. This project gathers, analyses and visualises research insight from the *Roskilde* music festival in Denmark in addition to the *ByLarm* and *Øya* music festivals in Norway. From both of these projects research publications will be incorporated in the courses of the study program and data sets are available to show case live, realistic data for the students.

The lecturers from the staff allocation document (i.e. Grønli, Fagernes, Vatrapu, Hussain, Mukkamala, Riegler) all do research and publish actively in the areas of the courses in Bachelor Data Science. The new recruitment positions listed, will further strengthen this research environment and these people will be included in the mentioned research groups and labs.

Vedlegg 12: Android eksamensoppgave

Eksamen

Denne delen av eksamen består av to applikasjoner med tilhørende dokumentasjon som skal leveres i henhold til frister oppgitt på WiseFlow.

Rammer:

- Eksamen er individuell – det er ikke mulighet for å levere som gruppe.
- Det forventes at du har lest, satt deg inn i, akseptert og fulgt reglene for intellektuell redelighet. Kilder og referanser brukt i oppgaven skal oppgis TYDELIG.
- Plagiering eller kopiering av en medstudent sin løsning, eller av tilsvarende løsninger funnet f.eks. på internett, fører automatisk til behandling etter normale administrative rutiner for fusk.

Innlevering:

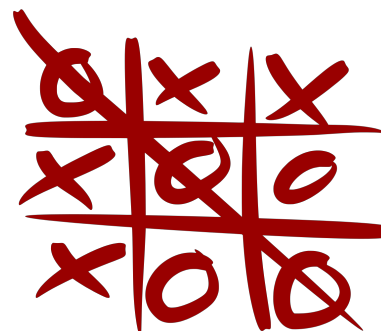
- Du skal levere på WiseFlow.
- Det skal lastes opp én (1) hovedzip-fil som inneholder:
 - Kildekoden til applikasjonene i to (2) **zip-filer**.
 - Applikasjonene sin APK-filer **utenfor** zip-filene.
 - Dokumentasjonen som en **pdf-fil**.
- Legg ved eventuelle instruksjoner som er nødvendig for kjøring / installasjon.

Karaktersetting:

- Applikasjon 1 teller 40 %
- Applikasjon 2 teller 40 %
- Samlet dokumentasjon / refleksjon teller 20 %

Applikasjon 1: Tic Tac Toe

Denne applikasjonen tar for seg et enkelt spill, *Tic Tac Toe* (også kjent som *Tre på rad* eller *Bondesjakk*), og går ut på å plassere X-er og O-er på et 3x3-brett inntil en av spillerne har 3 på rad eller brettet er fullt. Det er 8 mulige rekker med 3 på rad som man kan få; hver av de tre radene, de tre kolonnene og de to diagonalene.



Du skal lage spillet.

Spillet foregår slik:

1. På startskjermen skriver man inn navnet på spillerne og starter spillet.
2. Spillebrettet skal vise hvem som spiller og antall sekunder siden spillet startet.
3. Spiller 1 og 2 plasserer X eller O til noen har tre på rad eller brettet er fullt.
4. Når noen har vunnet, eller brettet er fullt, oppdateres resultathistorikken.
5. Resultathistorikken skal være en liste over navn på spillere som har vunnet, og rangert etter hvor mange ganger hver enkelt person har vunnet. (Når telefonen vinner i 1-player, skal navnet på resultatlisten være 'TTTBot')

Krav til innhold i applikasjonen:

1. Applikasjonen skal ha en Fragment-arkitektur.
2. Applikasjonen skal gjøre bruk av lokal lagring.
3. Applikasjonen skal ha logikk for å kunne spille alene mot telefonen.

Krav/mål:

1. Løsningen skal følge Android sine designprinsipper, samt arkitektur- og kodestandarder.
2. Løsningen skal beskrives i et tekstdokument, hvor du skal diskutere styrker og svakheter, forklare kodefilene som er lagt med og begrunne valgene du har gjort.
3. Arkitektur og flyt av applikasjonen skal modelleres og beskrives.
4. *Screenshots* av alle skjermflater skal legges i rapporten.

Ekstrapoeng:

Det er to muligheter for å skaffe ekstrapoeng i oppgaven. Disse poengene vil telle som bonus og / eller kunne veie opp for eventuelt andre mangler.

1. Betydelig vektlagt estetikk, design og utforming av applikasjonen.
2. Å implementere mobilens AI så godt at sensor ikke slår mobilen i løpet av 10 spill.

Applikasjon 2: <Ditt app-navn her>

Du skal designe og foreslå en valgfri applikasjon som minimum oppfyller kriteriene nedenfor. Du står fritt til å definere ide, omfang og konsept – men det skal være godkjent av foreleser.



Krav til innhold i applikasjonen:

1. Applikasjonen skal ha et tydelig definert konsept og formål
2. Applikasjonen skal ha en Fragment-arkitektur.
3. Applikasjonen skal gjøre bruk av et eksternt API (Firebase oppfyller **ikke** pkt 4.).
4. Applikasjonen skal gjøre bruk av en lokal database.
5. Applikasjonen skal ha multimedia innhold.
6. Applikasjonen skal ha lokasjonsdata og/eller kart.
7. Applikasjonen skal gjøre bruk av sensordata.

Krav/mål:

1. Løsningen skal følge Android sine designprinsipper, samt arkitektur- og kodenstandarder.
2. Løsningen skal beskrives i et tekstdokument, hvor du skal diskutere styrker og svakheter, forklare kodefilene som er lagt med og begrunne valgene du har gjort.
3. Arkitektur og flyt av applikasjonen skal modelleres og beskrives.
4. *Screenshots* av alle skjermflater skal legges i rapporten.

Ekstrapoeng:

Det er tre muligheter for å skaffe ekstrapoeng i oppgaven. Disse poengene vil telle som bonus og / eller kunne veie opp for eventuelt andre mangler i punktene 1-7 i kravlisten over.

1. Betydelig vektlagt estetikk, design og utforming av applikasjonen.
2. Applikasjonen er lansert til Google Play store (evt Beta).
3. Utførlig dokumenterte enhetstester, systemtester og brukertester.

Vedlegg 13: Kronikker fra Morgenbladet



HEI TOR-MORTEN GRØNLI

SØK

01.06-07.06

[AKTUELT](#)

[IDEER](#)

[KULTUR](#)

[BØKER](#)

[PÅFYLL](#)

[PORTALEN](#)

DEBATT / FORSKNING 00:00 - 25. mai 2018

På kompromiss med kvaliteten

Frode Eika Sandnes

Professor, Oslomet

Tor-Morten Grønli

Professor, Westerdals Oslo

ACT

Forkjempere for norsk fagspråk oppfordrer norske forskere til å skrive sin forskning på norsk, og særlig på nynorsk. Dette er et særdeles dårlig råd til unge forskere som ønsker å etablere seg i forskningsfronten.

Forskningsartikler skrevet på norsk blir ikke lest av andre enn norske forskere. Et konkret eksempel: i 30 år har Norsk Informatikkonferanse utgitt over 654 vitenskapelige arbeider både på norsk og engelsk. Av disse ble 42 arbeider skrevet på norsk. Ingen av disse norskspråklige arbeidene er sitert av andre forskere. Med andre ord, ingen andre forskere har bygget videre på disse norskspråklige arbeidene som blir oversett og glemt. Til sammenligning har totalt 1 259 forskere bygget videre på flere av de engelskspråklige bidragene. Telling av forskere som bygger videre på andres arbeid er et av de få konkrete målene vi har på relevans og kvalitet.

Det er faktisk egoistisk snylting å skrive på norsk.

En utenlandsk forsker som ikke kan norsk kan heller ikke lese en forskningsartikkel skrevet på norsk.

Utilgjengeliggjøring av forskning for internasjonale eksperter resulterer i at forskeren ikke får tilstrekkelig gode tilbakemeldinger, eller tilbakemeldinger i det hele tatt. Kvaliteten vil dermed lide over tid og føre til

isolering av forskeren. Det er gjennom fagfelle vurderingsprosessen at forskerne lærer av andre forskere. Tilbakemeldinger fra utlandet er uunnværlig i smale fagfelt der det i beste tilfelle kun er en håndfull eksperter i Norge. Forskere som kun publiserer sin forskning på norsk frarøver seg selv muligheten til korrigerende blick fra omverdenen.

Det er faktisk egoistisk snylting å skrive på norsk fordi forskeren da ikke gir noe tilbake til det internasjonale forskningsmiljøet der forskeren opprinnelig hentet sin inspirasjon og ideer.

Norske forskere har et samfunnsansvar for å formidle sin forskning til allmennheten og da helst på norsk eller nynorsk. Hver gang en forsker publiserer et vitenskapelig arbeid bør forskeren vurdere å formidle innholdet på norsk i tillegg. Forskere bør skrive norske lærebøker og videreutvikle fagspråket i fellesskap med studentene, slik at vi unngår teite oversettelser fra Språkrådet. Men vitenskapelige artikler må skrives på engelsk.

Frode Eika Sandnes**Tor-Morten Grønli**

Takk for at du har kjøpt Morgenbladet. Denne artikkelen er en gratis smakebit for alle, og krever ikke innlogging. [DEL DEN GJERNE!](#)

FORSKNING

SPRÅKDEBATT

SPRÅK

AKADEMIA

Annonse

Vil du være med å forme landets fremtidige kunstmiljø?

Kunsthøgskolen i Oslo søker

Dekan avdeling Kunstakademiet

Søknadsfrist: 10. november 2019

KUNSTHØGSKOLEN I OSLO
OSLO NATIONAL ACADEMY OF THE ARTS

LOGG INN [BESTILL ABONNEMENT](#)



MORGENBLADET

SØK

SISTE UTGAVE

[AKTUELT](#)

[IDEEER](#)

[KULTUR](#)

[BØKER](#)

[PÅFYLL](#)

[PORTALEN](#)



DEBATT / SPRÅK 00:00 - 08. juni 2018

Fagspråk i respirator

Frode Eika Sandnes

Professor, Oslomet

Tor-Morten GrønliProfessor, Westerdals Oslo
ACT

Direktør i Språkrådet Åse Wetås argumenterer i [Morgenbladet 1. juni](#) for at et fullverdig og levende fagspråk er avhengig av symbiosen mellom forskning på den ene siden og rapporter/NOU-er og andre ikke akademiske tekster på den andre. Hun skriver videre at politisk arbeid og samfunnsdebatt krever et norsk fagspråk som er like oppdatert som engelsk.

Det er få sjangre som oppleves mer dødt enn utredninger, NOU-er og standarder. Å bygge fagspråket rundt disse er som å holde fagspråket kunstig i live for at det skal kunne tas frem ved en og annen passende anledning. Levende og lekende norsk fagspråk, derimot, brukes av profesjonsutøverne: ingeniørene, helsearbeiderne, lærerne, og så videre i sitt daglige virke.

STEM-fagene (Science, Technology, Engineering and Mathematics) sine termer blir definert i fellesskap av de internasjonale fagmiljøene. Disse fagfeltene er preget av stor internasjonal konkurranse. Det er stor rift om de akademiske stillingene i Norge og forskere som ønsker å nå opp i konkurransen kan ikke unne seg luksusen å skrive på sine nasjonale språk.

Vi stenger oss inne i en nasjonal boble dersom vi kun bruker norsk.

Vi stenger oss inne i en nasjonal boble dersom vi kun bruker norsk. Innovasjon har ikke tid til å vente på at vi samler oss om et norsk vokabular, for så å oversette det når vi skal snakke med omverdenen. Formidling på norsk og utvikling av norsk fagspråk er viktig, men enda viktigere er det for forskerne å delta på den

internasjonale forskningsarenaen.

Hva er egentlig forskernes rolle? Wetås ser ut til å mene at det også er forskernes jobb å tilrettelegge for politikere og embetsverk. Det er ikke riktig. Forskere skal finne svar på spørsmål uavhengig av den rådende politiske agendaen.

Det er lenge siden politikuttforming var et rent nasjonalt anliggende. EØS-avtalen gjør oss bundet til mye av politikken på europeisk nivå. Det er også en undervurdering av politikere og embetsverk dersom man tror at de ikke kan forholde seg til faglige engelske tekster.

Kanskje også utredninger, NOU-er og politikuttforming blir bedre dersom vi begynner å skrive mer på engelsk?

Frode Eika Sandnes**Tor-Morten Grønli**

Du har nå lest en gratis smakebit fra Morgenbladet. Vil du ha mer godt lesestoff? Vi har inntil 40 prosent rabatt for nye abonnenter. [BLI ABONNENT](#)

SPRÅK

SPRÅKDEBATT

FORSKNING

SPRÅKRÅDET



Annonse

Mest lest

Siste

Vedlegg 14: Anbefaling fra instituttleder

Oslo, 30.09.2019

Anbefalelsesbrev merittert underviser

Jeg anbefaler herved professor Tor-Morten Grønli som merittert underviser. Han har over mange år fremmet instituttets undervisning i henhold til kravene, og har svært gode resultater og refleksjon om denne viktige rollen.

Dette innebærer å ha fokus på studenters læring, en klar kvalitetsutvikling av sitt arbeid over tid, en utforskende tilnærming, en kollegial holdning og praksis samt bransjenærhet og praksisnærhet.

Med vennlig hilsen



Eivind Brevik

*Instituttleder,
Institutt for teknologi
Høgskolen Kristiania*

Mobil: 90513650

Epost: eivind.brevik@kristiania.no