



CODING CLASS

Whitepaper

Kort om Coding Class projektet	2
1. Elevernes forståelse for den digitaliserede verden	3
2. Elevernes interesse for it	5
3. Elevernes evne til at arbejde kreativt og skabende med it	6
4. It, kodning og computationel tænkning ind i skolen	6
Referencer	8
Forfatterne	9

PROFESSIONSHØJSKOLEN

METROPOL



IT-Branchen



Kort om Coding Class projektet

"Kodning handler i høj grad om at forstå verden omkring os fra et nyt perspektiv. At forstå hvordan teknologien virker og påvirker os – og hvordan vi kan påvirke og ændre teknologien til at skabe helt nye løsninger." (Fra IT-Branchens hjemmeside om Coding Class projektet)

Dette whitepaper formidler kort resultaterne af evalueringen og dokumentationen af Coding Class projektet¹. Coding Class projektet blev igangsat i skoleåret 2016/2017 af IT-Branchen i samarbejde med en række medlemsvirksomheder, Københavns kommune, Vejle Kommune, Styrelsen for IT- og Læring (STIL) og med inspiration fra den frivillige forening Coding Pirates². Hovedrapporten og dette whitepaper er skrevet af Docent i digitale læringsressourcer og faglig leder af forsknings- og udviklingsmiljøet Digitalisering i Skolen (DiS), Mikala Hansbøl, fra Institut for Skole og Læring ved professionshøjskolen Metropol; og Lektor i læringsteknologi, interaktionsdesign, design tænkning og design-pædagogik, Stine Ejsing-Duun fra Forskningslab: It og Læringsdesign (ILD-LAB) ved Institut for kommunikation og psykologi, Aalborg Universitet i København. Hovedrapporten og nærværende whitepaper tager afsæt i vores evaluering og dokumentation af Coding Class projektet i perioden november 2016 til maj 2017.

Helt kort fortalt er Coding Class projektet et pilotprojekt, hvor en række skoler i København og Vejle kommuner har igangsat undervisningsaktiviteter med fokus på kodning og programmering i skolen. Evalueringen og dokumentationen af projektet omfatter kvalitative nedslag i udvalgte undervisningsinterventioner i efteråret 2016 og foråret 2017. Evaluering og dokumentation er nærmere beskrevet i hovedrapporten. Coding Class projektet har taget afsæt i Coding Pirates erfaringer med at skabe uformelle læringsrum for børn og unge, hvor fokus er på aktiviteter, der fremmer børn og unges skabende og kreative it-kompetencer. I denne korte sammenfatning har vi fokus på resultaterne af Coding Class projektet, hvor undervisningsinterventionerne havde fire formål:

1. at fremme elevernes forståelse for den digitaliserede verden, der omgiver dem nu og i fremtiden
2. at fremme elevernes interesse for it
3. at igangsætte undervisning, der fremmer elevernes evne til selv at arbejde mere kreativt og skabende med it i grundskolen
4. at sætte fokus på it, kodning og computationel tænkning som vidensdomæne i grundskolen

¹ Coding Class projektets hjemmeside: <https://itb.dk/articles/fremtidens-kompetencer/hvad-er-coding-class>

² Coding Pirates hjemmeside: <https://codingpirates.dk/>

1. Elevernes forståelse for den digitaliserede verden

Af regeringens kortlægning af "Virksomhedernes behov for digitale kompetencer", maj 2016³ fremgår det, at Danmark vil mangle ca. 19.000 it-specialister i 2030. Ifølge Danmarks Vækstråd (2016) vil Danmark, ud over it-specialister, i det hele taget mangle medarbejdere med både almene it-kompetencer og avancerede it-kompetencer. Fremtidens jobmarked spås at være sammensat af langt flere stillinger, hvor den primære jobfunktion ikke er it, men hvor avancerede it-kompetencer alligevel ses som væsentlige for at kunne udføre arbejdet. Funderet i den stigende digitalisering af samfundet og den globale verden, forudser flere internationale rapporter, at der vil udvikle sig et stort "digitalt gab" mellem de borgere, der har kundskaberne til at deltage i et højteknologisk samfund, og dem der ikke har kundskaberne. Med reference til en række rapporter peger Danmarks Vækstråd således på et behov for, at ikke bare grundskolen, men hele uddannelsessystemet, kommer mere aktivt i gang med at uddanne børn og unge, der kan bidrage aktivt og forholde sig kritisk og medskabende til fremtidens samfundsdannelser. Informatik og computationel tænkning kan betragtes som væsentlige elementer af arbejdet med teknologiforståelse.

Leder af It-Vest og tidligere leder for Center for Computational Tænkning ved Aarhus Universitet (AU), Michael Caspersen (2017), fremhæver, at informatik og computationel tænkning skal være en del af almindelsen, fordi "informatikkens grundlæggende principper, tænkemåder, udtryksformer og arbejdsformer" (s. 1) er allestedsnærværende i vores samfund. Forfatteren fremhæver, at informatik kan ses som:

- Det 21. århundredes mikroskop, der giver os radikalt nye erkendelsesmuligheder
- Det 21. århundredes udtryksværktøj, hvorigennem vi kan skabe helt nye udtryk,
- Et domæne, der overskrider etablerede grænser (fx fysiske, geografiske, tidslige, politiske), og giver radikalt nye muligheder for det sociale og fællesskabet.

Ifølge Michael Caspersen kan vi forstå informatik som 21. århundredes matematik og programmering som 21. århundredes skrivning. Derfor er informatik og computationel tænkning et nyt og helt centralt fag og almindennende område i hele uddannelsessystemet. Michael Caspersen fremhæver også, at det er væsentligt at beskæftige sig både med informatik som fag og i fag.

Coding Class projektet har netop sat fokus på elevernes forståelse for denne nye digitaliserede verden. Eleverne har mødt elementer af vidensdomænet informatik, når de via arbejdet med kodning, programmering, Scratch og spiludvikling fx har nedbrudt problemstillinger (dekomposition) og udtrykt enkeltdele af et komplekst hele (abstraktion), bygget programmeringssekvenser op, der eksekveres under rette betingelser (algoritmisk

³ Se VIRKSOMHEDERS BEHOV FOR DIGITALE KOMPETENCER, https://erhvervsstyrelsen.dk/sites/default/files/media/rapport_-_virksomheders_behov_efter_digitale_kompetencer.pdf

tænkning) og testet koden for at rette fejl (debugging).

Når elever via arbejdet med programmering og kodning skal skabe sekvenser, loops, hændelser, parallelitet, betingelser, data og operatorer, så arbejder eleverne med vidensdomænet informatik og computationel tænkning. Alt dette er dog primært på et indirekte niveau, hvis den kreative proces med at skabe spil, træder i forgrunden for eleverne.

Det er nødvendigt at arbejde direkte og tydeligt i undervisningen med elevernes forståelse for og deltagelsesmuligheder i den digitaliserede verden. Forståelse for og deltagelsesmuligheder i den digitaliserede verden handler både om at kunne læse, ændre og nyskabe den verden vi lever i:

"To learn to code students must learn the technicalities of programming language and common algorithms, and the social practices of programming communities... Education activist Paulo Freire once said that "reading the word is reading the world." He was right. Today, reading code is about reading the world. It is needed to understand, change, and remake the digital world in which we live."

(Kafai, 2016, s. 27)

Charoula Angeli med flere (2016) fremhæver, at et væsentligt aspekt af arbejdet med computationel tænkning er at designe et autentisk curriculum, hvor fokus er på at arbejde med meningsfulde og reelle problemer i verden. Computational tænkning er en tilgang til at forstå, hvordan det er muligt at håndtere problemer i verden via computationelle artefakter. I tråd med dette har Alfred Aho fremhævet fem forskellige udbredte argumenter for at inkludere computationel tænkning i skolens curriculum (Committee for the Workshops on Computational Thinking, 2011, s. 36-37):

1. Computational tænkning har indflydelse på stort set ethvert menneskeligt aspekt gennem anvendelse indenfor mange fagområder fx jura, lægevidenskab, arkæologi, journalistik og biologi.
2. Der er store samfundsmæssige og menneskelige risici forbundet ved dårlig computationel tænkning fx softwarefejl. I en verden fyldt med modelleringer og simulationer har vi brug for god software.
3. Med computationel tænkning kan vi udvikle nye og forbedrede måder at skabe, forstå og manipulere repræsentationer. Repræsentationer kan dramatisk ændre på den måde, vi forstår og ser problemer på.
4. Arbejdet med kreative programmeringsprojekter som afsæt for at udvikle computationel tænkning kan motivere elever til at søge videre uddannelse og arbejde med fokus på informatik / datalogi.
5. Relationer mellem informatik og uddannelser kan aktivere undervisning i nye områder, og understøtte arbejdet med nye metoder og nye idéer til problemløsning, der matcher videnspraksisserne, i det samfund som børn og unge skal deltage i.

2. Elevernes interesse for it

Elevernes it-interesse, interesse for (spil-)design og oplevelse af egne faglige og tekniske kompetencer har stor betydning for elevernes deltagelse i og udbytte af Coding Class aktiviteterne. Elever der har været demotiverede, har oplevet store udfordringer, haft moderat til lav it-interesse, og de har været uinteresserede i spil. En væsentlig undervisningsopgave er derfor at tilpasse udfordringsniveau til forskellige elever og - herunder - finde ud af, hvad der indholdsmæssigt kan fange eleverne. Det er generelt set udfordrende for alle eleverne at programmere et spil, hvorfor pædagogisk viden knyttet til computationel tænkning og tilknyttede arbejdspraksisser, der hjælper til at reducere kompleksitet (abstraktion, inkrementel og iterativ udvikling, samt et vokabular for området) er centrale.

Interessen for it stiger især hos eleverne, når det lykkes at skabe interesse- og praksisfællesskaber i klasserne, lærere og elever imellem. Det relationelle aspekt har en væsentlig betydning for elevernes motivation.

En del elever kan have en negativ forventning til fagområdet - og fx kodning, programmering og spiludvikling. Eleverne har mange forskellige grunde, såsom at det er svært, og at it ikke er noget for dem. Overordnet har langt de fleste elever i Coding Class projektet dog været begejstrede for aktiviteterne og flere eleverne peger på, at netop det skabende og kreative arbejde med it, programmering og kodning i undervisningen, har ændret deres it-interesser og -engagement henimod en mere producerende snarere end konsumerende tilgang til it. Dette viser vigtigheden af, at dette arbejde bliver obligatorisk for alle elever, hvis alle elever via skolen skal have adgang til informatik og computationel tænkning som almindelige områder.

Der skal arbejdes for at åbne og fastholde elevernes interesse i it som fag. Det kan være svært for eleverne at se fremtids- og samfundsperspektiverne i arbejdet med fx kodning, programmering, Scratch og spiludvikling. Derfor er det væsentligt at arbejde eksplicit med netop disse perspektiver og herunder transfer til andre fagområder. Undervisning med informatik og computationel tænkning skal vise eleverne, hvordan computationelle redskaber, computationel tænkning og digitaliseringen i samfundet har stor betydning for elevernes fremtidige liv, dannelse og deltagelsesmuligheder.

Mange elever er begejstrede for at få lov til at udfolde sig igennem skabende praksisser, som de fordyber sig i. Dermed giver den skabende og udforskende tilgang til arbejde med computationel tænkning potentiale for at involvere elever, der ikke tænder på teknologi i sig selv, men er begejstrede for de muligheder, de har for at skabe nye udtryk igennem teknologier.

Arbejdet med programmering er krævende og udfordrende for de fleste elever, og derfor hænger interessen for aktiviteterne meget nært sammen med de rammer der skabes

omkring aktiviteterne. Intense forløb med høje krav kan trætte på en måde, der demotiverer eleverne.

3. Elevernes evne til at arbejde kreativt og skabende med it

Arbejdet med elevernes skabende og kreative it-kompetencer, kodning og programmering er på mange måder nyt for både lærere og elever. Eleverne i klasserne spænder ofte over store it-interesse og kompetence forskelle. Samarbejdet i elevgrupperne og elevernes læring hænger derfor også meget nært sammen med den læringskultur, der etableres af lærere og elever - deres evne til at agere lærende fællesskaber. Lærere og elever har forskellige kompetencer at byde ind med og arbejdet med elevernes computationelle tænkning er komplekst og nyt for både elever og lærere. Vidensdomænet lægger op til en omstrukturering af de mere traditionelle skolehierakier, sådan at lærere og elever trækker på læringsstrategier, hvor læreren sætter sig i den lærendes og udforskendes sted sammen med eleverne, og samtidig udfordrer dem til at være ressourcer i undervisningen, der selvstændigt kan opsøge og skabe viden og ressourcer, hvorved elevernes tilgang til at skabe viden og erkende selvstændigt styrkes. Det er ikke lige nemt for lærere og elever at gå ind i disse friere og mere 'implicite' undervisningsrammer, med ændrede forventninger og tilgange til elev- og lærerrollerne i undervisningen. I arbejdet med elevernes computationelle tænkning er det centralt med en undersøgende tilgang til undervisningen frem for at læreren eller eleverne overtager og løser problemer for hinanden. Denne undervisningsform kræver både stort fagligt overskud og mod til at slippe eleverne fri med fokus på at medinddrage elevernes ressourcer i undervisningen.

Et er at igangsætte undervisningsaktiviteter, hvor eleverne udvikler computationelle artefakter, og dermed bringe elevernes kreative og skabende kompetencer med it i fokus. Dette sikrer imidlertid ikke at diskussioner om og udforskning af computationelle artefakter i verden også tildeles plads. Diskussioner om og udforskning af computationelle artefakter i verden og deres betydninger kan styrke relevansen af arbejdet for eleverne. Dette kan endvidere gøres ved at arbejde med at skabe forskellige computationelle artefakter. Her er robotteknologi oplagt i skolen (Helms & Majgaard, 2015), idet robotterne kan agere spejl for vores forståelse af menneske-maskin interaktion, og robotter og automatisering indgår i banebrydende nye teknologier som fx Tesla bilen. Samtidigt berører udviklingen af robotteknologi en række samfundsmæssige områder (sundhed, økonomi, undervisningssystemet, industri mv.).

4. It, kodning og computationel tænkning ind i skolen

Informatik og computationel tænkning er et nyt vidensdomæne, der knytter an til andre

vidensdomæner. Computational tænkning omfatter ifølge Charoula Angeli m.fl.(2016) kernebegreberne: abstraktion, generalisering, dekomposition, algoritmisk tænkning og debugging (fejlsøgning og korrigerende af fejl). J. M. Wing (2006) definerer computationel tænkning som en problemløsende tilgang til verden og dens computerrelaterede problemstillinger, der er særligt knyttet til STEM (Science, Technology, Engineering and Mathematics) fagene. Brennan & Resnick (2012) og Yasmin B. Kafai (2016) udvider Wings forståelse af computationel tænkning til også at involvere 'arts', og dermed betoner disse forskere også de humanistiske og musisk-kreative - de sociale, kreative og skabende - elementer ved vidensdomænet. I forlængelse heraf opsummerer Michael E. Caspersen (2017), at computationel tænkning (informatik) giver nye udtryks-, erkendelsesmuligheder og radikalt nye sociale og fællesskabsmuligheder. Brennan og Resnick (2012) beskriver endvidere, hvordan computationel tænkning omfatter computationelle begreber, design- og deltagelsespraksisser, samt computationelle perspektiver og et skabende og udforskende mindset.

Lærerkompetencer og computationel tænkning i skolen

For at kunne undervise i computationel tænkning skal læreren ifølge Charoula Angeli m.fl. (2016) have Content Knowledge (CK_{CT}) (viden om computationel tænkning og tilknyttede kompetencer); Learner Knowledge (LK_{CT}) (viden om, hvad eleverne kan have svært ved); Pedagogical Knowledge (PK_{CT}) (pædagogisk viden og pædagogiske greb knyttet til computationel tænkning); Technology Knowledge (TK_{CT}) (viden om og færdigheder til at kunne bruge, udvikle og tilpasse teknologier til undervisningen); samt Context Knowledge (CX_{CT}) (bevidsthed om værdien af computationel tænkning i skolen, for samfundet og for eleverne). Fokus på lærernes kompetenceudvikling inden for fagområdet er en nødvendighed for at kunne kvalificere arbejdet med aktiviteterne og sikre udbredelse af aktiviteterne i skolehverdagen fremadrettet.

Undervisning med computationel tænkning i skolen

Et væsentligt element i undervisning med computationel tænkning i skolen er lærernes erfaring, mod og tilgang til arbejdet med computere som eksplorative og skabende redskaber i undervisningen og verden. Dette arbejde forudsætter et sprog for, hvad computationel tænkning, programmering og design-baserede læreprocesser er. Arbejdet med fx spiludvikling, kodning og programmering med Scratch sikrer ikke i sig selv, at informatik og computationel tænkning som indholds- og læringsdimension fremtræder for eleverne. Elevernes computationelle tænkning udvikles via særlige eksplorative pædagogisk-didaktiske greb, der tydeliggør arbejdet med vidensdomænet for eleverne i undervisningen. Fokus i Coding Class projektet har været på elevernes kompetencer til at arbejde designorienteret med spiludvikling via kodning og programmering i Scratch. Lærernes kompetencer til at knytte informatik og computationel tænkning (dvs. computationelle begreber, design- og deltagelsespraksisser, samt computationelle perspektiver og et skabende og udforskende mindset) an til de allerede eksisterende fag og faglige aktiviteter i skolen skal udvikles.

Coding Class projektet har sat fokus på it som fag. Der er et potentiale for at videreudvikle koblingen mellem computationel tænkning og de eksisterende fag ved eksplicit involvering af

og tilknytning til disse fagligheder. Dette skal dog vægtes i forhold til, at der også er behov for at skabe et grundlag for at informatik og computationel tænkning etableres som sit eget vidensdomæne.

Igangsætte undervisning - skabe pædagogisk viden

Arbejdet med kodning, programmering og spiludvikling i undervisningen er svært tilgængeligt for de fleste lærere. Det har vist sig at være effektivt som opstart at køre et enkelt grundforløb (Musespillet⁴) med eleverne, men også lærerne har kunnet lukrere på dette forløb. Forløbet er designet så det reducerer kompleksiteten: Alle elever arbejder med samme udfordring, spillet er enkelt og viser, frem for at gennemgå, hvad man overordnet kan i det anvendte program. Forløbet kan være en god begyndelse, men er dog langt fra nok som introduktion til arbejdet med kodning, programmering, informatik og computationel tænkning i grundskolen. Endvidere knytter forløbet ikke an til fag, og forudsætter viden om spil som genre og spildesign.

Det kan være en god strategi for lærerne at indgå i et lærende fællesskab med eleverne, for derigennem i fællesskab at blive mere teknologividen og -erfarne. Den påkrævede viden om lærernes håndtering af læringsudfordringer; om pædagogiske greb og kontekstviden skal imidlertid understøttes på anden vis.

I Coding Class projektet har Coding Class instruktørerne været andre professionelle, der har deltaget i og bidraget til at udvikle skolehverdagen med fokus på at understøtte elevernes kreative og skabende it-kompetencer. Coding Class instruktørerne demonstrerer en eksplorativ tilgang til vejledning af eleverne, der støtter op omkring elevernes egne undersøgelsesprocesser, en tilgang som nogle lærere, men langt fra alle lærere, allerede anvender. Denne tilgang giver eleverne selvtillid og egen drift i undervisningen, hvilket igen skaber et godt læringsmiljø, hvilket er centralt, idet eleverne udfordres af aktiviteterne.

Fælles for de mere erfarne lærere og Coding Class instruktørerne er endvidere at undervisningen er mere problem- og interesseorienteret end målorienteret. Når skoler arbejder med kodning, programmering, informatik og computationel tænkning, er det væsentligt at have særligt fokus på lærernes kompetenceudvikling af eksplorative videnspraksisser i lærende fællesskaber med eleverne i undervisningen.

Referencer

Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 Computational Thinking Curriculum Framework: Implications for Teacher Knowledge. *Educational Technology & Society*, 19 (3), 47–57.

Brennan, K. and Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In: *Proceedings of the 2012 annual meeting of the*

⁴ Introduktion til Scratch og Musespillet, som blev brugt i Coding Class, ved Martin Exner: <https://www.youtube.com/watch?v=1-Jb8fKMalc>

American Educational Research Association (AERA), Vancouver, Canada.

Caspersen, M. E. (2017 forventet): 4.15 Informatik og Computational Thinking. I: Gymnasiepædagogik - En grundbog. Hans Reitzels Forlag. København.

Committee for the Workshops on Computational Thinking. (2011). Report of a Workshop on the Pedagogical Aspects of Computational Thinking. National Research Council. Washington, D.C.: National Academies Press.

Danmarks Vækstråd (2016): Rapport om kvalificeret arbejdskraft. København.

Helms, N. H., & Majgaard, G. (2016). Tema 1: Robotter i skolen. *Tidsskriftet Læring og Medier (LOM)*, 8(14).

Kafai, Y. B. (2016). From computational thinking to computational participation in K--12 education. *Communications of the ACM*, 59(8), 26-27.

Wing, J. M. (2006). Computational thinking. Viewpoint. In: *Communications of the ACM*. March 2006/Vol. 49, No. 3. S. 33-35.

Forfatterne



Mikala Hansbøl

Mikala er ansat som Docent og forskningskoordinator af FoU-miljøet Digitalisering i Skolen (DiS) ved Institut for Skole og Læring (SoL) på Metropol. DiS rummer ca. 30 undervisere fra læreruddannelsen. Forskning og udvikling i DiS har fokus på samspillet mellem it, læring, fag, kreativitet, didaktik, teknologiforståelse og digital dannelse/medborgerskab. Herunder kompetenceudvikling, spredning af it-didaktiske innovationer, og samspillet mellem it i læreruddannelsen, lærerfaglig teknologiforståelse og it i folkeskolen.

Mikala har en ph.d. i uddannelsesforskning, og har beskæftiget sig med forskning i og udvikling af relationer mellem it, læring og uddannelse i ca. 20 år. Mikala er medlem af Undervisningsministeriets Rådgivningsgruppe for teknologi i undervisningen.



Stine Ejsing-Duun

Stine Ejsing-Duun er Lektor på Aalborg Universitet København ved Institut for Kommunikation og Psykologi, K-ILD, ILD-LAB. Hun undersøger, hvordan (teknologiske) designprocesser og design tænkning kan anvendes som undersøgelsesformer, der hjælper os med at skabe en foretrukken fremtid. Hendes ambition er at beskrive, hvordan teknologier giver os mulighed for at overskride os selv. Hendes forskning har på forskellige områder været forbundet med at være kreativ igennem spillende og legende processer med teknologi.

Stine har deltaget en række projekter med teknologi, leg og læring som omdrejningspunkt de seneste 10 år. Blandt andet har Stine været en del af det to-årige forskningsprojekt Elevernes Egenproduktion og Elevlæring (et demonstrationsskoleprojekt), hvor digital produktion, herunder programmering var omdrejningspunkt for en række interventioner.

Stine Ejsing-Duun har en ph.d.-grad i HCI (Human Computer Interaction) fra DPU/AU. Projektet handlede om (interaktions-)design af lokationsbaserede spil.