



Karlstad Business School
Handelshögskolan vid Karlstads universitet

Jonathan Axelsson

Den digitala skolan

Fyra lärares uppfattningar kring programmering i
de tidiga skolåren

The digital school

Four teachers' perceptions about programming in the early school
years

Informatik

C-uppsats

Termin: VT-2018

Handledare: Katarina Groth Jansson

Karlstad Business School
Karlstad University SE-651 88 Karlstad Sweden
Phone: +46 54 700 10 00 Fax: +46 54 700 14 97
E-mail: handels@kau.se www.hhk.kau.se

Abstract

Syftet med studien har varit att exemplifiera faktorer som kan påverka hur lärare ser på programmering. För att få en djupare förståelse samt bidra med egna förförståelser har jag använt mig av kvalitativa intervjuer från ett hermeneutiskt perspektiv. Tätt ihop har jag även utgått från fenomenografin som ämnar att tolka olika människors uppfattning kring ett och samma objekt för att kunna belysa faktorer från olika synsätt.

Under analysen stod det klart att samtliga deltagare anser att elever bör lära sig programmering för kunna bidra och samverka i samhället. Det fanns även ett samband mellan inställningen till programmering och hur lärarna tycks betrakta sin egna kompetens och undervisningen i sig, där de som är mer positiva till programmering är benägna att prata i positiva termer om sin egna kompetens samt att utforska olika sätt att lära ut programmering. Avslutningsvis visar även studien på att oavsett inställning eller hur de ser på sin egna kompetens så anser samtliga att de programmeringsverktyg som används är roliga att använda och alla elever ges chansen att lära sig.

Nyckelord: *programmering, fenomenografi, lärares uppfattning, Blue-Bot, tidiga skolåren*

Innehållsförteckning

1 Introduktion	6
1.1 Bakgrund	7
1.1.1 Vad var ITiS?	7
1.1.2 Varför mer digitalisering?	7
1.1.3 Skolverkets rapport	8
1.2 Syfte	8
1.3 Problemformulering	8
1.4 Målgrupper	9
1.5 Metod	10
1.5.1 Val av metod	10
1.5.2 Urval	10
1.5.3 Datainsamling	12
1.5.4 Utformning av intervjuguide	12
1.5.5 Etiska överväganden	13
1.5.6 Analysmetod	13
2 Teori	16
2.1 Programmering, mer än bara kod?	16
2.2 Datalogiskt tänkande	16
2.3 Transfereffekt	18
2.4 Kompetensbrist hos lärare	18
2.5 Programmering bra för unga, demokratiskt samhälle	19
2.6 Analysmodell	19
3 Resultat: preliminär analys	21
3.1 Datalogiskt tänkande	22
3.2 Demokratiskt samhälle	23
3.3 Digital framtid	25
3.4 Kompetens	25
3.5 Programmering	26
3.6 Resurser	29
3.7 Transfereffekt	30
4 Analys	31
4.1 Programmering, vad menar du?	31
4.2 Jag kan ju faktiskt!	32
4.3 Framtiden är nu	33
4.4 Det sprider sig	34
4.5 Sammanfattning	35
5 Diskussion	37
5.1 Avslutande diskussion	37
5.2 Kunskapsbidrag	38
5.3 Fortsatt forskning	39
Omnämmande	40
Referenslista	41
Bilagor	44

Tabellförteckning

Tabell 1: Exempel över analysprocessen	14
Tabell 2: Redovisning av deltagare i studien med bakgrundsinformation	21

Figurförteckning

Figur 1: Faktorer som kan påverka uppfattningen av programmering	20
Figur 2: Resultatet av analysen i faktorer	36

1 Introduktion

Datorn och internet har gått från att användas av en liten grupp människor specialiserade på ett fåtal saker till att vara något som används dagligen; nästan hälften av världens befolkning har tillgång till internet - till skillnad från mindre än 1% i mitten på 90-talet (Internet live stats 2018). Datorn i sig har även den gått från att vara en stor box till att kunna få plats i fickan, vilket medför en lättillgänglig åtkomst till internet utan dess like. Vi har möjlighet att föra över valuta genom ett par knappklick och till och med legitimera oss genom digitala medel. De flesta av oss använder olika datorsystem vare sig vi vet om det eller inte, allt ifrån när vi blipper busskortet till att knappa in portkoden för lägenheten. I en ständigt förändrande värld är det viktigt att hålla sig uppdaterad och redan i en motion från 1984 (1984/85:731) går det att läsa "Det är självklart och nödvändigt att skolan skall ge eleverna grunderna på ett område som så genomgripande förändrar vårt samhälle som datoriseringen".

I Sverige har det gjorts flera försök att få in mer teknologi i den svenska skolan genom åren. Utöver den ovan nämnda motionen från 1984 bildades även KK-stiftelsen 1994 som satsade på informationsteknisk (IT)¹ utbildning inom skolan under andra halvan av 90-talet (Stiftelsen för kunskaps- och kompetensutveckling 2017). Men den största satsningen är nog IT i Skolan, eller kort ITiS, som var igång mellan 1999 och 2002. Det var en miljardsatsning på att få in mer IT inom skolan, både kompetens och verktyg, där ungefär halva Sveriges lärarkår arbetade i olika arbetslag och fick tillgång till både utbildning och dator. Det finns delade åsikter om hur ITiS egentligen gick, vissa lärare menar att de inte fick någon direkt utbildning för tekniken utan fokus låg mer på det pedagogiska lärandet. Överlag verkar de flesta vara överens om att ITiS ändå har bringat något positivt ur det hela och att IT används mer efter att ITiS tog slut än innan det påbörjades.

Så kom hösten 2015 och skolverket fick i uppdrag att lämna förslag om en ny IT- och digitaliseringsstrategi, där även programmering ska ingå. Nu har regeringen dragit igång en ny satsning och våren 2017 kom beslutet om en nationell digitaliseringsstrategi för skolväsendet där de presenterar tre fokusområden; digital kompetens för alla i skolväsendet, likvärdig tillgång och användning samt forskning och uppföljning kring digitaliseringens möjligheter (Regeringen 2017a). Dessa fokusområden ska uppnås till 2022 men de reviderade styrdokumentet för skolans uppdrag angående digitaliseringen och hur det ska läras ut börjar gälla redan från 1 juli 2018 där vissa skolor redan valt att påbörja ett större fokus på det digitala.

¹ Vidare genom rapporten skrivs förkortningen IT ut när det handlar om informationsteknik och informationsteknologi.

En av de delar som skiljer ITiS från nuvarande satsning är den statliga budgeten som för ITiS var ca 1,5 miljarder kronor och den nuvarande satsningen ser ut att sakna en särskild statlig budget. Det kan givetvis utgöra ett problem då varje kommun har olika förutsättningar i hur digitalt utrustade de redan är och enligt de nya styrdokumenterna ska varje elev få samma förutsättning för att kunna lära sig det digitala. Det betyder att varken ekonomi eller andra bakgrunder ska utgöra något hinder; ett angreppssätt kan vara att ge varje elev en surfplatta, men det är här kommunens budget kan utgöra ett hinder.

Det är även många som befarar att det kommer vara en brist på lärarkompetens när det ska in mer fokus på digitala verktyg och lärande. Programmering kommer att inkorporeras i flera olika ämnen, bland annat matematik där fokus för programmering ligger på problemlösandet. Programmering är så mycket mer än enbart kod; det handlar om just problemlösning, logiskt tänkande, kreativitet och strukturering. Förmågor som förhoppningsvis spiller över i andra delar av lärandet och livet.

1.1 Bakgrund

1.1.1 Vad var ITiS?

ITiS var en av vår tids mest omfattande satsningar på IT inom skolan och pågick mellan 1999 och 2002. Ca 70 000 lärare deltog och de fick alla en varsin dator att disponera (Tebelius, Aderklou & Fritzdorf 2003). En central poäng med ITiS var att öka kompetensen bland lärarna, det gjordes bland annat genom att arbeta tillsammans i olika arbetslag men även genom att ha seminarier med andra arbetslag (ibid.). Utöver kompetensutvecklingen var målet även att tillhandahålla e-postadresser till både lärare och elever.

1.1.2 Varför mer digitalisering?

“Det övergripande målet är att Sverige ska vara bäst i världen på att använda digitaliseringens möjligheter” (Digitaliseringsrådet 2018). Så lyder det övergripande målet för den svenska digitaliseringspolitiken och våren 2017 uttalar sig utbildningsminister Gustav Fridolin i ett pressmeddelande (Regeringen 2017b) att “[...] avsaknaden av ett samlat nationellt ansvar för digitaliseringen av skolan har gjort att skolan vidgat digitala klyftor i stället för att sluta dem”. Det kom i samband med den nya digitaliseringsstrategin för skolväsendet, en strategi som i samma pressmeddelande sammanfattas till:

- att programmering införs som ett tydligt inslag i flera olika ämnen i grundskolan, framför allt i teknik och matematik
- att eleverna blir stärkta i sin källkritiska förmåga

- att eleverna ska kunna lösa problem och omsätta idéer i handling på ett kreativt sätt med användning av digital teknik
- att eleverna ska arbeta med digitala texter, medier och verktyg
- att eleverna ska använda och förstå digitala system och tjänster
- att eleverna ska utveckla en förståelse för digitaliseringens påverkan på individ och samhälle

1.1.3 Skolverkets rapport

Skolverket har i uppdrag att vart tredje år följa upp IT i skolan, både kompetensmässigt och hur det används. Senaste rapporten skrevs 2016 och visar på en ökning av tillgång till dator², inklusive surfplatta och smartmobil, för både personal och elever med upp emot 50% i skolan jämfört med 2012 (Skolverket 2016a). Till skillnad från vad man skulle kunna tro så är det inte matematik som var det största användningsområdet för datorn i klassrummet; enligt undersökningen är det svenska och samhällsorienterade ämnen där datorn används mest.

Trots denna ökning av IT i skolan så upplever inte eleverna att de ökar sin IT-kompetens motsvarande ökningen av användandet. Dock betyder inte detta att den är på en låg nivå; nio av tio högstadiel elever tycker sig vara duktiga på informationssökning och ordbehandlingsprogram (Skolverket 2016a), vilket är likt resultatet för fyra år sedan.

Om vi istället ser till lärarnas IT-kompetens upplever åtta av tio att de har en bra kompetens, men de anser fortfarande att de behöver ett fortsatt kompetensutvecklingsbehov (Skolverket 2016a). Omkring 40% av grundskollärarna anser även att IT-supporten inte är tillräcklig och att krånglande IT-utrustning utgör ett hinder i deras undervisning (ibid.).

1.2 Syfte

Syftet med studien är att exemplifiera vissa faktorer som kan påverka lärares uppfattningar om programmering samt visa hur de faktiskt ser på programmering. Studien ämnar även att sätta den egna kompetensen i relation till uppfattningen för att studera dels hur en självskattad kompetens kan påverka arbetet och dels om kompetensen faktiskt är ett så stort hinder som flertalet aktörer får det att verka.

² Vidare inbegriper dator även surfplatta och smartmobil utöver den klassiska hemdatorn som de flesta är vana vid, om inte annat uttrycks. Detta eftersom det finns en dator i både surfplatta och mobiltelefon och med dagens teknologi är det möjligt att söka, räkna, utforska etc. även med dessa mindre datorer.

1.3 Problemformulering

Ur den inledande texten framgår att det finns en markant skillnad för den nuvarande satsningen och ITiS, nämligen budgeten. Kommunerna förväntas strukturera om sin budget för att kunna uppnå målen som är satta, vilket ger olika förutsättningar då alla kommuner inte är lika investerade inom IT i dagens läge. Inte bara verktygmässigt men även kompetensmässigt kan det skilja sig åt mellan olika lärare.

Efter ITiS fanns det de lärare som hade synpunkter på att de inte fick lära sig vidare mycket om just teknologin utan det var mer tryck på det pedagogiska lärandet; men eftersom teknologin utvecklas snabbt är det risk att det finns mer att lära sig än för 20 år sedan i proportion till vad de förväntas kunna och lära ut till elever. Det är även mer fokus på programmering som tidigare inte har funnits i så stort omfång som nu, även här kan kompetens spela stor roll. Dock är det, som tidigare nämnt, mycket annat än enbart kodning och lärare inom till exempel teknik och matematik har förmodligen med sig en hel del logiskt tänkande och problemlösande. Att lära sig själva kodnings- eller programmeringsbiten kanske då inte upplevs som ett problem när väl arbetet tar vid. Därav är det även av intresse att ta reda på inte bara hur de upplever den egna kompetensen och de stöd som finns att ta del av, men även, enligt dem själva, hur mycket kunskap det krävs för att lära ut.

Utifrån dessa två problem, budget och kompetens, känns det mest relevant att undersöka lärare vid för- och grundskolan där de redan har påbörjat ett mer digitalt arbete och lärande i enighet med de reviderade styrdokumentet. Gymnasieskolan utesluts här på grund av att programmering införs som ett eget ämne, till skillnad från de tidigare skolåren där det ingår i många andra ämnen.

Följaktligen så finns det tre huvudsakliga undersökningsfrågor som jag finner mest relevanta samt intressanta och är följande:

1. Vilka problem upplever lärarna att det finns med programmering i skolan?
2. Vilka möjligheter ser lärarna med programmering i skolan?
3. På vilket sätt uppfattar lärarna den egna kompetensen inom programmering?

1.4 Målgrupper

Tänkta målgrupper för rapporten är de som kan ha nytta av att se ur en lärares perspektiv på den nya satsningen; för att kunna förstå dennes tankar kring vilka möjligheter som finns framöver men även vilka hinder som behöver överkommas för att det ska bli en positiv erfarenhet. Dessa intressenter är då förmodligen inte bara styrande och beslutsfattande inom skolor och kommuner men även andra lärare som kan finna inspiration för sitt eget arbete.

1.5 Metod

1.5.1 Val av metod

Jag har i innevarande studie valt att lägga fokus på en djupare förståelse för enstaka lärares tankar och uppfattningar, att genom ett hermeneutiskt perspektiv försöka tolka och förstå. Patel och Davidson (2011, s.29) menar att den hermeneutiske forskaren använder sig av sin egna förförståelse som en tillgång under arbetet med att tolka och förstå objektet. Där det hermeneutiska perspektivet försöker tolka och förstå, och lärarnas egna uppfattningar är kärnan i arbetet, finns det ett annat perspektiv som nästan går hand i hand; nämligen fenomenografi. Marton (1981) förklarar det som studien av människors upplevelser av diverse fenomen. Marton, som även var den som ledde arbetet i utvecklingen av fenomenografi, skiljer därmed på fenomenologin som studien av fenomen och visar det utmärkande draget i fenomenografi till att studera själva uppfattningarna hos olika människor och inte bara en absolut sanning. Fenomenografin är även en central punkt i Segolssons (2006) verk där han med framgång studerar nio elevers uppfattningar kring programmering, vilket ökade min egna tro på fenomenografi som utgångspunkt för föreliggande studie. Det är dessa två förhållningssätt, hermeneutik och fenomenografi, som har varit centrala i innevarande studie för att verkligen komma underfund med syftet; hur lärarna upplever programmering.

För att samla in huvuddelen av materialet för studien använde jag mig av intervjuer för att kunna vara på plats och föra ett samtal med de enskilda lärarna. Dalen (2015, s.17) menar att forskaren alltid bidrar med egna åsikter och uppfattningar i mötet med intervjupersonen och framhäver vikten av att använda dessa för att öka förståelsen av det intervjupersonens utsagor. Då det är ett till viss del begränsat område som utforskas där huvudfokus ligger på programmering med kompetens, problem och möjligheter kring detta använde jag mig av en kvalitativ och semistrukturerad intervjuform, som Patel och Davidson (2011, s.82) beskriver; “[...] en lista över specifika teman som skall beröras, men intervjupersonen har stor frihet att utforma svaren”. Även Trost (2010, s.71) menar att intervjuaren lätt kan få fram följdfrågor under intervjuens gång, således kännetecknas den kvalitativa intervjun av flexibilitet och öppenhet, att kunna ställa följdfrågor som inte nödvändigtvis var där innan men som blir relevanta av nya upptäckter under intervjuens gång. Det hermeneutiska perspektivet vid kvalitativa intervjuer bygger även på att intervjuaren har en medkänsla för intervjupersonen där förförståelse kan bidra till egna uppfattningar och berika intervjun till den mån att intervjupersonens uppfattningar blir tydligare (Patel & Davidson 2011, s.84-85).

1.5.2 Urval

Först och främst var det skolor som redan påbörjat programmering i undervisningen som kändes relevanta då det är kärnan i ett par av frågeställningarna. Vidare var det

även frågan om vilka lärare, och då jag själv genom egna erfarenheter har märkt att det uppstår fler tankar kring problem och möjligheter när man väl påbörjat något, blev svaret snabbt att det är de som lär ut programmering eller kommit i närmare kontakt med det på ett eller annat sätt. Detta eftersom de förmodligen skulle komma att ha fler tankar och åsikter kring problem och möjligheter. På så sätt blev alla intervjuer likvärdiga i det avseende att de alla har gått från att ha en tidigare uppfattning om programmering till att utveckla en ny förståelse för det. Förmodligen gavs då en mer korrekt återspeglning av deras egenuppfattade kompetens inom området än om de ännu inte hade kommit i kontakt med programmering.

Då det handlar om kvalitativa intervjuer kommer inte generaliserbarheten vara likvärdig den som till exempel en kvantitativ enkätundersökning kan göra, vilket gör det slumpmässiga urvalet mindre viktigt. Jag valde således att ta kontakt med lärare jag redan visste om var aktiva vid för- och grundskolor där de har påbörjat införandet av programmering i undervisningen. Eftersom kommunerna själva står för omstruktureringen för att arbeta enligt den nya satsningen så har jag även valt att involvera lärare från minst två olika kommuner i värmland. Med detta i åtanke har jag även valt att begränsa till enbart kommunala skolor.

För intressantare jämförelser siktade jag på att få med lärare som var aktiva inom skolan vid satsningen ITiS för 20 år sedan då det kan bli fler tankar kring vilka problem/möjligheter de ser framför sig baserat på tidigare erfarenheter. Detta var inte ett lika viktigt krav då programmering inte fanns på bordet vid satsningen ITiS men det är fortfarande ett intressant ämne och jag försökte därmed få med två lärare som var aktiva vid ITiS.

Något som Patel och Davidson (2011, s.120) tar upp är den mängd textmaterial som uppstår på grund av den kvalitativa inriktningen på intervjuerna då de ofta kan pågå en längre stund och en vanlig metod är att transkribera ljudupptagningen i bearbetningsprocessen. Kopplat med det faktum att generaliserbarheten aldrig realistiskt kan uppnås i samma utsträckning som en kvantitativ metod samt den redan vida utsträckningen över kommuner, ålder samt roll, valde jag att begränsa antalet respondenter till fyra.

Slutresultatet blev två fritidspedagoger vid två olika kommuner där de använder programmering, samt en lärare i grundskolan som inte använder programmering i sin egen undervisning men som har kommit i kontakt med det och bedömdes ha nog värdefulla tankar för studiens syfte. Samtliga var aktiva under ITiS men enbart två av dem påverkades av det. Den fjärde deltagaren är markant yngre än de tre första och har befattningen lärarassistent, vilket innebär att det inte ligger ett huvudansvar på denne, men bedöms kunna ha annan värdefull syn på programmering på grund av

generationsskillnaden. En översikt av lärarna presenteras i tabell 1 under resultat (kapitel 3).

1.5.3 Datainsamling

Som tidigare nämnt har den nuvarande satsningen inte ett statligt bidrag och eftersom kommunernas olika engagemang för IT och digitalisering under tidigare år kan ha avsevärd påverkan, så avsåg jag även att ta reda på hur situationen ser ut på de skolor där intervjupersonerna är aktiva för att ha mer bakgrund att stödja analys och slutsats på. Avgränsningen görs här till att mestadels handla om vilket stöd lärarna har erhållit tidigare samt vilket stöd de upplever under den nya satsningen som pågår, i de fall där de själva erfarit att de har haft behov av det. Insamlingen av denna information skedde i samband med de intervjuer som utfördes med lärarna.

Intervjuerna ägde rum hemma hos lärarna för att ha en naturlig och trygg miljö för intervjupersonerna där de kunde känna sig hemmastadda samt för att försöka minimera distraktioner utifrån. Innan varje intervju påbörjades även transkriberingen av den föregående intervjuens ljudupptagning för att säkerställa att alla ämnesområden berördes och intervjuguiden fick möjlighet att justeras. Intervjuguiden ändrades aldrig, dock ställdes olika följdfrågor till olika deltagare baserat på vilka svar de gav, dessa uppkom i stil med Trosts (2010, s.71) tillvägagångssätt där de inte var nedskrivna på förhand. Under intervjuernas gång ställde jag även frågor efter behov för att undvika missförstånd och stämde av vissa delar i efterhand med respondenterna.

1.5.4 Utformning av intervjuguide

För att kunna ta reda på hur lärarna upplever fenomenet programmering och få en faktisk förståelse för vad de tänker, är det viktigt att ställa rätt frågor på rätt sätt. Dalen (2015, s.35) tar upp ett par exempel för hur intervjuaren kan gå tillväga i frågandet, där hon belyser "områdesprincipen" som innebär att frågandet startar i utkanten av de centrala delarna och arbetar sig inåt, för att sedan avsluta i mer generella frågor. Vidare ges även exempel för utformning av frågor där Dalen (ibid.) lyfter fram vikten av att ställa frågor på ett sätt som gör att intervjupersonen verkligen öppnar sig och svarar med egna ord. Detta kan bland annat ske genom att ge påståenden eller forskningsresultat och be intervjupersonerna yttra sig om dessa eller be dem beskriva ett fenomen. Båda dessa sätt använde jag mig av där jag började med med generella frågor kring deras uppfattning och tankar kring programmering för att senare hamna på mer precisa frågor, samt även förklara vissa fenomen. Bland annat fick samtliga deltagare en generell fråga angående vad de tror att programmering kan lära ut, för att sedan få transfereffekten förklarad för sig där jag gav båda sidor av forskningens syn på det och sedan lät intervjupersonerna uttala sig kring programmeringens effekter på nytt.

Trost (2010, s.71) diskuterar intervjuguidens struktur och menar att frågor inte alls är nödvändigt att ha på förhand, istället bör intervjuaren använda sig av ämnesområden som lärs in utantill och under intervjuens gång formulera frågor som faller under ämnesområdena denne går in med. Patel och Davidsson (2011, s.82) menar att det är den mest öppna formen av kvalitativa intervjuer, med minst struktur. Som tidigare nämnt valde jag att förhålla mig till en semistrukturerad intervjuform men höll även Trosts synsätt i åtanke då jag skapade olika ämnesområden med ett par fördefinierade frågor som bör besvaras för att vara säker på att all data som behövs faktiskt insamlades, men även med mycket utrymme för att ställa frågor som uppkommer under intervjuens gång. Den slutgiltiga intervjuguiden innehöll inte några utskrivna ämnesområden (se bilaga 1), för synlighet är de dock även utsatta under ungefärligt ämne under bilaga 2 (ungefärligt eftersom programmering, datalogiskt tänkande samt transfereffekt är så sammanvävda att ämnen mer bestäms utifrån svar snarare än fråga). Frågorna togs fram från den teori som återfinns i kapitel 2, men de var istället sorterade i dels lite oordning i ämnen men även sorterade så att en generell fråga om till exempel utbildning gavs före frågan om huruvida de tyckte sig ha nog med kompetens för undervisningen. Eftersom intervjuer styrs mycket av intervjupersonernas svar gavs dock inte alla frågor i samma sekvens för varje deltagare och vissa av dem svarade på en fråga innan den hann ställas på tänkt ställe.

1.5.5 Etiska överväganden

Intervjupersonerna kontaktades i god tid innan intervjuerna planerades äga rum, de fick även grundläggande information om vad intervjun skulle handla om i form av studiens syfte som ämnar ta reda på deras syn på programmering, att det är konfidentiellt och att de när som helst har rätten att ta tillbaka sitt samtycke samt att intervjun skulle komma att spelas in i form av ljud.

Då det inte finns någon vikt vid att veta vilka lärarna är kommer uppgifterna behandlas konfidentiellt, vilket innebär att namnen ersätts med vanliga svenska förnamn och varken kön eller skolornas namn kommer skrivas ut. Detta för att skydda lärarnas personliga integritet, både psykiskt och fysiskt, i det fall de känner sig oroliga över att prata om känsliga områden. Kön utelämnas då det inte framkom några skillnader kring om det var en kvinnlig eller manlig lärare vid analysen. Uppgifter om vilka årskurser de undervisar kommer redovisas då det för studiens syfte är av intresse att jämföra med vilken kompetens de tror sig behöva.

1.5.6 Analysmetod

Baserat på metoden jag valde, kvalitativa intervjuer, utfördes även analysen genom en kvalitativ innehållsanalys med en delvis induktiv ansats. Den induktiva metoden sker förutsättningslöst och inte baserat på ett kodschema eller liknande som utarbetats i

förväg (Lundman & Graneheim 2012, s.188). Dock är det inte en rent induktiv ansats jag använder mig av, utan den ligger enbart längre åt det induktiva hållet än det deduktiva i den mån att jag har förutbestämda kategorier baserat på den teori och analysmodell som återfinns i teorikapitlet, för att säkerställa att frågeställningarna besvaras. Analysmodellen låg som grund till den preliminära analysen som tog plats innan resultatet analyserades mot teori och studiens syfte. Det är heller inte utan en viss struktur som analysen genomfördes och Lundman och Graneheim (2012, s.189) listar följande begrepp som hör till kvalitativ innehållsanalys; analysenhet, domän, meningsenhet, kondensering, abstraktion, kod, kategori och tema. Dalen (2015, s.78) tar upp en kodningsprocess som till viss del liknar Lundman och Graneheims och menar att det är lätt att fastna i ett summeringsstadium av kategorier istället för att förstå vad det faktiskt handlar om och utforska nya kategorier. Detta togs hänsyn till i analysen i innevarande studie då även nya kategorier gavs chansen att uppstå utöver de förutbestämda.

På grund av den begränsade tiden och omfånget på innevarande studie valde jag att använda mig av Lundman och Graneheims lista av begrepp men modifierade denna efter behov och presenterar de delar jag använde mig av nedan³. Tabell 1 visar ett exempel på hur analysen med dess delar hamnar i ett slutgiltigt skede i analysmetoden.

Namn	Stycke	Nyckelord	Underkategori	Kategori
Erik	Att lära sig om dagens samhälle är bra för barnen	Bra för barnen	Förstå omvärlden	Demokratiskt samhälle
Anna	Det är bra så att dom förstår vad som händer i världen	Förstå världen	Förstå omvärlden	Demokratiskt samhälle
Lisa	Kul att både pojkar och flickor gillar det och det är väl bra	Alla lika	Genusskillnader	Demokratiskt samhälle

Tabell 1. Exempel över analysprocessen, styckena är ej faktiska citat av deltagarna.

Analysenhet i sammanhanget här menar på varje intervju, de fick även ett varsitt digitalt kalkylark tilldelat under bearbetningen av rådatan för att underlätta analysprocessen. Bearbetning av datan fortsatte kontinuerligt efter den första intervjun var genomförd och efter en initial genomläsning av varje intervju definierades domäner, enligt Lundman och Graneheim (2012, s.189) en grov struktur av innehållet. Dessa domäner inkorporerades sedan till kategorier under ett senare stadium. Upprepade genomläsningar

³ Kondensering, att korta ner meningen men bevara det centrala innehållet (Lundman & Graneheim 2012, s.190) utesluts då det inte är något större antal intervjuer att analysera samt för att inte av misstag utesluta någon väsentlig del. Även tema, att mer översiktligt visa på en röd tråd genom flera kategorier, utesluts på grund av ett mindre antal intervjuer där kategorier bedöms ge nog bra översikt.

skedde under analysens alla delar för att säkerställa en korrekt uppfattning av vad intervjupersonerna pratar om.

Efter en grov struktur följde en mer detaljerad indelning där första steget gick ut på att ta varje intervju för sig och finna meningsenheter, enligt Lundman och Graneheim (2012); “[...] ord, meningar och stycken av text som hör ihop genom sitt innehåll och sammanhang” (s.190). Dessa meningsenheter abstraheras och tilldelas nyckelord, en kod, för att kort sammanfatta vad de handlar om. Efter att alla stycken tilldelades nyckelord, och stycken som inte var relevanta sorterades bort, fortsatte analysen med nästa intervju.

När alla intervjuerna fick meningsenheter och koder påbörjades nästa steg i analysen; att finna kategorier. Varje kod sorterades vidare in i kategorier som gav ett större sammanhang tillsammans med besläktade koder och hänger ihop med de frågeställningar och områden som undersöks. Lundman och Graneheim (2012) menar att ingen data som är relevant ska uteslutas eller passa in i mer än en kategori, med fördel kan även underkategorier användas för att skilja på nära besläktade koder. Det var precis det som skedde under detta steg, där även domäner slogs ihop med kategorier och mer specifika underkategorier uppstod. Efter att alla citat var sorterade in i kategorier kopierades all text in i ett nytt kalkylark där alla intervjuer ingick. På så sätt gick det att sortera efter kategorier och få citat från alla intervjupersoner till varje kategori direkt.

Det sista steget i analysen innebar att leta efter mönster; att undersöka hur olika kategorier relaterar till varandra. Det är dessa mönster och sammanhang som är kärnan av resultatet och visar hur lärarna upplever de olika fenomenen samt en djupare förståelse för varför.

2 Teori

Kapitlet börjar med att förklara begreppen programmering, datalogiskt tänkande samt transfereffekt som är tätt sammanvävda men skiljer sig åt. Vidare förklaras två till teorier och slutligen presenteras en analysmodell utifrån teorierna som återfinns här, inklusive en lista för att skilja de tre förstnämnda begreppen åt.

2.1 Programmering, mer än bara kod?

Programmering i vardagligt tal är för många synonymt med att skriva kod, vilket, beroende på vem man frågar, inte är helt fel men heller inte helt korrekt. Flera forskare menar att det råder en otydlighet gällande vad programmering innebär (Mannila, Dagiene, Demo, Grgurina, Mirolo, Rolandsson, & Settle 2014; Kjällander, Åkerfeldt, & Petersen 2016), dock verkar det finnas en relativ samstämmighet bland forskarna själva kring det faktum att programmering är mer än att enbart skriva kod. Skolverket (2016b) menar att programmering utöver kodning även består av “[...] problemformulering, val av lösning, att pröva och ompröva samt att dokumentera [...]”, vilket fortfarande kan kännas som en otillräcklig formulering av vad programmering innebär. Mannila et al. (2014) har i sin rapport försökt definiera datalogiskt tänkande och hur det uppfattas i olika länder och i sin strävan efter att skilja begrepp åt skriver de bland annat att även analys och problembrytning⁴ ingår i programmeringens delar.

När det diskuteras kring programmering i en vidare bemärkelse likt föregående förklaringar menar Kjällander et al. (2016) att det oftast är datalogiskt tänkande som berörs. Var går då gränsen mellan programmering och datalogiskt tänkande? Voogt, Fisser, Good, Mishra och Yadav (2015) försöker i sin studie förklara vad datalogiskt tänkande innebär och menar att många studier om datalogiskt tänkande använder just programmering som kontext vilket gör att de ofta uppfattas som en och samma sak trots att de skiljer sig åt.

2.2 Datalogiskt tänkande

Den förste som tog upp datalogiskt tänkande var Papert (1980) som i sin studie försöker påvisa att barn som programmerar kan lära sig abstrakta kunskaper som kan föras över till andra områden. Wing (2006) pratar om datalogiskt tänkande i sin artikel som något som kan appliceras till alla fält för att lösa problem, dock är det inte begränsat till problemlösning; det är snarare ett sätt att tänka som hon även hävdar är en fundamental färdighet som alla bör kunna. Det som tycks skilja mellan deras förklaringar är hur de ser på programmeringens roll i datalogiskt tänkande där Papert (1980) menar att programmeringen i sig är det som får barnen att tänka mer abstrakt om hur de kan

⁴ Från engelskans “decomposition”, innebär att man bryter ner ett problem i mindre, mer hanterbara delar.

applicera förmågor erhållna från programmering till andra situationer. Wing (2006) menar däremot att det är mer ett abstrakt tänkande på flera nivåer; "Thinking like a computer scientist means more than being able to program a computer. It requires thinking at multiple levels of abstraction". Wings definition har fått kritik av bland annat Jones (2011), som menar att Wing aldrig ger en fast förklaring till vad datalogiskt tänkande är eller vad som skiljer det från andra sätt att tänka.

Det finns flera försök till att fastställa vad datalogiskt tänkande faktiskt innebär. International Society for Technology in Education [ISTE] och American Computer Science Teachers Association [CSTA] har sammanfattat datalogiskt tänkande till nio olika nyckelbegrepp; datainsamling, dataanalys, datarepresentation, problemlösningsbrytning, abstraktion, algoritmer och procedurer, automation, simulering samt parallellisering (ISTE & CSTA 2011). Denna definition är något som flera forskare hänvisar till (se Mannila et al. 2014; Kjällander et al. 2016) när de diskuterar datalogiskt tänkande och en snarlik definition ges av Heintz, Mannila, Nygårds, Parnes och Regnell (2015) när de menar att det behövs en annan term för att inbegripa de olika begreppen kring programmering som inte har fokus på just kodning. Voogt et al. (2015) kritiserar dock dessa sätt att definiera datalogiskt tänkande och menar att det är en för bred definition som riskerar att sudda ut gränserna till andra sätt att tänka, vilket går tillbaka till Jones (2011) kritisering av Wings (2006) definition av datalogiskt tänkande.

Enligt Mannila et al. (2014) kan programmering användas för att lära ut datalogiskt tänkande då flera delar av programmering är viktiga inom både programmering och datalogiskt tänkande och påstår även att välutformade programmeringsaktiviteter kan adressera alla nio nyckelbegrepp som ISTA och CSTA (2011) sammanfattar just datalogiskt tänkande till. En undersökning av Kalelioglu och Gülbahar (2014) där de involverade 49 elever från årskurs 5, visar dock att programmering genom Scratch⁵ inte hade några nämnvärda effekter på ökad problemlösningsförmåga hos unga. Dock hade det en, om än liten, inverkan på deras självförtroende för problemlösning; något som backas upp av Google (u.å.) som ett stöd för att lära sig datalogiskt tänkande. Voogt et al. (2015) menar samtidigt att programmering inte nödvändigtvis behöver finnas med i bilden för att utveckla datalogiskt tänkande; det ska gå att utveckla dessa förmågor inom andra områden än datavetenskap, men samtidigt ska man inte förringa programmeringens roll. Vidare ger Voogt et al. (2015) inte en egen definition av datalogiskt tänkande, utan påpekar istället att det är svårt att få till en definition på grund av två olika aspekter där kärnegenskaper och omgivande egenskaper ställs mot

⁵ Scratch är ett visuellt programmeringsspråk och verktyg som utformats vid MIT och låter användaren skapa sina egna interaktiva berättelser, spel och animationer. Det används världen över för att lära ut programmering på ett enkelt sätt som passar alla åldrar. <https://scratch.mit.edu/>

varandra. De poängterar även att för att integrera datalogiskt tänkande i utbildningen så behöver vi finna likheter med redan existerande områden (ibid.).

2.3 Transfereffekt

En av fördelarna som lyfts fram med att införa programmering i skolan är en så kallad transfereffekt (Kjällander et al. 2016), vilket innebär att kunskap erhållen genom programmering hjälper eller underlättar lärandet inom andra områden. Detta är något som Papert (1980) är inne på när han menar att barn lär sig tänka mer abstrakt och kreativt och för över kunskap från programmering till andra områden, i ett fall till att gå på styltor. Transfereffekter i sig är dock inte ett allmänt accepterat uttryck, det är snarare två sidor av forskning där flera menar på att transfereffekter inte existerar som rena överföringar. Kalelioglu och Gülbahar (2014) menar att det istället handlar om hur programmering lärs ut som avgör om elever får användning för kunskapen inom andra områden. Även andra forskare (Pea & Kurland 1984; Kurland, Pea, Clement & Mawby 1986) är skeptiska till att programmering i skolan skulle bidra till transfereffekter i den mån som Papert (1980) argumenterar för och menar även de att det handlar dels om hur programmering lärs ut men även vilken individuell nivå studenten är på.

En annan syn på transfereffekter ges av Perkins och Solomon (1989) som delar upp i "low-road" och "high-road" transfer där low-road är en högt och varierat tränad förmåga som gränsar till automation och kan appliceras till liknande situationer utan någon vidare ansträngning. High-road är istället en medveten abstraktion av koncept och reflektion över tidigare erfarenheter för att kunna tillämpa dessa på nya situationer. Perkins och Solomon (1989) poängterar dock att all den erfarenhet och varierad träning som krävs för low-road transfer inte är applicerbart i skolans undervisning, samtidigt som high-road transfer inte ofta naturligt sker utan behöver aktiveras för att fungera, det vill säga att bli uppmanad att tänka tillbaka och reflektera över tidigare händelser. Voogt et al. (2015) ifrågasätter då huruvida det är programmeringen i sig eller om det är lärarnas metodologi som gör att eleverna använder kunskapen i andra situationer.

Heintz et al. (2015) menar även att för att bättre kunna nyttja transfereffekter så bör datalogiskt tänkande införas i redan existerande ämnen istället för att läras ut som ett eget ämne, vilket går i hand med det Voogt et al. (2015), som tidigare nämnt, också är inne på, dock av andra orsaker.

2.4 Kompetensbrist hos lärare

Skolverkets rapport (2016a) visar på en kompetensbrist inom lärarkåren när det kommer till programmering, enbart en tredjedel anser att de inte behöver utveckla kompetensen alls inom programmering och mer än hälften vill ha mer kompetens inom grundläggande datorkunskap. Kjällander et al. (2016), som i sin översiktsstudie där de

studerar flera svenska forskningsbidrag och projekt som relaterar till lärande och programmering, lyfter samma poäng; Sveriges pedagoger har inte tillräcklig kompetens för att undervisa i programmering. Rolandsson (2015), som i sin avhandling hanterar programmering utifrån en lärares perspektiv, menar att mer kunskap inom IT inte räcker, det krävs även djupare ämnesdidaktiska kunskaper. Heintz et al. (2015) tar i sin studie upp ett antal projekt som genomförs i Sverige med fokus på att lära ut datavetenskap vid skolor. Vid ett av projekten visade det på att många av lärarna som deltog var till en början osäkra på om de skulle klara av att lära ut programmering inom sina ämnen. Men efter att ha sett hur det fungerar och hur eleverna interagerar med programmering menar de på att många lärare ändrar sin åsikt och istället ser möjligheter (ibid.).

2.5 Programmering bra för unga, demokratiskt samhälle

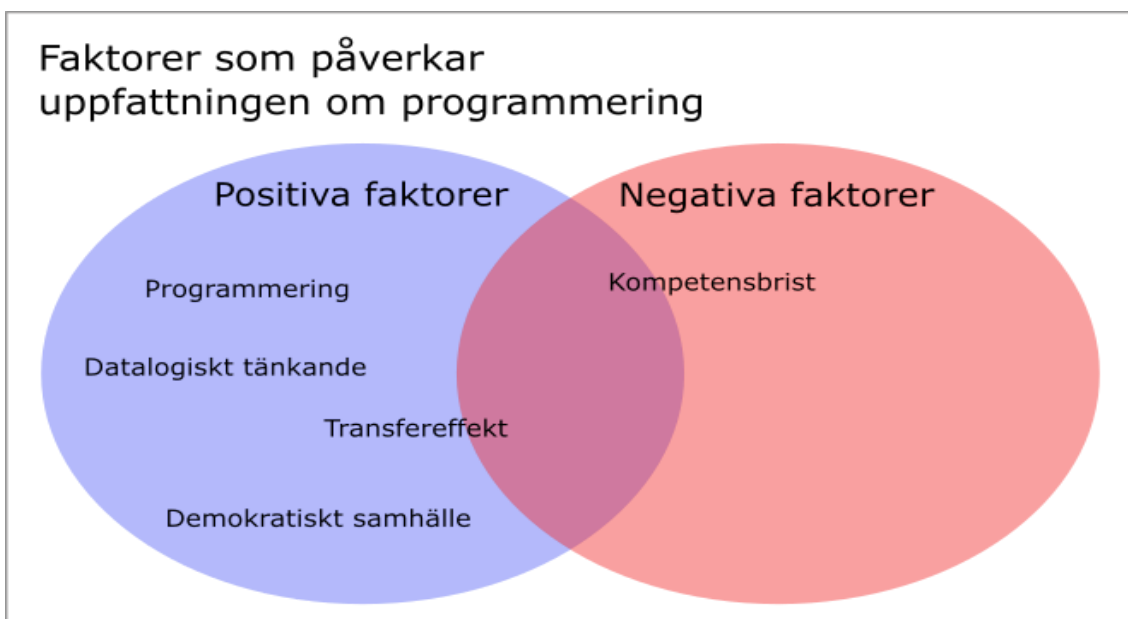
Det demokratiska samhället är en viktig punkt i införandet av programmering i skolan; för att kunna delta i ett modernt samhälle krävs att man besitter kunskap inom olika områden, och i vår tids globala digitalisering är allt mer och mer uppbyggt med hjälp av programmering. För att kunna påverka krävs förståelse och det är flera som belyser vikten av programmering av just den orsaken (se Grönlund, Ranerup & Gustavsson 2003; Regeringen 2017b; Rolandsson 2015). Rolandsson (2015) tar upp demokrati i relation till läroplanen och menar att i framtiden kommer förmodligen alla studenter ha någon form av programmering på schemat, men tar samtidigt upp svårigheten med att ta fram läroplaner för ett så förändrande område som IT. Detta är även något som Carr (1998) tar upp när han menar att det är bara genom en ständigt förändrande både läroplan och demokrati som det faktiskt kan bli en läroplan *för* demokrati.

Det finns dock ett problem med programmering i läroplanen, i relation till det demokratiska perspektivet, vilket Rolandsson (2015) fick fram och förklarar som två sidor av undervisningen; där ena sidan är de lärare som lär ut till alla elever efter deras förmåga och den andra sidan är de som sållar ut de elever som lärarna inte tror kan lära sig programmering.

2.6 Analysmodell

Baserat på teorin och den valda analysmetoden där ansatsen är mer eller mindre förutsättningslös, arbetades det fram en lättviktig analysmodell som innehåller de delar från teorin som krävdes för att studiens syfte skulle uppnås. Analysmodellens faktorer är således de övergripande kategorier som analysmetoden använder sig av som startpunkt vid analysen av intervjusvaren och visas i figur 1 nedan. För att skilja tre tätt sammanvävda begrepp åt, programmering, datalogiskt tänkande samt transfereffekt, så definieras de i innevarande studie enligt följande:

- Datalogiskt tänkande avser kunskap som kan abstraheras, till exempel logiskt tänkande, och även stödjas av personlig utveckling som till exempel ökat självförtroende. Det är inte nödvändigtvis uppkommet från programmering, dock är det programmering som har fokus i innevarande studie och därmed kommer datalogiskt tänkande och transfereffekter vara tätt sammanvävda trots att de är två olika begrepp.
- Programmering avser själva “görandet”. Då majoriteten av de elever som medverkande lärare undervisar är i så pass låg ålder att den programmering de kommer i kontakt med är just Blue-Bots⁶, exemplifieras detta “görande” till; se och analysera problemet (till exempel en bana) till att försöka lösa problemet (räkna antal steg och riktningar), skapa lösningen (koda in via knappar på roboten) och slutligen testa (starta roboten).
- Transfereffekter avser då att överföra kunskap erhållna från programmering till andra områden, att det är programmeringen i sig som ligger till grund för transfereffekten.



Figur 1. Analysmodell med faktorer som kan påverka uppfattningen av programmering.

Den överlappande ovalen symboliserar att det delvis räknas till en faktor men delvis ej, där transfereffekt är förespråkade av bland andra Papert (1980), men även motsägs av bland andra Kurland et al. (1986). Kompetensbrist överlappar på grund av Heintz et al. (2015) som menar att lärare som påbörjade programmering överkom sin osäkerhet kring kompetensbrist.

⁶ Blue-Bot är en liten sköldpaddslänkande robot med knappar på ryggen, den programmeras genom att trycka på knapparna som lagrar sekvensen i ett minne. När roboten sedan startas aktiveras koden som trycktes in genom knapparna och sedan förflyttar sig enligt hur det programmerades. Det finns medföljande banor som är utritade på mattor men det går utmärkt att hitta på egna moment själv; den är inte bunden till något förutom plant underlag. Det går även att styra roboten genom en applikation för till exempel surfplattor där det skickas signaler via bluetooth.

3 Resultat: preliminär analys

Kapitlet inleds med en beskrivning av intervjupersonerna samt upplägg av resultatet. Vidare lyfts de, för studiens syfte, viktigaste kategorier och underkategorier som uppkom vid den preliminära analysen av intervjumaterialet under egna rubriker, tillsammans med förklarande citat från intervjuerna för att tydliggöra innebörden.

Alla citat har försiktigt redigerats för att ta bort dialekt samt utfyllnadsord som kan hindra läsbarheten, pauser markeras med två punkter och överhoppade ord eller meningar markeras med tre punkter inom hakparentes, även mina förklarande ord markeras på samma sätt som borttagna ord, exempelvis ersätts “det” med [vad “det” syftar på] för att sätta citatet i kontext. Intervjupersonerna med deras fingerade namn redovisas i tabell 1 tillsammans med bakgrundsinformation.

Namn	Ålder	Undervisar årskurs	Befattning	ITiS?
Lisa	45-65	Fritidshem och förskoleklass	Fritidspedagog	Ja
Erik	45-65	åk 4-5	Grundskollärare	Ja
Anna	20-30	Fritidshem, upp till åk 3	Läroassistent	Nej
Emma	45-65	Förskolan	Fritidspedagog	Ja

Tabell 2. Redovisning av deltagare i studien med bakgrundsinformation.

Lisa och Erik är verksamma inom samma kommun och båda två upplevde skillnader jämfört med ITiS på ett eller annat sätt, Erik undervisar dock inte i programmering men har tack vare ett nära samarbete med andra lärare viss kunskap inom området och förde en bra diskussion kring ämnena programmering och digitalisering. Anna, Lisa och Emma använder Blue-Bot som enda⁷ verktyg för att lära ut programmering, med undantag för QR koder som Emma använder i samband med programmeringen.

Anna har befattningen läroassistent och befinner sig mestadels på fritidshemmet men är även ibland med på vissa klasser i lågstadiet. Det finns inga kommentarer om ITiS från henne då hon ej var verksam vid den tiden. Emma var verksam under tiden ITiS pågick men berördes aldrig av det, troligtvis på grund av att hon är verksam inom förskolan, så ej heller några kommentarer finns där.

Under den preliminära analysen arbetades det fram ett fåtal kategorier i tillägg till de som redan återfinns i analysmodellen och samtliga presenteras i kronologisk ordning:

⁷ Då Blue-Bot är det enda robotverktyg som används kommer det refereras till även som “roboten” hädanefter.

datalogiskt tänkande, demokratiskt samhälle, digital framtid, kompetens, programmering, resurser och transfereffekt.

3.1 Datalogiskt tänkande

Likheter med matematik

Samtliga respondenter var mer eller mindre eniga om att programmering på ett eller annat sätt kan bidra till att generera ett mer abstrakt tänkande, i stil med datalogiskt tänkande. Erik såg likheter med matematik och menade att våran typ av matematikräknande ligger nära problemlösning, citatet nedan är taget från en längre berättelse om hur en elev som kommit från Palestina ligger efter i matematiken:

Vi har nåt matematikmaterial nu [...] och det kan jag säga det är nog väldigt stor skillnad på arabländer och öststater och det vi håller på med i matematik. Att dom är mycket mer fokuserade bara på matematikräknande, att lösa algoritmer, och vi har mer problemlösning, mycket mer samarbete och problemlösning än bara rent manuellt räknande.

Även Anna pekade på ett samband till matematik där just logiken och hur utförandet sker påminner om programmeringen:

[...] att se logiken i hur saker utformar sig, det blir mer konkret, trycker ja vänster så svänger den vänster. [...] det blir någon sorts logik i det, precis som det blir i allt annat, eller som ett mattetal eller vad som helst, att man ser att det blir lite.. att det följer en logisk väg.

Ett fascinerande angreppssätt som Emma berättade om, är att blanda in andra delar i programmeringen:

Så har vi blandat upp den där banan med QR koder också, så de måste ta fram iPaden då och scanna av, så får de ett litet matteproblem när de kommer [till en QR kod] - det är också spännande då när de stöter på lite annat under vägen.

Ingen av de andra tog upp att det går att mixa ämnen på det sättet, men det tycks vara uppskattat bland eleverna och Emma kändes väldigt engagerad när hon pratade om det.

Den personliga utvecklingen

Det är inte bara den direkta relationen till andra ämnen utan även den personliga utvecklingen som flera av de tog upp. Flera såg både direkta och indirekta fördelar där de bland annat tog upp hur barnen lär sig principen för hur man tänker och att det kan öka självförtroendet med en större kunskap. Emma listade ett flertal kunskaper som hon anser utvecklas genom programmering:

Dom lär sig att koncentrera sig och fokusera, dom lär sig strategiskt tänkande, tänka framåt och räkna ut saker, lär sig mycket matematik och lägesbegrepp.

Erik var mer inne på de indirekta fördelarna med programmering; “Att man får en större kunskap om omvärlden, bygger väl folk på insidan också, det stärker ju självförtroende och egot så det är väl positivt, absolut”.

3.2 Demokratiskt samhälle

Förståelse och säkerhet

När de fick frågan om vilken uppfattning de hade kring varför programmering skulle läras ut var samtliga väldigt eniga om att det är på grund av utvecklingen av samhället och att i dagens och framtidens samhälle är det något man behöver lära sig. Tre av de fyra deltagarna var positiva till utvecklingen som vi har och var även positiva till att ge eleverna en förståelse för omvärlden. Lisa, som är den fjärde deltagaren, uttryckte sig dock lite negativt mot samhällsutvecklingen vi har men menade trots det att eleverna behöver kunskapen:

Ja alltså eftersom vi är där vi är tycker jag [att programmering ska läras ut], sen kan jag erkänna att jag gillar inte den här samhällsutvecklingen.. för det är så himla mycket fokus på digitalisering hela tiden, men nu är vi där och då behöver dom lära sig det. Så ungefär tycker jag.

Flera av dem menade även att programmering behövs för att förstå omvärlden, att veta hur ting är uppbyggda:

Det är bra för att du får ju en insikt i att veta hur någonting fungerar, varför blir det som det blir? Det är ju för att nån har sagt till den här maskinen eller det här programmet att göra på det här sättet. Det är inte så att en maskin är smart på det viset, utan man säger att den tänker men det gör den ju inte, den bara bearbetar fakta om man säger så. (Erik)

Något som stack ut lite är att Anna tog upp en sak som de andra inte berörde och tänkte tillbaka på när hon själv gick i skolan och det fanns en viss grupp som höll på med programmering:

Och sen är det ju lite kul för då tar man bort det här som jag tycker, när jag var liten, [...] det var visst typ av folk som höll på med programmering, [...] så har alla nu liksom programmering att, det blir mer vinklat på det sättet [...] det blir det här - alla kan hålla på med det, [...] det blir som ett.. ett okej yrke eller vad man ska säga.

Emma tog även upp hennes roll i förhållande till att lära ut till elever och menade att det är viktigt att introducera programmering för barnen, “[...] det tycker jag är min roll då i alla fall, även om jag inte kan mycket själv, att man ändå introducerar det och visar” (Emma), oavsett egen kunskapsnivå och förklarade samtidigt att det är roligt för de vuxna att lära sig också.

Enbart en av deltagarna, Lisa, pratade om lite säkerhet kring att lära sig programmering, att kunna använda det och kontra dem som använder det för ett ont syfte:

[...] det är viktigt att det finns kunniga människor som kan ligga steget före, [...] vapenindustrin, sociala medier, allt - allt som man kan använda positivt kan man ju nästan alltid använda negativt också.

Hon uttryckte även lite oro inför framtiden och ifrågasatte vilken kunskap som är viktigast:

Men nu kommer det i brevlådan krisberedskap och man kanske ska lära barnen sånt också, göra en eld, veta vad man ska äta i naturen, det är också samhällsnytta. [...] Jag kan egentligen tycka att den där baskunskapen, livskunskapen, egentligen är viktigare.

Genus och social bakgrund

Ett positivt mönster jag fann, som jag inte förväntade mig, var att ingen kände att det fanns några skillnader mellan varken kön eller social bakgrund. Dock kände Emma att det fanns en viss skillnad i hur pojkar och flickor betar sig när de får något nytt presenterat för sig och menade att det ligger en stor roll på de vuxnas axlar att ge alla chansen:

Jag tror att vi har stor påverkansmöjlighet på förskolan. För att man märkte när man först tog fram det och presentera det, att det var pojkarna [...] som va mest liksom sög sig till det här som flugor [...] men sen gjorde vi möjlighet, vi dela in dom i lite grupper o så, att alla, ja såg till att alla fick prova, o då [...] spred det sig så att då var det flera flickor som faktiskt tyckte det var roligt o ville pröva igen o igen. [...] men det var för att vi vuxna agera o skapa förutsättningar för det [...] Lite den känslan får man, det gäller att skapa möjligheter, ta det lite lugnt och skapa små grupper.

Annars verkar deltagarna överens om att alla elever har möjligheten, flera uttryckte sig i stil med “Alla har datorer hemma, iPads hemma, o har den möjligheten” (Emma) och “Det är ingen, asså över kön o sådär finns det ingen skillnad heller, om dom är intresserade.. utan alla är intresserade” (Anna).

3.3 Digital framtid

Framtiden

Tre av fyra deltagare pratade om framtiden med teknologin och digitalisering där en av dem menade att det finns ett slags ekonomisk trygghet i att lära sig programmering “[...] i alla möjliga.. sektorer om man säger så, behöver man kunna programmering för att klara sig” (Lisa). Emma pratade om att det finns mycket mer att uppfinna, upptäcka samt utnyttja och hänvisar till att det är barnens framtid. Den tredje deltagaren som pratade om framtiden, Erik, gjorde det på ett nostalgiskt sätt där han just nu håller på och slänger gamla uppslagsverk som ingen använder längre då det bara är att “googla” efter information, men poängterade även vikten av att vara källkritisk. Han menade även att det är en bättre satsning nu än under ITiS; “[...] då hade ju inte alla tillgång till datorer.. men det skulle köpas in men då fanns det inget riktigt vad man skulle göra med dom” (Erik).

3.4 Kompetens

Kompetensbrist

På frågan om de fått någon utbildning som ska hjälpa dem i undervisningen svarade samtliga nej och känslan av att de inte kände sig säkra nog i sin kompetens var överhängande; två av dem nästan skrattade åt tanken över att ha fått fortbildning. Dock pratade två av dem att de har gått någon kurs, men inte som handlar om programmering. Flera av dem poängterade att det finns en generell brist på kompetens vid deras respektive arbetsplats, Lisa menade att det finns de som knappt kan använda en dator och Anna menade att de hade tur då han som är teknikansvarig på skolan även är med på fritidshemmet hon är vid. Emma lyfte även en intressant poäng när hon menade att det finns en motvillighet till att lära, “[...] dels personalens, vad ska ja säga.. rädsla eller.. motstånd till och lära och orka ta reda på vad det är för någonting [...]” (Emma). Lisa menade även att eleverna är före pedagogerna i kunskap:

Alltså eleverna är redan där, de är många gånger redan långt före pedagogerna i kunnighet inom vissa, surfplattor och telefoner och så. Kanske inte just datorer och så dom är inte, dom är mer inne på appar och det där.

Kompetensen finns ändå

Trots den uppenbara bristen på fortbildning tror sig ändå samtliga ha tillräcklig kompetens för att kunna undervisa på den nivå de befinner sig och som Lisa

sammanfattade alla deras tankegångar ganska bra; “Ja men eftersom jag är på förskoleklass och fritidshemmet känner jag att [då] räcker det” (Lisa).

Det finns dock en vilja att lära sig mer, där en av drivkraften är att kunna locka fler elever genom mer avancerad kunskap eller programmering. En annan drivkraft är att kunna lära ut mer under tiden de håller på med det:

Men hade ja haft mer kunskap om hur det funkar, hur det är uppbyggt och sådär, då hade man under tiden som man använder dom kunnat pratat om hur det faktiskt går till, så. [...] vet man mer runtomkring så kan man prata mer runtomkring och lära ut mer runtomkring än att bara såhär, ah men du får trycka på höger. (Anna)

Både Lisa och Emma tog även upp att det finns kurser de får gå, enligt Lisa var det dock ett ganska dåligt utbud på just programmeringskurser och Emma påpekade att hon själv skulle vilja lära sig mer avancerade robotar men uttryckte även lite missnöje över hur det är upplagt:

Det är mer riktat mot skolan tycker jag, än förskolan, för jag ringde en gång för att lära mig [Dash och Dot⁸], men då sa hon att det var mer mot dom som kunde lite mer i skolan då, så att det verkar som det kanske ligger liiite efter på förskolesidan. Det är lite synd då. (Emma)

3.5 Programmering

Vad betyder det?

Samtliga fick frågan om vad programmering innebär för dem, samtliga gav snarlika men olika svar:

När jag hör programmering så tänker jag något jättesvårt, att man ska programmera dataprogram och sånt, fast när jag provat programmering på en enkel nivå med Blue-Bot så var det väldigt lätt och väldigt kul, [...] fast i mitt huvud är det jättestort och svårt. (Lisa)

Att du tillverkar ett program kanske, eller att du styr en robot eller nåt sånt. Att du styr nånting med hjälp av programmering, att du säger till vad den ska göra. (Erik)

Att man, gör olika koder.. håller på lite med teknik.. ettor och nollor. (Anna)

Programmering för mig innebär att.. vi erbjuder barnen att bekanta sig med det på förskolan.. det är Blue-Botar.. har vi, så att dom.. räknar ut vart den här lilla Blue-Boten ska åka på en matta, och så får de trycka in på knappar

⁸ En annan typ av robot som är snarlik Blue-Bot men lite mer avancerad.

då, vart dom ska åka. Mm, det är ungefär det jag kan om programmering, att strategiskt räkna ut. (Emma)

Tre av dem nämnde uteslutande att man styr eller programmerar något, Lisa påpekade även att hon inte helt förstår vad programmering är men att hon förstår principen för det, och den fjärde, Emma, pratade även om att strategiskt räkna ut vart Blue-Boten ska åka, vilket visar på en ganska enformig uppfattning kring ett relativt komplext begrepp. Emma är även den enda som pratade om hur det går att använda programmering i kombination med andra ämnen och säger; “Jag tycker att det tillför mycket, det är ett roligt sätt att jobba” (Emma).

Den röda tråden är dock att alla har ändrat uppfattning kring vad programmering betyder sedan de blivit bekanta med det. På frågan om vad de hade för uppfattning från början kring programmering så var det mer i stil med svåra, avancerade program, “[...] förut kändes det mer yrkesgrej, man var tvungen att va expert, man läser till nånting för att bli expert på det, så, asså ingenjörer” (Anna), med en utmärkande deltagare, Emma, som uppgav att denne inte hade någon uppfattning alls kring programmering sedan tidigare. Även Lisa gav ett intressant svar som inte de andra berörde:

Jag hade nog inte kallat det för programmering med Blue-Bot om jag bara fått dom i handen och man skulle trycka in koderna [...] jag hade nog inte tänkt det programmering.

Inställning

Som tidigare nämnt ansåg samtliga deltagare att det är bra och nödvändigt att lära ut programmering, men en deltagare gav följande svar på frågan om hon använder något annat i undervisningen:

Nej, det är inget intresse jag har heller, alltså jag använder sociala medier, ute och googlar ibland [...] men programmering gör jag inte mer än man måste. (Lisa)

Trots detta tyckte hon dock att det var väldigt kul med programmering på den nivå som Blue-Bot ligger på och flera uttryckte ett nöje i att programmera med Blue-Bots; “Vi gör det på skoj eller vad man ska säga, så det är kul faktiskt” (Anna).

Både Lisa och Erik menade även att det finns ekonomiska intressen som ligger bakom att införa programmering i skolan; “Men sen så att det är väl också så att det är någon som tjänar jävligt mycket pengar på hela den här karusellen” (Erik).

Anna lyfte ensam en punkt som är viktig att notera och menade att när de är så unga får de inte riktigt lära sig själva tekniken:

Nu känns det som att dom här grejjerna som vi har, eftersom att det är så små elever, det känns som att då får dom mycket serverat, de behöver inte tänka så mycket över vad det faktiskt är dom gör, på det sättet.

I kontrast till Annas kommentar menade dock Emma att det viktiga för dem på förskolan är att introducera barnen till programmering, att “Det viktigaste för oss är att presentera materialet, att dom har sett det, det skapar igenkänningar när de kommer till skolan” (Emma).

Elevernas reaktioner

Eleverna tycks ta till sig programmeringen väl, ingen av de tillfrågade deltagarna kunde dra sig till minnes att det är någon som inte gillar att åtminstone testa det. Samtliga använde även ord likt “jättekul” och “superkul” när de pratade om hur eleverna reagerar på programmering. Lisa upplevde att de är ivriga och impulsiva, även om de är två åt gången på en robot:

Vi sa till dom att dom skulle samarbeta och va två och två på dom, [...] de är så ivriga o impulsiva så de har svårt att vänta på att nån kompis ska få göra färdigt själv, man vill berätta hur man ska göra om man kommit på [...] hur man ska göra och så där, eller nästan rycker de ur händerna på varann.

Den initiala nyfikenheten verkar dock dö ut hos en del elever, dels menade några av deltagarna att eleverna börjar åka runt lite överallt med robotarna efter en stund, eller rent av ger upp för att det är för svårt, dels verkar det finnas inrotade mönster där elever gör det de alltid brukar göra. Anna, som är på fritidshem där de oftast har över 50 elever åt gången, förklarade det så här; “[...] dom som spelar innebandy, de kanske inte kommer och programmerar utan de kanske spelar innebandy eftersom de alltid spelar innebandy” (Anna).

Samtidigt finns det de elever som alltid vill leka med robotarna, eller sitter länge och fokuserat för att finna lösningar. Emma berättade om en elev som flyttade roboten samtidigt som han knappade in vägen och när hela vägen var färdig flyttades roboten tillbaka till start och startades; “Dom andra barnen var inte så noggranna, det var en väldigt fascinerande strategi, han hade ju verkligen fokus och koncentration och tålamod” (Emma). Anna och Emma berättade även om hur eleverna sporrar sig själva genom att göra svårare banor eller få roboten att gå baklänges och Emma känner ibland att hon måste hejda eleverna:

Dom prövar sig mycket fram där, själva, trycker och sådär så man får nästan hejda dom då istället och tänk nu [...] ‘men är det den eller den?’ jamen pröva får du se, ‘nej det var inte den’, så gör dom om, dom lär sig mycket genom learning by doing så. (Emma)

3.6 Resurser

Brist på utrustning

Samtliga deltagare uttrycker en brist på resurser, Emma och Lisa tog upp att de lånar Blue-Bots via SLI medicenter men att de ofta är upptagna så de behöver vara ute i god tid, men även då blir det bara någon gång per termin. Anna tog upp en värre situation; “Ja asså.. nån gång när vi testa hade vi två, då får de ju vänta och så blir de osams och så där” (Anna). Att barnen blir osams över robotarna verkar inte höra till normen, men hon berättade även att ibland har de enbart haft två robotar när det har varit uppemot 90 barn på fritidshemmet.

En ljuspunkt är dock Lisa som berättade om några lärare som gick ihop och höll en workshop, som till slut ledde till att de fick ett ja till att köpa in ett tiotal robotar till kommunen som kunde cirkulera runt mellan skolorna. Dock verkar det vara just ett ekonomiskt hinder som står i vägen för utrustningen, både Lisa och Erik tog upp just det; “Ja utrustning, pengar, är ju ett hinder helt klart” (Lisa), “[...] men då säger dom att ‘nej, det finns inga pengar’ [...]” (Erik).

I relation till ITiS kände dock Erik att det är en mer genomtänkt satsning nu och hänvisade till det faktum att det finns mer teknik och digitala verktyg; vilket leder till att undervisningen i sig inte saktas ner av att elever behöver stå på kö för att använda till exempel en dator. Lisa kände däremot lite tvärtom, att de fick mer utrustning under ITiS, dock präglas deras svar av deras tankegångar; Erik tänkte på undervisningen i sig medan Lisa syftade på arbetslagens⁹ resurser.

Brist på tid

Den enda deltagaren som inte uttryckligen sa att det inte fanns någon tid för att sätta sig in i det nya var Anna, vilket kan bero på att hon är den yngsta av deltagarna men även den som är lärarassistent och inte huvudansvarig för undervisningen. Resterande respondenter vittnade om en tydlig tidsbrist, i förhållande till olika aspekter där fortbildning och fullt schema ligger i centrum:

Det som är svårt också är att man ska ha tiden till det i skolan, för det ingår i timplanen men samtidigt ska man hinna med allt annat också, som personal, som lärare ska man hinna med allt annat och sätta sig in i och kunna. (Lisa)

Erik menade även att “Ingenting annat ska tas bort under tiden utan det bara läggs på, så det är väl ett problem inom skolan” (Erik). Emma lyfte en punkt där hon menade att mer

⁹ Se kapitel 1.1.1

avancerade programmeringsverktyg inte riktigt hinns med, mestadels på grund av tiden kring allt praktiskt arbete:

Några som heter [Dash och Dot], men [...] vi hann inte att sätta oss in i och få det att fungera då, för vi fick det i slutet av terminen, [Blue-Bots] var ju bara att trycka direkt på med en gång. [...] sen så är det ju.. [...] tiden, det är så mycket runtomkring, allt praktiskt arbete som stjälar tid i från undervisningen som vi ska göra, för att ha det och fungera under dagen. Så vi gör så mycket annat än att undervisa och det tycker jag är helt förkastligt.

3.7 Transfereffekt

Finns den?

En av deltagarna uppgav att det var svårt att svara på om programmering kan generera kunskaper som kan användas mer generellt, och efter att ha fått transfereffekt förklarar för sig svarade:

Det kan ju också va så att dom har en annan mognad i hjärnan som gör att dom lär sig programmera fortare [...] För det känns ju som att när vi nyss har börjat med det finns det inget att jämföra med, [...] om vi gör det i skolan i våran grupp så har vi ingen jämförelsegrupp. (Lisa)

Samtidigt var de andra positiva till att det på något sätt skulle kunna finnas transfereffekter, i olika mån. Erik menade att eftersom vår matematik är så problemlösningsbaserad så finns det en chans att “[...] programmering skulle mer främja vår matematik om man säger så” (Erik) och Anna jämförde också med matematik:

Speciellt matte är en sån sak, ställa upp ett tal och se att det faktiskt leder någonstans, precis som när man programmerar sin Blue-Bot, [...] man får börja någonstans, för att den ska komma dit, det går inte bara flytta den till mål så är man klar.

Dock uttalade sig varken Erik eller Anna om några andra ämnen än matematik, medan Emma pratade mer generellt om transfereffekt, och även genom hela intervjun har Emma uttryckt det största intresset för programmering överlag:

Det här är nånting som majoriteten tycker är väldigt roligt, det blir på lek, det är intressant, trycka på knappar och det händer nånting [...] det verkar som att det tilltalar väldigt många barn i alla fall, [...] och allting som är roligt gör ju saker och ting lättare att lära sig och om det får spridningseffekt då eller inte så.. ah det är väl bara positivt om det får det. [...] Ja tror, tycker man att det är roligt och.. håller på med det här Blue-Botar eller vad de nu kan göra [...], då får man massa mer på köpet som handlar om nånting annat, [...] att det blir nån slags spridningseffekt, tror jag, absolut.

4 Analys

I detta kapitel jämförs resultatet med teori och studiens syfte, för att försöka svara på de frågeställningar som gavs i kapitel 1. Kapitlet avslutas sedan med en kort sammanfattning av analysen.

4.1 Programmering, vad menar du?

När deltagarnas svar på frågan om vad programmering innebär jämförs, syns det direkt att det skiljer sig markant från de definitioner som brukar ges i den vetenskapliga världen, trots att det även där råder en oenighet kring definitionen så är de överens om att programmering är mer än koden (Mannila et al. 2014; Kjällander et al. 2016). Istället för bland annat problemlösning, design av lösning, dokumentation och kod så gav tre av fyra ett svar som mer tyder på att det enbart är koden, programmerandet av något, som blir synonymt med programmering. Med undantag för Emma som svarade att det även var ett strategiskt tänkande, där det går att läsa in lite design samt problemlösning. Men går vi djupare än så går det att finna en bredare förklaring. Både Erik och Anna jämförde programmering med matematik, att det har mer logik i sig och Erik menade till och med att den problemlösning som återfinns i matematiken främjas av programmering. Vi ser alltså en förklaring som närmre och närmre stämmer överens med vad både Skolverket (2016b) och Mannila et al. (2014) menar att programmering är.

Trots den relativt enformiga förklaring av vad programmering innebär, menade samtliga respondenter att eleverna lär sig allt från grunden till hur man tänker, till en rad olika förmågor. Detta visar på att de ändå tänker kring programmering på ett större plan än att enbart "styra något", vilket enligt Kjällander et al. (2015) gör att begreppet börjar gå över till datalogiskt tänkande snarare än programmering. Många av dessa förmågor återspeglas även i det datalogiska tänkandet som definierat av ISTE och CSTA (2011) och går även i hand med Paperts (1980) syn på datalogiskt tänkande där programmering är källan till att kunna abstrahera kunskapen. Vi ser även att Emma använder sig av olika medium i programmeringen där hon samtidigt sätter ut delmål längs banan där eleverna scannar in en QR kod för att sedan få ett matematikproblem att lösa, vilket ligger nära det Voogt et al. (2015) menar att undervisningen i sig borde sträva efter; att inkorporera datalogiskt tänkande i redan existerande ämnen.

Dock börjar det även närma sig den gräns som Voogt et al. (2015) verkar förhålla sig till där de menar att synen på även datalogiskt tänkande blir alltför bred och det inte längre går att urskilja datalogiskt tänkande från andra sätt att tänka. Även Jones (2011) har tidigare kritiserat liknande synsätt (se Wing 2006) för att vara för breda och menar att det behöver konkretiseras mer. Speciellt förklaring från Emma där hon menar att

eleverna lär sig flertalet abstrakta kunskaper kan ifrågasättas huruvida det faktiskt är datalogiskt tänkande som ligger i fokus eller om det kan vara liknande sätt att tänka.

Det var flera som pratade på ett eller annat sätt om personlig utveckling men Erik var den som mest tog upp en specifik fördel med programmering när han pratade om att det stärker självförtroendet på insidan; vilket spelar i händerna på Kalelioglu och Gülbahar (2014) samt Google (u.å.) som menar att vissa personliga egenskaper kan förstärka datalogiskt tänkande.

4.2 Jag kan ju faktiskt!

För att återgå till syftet med innevarande studie så tycks samtliga uppleva att de inte har nog utbildning eller tillräcklig kompetens för att lära ut programmering när de först fick frågan, vilket stämmer bra överens med flera andra aktörer (se Skolverket 2016a; Kjällander et al. 2016; Rolandsson 2015). Lisa påpekade även att det är en stor generell kompetensbrist i kommunen där hon arbetar och menade att flertalet lärare inte kan mycket mer än att skicka ut e-post. Både Lisa och Emma tog upp ett hinder för fortbildningen där brist på kurser ligger i fokus. Vad kompetensbristen bottnar i är svårt att säga, dock hade Emma en teori kring det när hon tog upp motvilligheten eller rädslan till att lära sig det nya.

Det fanns även en generell attitydskillnad där Anna och Emma var mer positiva till programmering i helhet, trots brist på resurser. Lisa och Erik var lite mindre positiva, dock mestadels i samband till brist på stöd från Eriks sida medan Lisa även motsatte sig det digitaliserade samhället som helhet och programmeringen i sig där hon uppgav att det höll hon inte på med mer än hon måste. Detta avspeglades även i vad respektive deltagare tror att programmering kan göra för eleverna där endast Lisa inte fullt tycks tro att det går att erhålla några mer abstrakta kunskaper eller transfereffekter från programmering, för henne handlade det snarare om att väcka ett intresse.

Dock hände något intressant när de fick frågan kring deras kompetens i relation till undervisningen de bedriver idag; samtliga deltagare var då överens om att kunskapen räcker för den nivå de är på. Emma lyfte även en viktig punkt när hon sa “[...] det tycker jag är min roll då i alla fall, även om jag inte kan mycket själv, att man ändå introducerar det och visar” vilket visar på att för Emma så är inte den egna kompetensen central; så länge hon lyckas förmedla innebörden så har hon lyckats med sin uppgift. Som hon även tog upp så finns det samtidigt en viss skillnad i kompetensen som krävs för olika typer av programmering, “Vi hann inte att sätta oss in i och få det att fungera då, för vi fick det i slutet av terminen, [Blue-Bots] var ju bara att trycka direkt på med en gång” (Emma). För studiens räckvidd var det inte några mer avancerade robotar med, vilket kan påverka det faktum att alla ändå tror sig ha tillräcklig kompetens. Det går även emot den initiala uppfattning om att det skulle finnas en utbredd kompetensbrist

(se Kjällander et al. 2016; Rolandsson 2015; Skolverket 2016a), istället verkar det snarare som att de som är verksamma inom lägre skolåren faktiskt har kompetensen som krävs eller att utrustningen är såpass enkel att lära sig att det inte kräver någon omfattande utbildning för att komma igång.

Det finns dock en vilja att lära sig mer, även Lisa finner ett nöje i att programmera med Blue-Bots och påpekade att hon skulle vilja ha något mer avancerat för att väcka intresse hos lite äldre elever. Anna är den enda som inte pratade om andra typer av programmering utan ville istället ha en djupare förståelse för just Blue-Bot för att kunna prata om den samtidigt som hon använder den med eleverna. Dessa upptäckter, viljan att lära mer samt det faktum att de anser sig kunna nog, ligger som bevis för den teori som Heintz et al. (2015) har om att lärarna som påbörjar programmering ändrar sin åsikt och istället börjar se möjligheter, de inser att det inte behöver vara så svårt eller stort som de först hade föreställt sig.

4.3 Framtiden är nu

Det finns en överväldigande övertygelse om att programmering ska läras ut, samtliga deltagare menade att det ligger i tiden; dagens samhälle är så pass avancerat att det är nödvändigt ur ett demokratiskt perspektiv för att kunna följa med utvecklingen och bidra. Erik uttryckte sig i stil med att förstå omvärlden, att få en insikt i hur något fungerar och att det faktiskt är människor som säger åt alla program och saker vad de ska göra, ett synsätt som delas av bland annat Grönlund et al. (2003) och Rolandsson (2015). Oavsett syn på programmeringen i sig så är de överens om att eleverna ska lära sig vilket gör att vid en jämförelse med Rolandssons (2015) upptäckt faller samtliga deltagare under kategorin för lärare som lär ut till eleverna efter deras förmåga. Emmas argument att det ligger mycket ansvar på de vuxnas axlar är dock viktigt att tänka på; alla elever behöver få möjligheten att programmera, oavsett bakgrund eller kön och det är upp till lärarna att skapa dessa möjligheter.

Lisa fäste även stor vikt vid att utnyttja programmering i ett gott syfte för att motverka dem som har ett ont uppsåt och menade att det är viktigt att ha dessa kunniga personer i samhället. Emma tog även upp alla framtida uppfinningar och upptäckter som vi ännu inte har utvecklat och menade att det är barnens framtid.

Intressant är även deras tankar kring det faktum att programmering nu kommer med i timplanen, något Lisa nämnde i relation till ett fullt schema där allt ska hinnas med och även Erik samt Emma tog upp problemet med tidsbrist. Erik speciellt menade på att allt läggs på men inget tas bort, vilket pekar mot att undervisningen kan bli lidande och det går då att ifrågasätta hur demokratiskt det blir; detta är något Carr (1998) berör när han

pratar om läroplan i förhållande till demokrati och även Rolandsson (2015) belyser svårigheterna med att utveckla läroplaner för ett så föränderligt ämne som IT.

4.4 Det sprider sig

Att programmering skulle generera mer abstrakt kunskap var som sagt de flesta mer eller mindre överens om, men om det går att sprida den här kunskapen till andra ämnen har de lite mer skilda meningar kring. Erik gav exemplet om en elev från en annan kultur som låg två årskurser bakom våran matematik, förmodligen på grund av att matematiken skiljer sig, och menar att programmering skulle främja våran matematik. Detta går ihop med Papert (1980) när han menar att programmering ökar problemlösningsförmågan som kan överföras till andra områden, i hans exempel till att gå på styltor. Emma var väldigt positiv till att det skulle finnas transfereffekter och menade att alla barnen tycker det är väldigt kul med programmering och om något är kul så lär man sig bättre. Går vi tillbaka till det flera forskare säger kring transfereffekten (se Pea & Kurland 1984; Kurland et al. 1986; Kalelioglu & Gülbahar 2014) ser vi även hur de menar på att det är själva processen, hur istället för vad, som avgör om det uppstår någon transfereffekt; vilket kan vara varför Emma har en så mycket positivare syn då hon även lär ut programmering på ett annat sätt genom att involvera flera moment och ämnen. Detta speglas delvis i det Voogt et al. (2015) tar upp när de menar att det kanske snarare beror på lärarnas metodologi än programmering i sig som den avgörande faktor kring uppkomst av transfereffekter. Samtliga deltagare tyckte dock att just dessa Blue-Bots är roliga att använda, men de har olika inställning som kan påverka på en individuell nivå när de lär ut.

Lisa, som inte tycker sig kunna se någon transfereffekt, mycket på grund av att hon inte känner sig insatt, gav även förklaringen att “[...] det kan också va så att dom har en annan mognad i hjärnan som gör att dom lär sig programmera fortare [...]” efter att ha fått Paperts (1980) exempel med styltor förklarat för sig, något som till viss del backas upp av Kurland et al. (1986) som pratar om studenternas individuella nivåer i förhållande till transfereffekten.

Det tredje synsättet, som definierat av Perkins och Solomon (1989)¹⁰, tycks ingen kunna relatera till då deras elever är så pass unga att de inte reflekterar så mycket över vad de gör, Lisa menade att för dem är det nog mer en leksak och Anna pratade om hur eleverna får så mycket serverat i den låga åldern att de inte behöver tänka över vad de gör. Emma uppgav att hon fick ibland hejda sina elever och uppmanade dem att tänka till lite, men trots det tycks de till största delen testa sig fram.

¹⁰ Se kapitel 2.3

4.5 Sammanfattning

Ingen av dem ger en förklaring på programmering som är fullständig enligt flera insatta, bland andra Mannila et al. (2015). Dock visar flera av dem en förståelse att det är mer än att skriva kod eller styra en robot, de reflekterar mer kring det när de uppmanas svara på vad eleverna lär sig genom programmering. Att programmering kan generera mer abstrakt kunskap var dock samtliga överens om i olika grad där Emma är den som ser mest positivt på det. Inställning till programmering i sig influerar deras svar och tycks även korrelera med inställningen till barnens lärande där en mer positiv attityd även pratar mer entusiastiskt kring programmering i undervisning och dess fördelar. Även transfereffekter, att programmering är en direkt källa till kunskap som används inom andra ämnen, är det återigen Emma som är den mest positiva till. Emma är även den som verkar uttrycka störst intresse för programmering och beklagar sig över att inte ha hunnit med att lära sig mer avancerade verktyg. Både Lisa och Emma pratar även om att väcka intresse och nyfikenhet hos barnen och Emma menar att det är hennes roll i programmeringen.

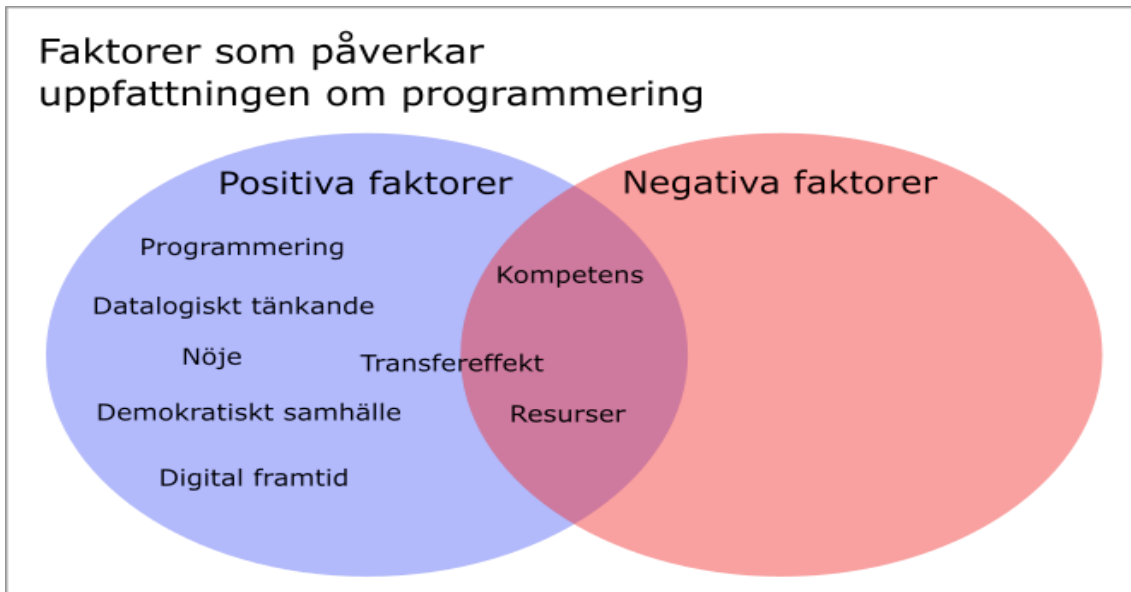
Samtidigt är den yngsta, Anna, den som inte upplever så mycket negativt i förhållande till resurser och stöd, vilket dock kan bero på hennes befattning som lärarassistent och inte huvudansvarig lärare. Både Erik och Lisa upplevs som mer benägna att prata i termer som uppfattas negativt när de pratar om resurser, både tid och material, och de är även verksamma inom samma kommun vilket leder till att tro att det kan vara kommunen som inte är lika framåt som de övrigas, vilket även kan påverka lärarnas syn på hela satsningen. De enda kommentarer kring ITiS var även de från Erik och Lisa, där Lisa saknade de resurser som fanns att tillgå för lärarna och Erik istället menade att den nuvarande satsningen är mer genomtänkt.

Den självskattade kompetensen verkar enligt samtliga deltagare räcka, trots den initiala brist som gavs. Sedan om det är en generell brist i lärarkåren eller ej, kan inte svaras på i innevarande studie, även om en specifik lärare upplever att så är fallet. I stil med Heintz et al. (2015) så verkar de snarare se möjligheter och även Lisa, som är minst positiv till programmering och en ökad digitalisering i samhället, gillade Blue-Bots och ville ha mer avancerat material. Tyvärr tycks utbudet på fortbildning vara relativt lågt, speciellt i lägre åldrar där de även inte anses kunna lära sig mer avancerade verktyg vilket även går ut över lärarna.

Samtliga anser att det är viktigt att kunna och lära ut programmering i grundskolan, där Erik var den som pratade mest kring det faktum att elever behöver lära sig hur saker och ting fungerar, att det är viktigt även för att kunna förmedla sig. Anna tog även upp resonemanget att genom att lära ut programmering till alla tas den minoritet bort som har varit innan, att lite eliminera stigma kring att sitta framför datorn och vara en nörd. Ingen av de andra tog upp samma punkt som Anna, förmodligen på grund av

ålderskillnaden som gör att de övriga tre inte har upplevt samma situation när de gick i grundskolan själva.

Analysen kan även sättas in i en modell likt analysmodellen, där den omarbetade kategorin kompetens samt de nya kategorierna resurs och digital framtid, visas i relation till positiva och negativa faktorer (se figur 2).



Figur 2. Resultatet av analysen i faktorer, baserat på ursprungsfiguren av analysmodellen i kapitel 2.

Kompetens och resurser ligger helt inom båda fälten på grund av deras två sidor där det upplevdes både en brist och ett stöd inom vardera faktor. Ingen faktor var således helt negativ, och vi ser en överväldigande positiv syn där transfereffekt fortfarande delvis ligger mot det negativa hållet men symboliserar en positiv-neutral faktor snarare än positiv-negativ. Notera även att det är de övergripande kategorier som är utplacerade, med tillägget av *Nöje*, som även inkluderar en form av intresse, för att utmärka att det är en faktor som påverkar, men som inte var övergripande nog att bli en egen kategori under analysens gång.

5 Diskussion

Bakgrunden till studiens syfte, att ta reda på hur lärarna uppfattar programmering och även hur de ser på sin egen kompetens, grundar sig i mitt egna intresse av programmering samt min egna erfarenhet av lärare inom programmering i de högre skolåren och högskola där det kraftigt har varierat i kompetensnivå. När programmering står på agendan för hela skolväsendet var det således intressant att ta reda på hur lärarna känner och ser på det, framförallt i de lägre skolåren där det är viktigt att introducera programmering utan att ge avsmak för det.

Syftet med studien har varit att exemplifiera ett par faktorer som kan påverka hur lärarna uppfattar programmering. I analysen togs flera av dessa upp och visar även att det inte bara är konkreta begrepp som datalogiskt tänkande eller resursbrist som påverkar, även inställning och om de finner ett nöje i det påverkar lärarnas uppfattning. Den punkt som band samman samtliga deltagare var dock den demokratiska aspekten; elever behöver lära sig programmering för att förstå vår värld som den är, och utvecklas till, det var inget tvivel om saken bland lärarna.

5.1 Avslutande diskussion

Jag gick in med en förväntan om en bred kompetensbrist, kanske föga förvånande då flera forskare och lärarorganisationer praktiskt taget skriker att det saknas kompetens inom IT¹¹. Jag blev dock glatt motbevisad; som framgår i resultatet och analysen kändes det först som att mina farhågor skulle besannas, men efter vidare utfrågning visade det sig att samtliga lärare ändå känner sig trygga nog i sin kompetens i förhållande till den nivån de undervisar. Kompetensbristen kan mycket väl vara sann, men för innevarande studies omfattning är det något som inte existerar i ett större omfång utan begränsas mer till att lärarna har kunskap om ett fåtal verktyg istället för en uppsättning av verktyg att använda i undervisningen. Om studien hade genomförts i högre åldrar är det möjligt att resultatet hade sett annorlunda ut, där både verktygen samt kunskapen som krävs är mer avancerade.

Förvånande var även den breda tron på att programmering skulle kunna mynna ut i ett datalogiskt tänkande eller rentav hjälpa andra ämnen genom transfereffekter, förvånande just på grund av hur enkel Blue-Bot är jämfört med en mer klassisk syn på programmering där ett programspråk som till exempel Java eller C används. Studien tar inte reda på om det faktiskt finns några av de nämnda effekterna, enbart hur lärarna uppfattar det. Den lärare som inte förutsatte några nämnvärda transfereffekter gav även en liknande förklaring som Kurland et al. (1986) ger där det handlar mer om individen än ämnena.

¹¹ Se kapitel 2.4

Föga förvånande var dock att tid och utrustning sågs som ett hinder, bland annat på grund av att kommunerna själva står för satsningen, dock var det en kommun där lärarna tog initiativ och fick tillåtelse att köpa in ett flertal robotar vilket ger en strimma av hopp för att flera ska få tillgång till mer utrustning för en bättre undervisning.

Angående möjligheter med programmering, så är det flera som tas upp; bland annat de kunskaper som genereras, men även att kunna använda programmering tillsammans med andra ämnen där de till exempel ligger som extra utmaningar längs en väg som roboten ska gå och behöver lösas innan de kan gå vidare. En av deltagarna fick även som hastigast Scratch förklarat för sig och såg en möjlighet att locka äldre elever genom att använda ett verktyg som är lite mer avancerat.

Jag ser även en liten korrelation mellan deltagarnas självskattade kompetens och deras undervisning, detta syns dock tydligare när det sätts i samband till deras inställning till programmering och hur det påverkar både hur de uttrycker sig kring sin egna kompetens och även i förhållande till undervisningen. Tre av fyra var mer optimistiska och den fjärde kände att samhällsutvecklingen i sig var på väg åt fel håll med för mycket fokus på digitalisering. Denna inställning tycks även avspeglas i hur programmering uppfattas och hur de förhåller sig till att lära sig mer avancerad kunskap. Det positiva att märka här är dock att oavsett hur de förhåller sig till programmering så strävar samtliga efter att ge eleverna undervisning i det, att skapa möjligheter för alla att stilla sin nyfikenhet.

5.2 Kunskapsbidrag

Den kvalitativa inriktningen gav mig möjlighet att få en djupare förståelse, att genom ett fenomenografiskt perspektiv ta reda på hur programmering uppfattas bland lärare. Eftersom programmering är ett ganska statiskt objekt känns det inte som att deras uppfattningar kring fenomenet skulle kunna avse olika typer av programmering, snarare mer hur de uppfattar samma objekt på olika sätt, vilket är just vad fenomenografien ämnar ta reda på.

Då det inte fanns någon större mängd forskning kring ämnet programmering i skolan och det har blivit väldigt aktuellt på grund av regeringsbeslutet att införa programmering senast sommaren 2018, såg jag ett behov där det gick att tillföra reliabel forskning, framförallt i de lägre skolåren. Studien är inte helt generaliserbar på grund av den kvalitativa inriktningen och en smärre mängd deltagare samt avgränsningen till lägre skolåren, men visar ändå på vissa röda trådar som kan antas existera i en större population.

5.3 Fortsatt forskning

Då jag i min studie fann att det inte alls råder den kompetensbrist som många menar, ser jag en stor nytta i att utöka upplägget till två olika områden:

1. Fortsätt studera kompetensnivån i de lägre skolåren, fast i en vidare omfattning än innevarande studie. Med fördel skulle enkäter kunna användas för att samla in ett stort antal svar, dock kan det även påverka svaren när det inte blir samma intima och mer spontana situation som en intervju kan generera.
2. Gå upp ett par klasser och undersök i större omfattning lågstadiet eller mellanstadiet; se om det finns ett mönster där kompetensbristen fortfarande är ett slags illusion eller om det är då den manifesterar sig.

Omnämmande

Stort tack till er som deltog för intervjuerna, det är ni som är kärnan av arbetet. Tack även till min handledare Katarina Groth Jansson som har hjälpt mig komma på rätt väg.

Speciellt tack till min sambo Katrin för att ha funnits där när jag har behövt ventilera eller få feedback under dessa månader. Tack även Alice, Valpen och Caesar för många gosiga skrivstunder!

Jonathan Axelsson

Referenslista

- Carr, W. (1998). *The curriculum in and for a democratic society*. Curriculum Studies, 6(3), 323-340. doi: 10.1080/14681369800200044
- Dalen, M. (2015). *Intervju som metod*. 2.uppl. Malmö: Gleerups Utbildning AB.
- Digitaliseringsrådet (2018). *För ett hållbart digitaliserat Sverige – en digitaliseringsstrategi*.
<http://www.digitaliseringsradet.se/digitaliseringsstrategin/> [2018-03-20]
- Google (u.å.). *CT Overview*.
<https://edu.google.com/resources/programs/exploring-computational-thinking/#!ct-overview> [2018-05-02]
- Grönlund, Å., Ranerup, A. & Gustavsson, P. (2003). *IT och demokrati - Erfarenheter och framtida vägar*. Delrapport till ITPS utvärdering av den svenska IT-politiken.
- Heintz, F., Mannila, L., Nygårds, K., Parnes, P. & Regnell, B. (2015). Computing at School in Sweden - Experiences from Introducing Computer Science within Existing Subjects. I A. Brodnik, & J. Vahrenhold (red.). *Informatics in Schools. Curricula, Competences, and Competitions /Lecture Notes in Computer Science and General Issues*. Ljubljana, Slovenia, September 28 to October 1 2015. Springer, ss. 118-130.
<https://www.ida.liu.se/divisions/aiics/publications/ISSEP-2015-Computing-At-School.pdf> [2018-06-08] doi:10.1007/978-3-319-25396-1_11
- International Society for Technology in Education [ISTE] & American Computer Science Teachers Association [CSTA] (2011). *Computational thinking in K-12 education*. 2.uppl.
http://c.ymcdn.com/sites/www.csteachers.org/resource/resmgr/472.11CTTeacherResources_2ed.pdf [2018-03-25]
- Internet live stats (2018). *Internet Users*.
<http://www.internetlivestats.com/internet-users/> [2018-03-20]
- Kalelioglu, F. & Gülbahar, Y. (2014). *The Effects of Teaching Programming via Scratch on Problem Solving Skills: A Discussion from Learners' Perspective*. Informatics In Education, 13(1), 33-50
- Kjällander, S., Åkerfeldt, A. & Petersen, P. (2016). *Översikt avseende forskning och erfarenheter kring programmering i förskola och grundskola*.

- Kurland, D.M., Pea, R.D., Clement, C. & Mawby, R. (1986). *A study of the development of programming ability and thinking skills in high school students*. Journal of educational computing research, 2(4), 429-458.
- Lundman, E. & Graneheim, U.H. (2012). Kvalitativ innehållsanalys. I Granskär, M. & Höglund-Nielsen, B. (red.). *Tillämpad kvalitativ forskning inom hälso- och sjukvård*. 2.uppl. Lund: Studentlitteratur AB, ss. 187-202.
- Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L. & Settle, A. (2014). Computational Thinking in K-9 Education. Proceedings of the Working Group Reports of the 2014 conference on *Innovation & Technology in Computer Science Education*. Uppsala, Sweden, June 23-25th, 2014. doi:10.1145/2713609.2713610
- Marton, F. (1981). *Phenomenography - describing conceptions of the world around us*. Instructional Science, 10, 177-200.
- Motion 1984/85:731. *Grundskolan och gymnasieskolan*. Stockholm: Folkpartiet
- Papert, S. (1980). *Mindstorms - Children, Computers, and Powerful Ideas*. Brighton: Harvester Press.
- Patel, R. & Davidson, B. (2011). *Forskningsmetodikens grunder*. 4.uppl. Lund: Studentlitteratur AB.
- Pea, R.D. & Kurland, D.M. (1984). *On the cognitive effects of learning computer programming*. New Ideas Psychology, 2(2), 137-168.
- Perkins, D.N. & Solomon, G. (1989). *Are Cognitive Skills Context-Bound?* Educational Researcher, 18(1), 16-25.
- Regeringen (2017a). *Nationell digitaliseringsstrategi för skolväsendet*. <http://www.regeringen.se/informationsmaterial/2017/10/regeringen-beslutar-om-nationell-digitaliseringsstrategi-for-skolvasendet/> [2018-03-24]
- Regeringen (2017b). *Stärkt digital kompetens i läroplaner och kursplaner*. <http://www.regeringen.se/pressmeddelanden/2017/03/starkt-digital-kompetens-i-laroplaner-och-kursplaner/> [2018-02-28]
- Rolandsson, L. (2015). *Programmed or not. A study about programming teachers' beliefs and intentions in relation to curriculum*. Diss. Stockholm: Kungliga tekniska högskolan.
- Segolsson, M. (2006). *Programmeringens intentionala objekt: Nio elevers uppfattningar av programmering*. Diss. Karlstad: Karlstads universitet.
- Skolverket (2016a). *IT-användning och IT-kompetens i skolan*. Stockholm: Skolverket.

- Skolverket (2016b). *Redovisning av uppdraget om att föreslå nationella it-strategier för skolväsendet – förändringar i läroplaner, kursplaner, ämnesplaner och examensmål*. Stockholm: Skolverket.
- Stiftelsen för kunskaps- och kompetensutveckling (2017). *Om KK-stiftelsen*. <http://www.kks.se/om-oss/om-kk-stiftelsen/> [2018-03-20]
- Tebelius, U., Aderklou, C. & Fritzdorf, L. (2003). *ITiS som incitament till skolutveckling: den nationella utvärderingen av IT i skolan : surveystudien, 2000-2003*.
- Trost, J. (2010). *Kvalitativa intervjuer*. 4.uppl. Lund: Studentlitteratur.
- Voogt, J., Fisser, P., Good, J., Mishra, P. & Yadav, A. (2015). *Computational thinking in compulsory education: Towards an agenda for research and practice*. *Education and Information Technologies* 20, Issue 4, pp 715-728.
- Wing, M.J. (2006). *Computational Thinking*. *Communications of the ACM*, 49(3), 33-35.

Bilagor

Bilaga 1. Intervjuguide

Hur länge har du tjänstgjort som lärare?

Vilken årskurs undervisar du?

Hur många elever är det åt gången?

Var du verksam under ITiS?

1. Vad innebär programmering för dig?
2. Hade du en annan uppfattning om programmering innan du började med det i undervisningen?
3. Ser du några möjligheter med programmering?
4. Hur upplever du att eleverna reagerar på programmering?
5. De barn som påbörjat programmering, kan du se någon skillnad i hur de agerar jämfört med barn som inte testat på det ännu?
6. Känns eleverna mer självsäkra efter att ha börjat programmering än innan de prövade på det?
7. Har du någon uppfattning om varför programmering ska läras ut?
8. Anser du att elever bör lära sig programmering?
9. Vad upplever du att eleverna lär sig genom att programmera?
10. Upplever du att programmering genererar kunskaper som kan användas mer generellt?
11. Ser du någon skillnad mellan eleverna med avseende på deras sociala bakgrund, kön eller liknande i förhållande till programmering?
12. Finns det några hinder som försvårar arbetet med programmering i undervisningen?
13. Har ni utrustningen som krävs för undervisningen?
14. Jämfört med ITiS, har ni mer/mindre resurser nu?
15. Använder du något speciellt i undervisning för att lära ut programmering?
16. Har du fått någon utbildning som ska hjälpa din undervisning?
17. Känner du att du kan tillräckligt mycket om programmering för din undervisning?
18. Jämfört med ITiS, upplever du ett annorlunda stöd nu? Bättre/sämre
19. Är det något du vill tillägga som inte tagits upp?

Bilaga 2. Intervjufrågor efter ämnesområde

Generella problem med programmering

12. Finns det några hinder som försvårar arbetet med programmering i undervisningen?
13. Har ni utrustningen som krävs för undervisningen?
14. Jämfört med ITiS, har ni mer/mindre resurser nu?

Generella möjligheter med programmering

3. Ser du några möjligheter med programmering?

Programmering

1. Vad innebär programmering för dig?
2. Hade du en annan uppfattning om programmering innan du började med det i undervisningen?
4. Hur upplever du att eleverna reagerar på programmering?
15. Använder du något speciellt i undervisning för att lära ut programmering?

Datalogiskt tänkande

5. De barn som påbörjat programmering, kan du se någon skillnad i hur de agerar jämfört med barn som inte testat på det ännu?
6. Känns eleverna mer självsäkra efter att ha börjat programmering än innan de prövade på det?

Transfereffekter

9. Vad upplever du att eleverna lär sig genom att programmera?
10. Upplever du att programmering genererar kunskaper som kan användas mer generellt?

Kompetens

16. Har du fått någon utbildning som ska hjälpa din undervisning?
17. Känner du att du kan tillräckligt mycket om programmering för din undervisning?
18. Jämfört med ITiS, upplever du ett annorlunda stöd nu? Bättre/sämre

Demokratiskt samhälle

7. Har du någon uppfattning om varför programmering ska läras ut?
8. Anser du att elever bör lära sig programmering?
11. Ser du någon skillnad mellan eleverna med avseende på deras sociala bakgrund, kön eller liknande i förhållande till programmering?

Övrigt

19. Är det något du vill tillägga som inte tagits upp?