

SKOLPORTENS NUMRERADE ARTIKELSERIE
FÖR UNDERVISNING, LÄRANDE OCH LEDARSKAP

PROGRAMMERINGS- UNDERVISNINGENS EFFEKTER ENLIGT LÄRARNAS SJÄLVA

En effektstudie av möjligheter
och utmaningar med
programmeringsundervisning

FÖRFATTARE:

Helén Viebke



SKOLPORTEN

LEDA & LÄRA

6/2020

SAMMANFATTNING

DENNA STUDIE KAN förhoppningsvis bidra med ökade kunskaper kring vilken effekt programmeringsundervisning har, ur ett lärarperspektiv och ge en bild av olika programmeringsmetoders effekter. Rektorer och lärare kan använda artikeln som ett discussionsunderlag för utvecklingen av programmeringsundervisning på i sin skola eller i sin kommun. Studien är gjord i en mellanstor kommun där lärare från årskurs 1 till och med årskurs 9 systematiskt har dokumenterat sina lektioner i programmering. De främsta effekterna i studien visar bl.a. att pojkar och flickor är lika aktiva, eleverna är mer aktiva generellt och de hjälper varandra mer än vanligt. Eleverna verkar även öka sin problemlösningsförmåga, enligt lärarna i studien. Dock verkar inte datalogiska begrepp inom programmering ännu vara kända begrepp för lärarna. Ett fortsatt behov av kompetensutveckling och beprövad verksamhet inom området behövs för att utveckla programmeringsundervisningen och dess didaktiska utveckling i den svenska skolan.

Helén Viebke är utvecklingsstrateg och arbetar på Barn- och utbildningsförvaltningen i Åstorp. E-post: helen.viebke@astorp.se

Denna artikel har den 14 september 2020 accepterats för publicering i Skolportens numrerade artikelserie för utvecklingsarbete i skolan. Artikeln har granskats av en forskare som ingår i Skolportens granskargrupp.

Fri kopieringsrätt i ickekommersiellt syfte för kompetensutveckling eller undervisning i skolan och förskolan under förutsättning att författarens namn och artikelns titel anges, samt källa: Skolportens artikelserie. I övrigt gäller copyright för författaren och Skolporten AB gemensamt.

Denna artikel är publicerad i Skolportens artikelserie Leda & Lära:
www.skolporten.se/forskning/utveckling/

Aktuella Författaranvisningar & Skrivregler:
www.skolporten.se/forskning/skolutveckling/skolportens-utvecklingsartiklar/

Vill du också skriva en utvecklingsartikel? Mejla till redaktionen@skolporten.se

INNEHÅLL

INLEDNING, SYFTE OCH FRÅGESTÄLLNINGAR	7
Datalogiskt tänkande	8
Effekter av programmering i undervisningen	8
METOD OCH GENOMFÖRANDE	11
METODER OCH URVAL	13
Beskrivning av den digitala dokumentationen	13
Styrkor och svagheter med de olika insamlingsmetoderna	15
RESULTAT	17
Resultat av förundersökning hösten 2018 och våren 2019	17
Starkaste indikationerna på eleveffekter	18
Svagaste indikationerna på eleveffekter	20
Programmeringsmetodernas olika effekter	20
Möjligheter och utmaningar med programmeringsundervisning	21
DISKUSSION	23
Metoddiskussion	23
Starkaste och svagaste indikationerna	23
Möjligheter och utmaningar med att implementera och följa upp programmeringsundervisning	24
REFERENSLISTA	27
APPENDIX	31
Bilaga 1: Effektaggar	31
Bilaga 2: Intervjuguide	31
Bilaga 3: Enkät	32

INLEDNING, SYFTE OCH FRÅGE- STÄLLNINGAR

DET FINNS ÄNNU inga större studier i Sverige vilket lärande elever utvecklar när de har lektioner med programmeringsinslag. Den forskning som finns är främst internationell, vilken fokuserar på det datalogiska lärandet och mycket lite på det generella lärandet. Ofta fokuserar forskarna även på enskilda programmeringsmetoder och deras effekter (Zhang & Nouri 2019; Kalelioğlu 2015; Moreno-León, Robles och Román-González 2016). Från lärare och elever kommer signaler om att programmering medför ett positivt lärande där alla eller flertalet elever är motiverade och visar på ökad samarbetsförmåga. Är det för att programmering är nytt och spännande eller finns det ingredienser i undervisningen som medför ett tillsynes mer engagerat lärande? Följande artikel kommer att belysa vilka effekter lärarna tagit fasta på när det gäller vad elever lär sig och utvecklar under en programmeringslektion. Artikeln kommer att visa vilka datalogiska kompetenser eleverna utvecklar samt vilka övriga kompetenser och förmågor eleverna verkar använda och utveckla. Studien kommer även att visa resultat vad olika val av programmeringsmetoder ger för effekt i elevernas lärande.

Syftet med studien är att undersöka lärarnas uppfattningar kring vad eleverna lär sig under lektioner med programmeringsinslag och vilka möjligheter och utmaningar som finns i införandet av programmering i undervisningen.

Under 2017–2020 genomfördes Ifous-programmet *Programmering i ämnesundervisningen*. Stockholms universitet med forskare inom programmering och datalogi har deltagit tillsammans med 5 skolhuvudmän inklusive Åstorps kommun där samtliga grundskolor

samt vuxenutbildningen (IM) ingår. Artikeln utgår från dokumenterade lektioner i Åstorps kommun. Syftet med FoU-programmet är att utveckla didaktiska arbetssätt, uppgifter och innehåll för undervisning i programmering.

Från och med hösten 2018 infördes programmering som en del i läroplanens kursplaner i matematik och teknik (Lgr 11 2018) vilket är en del av Skolväsendets nationella digitaliseringsstrategi (Utbildningsdepartementet 2017). Det handlar inte bara om att lära eleverna att programmera och att utveckla en datalogisk förmåga och datalogiskt tänkande utan också att ge dem förutsättningar att som demokratiska medborgare kunna använda och förstå digitala system och tjänster. Mannila (2017) beskriver hur programmering bör vara en del av allmänbildningen i ett allt mer digitaliserat samhälle, och poängterar skolans roll i att ge alla lika möjligheter att under sin skolgång tillgodogöra sig kunskaper och förståelse av det digitaliserade samhället, dess möjligheter och begränsningar. Här poängteras att en väsentlig del av förståelsen handlar om en insikt om vilka uppsättningar av regler som ligger till grund för digitala system och algoritmers roll i samhällsutvecklingen. Mannila (2017) menar vidare att det saknas långvarig forskning om programmering i skolan och systematisk forskning kring vilka programmeringsmetoder och modeller som ger vilket lärande samt vilken progression som kan stödja lärarnas planering och genomförande av undervisning i programmering. Enligt Olsson (2018) är aktionsforskning en av flera metoder för att pröva och bepröva undervisningen med vetenskaplig grund. Man formulerar frågor kring sin undervisningsprak-

tik tillsammans i ett arbetslag eller på en skola, sedan startar man upp undervisningsprocesser som man kollegialt följer upp, dokumenterar och reflekterar över

vad som sker och därigenom får kunskap om den gemensamma praktiken.

DATALOGISKT TÄNKANDE

DATALOGISKT TÄNKANDE ÄR en av de kompetenser som lyfts fram när man pratar om vad elever ska lära sig av programmering och används som argument för införandet av programmering i skolan. Begreppet datalogiskt tänkande kommer bl. a. från matematikern Seymour Papert som på 1980-talet fick ett stort internationellt genomslag tillsammans med programmeringsspråket Logo som Papert var med och skapande. Papert menade att när barn lär sig programmera utvecklar de kunskaper och förmågor, såsom ett abstrakt, logiskt och analytiskt tänkande, som även kan användas inom andra områden. (Lilja, Godhe & PlayerKoro 2016).

Inför läroplansrevideringen 2018 (Lgr 11 2018) skapades det ett samarbetsprojekt mellan Skolverket, universiteten, företrädare för skolan och mjukvaruindustrin som fick namnet Tripple Helix. Under ett antal möten och workshops diskuterades innehåll och formuleringar i den kommande läroplansrevideringen. Universitetens datavetare och deltagare från industrin framförde vikten av att låta elever utveckla ett så kallat datalogiskt tänkande. Kontentan av samarbetet kan sammanfattas i nedanstående citat.

Vi menar att fokus ska vara datalogiskt tänkande som en naturlig del av alla ämnen och ha ett fokus på experimenterande så att möjligheterna med mjukvara och teknik blir konkret och spännande. Datalogiskt tänkande innefattar generella färdigheter som har sin grund i datavetenskap som att dela upp och lösa problem, hitta mönster, tänka logiskt samt arbeta strukturerat och kreativt. (Runeson, Andersson, Heintz, Mannila & Rolandsson 2015 se Björklund 2019, s 6)

Flera forskare menar även att många lärare redan ger elever möjlighet att utveckla dessa ovan nämnda förmågor i den undervisning som sker idag (Heintz, Mannila, Nygårds, Parnes & Regnell 2015; Mannila, Dagiene, Demo, Grgurina, Mirolo, Rolandsson & Settle 2014). Heintz et al. (2015) menar att synen på programmering bör vidgas från att ge barn och elever möjlighet att endast skriva kod till att även betona förmågor som analytiskt och abstrakt tänkande samt problemlösning. Begreppet datalogiskt tänkande bör etableras som en uppsättning allmänna färdigheter där programmering kan vara ett verktyg för att träna dessa (ibid.). Nygårds (2015) menar att färdigheterna och förmågorna bör användas på ett tvärvetenskapligt sätt, genom hela läroplanen.

EFFEKTER AV PROGRAMMERING I UNDERVISNINGEN

EN OMDEBATTERAD FRÅGA är om, och i så fall vilka effekter programmering i skolan leder till. Olika forskningsrapporter visar på olika resultat (Mannila 2017). I Sverige, liksom i Finland, undervisas programmering ämnesintegrerat. Detta innebär att programmering ingår som en del i andra skolämnen (I Sverige t.ex. i matematik och teknik). I vissa länder, exempelvis

i England, är programmering ett eget ämne. Frågan är om ämnesintegreringen av programmering i den svenska skolan innebär att vi hoppas på en så kallad transfereffekt? Vilket skulle innebära att t.ex. det datalogiska tänkandet och problemlösningens förmågan, som sägs utvecklas av undervisningen i programmeringen, kommer till nytta i elevens lärande även i andra äm-

nen. Transfereffekt är ett begrepp som nämns i programmeringssammanhang (Kjällander, Petersen & Åkerfeldt 2016). Det uppstår när eleverna kan ta med sig förvärvat ämnesspecifik kunskap från ett ämne till ett annat. Som t.ex. när elever tränar att stegvis lösa ett problem genom programmering under en tekniklektion så kan det innebära att eleven känner igen utmaningen och kan överföra den kunskapen under en matematiklektion med problemlösningssuppgifter. Transfer är en kognitiv process och för att den ska vara möjlig måste eleven se likheter mellan de olika lärsituationerna.

Stolpe (2018) menar att eleverna inte per automatik utvecklar ett datalogiskt tänkande när de programmerar. Hon säger bl. a:

Flera studier visar att elever bara marginellt, om ens alls, utvecklar sin mer generella förmåga att till exempel lösa problem och tänka logiskt (Brenna, 2015; Kalelioglu & Gülbahar, 2014; Straw, Bamford & Styles, 2017; Sullivan & Heffernan, 2016). Eleverna lär sig dock att hantera de digitala verktyg som de får öva på att använda sig av i skolan. Elever som får arbeta med att programmera med hjälp av Scratch blir bättre på att lösa uppgifter i Scratch. Det betyder att det som eleverna får öva på, det blir de också bättre på. Forskningen visar också att det är svårt för eleverna att överföra sina kunskaper till andra sammanhang. Det handlar både om när eleverna byter programmeringsmiljö och när de ska tillämpa sina kunskaper i situationer som inte alls handlar om programmering. Det verkar alltså som att det kanske finns en övertro bland somliga för vad programmering faktiskt ska kunna bidra med. (Stolpe 2018, s 10)

Vad som skulle utgöra en bra grund i undervisningen för att uppnå en transfereffekt av datalogiskt tänkande är inte funnet i denna studie.

Ett exempel på studie som visat på att programmering i skolan leder till positiva effekter på elevernas problemlösningsförmåga är Moreno-León, Robles och Román-González (2016) studie där de bl.a. genom ett utvecklat online verktyg, Dr Scratch, mäter elevernas utveckling av datalogiskt tänkande. Verktøget visar hur avancerat eleverna programmerar och vilken nivå i datalogiskt tänkande de är på. I detta projekt såg man framförallt en utveckling hos de äldre eleverna (åk 6) i logiskt tänkande, användar-interaktivitet, datarepresentation, flow kontroll och problemlösning. Andra studier visar att programmering inte leder till att eleverna blir bättre på t.ex. problemlösning efter programmeringsundervisningen. (Kalelioğlu & Gülbahar 2014). Dock visar dessa olika studier på hur viktig lärarens roll är för undervisningens effekt. Lärarens attityd till ämnet och förmåga att bygga upp stödstrukturer kring eleverna är väsentligt för att eleverna ska kunna utveckla sina kunskaper inom programmering.

METOD OCH GENOMFÖRANDE

SYFTET MED STUDIEN är att undersöka lärarnas uppfattning kring vad eleverna lär sig under lektioner med programmeringsinslag och vilka möjligheter och utmaningar som finns i införandet av programmering i undervisningen. Huvudundersökningen i studien är genomförd under hösten 2019, medan förundersökningarna är genomförda under hösten 2018 respektive våren 2019 och finns med som bakgrund och referensmaterial för att stärka resultaten. Hela processen är genomförd i Åstorps kommuns grundskolor i samband med deltagande i Ifous-programmet *Programmering i ämnesundervisning*. Under hösten 2018 gjordes en mindre förundersökning av elevernas lärande under lektioner i programmering med hjälp av ett internetbaserade IT-stöd, *LoopMe* som bl.a. forskaren Martin Lackeus på Chalmers högskola varit med och arbetat fram (Lackeus 2018; Lackeus, Sävétun & Westlund 2020). En inledande del i förarbetet har varit att bedöma hur IT-stödet och metodiken kring detta verktyg skulle kunna ge ett starkt och nyanserat resultat med tillgång till både kvalitativa och kvantitativa data. Data erhålls genom att lärare digitalt dokumenterar sina reflektioner och upplevda resultat i IT-stödet *LoopMe*. Resultatet från förundersökningarna var mycket positivt. Bland annat visades sig IT-stödet vara tidseffektivt och enkelt för lärarna att dokumentera i. Det gav ett synliggörande av effekter och görande i undervisningen som var kommunicerbart på både individ- och gruppnivå. Detta medförde att både uppdrag och effekter kunde förbättras tillsammans med lärarna. En utökad undersökning genomfördes under vårterminen 2019. Under processen från hösten 2018 och våren 2019 gjordes justeringar, bl.a. i form av ändrade effekter och utökning av antal respondenter samt mängden dokumenterade lektioner. En ”tagg” i den digitala dokumentationen i IT-stödet är en indikator eller en effekt som man tagit fram ur en

hypotes, erfarenhet och/eller forskning därav namnet effekttagg i denna studie. Effekttaggarna används i efterföljande analys för att kunna fånga samband mellan orsaker och effekter, där orsaken i denna studie är en lektion i programmering, i olika åldrar och med olika programmeringsmetoder. Effekttaggarna fångar sedan vad lärarna uppfattar att eleverna lär. En lektion dokumenteras som en loop i IT-stödet (Lackeus, Sävétun & Westlund 2020). Efter våren 2019 gjordes ytterligare justeringar, bl. a. tillsammans med forskaren Jalal Nouri (forskningsledare i Ifous-programmet, Stockholms Universitet) inför huvudundersökningen. Frågeställningarna i de så kallade uppdragen (se bild 2) justerades och effekttaggarna utökades med forskningsbaserade taggar (Brennan & Resnick 2012). De 33 effekttaggarna (se bilaga 1) utgör de beroende variablerna. Effekttaggarna mäter elevers datalogiska tänkande, samt elevernas lärande av mer generell art såsom engagemang, samarbetsförmåga, uthållighet, etc. De åtta oberoende variablerna (se bild 1) utgörs av didaktiska metoder inom programmeringsundervisning som används i de dokumenterade lektionerna med elever.

METODER OCH URVAL

FÖR ATT UNDERSÖKA det studerande fenomenet har tre olika datainsamlingsmetoder kombinerats: digital dokumentation genom IT-stöd, gruppintervju och enkät. Datainsamling och analys-metod består av fem olika steg: (1) digital datainsamling av lärarnas reflektioner, effekttaggarna och känslotillstånd, (2) förberedelse och genomförande av gruppintervjuer samt transkribering av intervjuer med utvalda intervjugrupper (3) tillverkning av enkät, genomförande av enkät, (4) sortering av data, (5) analys av genererad data från digital datainsamling, intervju och enkät.

Den digitala dokumentationen, som dokumenterats via det så kallade IT-stödet, är den centrala resultatinsamlingsmetoden i studien. Undersökningarna hösten 2018 och våren 2019 fungerade enbart som förstudier, som nämnts ovan. IT-stödet har använts av lärare som har dokumenterat sina lektioner i programmering. Under hösten 2018 är respondenterna

i IT-stödet 12 lärare från Ifous-programmet. Under våren 2019 och hösten 2019 är det 48 lärare i Åstorp som har deltagit som respondenter med hjälp av IT-stödet. Fyra gruppintervjuer har genomförts. Urvalet till intervjurespondenterna utgörs av matematik- och NO-lärare. Två grupper på två högstadieskolor (totalt 13 lärare) samt två grupper på lågstadiet (totalt 19 lärare). Denna spridning har valts ut för att fördjupa förståelsen av de olika åldrarnas resultat av kvantitativ och kvalitativa data som erhållits innan intervjuerna. Gruppintervjuerna varade i ca 60 minuter vardera. En enkät har genomförts med fem korta frågor kring programmering. Den har besvarats av de 32 lärare som deltagit under intervjuerna (bilaga 2). De deltagande lärarna (inkl. skolans ledning) har informerats om studiens syfte och att det insamlade materialet har oidentifierats och presenteras på gruppnivå.

BESKRIVNING AV DEN DIGITALA DOKUMENTATIONEN

EFTER ATT LÄRARNÄ genomfört sin programmeringslektion går de in i IT-stödet via en dator eller en applikation i mobiltelefonen och dokumenterar lektionen. I IT-stödet väljs den programmeringsmetod de använt under lektionen (se bild 1) och därefter besvarar lärarna de frågor som finns i uppdraget (se bild 2). Detta har utformats för att undersöka lärarnas uppfattningar och reflektioner kring vad elever lärt sig under programmeringslektionen (kvalitativ data). Lärarna väljer sedan och "klickar i" effekttaggarna och de så kallade känslotaggarna. Effekttaggarna visar vad lärarna uppfattar att eleverna lär sig, d.v.s. förmågor,

kompetenser och attityder som eleverna visar under lektionen (kvantitativ data, se Bilaga 1). Känslotaggarna mäter lärarnas känsloläge (sinnesstämning) med hjälp av glada och ledsna emojis (Se bild 3).

Resultatet av den digitala dokumentationen genom IT-stödet har tagits fram ur kategorisering av de olika programmeringsmetoderna, ålder på elever, de olika effekttaggarna och kvalitativa data som består av lärarnas reflektioner kring lektioner samt känslotaggarna. En summering är gjord av antalet lektioner (loopar), antal effekttaggarna, procentuell fördelning av andel effekttaggarna som använts och vilka effekttaggarna

PROGRAMMERINGSMETODER SOM LÄRARNAS INLEDNINGSVIS VÄLJER I IT-STÖDET LOOPME.

1. Analog programmering – utan digitala hjälpmedel
2. Robotprogrammering – Bluebot, Beebots, Blue-bot, Lego mindstorms, etc.
3. Blockprogrammering utan struktur i Scratch och liknande
– utan den inbyggda förbestämda lektions-struktur som finns i code.org
4. Blockprogrammering med struktur- Lektionen ska innehålla en inbyggd förbestämd lektionsstruktur/arbetssätt, så som det är i code.org.
5. Mikrokontrollers - arbete med Microbit, Arduino eller liknande.
6. Textbaserad programmering - i språk som Python, Java, osv.
7. Kombinerade didaktiska verktyg - Block + textbaserad programmering genom exempelvis mikrokontrollers.
8. Övrig didaktisk metod - Exempelvis Excel eller Google kalkylblad)

BILD 2: LÄRARNAS UPPDRAG I IT-STÖDET LOOPME OAVSETT VILKEN PROGRAMMERINGSMETOD DE VALT ATT ARBETA MED.

▼ ⓘ 01. Analog Programmering: Reflektera kring genomförd lektion ⓘ

Genomför en valfri lektion med analog programmering ihop med elever, alltså helt utan digitala hjälpmedel.

Berätta först kortfattat här i LoopMe vad ni gjorde. Ange vilket programmeringsverktyg som användes av eleverna. Reflektera sedan kring hur det fungerade för dig samt för eleverna. Utgå i din reflektion så mycket du kan från följande frågor:

Varför tror du att det blev som det blev?
Vilka effekter kunde du se hos eleverna?
Hur tog du reda på vad eleverna lärt sig?
Var det något som överraskade dig?

Uppföljningsfråga: Vad upplevde du att eleverna hade lätt för respektive svårt för?

som använts flest antal gånger samt minst antal gånger tillsammans med de olika programmeringsmetoderna.

För att kunna göra en kvalitativ analys av det insamlade materialet är det till stor del öppna frågor som använts med låg grad av strukturering av svar (Patel & Davidsson 1994). De två informanterna gavs stort utrymme att svara fritt i både IT-stödet, enkäter och under intervjuerna. Frågorna i enkäten och gruppintervjuerna är utformade för att spegla lärarnas tankar och uppfattningar om programmering, programmeringslektioner och elevernas lärande.

Intervjufrågorna är öppna och standardiserade (Patel & Davidsson 1994). Följdfrågor användes för att klargöra eller fördjupa svaren. Intervjuerna har spelats in digitalt till ljudfiler med hjälp av en mobiltelefon. Samtliga intervjuerna har skett på plats, ute på den

skola där respondenterna arbetar. De intervjuade har inte fått se intervjufrågorna i förväg. På så sätt har de inte haft någon möjlighet att förbereda sina svar. Det är respondenternas spontana åsikter, reaktioner och svar på frågorna som efterfrågats. Materialet är noggrant genomlyssnat och transkriberat. Svaren har kategoriserats för att få fram mönster och slutsatser. Intervjuerna har även bearbetats så att citat från intervjuerna kan förtydliga enskilda respondenters uppfattningar och erfarenheter i resultatåtergivningen.

Enkätundersökningen genomfördes med hjälp av en analog enkät med fem frågor för relatera resultatet till de starkaste respektive svagaste eleveffekterna. De individuella enkätsvaren har sammanställts manuellt i en Excel-fil där sammanställningar gjorts för att underlätta analys av svaren.

BILD 3: IT-STÖDETS APPLIKATION I MOBIL MED UPPDRAG, UPPFÖLJNINGSPRÅGA OCH TAGGAR.

Uppdrag

Uppföljningsfråga

Känslotaggar

Effekttaggar

STYRKOR OCH SVAGHETER MED DE OLIKA INSAMLINGSMETODERNA

GENOM ATT ANVÄNDA flera metoder för att undersöka ett fenomen, så kallad triangulering, kan resultatet från studien styrkas. Genom att lyfta fram resultatet av samtliga dokumenterade lektioner till lärarna under gruppintervjuerna kunde dessa matchas och ge svar på om lärarna uppfattade att IT-stödets digitala dokumentation stämde överens med deras generella uppfattning eller inte. Lärarna har även kunnat ge sin syn på vad de tror ligger till grund för resultatet.

Fördelen med dokumentationen från IT-stödet är att även med ett stort antal respondenter erhåller man ett systematiserat och väldokumenterat kvantitativt och kvalitativt underlag. IT-stödets styrka är att man får möjlighet att samla in en stor mängd data samtidigt som det ger möjlighet att svara direkt till den enskilda individens data och där kunna få förtydliganden av eventuellt oklara svar. Det är enkelt och tidseffektivt för lärarna att dokumentera i IT-stödet via dator eller app. i mobiltelefonen. IT-stödets dataunderlag är dessutom enkelt att hantera, sortera och analysera.

Gruppintervjuns data är avsedd att vara ett komplement till IT-stödets data. Styrkan i detta är att det

ges möjlighet att upptäcka detaljer som förbisetts i dokumentationen och den ökar förståelsen för respondenternas perspektiv. Intervjun ger i viss utsträckning möjlighet att kontrollera om man förstått saker rätt, klara ut missförstånd och fördjupa de svar som getts i IT-stödet (Cato & Björndal 2005). Gruppintervjun har styrkan att få fleras synpunkter på en och samma gång. Svaghet med gruppintervjun är att en del är tysta och andra tar mer talutrymme. Svagheten med intervjuer är att de tar tid att förbereda och bearbeta genom att lyssna igenom inspelat material och transkribera det till text. Man riskerar även att påverka de som intervjuas så att den information man får färgas av egna uppfattningar. Informationsinhämtningen är ett växelspel mellan intervjuare och deltagare vilket är en större utmaning när man intervjuar en grupp än när man intervjuar en enskild individ. Cato & Björndal (2005) menar att enkätundersökning i jämförelse med intervjun som metod har den fördelen att informationen blir registrerad i skriftlig form då respondenten avger sina svar. En möjlig svaghet är frågornas konstruktion.

RESULTAT

NEDAN REDOVISAS RESULTATET av undersökningen. Utifrån analysen av materialet som samlats in med hjälp av tre valda metoderna har jag valt att dela in resultatredovisningen under rubrikerna:

- ★ Resultat förundersökning hösten 2018 och våren 2019

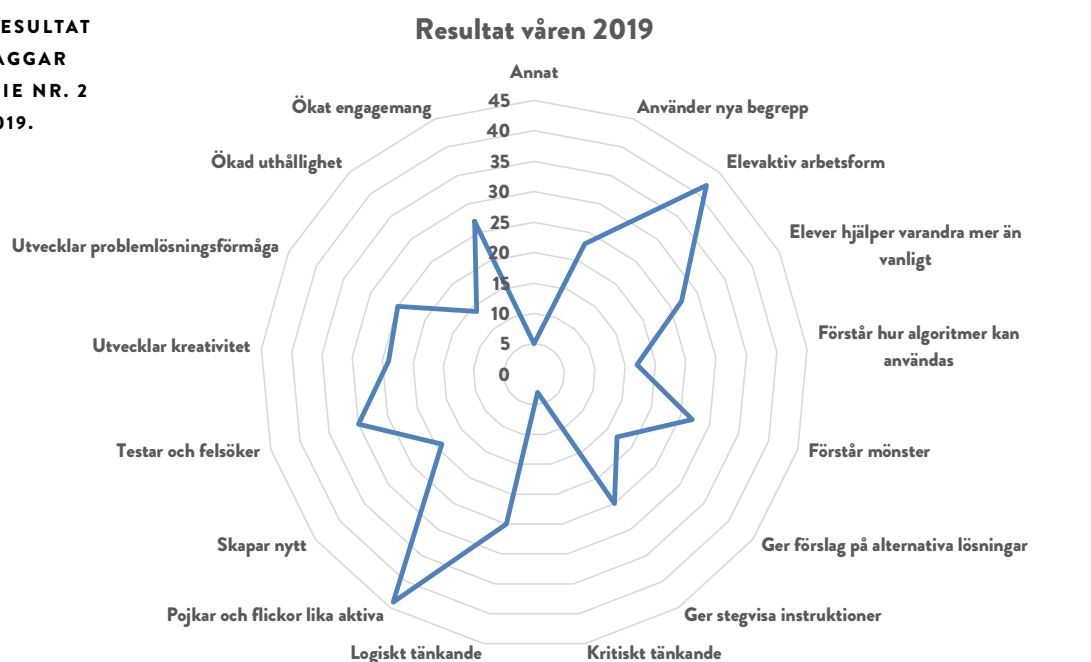
- ★ Starkaste indikationerna på eleffekter
- ★ Svagaste indikationerna på eleffekter
- ★ Programmeringsmetodernas olika effekter
- ★ Möjligheter och utmaningar att undervisa i programmering

RESULTAT AV FÖRUNDERÖKNING HÖSTEN 2018 OCH VÅREN 2019

UNDER FÖRUNDERÖKNINGEN 2018 arbetade vi med frågan: Vad lär sig eleverna när de programmerar förutom programmering? Resultatet visade främst att eleverna samarbetar, använder logiskt tänkande, visar att de kan lösa problem, omsätter idéer i handling och kan ge stegvisa instruktioner. Detta resultat jämfördes bl.a. med Skolverkets samlade forskning kring effekt

av digital undervisning (Kjällander, Petersen & Åkerfeldt 2016) och Utbildningsutskottets utredning, *Digitalisering i skolan - dess påverkan på kvalitet, likvärdighet och resultat i utbildningen* (Riksdagen 2015) vilket medförde en revidering av effekttaggarna under våren 2019 och ett nytt resultat i förstudie nummer 2 (se bild 4).

**BILD 4: RESULTAT
EFFEKTTAGGAR
FÖRSTUDIE NR. 2
VÅREN 2019.**



TABELL 1: EFFEKTER MED VALDA PROGRAMMERINGSMETODER I STUDIEN AV 68 LEKTIONER HÖSTEN 2019.								
Effekttaggar	Analog Programmering	Robotprogrammering	Blockprogrammering UTAN struktur	Blockprogrammering MED struktur	Mikrocontrollers	Övrig didaktisk metod	Fria loopar	Andel loopar där effekttaggen har använts:
Elever ansluter till sociala nätverk och samarbetar när de programmerar	0%	0%	0%	17%	0%	0%	0%	1%
Elever använder operatörer (< > & ! etc)	0%	0%	8%	0%	0%	0%	0%	1%
Elever ger stegvisa instruktioner	60%	48%	69%	33%	100%	20%	100%	54%
Elever gör modulariserad kod	5%	0%	8%	0%	0%	0%	0%	3%
Elever kan arbeta inkrementellt och iterativt (öka stegvis och upprepa)	5%	5%	0%	0%	0%	20%	0%	4%
Elever kan gestalta och uttrycka sig själva genom programmering	5%	0%	0%	33%	0%	0%	0%	4%
Elever kan modularisera (dela upp)	10%	0%	8%	0%	0%	0%	0%	4%
Elever kan återanvända och remixa kod	0%	0%	8%	17%	0%	0%	0%	3%
Elever ger förslag på alternativa lösningar	55%	43%	69%	67%	100%	40%	50%	54%
Elever hjälper varandra mer än vanligt	50%	48%	62%	67%	100%	80%	0%	54%
Elever testar och felsöker	50%	76%	77%	67%	100%	100%	50%	69%
Elever utvecklar sin kreativitet	50%	48%	62%	83%	100%	40%	0%	53%
Elever utvecklar sin problemlösningsförmåga	40%	71%	77%	50%	100%	80%	50%	62%
Elever utövar logiskt tänkande	30%	62%	85%	33%	100%	60%	50%	54%
Elever visar ökat engagemang	55%	48%	69%	33%	100%	60%	0%	53%
Elever är mer aktiva	70%	57%	69%	50%	100%	60%	0%	62%
Pojkar och flickor är lika aktiva	80%	81%	100%	83%	100%	100%	50%	85%
Annat	0%	10%	0%	0%	0%	0%	0%	3%
Totalt antal valda effekttaggar	174	184	156	64	16	51	9	68 loopar
Antal loopar (lektioner)	20	21	13	6	1	5	2	68
Medelantal effekttaggar per loop	9	9	12	11	16	10	5	10
Känslomått*	1,2	1,3	1,1	1,5	1,0	0,8	1,0	1,1

*Känslomåttet går mellan -2 och 2

STARKASTE INDIKATIONERNA PÅ ELEVEFFEKTER

EFTER DE BÅDA förstudierna genomfördes huvudstudien med utgångspunkt av 68 lektioner i programmering. Lektionerna benämns som loopar i IT-stödet och visas i resultatet nedan.

Resultatet från den digitala dokumentationen (tabell 1), redovisar de effekter som taggats i mer än 53 procent av tillfällena och de som taggats mindre än 5 procent av tillfällena. Lärarna väljer mer frekvent (83,5 procent) de taggar som lärarna själv varit med och tagit fram genom prövning i Ifous-programmet än de taggar som är mer forskningsbaserade och av datalogisk karaktär* (se bilaga 1). De åtta första effekterna i tabell 1 är av mer datalogisk karaktär. Resultatet från de 68 dokumenterade lektioner hösten 2019 stämmer väl överens med resultatet av de 48 dokumenterade lektionerna under förundersökningen våren 2019.

De fem främsta effekterna i elevers lärande i den digitala dokumentationen är:

- ★ Pojkar och flickor är lika aktiva (85 procent)
- ★ Elever testar och felsöker (69 procent)
- ★ Elever utvecklar sin problemlösningsförmåga (62 procent)
- ★ Elever är mer aktiva (62 procent)
- ★ Elever hjälper varandra mer än vanligt (54 procent)

Effekterna av elevernas lärande och uppvisande av förmågor under programmeringslektioner har visat sig vara både av datalogisk karaktär och generell lärande karaktär.

* Datalogisk karaktär benämns här med de effekttaggar som har fokus på datalogiskt kunnande såsom kritiskt tänkande, ansluta till sociala nätverk och samarbetar när de programmerar, använda operatörer (< > & ! etc.), använda slingor/loopar, använda villkorssatser, ge stegvisa instruktioner, göra händelsestyrd kod, göra modulariserad kod och hantera data med variabler.

Pojkar och flickor är lika aktiva taggas i 85 procent av samtliga lektioner. Flertalet lärare håller under intervjuerna med om att pojkar och flickor är lika aktiva under programmerings-lektionerna oavsett programmeringsmetod. I den digitala dokumentationens resultat framkommer ingen skillnad och vid intervjuerna säger mer än 2/3 av lärarna att de inte ser någon skillnad i aktiviteten på pojkar och flickor. Dock säger 1/3 av lärarna att de ser tydliga skillnader beroende av kön i själva uppgiftsutförandet.

"Flickorna letar fram de gulligaste figurerna och ska ha vissa färger medan pojkarna vill spela och är mer tävlingsinriktade" (Från intervju med undervisande lärare F-3).

Någon högstadielärare belyser att hen inte uppfattat skillnad hos könen hos de grupper hen har. I dessa fall säger läraren att det är framförallt i de välfungerande grupperna som skillnaden inte märks.

Testa och felsöka samt att ge stegvisa instruktioner är de datalogiska färdigheter som framträder mest. Effekten **Elever testar och felsöker** taggas i 69 procent av samtliga lektioner och förekommer framförallt vid robotprogrammering och blockprogrammering utan struktur. Detta kan jämföras med effekten **Eleven utvecklar sin problemlösningsförmåga** som taggats i 62 procent av samtliga lektioner och förekommer vid robotprogrammering, blockprogrammering utan struktur och övrig didaktisk metod (Google kalkylark). Här finns även en diskrepans mellan lågstadielärare respektive högstadielärare syn på vad problemlösning är.

I intervjuerna framkommer det att lågstadielärarna anser att testa och felsöka är en del av problemlösningen. Detta verkar inte vara den allmänna uppfattningen inom högstadiegrupperna. I intervjuerna med lärarna i högstadiegrupperna säger man att problemlösningsförmågan beror väldigt mycket på typ av uppgifter och programmeringsspråk. En högstadielärare säger:

"Elever testar och felsöker men tveksamt hur mycket problemlösning som sker" (Från intervju med undervisande lärare 7-9).

Elever är mer aktiva har taggats i 62 procent av samtliga lektioner hösten 2019 (jmf 57 procent våren 2019) fördelat på alla programmeringsmetoder. I

den digitala dokumentationens kvalitativa kommentarer beskriver lärarna i positiva ordalag om hur eleverna har tyckt att det varit roligt med programmeringslektionerna, varit motiverade och visat en hög samarbetsvilja och skaparglädje. Flera lärare har blivit överraskade över hur snabbt deras elever lärde sig programmering och flera säger att alla elever vågade pröva sig fram.

"Även de mest svaga eller omotiverade eleverna skapade program efter sin förmåga genom att testa sig fram" (Från intervju med undervisande lärare 7-9).

Lärarkommentarer från metoden blockprogrammering med struktur i den digitala dokumentationen genom IT-stödet, beskriver hur läraren som inte är så säker på användningen av Scratch JR blir en jämlike med eleverna och att detta skapade ett samarbetsklimat i klassrummet där många elever hjälptes åt.

"Alla börjar på noll" (Från intervju med undervisande lärare F-3).

I intervjuerna med lågstadielärarna tror de att elevernas engagemang beror på att programmering är nytt och att det blir "mer fria" vilket gör att eleverna vågar sig på problemen.

"Efter en bugg testar de hur mycket som helst" (Från intervju med undervisande lärare F-3).

Lärarna lyfter även fram den omedelbara feedbackens påverkan på elevernas engagemang och motivation.

"I programmeringen får eleverna omedelbart resultat. Omgående feedback. Detta att jämföra med i t ex matematiken där läraren är den som ger rätt svar" (Från intervju med undervisande lärare F-3).

En av högstadielärarna tror att engagemanget även kan bero på att eleverna får göra något kreativt och att eleverna får arbeta i projekt (tidsavgränsad arbetsuppgift) med problemlösning där eleverna får hitta sina egna metoder och vägar. En större andel lärare, både på lågstadiet och högstadiet, säger i intervjuerna att lärarna känner att de inte räcker till och att detta troligen gör att eleverna engagerar sig mer och hjälper varandra för att komma vidare.

SVAGASTE INDIKATIONERNA PÅ ELEVEFFEKTER

DE FYRA SVAGASTE indikationerna på eleveffekter visar sig vara de effekter av mer datalogisk karaktär.

- ★ Elever ansluter till sociala nätverk och samarbetar när de programmerar (1 procent)
- ★ Elever använder operatorer (< > & ! etc) (1 procent)
- ★ Elever gör modulariserad kod (3 procent)
- ★ Elever kan återanvända och remixa kod (3 procent)

Det är tydligt att de datalogiska effekterna överlag är minst dokumenterade av lärarna. Lärarna säger sig känna brist på kunskap och kompetens när det gäller datalogiska begrepp och fördjupad kompetens i programmering. Lärarna menar att detta är orsaken till att de inte har valt att dokumentera effekterna

med datalogiska begrepp. I Intervjuerna säger bland annat lärarna:

"Vi använder inte begrepp som iterativt, modulariserad kod etc" (Från intervju med undervisande lärare 7-9).

"Vi förstår inte dessa begrepp och därför har vi inte klickat i dem" (Från intervju med undervisande lärare 7-9).

Lärarna hänvisar till att det i läroplanen står problemlösning som begrepp men att t.ex. iterativt, modulariserad kod, remixa kod inte nämns i läroplanen. Lärarna, framförallt på högstadiet, uppfattar att de inte "äger" språket för programmering och att taggar av datalogisk karaktär är otydliga samt att det är för många taggar att förhålla sig till.

PROGRAMMERINGSMETODERNAS OLIKA EFFEKTER

I NEDANSTÅENDE TABELL förtydligas lärarnas val av de specifika programmeringsmetoderna som kompletterar till tabell 1. Den visar vilka de olika metodernas främsta effekter är och i vilka årskurser lektionerna har genomförts.

I samtliga programmeringsmetoder är effekten "pojkar och flickor lika aktiva" markerad flest antal gånger i IT-stödet. Endast tre av programmeringsmetoderna visar på likhet i de främsta effekterna. Det är robotprogrammering, blockprogrammering utan struktur samt övrig didaktisk metod. Dessa tre metoder visade även främst resultat i effekterna:

- ★ Elever testar och felsöker
- ★ Elever utvecklar sin problemlösningsförmåga

I de yngre åren används till stor del analog programmering (16 av 20 lektioner i årskurs F-3) där elevernas programmeringsuppgift oftast är att "programmera" varandra. Vid dessa lektioner får de i

uppgift att förflytta sina kamrater genom rummet med stegvisa instruktioner som är så tydliga som möjligt. Även robotprogrammering (ex robotarna BeetBot, Blue-bot, Ozobot) används till stor del i årskurs F-3 (16 av 21 lektioner). I årskurs 4-6 arbetar man med analog programmering (4 av 25 lektioner), robotprogrammering (7 av 25 lektioner) och med blockprogrammering (Scratch) (12 av 25 lektioner). Blockprogrammering utan struktur och med struktur har valts som programmeringsmetod av lärare som använder Scratch och Scratch Jr. På högstadiet har man använt Mikrokontrollers eller Google Kalkylblad i de lektioner som dokumenterats digitalt genom IT-stödet.

TABELL 2: FÖRDELNING AV DE OLIKA PROGRAMMERINGSMETODERNAS FRÄMSTA EFFEKTER.

Pojkar och flickor lika aktiva	Antal lektioner	Fördelning Årskurs	Främsta effekter (antal taggningar/ antal lektioner)	
Elever ger stegvisa instruktioner	20	F-3 16 st 4-6 4 st	Pojkar och flickor lika aktiva Elever är mer aktiva Elever visar ökat engagemang Elever ger förslag på alternativa lösningar	16/20 14/20 11/20 11/20
Elever ger förslag på alternativa lösningar	21	F-3 16 st 4-6 7 st Blandade åldersgruppervid 2 tillfälle	Pojkar och flickor lika aktiva Elever testar och felsöker Elever utvecklar sin problemlösn.förmåga Elever utövar logiskt tänkande	17/21 16/21 15/21 13/21
Elever hjälper varandra mer än vanligt	13	F-3 2 st 4-6 7 st 7-9 2 st år 3 + år 5 2 st	Pojkar och flickor lika aktiva Elever utövar logiskt tänkande Elever utvecklar sin problemlösn.förmåga Elever testar och felsöker	13/13 11/13 10/13 10/13
+ 12 taggar till 1/1	6	4-6 5 st 7-9 1 st	Pojkar och flickor lika aktiva Elever skapar nytt Elever utvecklar sin kreativitet	5/6 5/6 5/6
Mikrokontrollers	1	4-6 1 st	Pojkar och flickor lika aktiva Elever ger stegvisa instruktioner Elever ger förslag på alternativa lösningar Elever hjälper varandra mer än vanligt + 12 taggar till 1/1	1/1 1/1 1/1 1/1
Textbaserad programmering	0	5	5x30 min	0
Kombinerade didaktiska verktyg	0	5	5x30 min	0
Övrig didaktisk metod (ex Google Kalkylark	5	1-6 1st 4-6 1 st 7-9 3 st	Pojkar och flickor lika aktiva Elever testar och felsöker Elever hjälper varandra mer än vanligt Elever utvecklar sin problemlösn. förmåga	5/5 5/5 4/5 4/5
Fria Loopar = ej valt programmeringsmetod	2	F-3 2 st	Elever ger stegvisa instruktioner (Metod programmera människor att gå)	2/2
Totalt antal lektioner	68			

MÖJLIGHETER OCH UTMANINGAR MED PROGRAMMERINGSUNDERVISNING

LÄRARNAS REFLEKTIONER I den digitala dokumentationen tillsammans med intervjuerna belyser vilka möjligheter och utmaningar som lärarna möter när de undervisar i programmering. De positiva reaktionerna från lärarna handlar om att de uppfattar att eleverna tycker att det är roligt att arbeta med programmering. Lärarna skriver i den digitala dokumentationen genom IT-stödet om hur eleverna var moti-

verade, visade en hög samarbetsvilja och skaparglädje under lektionerna.

”Eleverna verkade tycka att det var roligt, de ville genast börja och höll på länge” (Från intervju med undervisande lärare F-3).

Både i den digitala dokumentationen i IT-stödet och under intervjuerna säger flera lärare att de har blivit överraskade över hur lätt och snabbt deras

elever har att lära sig programmering. En lärare på mellanstadiet beskriver att eleverna fått göra en utvärdering av lektionen med en s.k. EXIT-ticket där eleverna bl.a. skrivit att de längtade till nästa programmeringslektion.

Möjligheter lärare har identifierat är att de elever som annars behöver mycket stöd av läraren under en ”vanlig lektion” verkar klara av programmeringslektionerna bättre. Det gäller såväl elever med bristande språkkunskaper som elever med inlärningssvårigheter. En högstadielärare skriver:

”Jag tror att de omotiverade eller kanske snarare osäkra eleverna tyckte att uppgiften var mer begriplig och rolig. Det fanns inget rätt eller fel utan de kunde prova sig fram och fick direkt feedback eftersom idén antingen fungerade eller inte” (Från intervju med undervisande lärare 7-9).

Lärarna menar att hjälpsamheten, samarbetet och viligheten att ta emot hjälp klart verkar öka under programmeringslektionerna. Utmaningar som lärare har identifierat för de yngre eleverna är att de inte alltid kan höger och vänster och att elevers bristande språkkunskaper ibland är ett hinder. Inom samtliga metoder och åldrar skriver lärarna i IT-stödet att eleverna är ovana vid att läsa, ge och följa instruktioner. I samtliga undersökningar säger lärarna att tiden inte räcker till. Lektionstiden, särskilt för högstadiet, är för

kort och tid behövs för att hitta bra uppgifter och skapa en lektionsbank med uppgifter på olika nivåer. Samt att ha tid själv för att lära sig om programmering och fördjupa sina kunskaper som lärare.

När lärarna får svara på frågan vad som behövs för att de ska kunna bli bättre och bekvämare i att undervisa programmering är det framförallt tre saker som sticker ut:

- ★ Tid att hjälpa och lära varandra kollegialt.
- ★ En slags arbetsbeskrivning och en bank med bra uppgifter för att garantera någon slags likvärdighet vad eleverna ska lära sig. Gärna en bild på progressionen i elevernas lärande i programmering.
- ★ Fungerande teknik.

För att skapa likvärdighet bör fokus vara på grundläggande förutsättningar som teknik och kompetensutveckling. Utan dessa är det en utmaning för en enskild lärare att utveckla undervisningen i programmering och skillnaderna blir stora. När lärarna själva fått utvärdera vad som hjälpt dem mest med att använda IT i undervisningen rankas kollegialt utbyte och stöd inom skolan högst. Många studier visar att kollegialt samarbete är en nyckelfaktor för lyckad skolutveckling liksom betydelsen av stöttning från ledningen (Skolverket 2018).

DISKUSSION

METODDISKUSSION

DEN DIGITALA DOKUMENTATIONEN genom IT-stödet har visats sig vara systematisk, dokumenterad, kommunicerbar och möjlig att kritiskt granska. I den digitala dokumentationen är både kvantitativ och kvalitativ data tillgänglig när lärarna beskriver undervisningen som skett i programmering. Att som undersökningsledare kunna följa, svara och ställa frågor till varje lärare ger en möjlighet att få en tydlig bild av undervisningens effekter och förutsättningar. Lärarnas skrivna reflektioner tillsammans med redovisade effektaggarna ger en fördjupad bild av deras undervisning.

I den digitala dokumentationen, genom IT-stödet, har effektaggarna tagits fram genom prövning samt befintlig forskning inom området av elevers lärande i programmering (Brennan & Resnick 2012). Urvalet av taggar kan begränsa resultatet av effekter i elevernas lärande. Utgångspunkten i taggarna har därför prövats och omprövats och reviderats i tre omgångar

och ökat i antal inför den sista omgången av dokumenterade lektioner. Det har även funnits möjlighet för läraren att tagga effekten "annat" för att säkerställa fler möjliga effekter. Dock är det enbart 2 taggningar av 654 på "annat" vilket verkar visa att studien har täckt in effekterna av elevernas lärande relativt väl. Att få effekterna på "förhand" kan göra det lättare att upptäcka och sätta ord på det man själv upplevt av sin lektion. Men det verkar som att begreppen behöver vara begripliga för att man ska välja dem. I Brennan och Resnicks forskning (2012) omvandlade man elevsvaren i sina intervjuer till datalogiska begrepp och händelser. I detta fall hade man behövt göra tvärtom för att se om lärarna upptäckt det datalogiska tänkandet och görandet i undervisningen. Att ha ord och begrepp för att upptäcka, reflektera, analysera och förstå sammanhang är viktigt oavsett ålder och kontext.

STARKASTE OCH SVAGASTE INDIKATIONERNA

STUDIENS RESULTAT VISAR på fem starka indikationer och fyra svaga indikationer på elevers lärande. Den mest utmärkande är att **pojkar och flickor är lika aktiva** vid samtliga olika programmeringsmetoder. Forskning som är gjord inom området visar att programmeringsundervisning kan påverka stereotypa könsmonster i lärandet (Berner 2003). Kjällander et al. (2018) lyfter frågan om jämställdhet i sina studier och menar att skillnaden mellan könen är liten när det kommer till programmering. Frågan är vad det är som gör att pojkar och flickor är lika aktiva under programmeringslektionerna? Kan upplägget av

programmeringsundervisningen som bygger på att eleverna får lösa problem, arbeta praktiskt, självständigt och tillsammans med andra samt få omedelbar feedback, vara de ingredienser som påverkar elevernas motivation, nyfikenhet, uthållighet och stereotypa könsmonster? Lärarna säger i intervjuerna att eleverna har dialog med varandra under lektionerna, de prövar sig fram, de är fysiskt aktiva och eleverna ser omedelbara resultat i sitt provande. Är detta ett resultat av en förändrad lärarrollen? Flera lärare i denna studie säger att de "tar ett steg tillbaka" och låter eleverna vara "friare" i lärandet och förhållan-

det lärare- elev blir mer jämlikt i lärmiljön. Invanda beteenden och förväntningar verkar förändras och eleverna vågar prova sig fram mer utan att "leta efter ett rätt svar i facit". Forskning visar att autonomi, känsla av att bemästra uppgiften och att ha ett tydligt syfte med uppgiften är avgörande faktorer för den inre motivationen. Även känslan av framsteg är viktig för den inre motivationen (Thomas 2009; Thomson & Wery 2013). Något i programmeringsundervisningen verkar frigöra är elevernas mod och förmåga att ta för sig i lärandet, oavsett kön eller utmaning i lärandet.

I denna studie är det tydligt att de effektaggar som handlar om elevernas förmåga att träna och ut-

veckla datalogiskt tänkande är minst taggade, vilket kan bero på att lärarna känner brist på kunskap och kompetens när det gäller datalogiska begrepp. De har inte det vetenskapliga språket och den fördjupade kompetensen i programmering, vilket gör att de inte har internaliserat dessa begrepp i sin vardagspraktik. Lärarna tror själv att detta är orsaken till att de inte har taggat effekterna av mer "datalogisk karaktär". De datalogiska begreppen som finns med i taggarna är troligen för avancerade för den nivå av programmering som lärare och elever nu befinner sig på.

MÖJLIGHETER OCH UTMANINGAR MED ATT IMPLEMENTERA OCH FÖLJA UPP PROGRAMMERINGSUNDERVISNING

PROGRAMMERING HAR INFÖRTS som obligatoriskt ämne i mer än 16 europeiska länder. Implementering och uppföljning av införandet ser mycket olika ut. I vårt grannland Danmark gjordes en satsning en satsning på bred nationell nivå med fokus på gemensam teknik, en så kallad Ultra:bit (en slags micro:bit), som successivt införts från årskurs 4 och uppåt. Samtidigt har man satsat på kompetensutveckling i programmering för lärare under en längre tid. Uppföljningen i Danmark visar på positiv respons från både elever och lärare där det bl. a. visat sig att redan efter två år har 83 procent av lärarna börjat att använda programmering i undervisningen och 95 procent av lärarna säger att eleverna är intresserade av att lära sig mer om programmering och digital teknologi (Pålsson 2018).

England var det första Europeiska land som införde programmering i läroplanen 2014 och skälet är det samma som i Sverige – brist på rätt kompetens och se till att elever får en samhällsmedborgerlig kompetens som är nödvändig i den nya digitaliserade världen. Dock har man i England sett att elevernas nyfikenhet och lust att lära programmering drastiskt har minskat (Wohl 2018). Orsaken till nedgången menar Wohl (2018) beror på innehållet i undervisningen. Det förbereder inte unga för verkligheten. Barnen upplever inte att de lär sig något de har nytta av.

Min forskning visar att elever är frustrerade framför inspirerade av förändringen i läroplanen. Det finns saker barn och unga faktiskt vill lära sig om datorer, kunskap som gör direkt nytta, som att kunna nätsäkerhet, rensa datorn på virus, kort-kommandon, Excel och andra mjukvaruprogram, men det ingår inte i kurserna i dag. (Wohl 2018)

Wohl (2018) pekar ut två viktiga lärdomar från den engelska erfarenheten:

- ★ Att innan implementering veta vad det är man vill att programmering i läroplanen ska uppnå. Vad barnen ska lära sig, och varför? Vet man inte det så är det svårt att utvärdera sedan.
- ★ Att innan införandet ha en period av intensiv träning för de lärare som berörs. För många lärare är det helt ny kunskap som ska inhämtas och ämnet som sådant kan vara svårt att lära ut. Utfallet av satsningen kan påverkas av att lärare inte känner sig redo.

Implementeringen av programmering i ämnesundervisningen i Åstorp verkar ha kommit en bit på vägen. I dokumentationen blir det tydligt vilka skolor som har kommit igång och vilka skolor som har en bit kvar för att det ska kännas stabilt och inte endast vara ett ansvar på enskilda lärare.

Vi har fortfarande inte tillräckligt med stöd i forskningen för hur bra relevant programmeringsundervisning bör gå till. Vi behöver diskutera didaktikens frågor. Vad är det vi vill att eleverna ska lära sig och varför? Hur tränar vi eleverna att utveckla de datalogiska förmågorna både med och utan programmering? Hur kan vi använda oss av elevernas nyfikenhet och engagemang som de visar under programmeringslektionerna? Hur undviker vi digitalt yttflyt? Vad behövs i undervisningen för att transfereffekter skall uppnås? Programmeringsundervisningen bör utgå från elevernas vardag så att eleverna ser relevansen i innehåll och metod (Wohl 2018). Risken finns annars att programmering blir ett isolerat innehåll som ska checkas av, istället för att bli en relevant och intressant ingång för att arbeta med t.ex. teknikämnets alla delar

eller utvidga matematikämnets kunskapsområden. För att kunna besvara ovanstående frågor behövs kollegiala samtal, fler studier och mer beprövad undervisning.

I framtida studier av programmering i undervisningen vore det intressant att med stöd av digital dokumentation genom IT-stödet LoopMe, låta lärare och elever parallellt reflektera över undervisningen och dess effekter i lärandet. Den digitala dokumentationen tillsammans med kollegiala diskussioner skulle kunna ge en väl beprövad undervisning, kollegialt lärande mellan skolhuvudmän och grund för forskning. Olsson (2018) menar att pröva och bepröva det nya man inför i undervisningen borde vara en självklar del i en verksamhet som skall bygga på vetenskaplig grund och beprövad erfarenhet.

REFERENSLISTA

- ★ Berner, B. (2003). *Vem tillhör tekniken?: kunskap och kön i teknikens värld*. Lund: Arkiv förlag.
- ★ Björklund, L. (2019). Datalogiskt tänkande: Ett förhoppningsvis baserat argument för att införa programmering i skolan. *Föreningen svensk undervisning*, 4 maj. <https://undervisningshistoria.se/datalogiskt-tankande-ett-forhoppningsbaserat-argument-for-att-infora-programmering-i-skolan/>
- ★ Brennan, K. & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *In Proceedings of the 2012 Annual Meeting of the American Educational Research Association*, Vancouver, Canada (pp. 1-25).
- ★ Cato, R. & Björndal, P. (2005). *Det värderande ögat*. Stockholm: Liber.
- ★ Heintz, F., Mannila, L., Nygårds, K., Parnes, P. & Regnell, B. (2015). Computing at school in Sweden – experiences from introducing computer science within existing subjects. *In International Conference on Informatics in Schools, Evolution and Perspectives* (pp.118-130) Springer, Cham. <http://www.diva-portal.org/smash/get/diva2:1005958/FULLTEXT01.pdf>
- ★ Kalelioğlu, F. & Gülbahar Y. (2014). The Effects of Teaching Programming via Scratch on Problem Solving Skills: A Discussion from Learners' Perspective. *Informatics in Education*, 2014, Vol. 13, No. 1, 33–50. Vilnius: Vilnius University. <https://files.eric.ed.gov/fulltext/EJ1064285.pdf>
- ★ Kalelioğlu, F. (2015). Ett nytt sätt att undervisa programmeringsfärdigheter för K-12 studenter: Code.org. *Datorer i mänskligt beteende*. Volym 52, s 200-210. <https://doi.org/10.1016/j.chb.2015.05.047>
- ★ Kjällander, S., Peteresen, P. & Åkerfeldt, A. (2016). *Översikt avseende forskning och erfarenheter kring programmering i förskola och grundskola*. Stockholm: Skolverket.
- ★ Kjällander, S., Selander, S. & Åkerfeldt, A. (2018). *Programmering: introduktion till digital kompetens i grundskolan*. Första upplagan. Stockholm: Liber.
- ★ Lackeus, M. (2018). *Collecting Digital Research Data through Social Media Platforms*. Göteborg: Chalmers universitet.
- ★ Lackeus, M., Sävetun, C. & Westlund, C. (2020). *Lärares vetenskapliga lärande med IT-stöd – vad, varför, hur?* Göteborg: Me Analytics.
- ★ Lgr 11 (2018). *Läroplan för grundskolan samt för förskoleklassen och fritidshemmet: reviderad 2018*. Stockholm: Skolverket.
- ★ Lilja, P., Godhe, A-L. & PlayerKoro, C. (2016). Programmeringens återkomst. *Lärande och It-gruppens blogg* [blogg], 11 mars. <https://lit.blogg.gu.se/2016/03/11/programmering1/> [2020-05-20]

- ★ Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L. & Settle, A. (2014). Computational Thinking in K-9 Education. *ITiCSE '14. Föreläsningar av konferensen om innovation & teknik inom datavetenskaplig utbildning*. s. 1-29. ITiCSE. Uppsala 23 juni 2014. <https://doi.org/10.1145/2713609.2713610>
- ★ Mannila, L. (2017). *Att undervisa i programmering i skolan*, Lund: Studentlitteratur.
- ★ Moreno-León, J., Robles, G. & Román-González, M. (2016). Code to Learn: Where Does It Belong in the K-12 Curriculum? *Journal of Information Technology Education: Research, Volume 15*, 283-303. <http://www.jite.org/documents/Vol15/JITEv15ResearchP283-303Moreno2233.pdf>
- ★ Nygårds, K. (2015). *Koden till digital kompetens*. Stockholm: Natur & Kultur.
- ★ Olsson, L. M. (2018). *Från tyst kunskap till beprövad erfarenhet*. Stockholm: Lärarförlaget.
- ★ Patel, R. & Davidson, B. (1994). *Forskningsmetodikens grunder*. Lund: Studentlitteratur.
- ★ Pålsson, S. (2018). Programmering och digitalt skapande i skolan. *Veckans spaning vecka 40* [blogg], 5 oktober. [http://www.spaningen.se/veckans-spaning-vecka-40-programmering-och-digitalt-skapande-i-skolan/\[2020-05-20\]](http://www.spaningen.se/veckans-spaning-vecka-40-programmering-och-digitalt-skapande-i-skolan/[2020-05-20])
- ★ Riksdagen (2015). Rapport 2015/16:RFR18. *Digitaliseringen i skolan – dess påverkan på kvalitet, likvärdighet och resultat i utbildningen*. Stockholm. https://www.riksdagen.se/sv/dokument-lagar/dokument/rapport-fran-riksdagen/digitaliseringen-i-skolan--dess-paverkan-pa_H30WRFR18
- ★ Runeson, P., Andersson, S., Heintz, F., Mannila, L. & Rolandsson, L. (2015). *Hur skall det svenska utbildnings-systemet möta framtidens utmaningar? Helhetssyn på svensk utbildning om digitalisering och programmering*. Swedsoft. <http://swedsoft.se/wp-content/uploads/sites/7/2015/05/2015-01-Utbildning-white-paper.pdf>
- ★ Skolverket (2018), *Digitaliseringen i skolan-möjligheter och utmaningar*. Stockholm.
- ★ Stolpe, K. (2018) Programmering- elever blir bra på det de övar! *CETIS Nyhetsbrev*, nr 1, ss.10-11. <https://liu.se/cetis/nyhetsbrev/2018-1-programmering.shtml>
- ★ Straw, S., Bamford, S., & Styles, B. (2017). *Randomised controlled trial and process evaluation of code clubs*. Slough: NFER.
- ★ Sullivan, F. R., & Heffernan, J. (2016). Robotic Construction Kits as Computational Manipulatives for Learning in the STEM Disciplines. *Journal of Research on Technology in Education*, 48(2), 105-128.
- ★ Thomas, K.W. (2009). *Intrinsic motivation at work*. 2.uppl. Oakland: Berrett-Koehler.
- ★ Thomson, M. & Wery, J. (2013). Motivational strategies to enhance effective learning in teaching struggling students, British Journal of Learning Support. *Support for Learning*, v28 n3 p103-108 aug 2013. North Carolina State University: North Carolina. https://www.academia.edu/22362520/Motivational_strategies_to_enhance_effective_learning_in_teaching_struggling_students

- ★ Utbildningsdepartementet (2017). Nationell digitaliseringsstrategi för skolväsendet. Stockholm.
<https://www.regeringen.se/4a9d9a/contentassets/00b3d9118b0144f6bb95302f3e08d11c/nationell-digitaliseringsstrategi-for-skolvasendet.pdf>
- ★ Wohl, B. (2018). Varför ska barn lära sig programmera i skolan? *Dagens arena*, 22 mars. <https://www.dagensarena.se/innehall/varfor-ska-barn-lara-sig-programmera-skolan/>
- ★ Zhang, L. & Nouri, J. (2019). *A systematic review of learning computational thinking through Scratch in K-9*. Department of Computer and Systems Sciences, Stockholm: Stockholms universitet.

APPENDIX

BILAGA 1: EFFEKTTAGGAR

DEN ENA HÄLFTEN av effekttaggarna är framtagen med stöd av Åstorps IFOUS-lärargrups resonemang kring forskningsartiklars visade eleffekter samt egen prövning av effekter i elevernas lärande. Den andra hälften utgår framförallt från Brennans och Resnicks forskning (2012) utifrån datalogiska kompetenser.

Ifous-gruppens effekttaggarna är effekter av mer generell lärande karaktär. Det är färdigheter, attityder och s.k. "mjuka" förmågor som engagemang, uthållighet etc.

- ★ Pojkar och flickor är lika aktiva,
- ★ Elever är mer aktiva,
- ★ E visar ökat engagemang,
- ★ E visar ökad uthållighet,
- ★ E utövar logiskt tänkande,
- ★ E utvecklar sin problemlösningsförmåga,
- ★ E utvecklar sin kreativitet,
- ★ E testar och felsöker,
- ★ E stärker sin pedagogiska kommunikationsförmåga,
- ★ E skapar nytt,
- ★ E hjälper varandra mer än vanligt,
- ★ E ger förslag på alternativa lösningar,
- ★ E förstår mönster,
- ★ E använder nya begrepp,
- ★ Annat

De datalogiska effekttaggarna är de som visar effekter på elevers datalogiska tänkande och görande;

- ★ Elever utövar kritiskt tänkande,
- ★ E ansluter till sociala nätverk och samarbetar,
- ★ E använder operatorer (< > & ! etc),
- ★ E använder slingor/loopar,
- ★ E använder villkorssatser,
- ★ E ger stegvisa instruktioner,
- ★ E gör händelsestyrd kod,
- ★ E gör modulariserad kod,
- ★ E hanterar data med variabler,
- ★ E kan abstrahera (dölja, förenkla),
- ★ E kan arbeta inkrementellt och iterativt (öka stegvis),
- ★ E kan gestalta och uttrycka sig själva genom programmering,
- ★ E kan modularisera (dela upp),
- ★ E kan återanvända och remixa kod,
- ★ E programmerar parallella aktiviteter

De 15 senast uppräknade effekttaggarna har lagts in med en förklarande bilaga i IT-stödet.

Övriga taggar: Årskurs F-3, Årskurs 4-6, Årskurs 7-9 (Komvux)

BILAGA 2: INTERVJUGUIDE

INLEDNINGSFRÅGOR:

1. Har alla haft lektion med programmering?
2. Vad är den största utmaningen med att ha lektion i programmering? Nämn 3 saker.
3. Nytt för elever att kunna programmering?

POWERPOINT VISAS MED RESULTAT FRÅN IT-STÖDETS DOKUMENTATION VT. 2019 OCH HT. 2019.

Vad ser ni för resultat i dessa bilder med diagram?

Vad tänker ni ligger bakom de mest i klickade taggarna?

BILD 6 KÄNSLOLÄGE +2 TILL -2

Lågstadieskola Y medelkänsla +2

Högstadieskola X medelkänsla +1

Medelkänsla samtliga skolor + 1,4

4. Vad tror ni att detta beror på?

BILD 8 OCH 9 STARKAST INDIKATIONER

- ★ Pojkar och flickor lika aktiva
 - ★ Elever testar och felsöker
 - ★ Elever utvecklar sin problemlösningsförmåga
 - ★ Elever är mer aktiva
 - ★ Elever hjälper varandra mer än vanligt
5. Stämmer denna bild med er uppfattning?

6. Är flickor och pojkar lika aktiva på samma sätt?
7. Vad tror ni gör att eleverna är mer engagerade/motiverade under programmeringslektioner?

BILD 8 OCH 9 SVAGASTE INDIKATIONER

- ★ Elever ansluter till sociala nätverk och samarbetar när de programmerar
 - ★ Elever använder operatorer (< > & ! etc)
 - ★ Elever gör modulariserad kod
 - ★ Elever kan återanvända och remixa kod
8. Varför tror ni att dessa är de svagaste indikatorerna?

BILAGA 3: ENKÄT

ENKÄT IFOUSPROJEKT PROGRAMMERING I ÄMNESUNDERVISNINGEN

Detta material tillsammans med gruppintervjun kommer att behandlas helt konfidentiellt av mig Helen Viebke. Intervjun kommer att kopplas till det dokumenterade resultat av 128 lektioner i programmering som Åstorps lärare gjort 2018–2019 för att visa på vad elever lär under programmeringslektioner och vad lärare tänker och känner inom detta område.

Det är frivilligt att skriva namn:

1. Vad tänker du när du hör ordet programmering?
2. Vad känner du inför att ha lektioner i programmering?
3. Vilket tycker du är det främsta skälet till att elever ska lära sig om programmering i skolan?
4. Vad är beprövad erfarenhet enligt dig?
5. BILD Effekter av elevers lärande i programmering – stämmer detta enligt din uppfattning?

Starkast indikation på eleveffekter

- ★ Pojkar och flickor är lika aktiva (85 % av looparna har denna tagg)
- ★ Elever testar och felsöker (69 % av looparna har denna tagg)
- ★ Elever utvecklar sin problemlösningsförmåga (62 % av looparna har denna tagg)
- ★ Elever är mer aktiva (62 % av looparna har denna tagg)

Svagast indikation på eleveffekter

- ★ Elever ansluter till sociala nätverk och samarbetar när de programmerar (1 % av looparna har denna tagg)
- ★ Elever använder operatorer (< > & ! etc) (1 % av looparna har denna tagg)
- ★ Elever gör modulariserad kod (3 % av looparna har denna tagg)
- ★ Elever kan återanvända och remixa kod (3 % av looparna har denna tagg)



SKOLPORTEN