



PROGRAMMERING I SKOLAN VAD, HUR, NÄR OCH VARFÖR?

2020:5 – Slutrapport från FoU-programmet
Programmering i ämnesundervisningen

Ifous rapportserie 2020:5

Stockholm, oktober 2020

ISBN: 978-91-985535-7-4

Redaktör: Anette Jahnke

Fotografier: Ola Jacobsen (omslagsfoton), Jalal Nouri (i kapitel
Forskningsinsatsens genomförande och resultat)

Grafisk form & produktion: Per Isaksson

Ansvarig utgivare: Ifous

Fri kopieringsrätt i ickekommersiellt syfte för kompetensutveckling
eller undervisning i skolan och förskolan under förutsättning att
författarens namn och rapportens titel anges, samt källa. I övrigt
gäller copyright för författarna och Ifous AB gemensamt.

INNEHÅLL

Förord	5
Sammanfattning	7
<i>Anette Jahnke, projekt- och processledare Ifous</i>	
Detta är FoU-programmet Programmering i ämnesundervisningen	9
Fem huvudmän tog täten	11
Freinetskolan Hugin: Basnivån hos eleverna högre och spjutspetsarna skjuter än högre	13
<i>Teresa Sundström, lokal processledare Freinetskolan Hugin</i>	
Stockholm: Föräldrar hör av sig – barnen berättar om programmering	17
<i>Per Anderhag, central processledare, lektor FoU-enheten Stockholm stad</i>	
Simrishamn: Eleverna mer positiva till programmering ju mer undervisning de får	21
<i>Pia Svensson, rektor, Simrishamns kommun</i>	
Tyresö: Eleverna har lärt sig begrepp och har precis som lärarna nu ett gemensamt språk	23
<i>Sara Vikslund, central processledare, IT-strateg, Tyresö kommun</i>	
Åstorp: Alla elever i kommunen har provat på programmering	27
<i>Annika Hoppe, ordförande i styrgruppen, förvaltningschef. Åstorps kommun</i>	
Fem huvudmäns utvecklingsarbete	31
<i>Anette Jahnke, projekt- och processledare Ifous</i>	
Forskningsinsatsens genomförande och resultat	35
<i>Jalal Nouri, docent, Stockholms universitet</i>	
Utvecklade lektioner med hjälp av Lesson studies	51
<i>Anette Jahnke, projekt- och processledare Ifous</i>	
Lärares utvecklingsarbete i artiklar	57
<i>Anette Jahnke, projekt- och processledare Ifous</i>	
Slutsatser från utvärderingen	59
<i>Lejla Mundzic, Emerga Institute</i>	
Litteraturlista	61
Bilaga 1: Planeringsmall inför prövning av lektion	65
Bilaga 2: Debriefingmall efter prövning av lektion	67
Bilaga 3: Presentationsmall	69

FÖRORD

Frågan om vad barn och unga ska lära sig i skolan engagerar många. Mycket anses vara så viktigt att det har lett till en betydande stofffrånsel i skolans kurs- och ämnesplaner. Även om det är trångt om utrymmet är nog alla överens om att eleverna måste få möjlighet att utveckla sin digitala kompetens. Både EU och OECD definierar digital kompetens som en nödvändig förutsättning för att aktivt kunna delta i ett kunskapsbaserat samhälle. De flesta är nog också eniga om att man i skolan bör få grundläggande kunskaper om programmering. Inte för att alla ska bli programmerare utan för att eleverna ska få möjlighet att utveckla sitt datalogiska tänkande, sin förmåga att uppfatta abstraktioner och se mönster, och förstå vad en algoritm är med alla dess effekter för oss som individer och för samhällsutvecklingen.

Det var mot denna bakgrund som regeringen år 2017 beslutade att programmering skulle införas som ett inslag i ämnena matematik och teknik i grundskolan. Hur denna förändring skulle implementeras var en fråga som många lärare och skolledare ställde sig utan att få några konkreta svar. Någon forskningsbaserad kunskap som skulle kunna ge vägledning fanns inte heller att hämta. Med tanke på att lärare och skolledare har ansvar för att utbildningen ska vila på vetenskaplig grund och beprövad erfarenhet blev uppgiften därför inte helt lätt att hantera.

Om inte kunskap finns måste den byggas! Därför startade Ifous ett forsknings- och utvecklingsprogram som skulle ge fördjupad kunskap om hur programmering i ämnesundervisningen kan utformas och utvecklas. Arbetet har bedrivits tillsammans med fem skolhuvudmän och en forskargrupp vid Stockholms universitet. Det har letts av Anette Jahnke, projekt- och processledare på Ifous. Anette är också redaktör för och författare i den här slutrapporten. Jalal Nouri, som är docent vid Stockholms universitet och som har lett forskningsinsatsen har bidragit med ett kapitel om programmets forskningsresultat. Det har granskats vetenskapligt av Staffan Selander, professor vid Stockholms universitet. Även representanter för de fem medverkande skolhuvudmännen har bidragit med texter liksom Lejla Mundzic, Emerga Institute, som har ansvarat för utvärdering av programmet. Rapporten som helhet har kvalitetsgranskats av Karin Hermansson, FoU-ansvarig på Ifous.

Efter tre års arbete återstår många frågor och det finns fortsatt behov av fördjupad kunskap och forskningsbaserad utveckling. Detta FoU-program måste ändå betraktas som ett viktigt bidrag i arbetet med att bygga den vetenskapliga grunden och bepröva erfarenheterna när det gäller undervisning och lärande i programmering. Denna slutrapport ger också underlag för diskussion om utformning och implementering av policybeslut i skolan utifrån ett generellt, kritiskt men ändå konstruktivt perspektiv. Förhoppningsvis kommer rapporten att vara av intresse för många.

Rapporten får gärna citeras med angivande av källa.

Stockholm i oktober 2020

Marie-Hélène Ahnborg
VD Ifous

SAMMANFATTNING

Anette Jahnke, projekt- och processledare Ifous

Tänk dig att du plötsligt ska undervisa i ett ämne du har väldigt lite kunskap om och erfarenhet av. Tänk dig att du dessutom under denna process ska dela dina erfarenheter, inte bara med dina kollegor utan också med andra lärare från andra skolor och huvudmän. Tänk att du är rektor eller skolchef som plötsligt ska säkerställa att rätt förutsättningar ges för något som få verkar veta vad det innebär. Utöver detta, ska ni alla dela ert arbete och erfarenheter med forskare, som precis som ni saknar kunskap om vad, varför, hur och när programmering ska undervisas och läras i den svenska grundskolan. Men att alla elever i Sverige ska få möta programmering i undervisningen, det beslutades av regeringen och verkställdes av Skolverket genom nya styrdokument 2017. Med dessa utgångspunkter startade Ifous tillsammans med fem huvudmän och en forskargrupp vid Stockholms universitet ett treårigt FoU-program.

Syftet med FoU-programmet var att utveckla didaktiska arbetssätt, uppgifter och innehåll för undervisning i programmering. Syftet var också att genom samarbete mellan lärare och forskare låta lärares kunskaper och erfarenheter utgöra grund för gemensam kunskapsutveckling och ny forskning.

Målen för utvecklingsarbetet var att utveckla kollegiala arbetsmetoder för undervisningsutveckling, och beprövade och vetenskapligt förankrade undervisningsmetoder så att eleverna utvecklade kunskaper. En tydlig progression skulle utvecklas i undervisningen. Utvecklingsarbetet skulle också leda till att lärare publicerade sitt arbete i utvecklingsartiklar.

Utvecklingsarbete hade också som mål att huvudmännen skulle ha skapat förutsättningar för fortsatt FoU-arbete efter programmets slut samt utformat en plan för delning av kunskaper från FoU-programmet till andra skolor.

Målen med forskningen var att synliggöra forskningsfrågor och besvara dessa genom samarbete med lärare, att forskningsresultat publicerades och att metoder för praktisk forskning vidareutvecklades.

FÖRÄNDRAT FOKUS OCH FÖRHÅLLNINGSSÄTT

Under tre år har 135 lärare, rektorer och ledare på huvudmannanivå från Freinetskolan Hugin, Simrishamns kommun, Stockholms stad, Tyresö kommun

och Åstorps kommun arbetat i enlighet med Ifous modell för samverkan (Ifous, 2020) utifrån ovan nämnda syfte och mål. Deltagarna har varit fördelade i 15 utvecklingsgrupper från olika skolor. Förutom allt lokalt arbete som skett inom huvudmän, skolor och klassrum, och inom forskargruppen, så har sex gemensamma utvecklingsseminarier, elva processledarmöten och 15 styrgruppsmöten genomförts under ledning av Ifous.

Under de tre årens arbete har innehåll och fokus förändrats, från teknik till didaktik. Oron vid starten byttes mot insikten att inte vara ensam, utan lärare fann stöd i de som redan hade viss erfarenhet av undervisning i programmering, och i forskarnas nyfikenhet och undersökningar. Undervisningsmodellen lesson studies introducerades och gav det kollegiala samarbetet struktur och fokus. Osäkerheten vid starten byttes sakta mot en allt större säkerhet i vad programmering kan innebära, och ett mod i att tillsammans med elever och kollegor pröva, och pröva igen, olika lektionsupplägg.

ÖKAD MOTIVATION HOS ELEVERNA

Den externa utvärderingen konstaterar att målen för programmet till mycket stora delar uppnåtts. Gemensamt för huvudmännen är att utvecklingsarbetet har lett till att undervisning i programmering i större utsträckning genomförs och kvaliteten har höjts med ett större fokus på elevernas lärande. Elevernas inställning är mer positiv, de visar ett större intresse och engagemang för programmering. Det ges också exempel på en mer positiv inställning till matematikämnet. Ett annat gemensamt resultat är att lärarna har utvecklat sin professionalitet, genom ett nytt gemensamt språk, stärkt vetenskapligt förhållningssätt till undervisningsutveckling och fördjupat samarbete med kollegor och elever. Totalt har 90 lesson studies genomförts och dokumenterats. Två utvecklingsartiklar har publicerats av deltagande lärare och ytterligare fler är på gång.

PROGRESSION I PROGRAMMERINGS- UNDERVISNINGEN

Forskningsarbetet tog avstamp i att besvara de mest grundläggande didaktiska frågorna: Vad ska undervisas, hur, när och varför? Totalt utfördes 50 klassrumsobservationer och lärarintervjuer, 70 elevintervjuer samt analys av dokumentation från 90 lesson studies. Forskningsresultaten är så här långt publicerade i 15 vetenskapliga publikationer (varav några tillsammans med lärare). Resultaten handlar om hur programmering har förändrat lärares och elevers skolpraktik; effekter på elever förhållnings-sätt, kunskaper och kompetenser; utmaningar associerade med införandet och utförandet av programmeringsundervisning; kategorisering av didaktiska arbets-sätt; samt en progressionsmodell.

Fem didaktiska arbets-sätt har framträtt, namngetts och beskrivits utifrån använt programmeringsverktyg: analog programmering; robot- och app-programmering; blockprogrammering; mikro-kontroll-programmering; och textprogrammering. För var och en av dessa finns olika användningssätt/områden i termer av utveckling av elevers datalogiskt tänkande och för allmänna ändamål.

En progressionsmodell för årskurs F-9 har utarbetats för användning av lärare och rektorer vid planering av programmeringsundervisning. Modellen består av tre dimensioner – datalogiskt tänkande, digital kompetens och programmeringsverktyg – samt fyra årskursnivåer. Modellen ger vägledning kring vilka programmeringsverktyg som kan vara relevanta på de olika nivåerna och vilka datalogiska kunskaper och färdigheter som eleverna ges möjlighet att utveckla.

Den forskning som bedrivits har varit praktiktäna, i betydelsen att delar av forskningsprocessen varit ”nära” praktiken genom att en av de valda datain-

samlingsmetoderna utgjordes av dokumenterade lesson studies. Detta har samtidigt inneburit att utvecklingsarbetet i skolorna skett på ett forskningsbaserat sätt. Forskningen har också varit ”nära” praktiken i betydelsen att preliminära resultat från olika delstudier återkopplats på seminarier, med dialog kring relevans och användbarhet. Detta har medfört att programmets forskningsresultat redan har börjats användas och utvecklas av deltagarna själva, och kan bedömas vara relevanta och användbara för alla skolans yrkesverksamma med ansvar för undervisning i programmering.

VIKTIGT ATT FORTSÄTTA FOU-ARBETET

Både programmets deltagare och forskare understryker att mycket återstår att utveckla och forska kring när det gäller undervisning och lärande i programmering. Till exempel krävs mer FoU-arbete kring att synliggöra och följa vilka former av kunskaper elever utvecklar – och med vilken kvalitet – genom undervisning om, med och i programmering. Speciellt behövs mer kunskap om hur elevers kunskaper påverkas i de ämnen där programmering påstås vara ett medel, till exempel i matematik. FoU-programmets resultat väcker frågan om det till exempel ens är möjligt att lära sig matematik genom programmering i grundskolans tidigare årskurser i enlighet med de skrivningar som finns i dagens kursplaner? Vår förhoppning är att resultaten av FoU-programmet inte enbart används av skolans yrkesverksamma, utan också av de som har ansvar för att initiera och utforma styrdokument – dokument vars innehåll också behöver vila på vetenskaplig grund och beprövad erfarenhet. En grund som nu efter tre års arbete blivit lite stadigare att stå på.

DETTA ÄR FOU-PROGRAMMET PROGRAMMERING I ÄMNES- UNDERVISNINGEN

Regeringen beslöt 2017 att justera skolans styrdokument för att säkerställa att programmering skulle utgöra ett inslag i undervisningen. Men hur skulle dessa ambitioner realiseras så att eleverna verkligen lärde sig det vi önskar? Allt fler lärare började använda sig av olika programmeringsmiljöer och skaffade sig erfarenheter och kunskaper kring undervisning i, och med, programmering. Men kunskapen som växte fram behövde synliggöras, bli beprövad och delas av fler.

Den didaktiska praktikhäna forskningen kring undervisning och lärande i programmering, speciellt inom ramen för olika skolämnen var – och är till viss del fortfarande – mycket begränsad. En modern programmeringsdidaktik saknades. Genom FoU-programmet fick huvudmannen stöd att utifrån nya styrdokument utveckla en undervisning som i högre grad kunde vila på vetenskaplig grund och erfarenheter som blivit beprövade.

Syftet med FoU-programmet var att utveckla didaktiska arbetsätt, uppgifter och innehåll för undervisning i programmering. Syftet var också att genom samarbete mellan lärare och forskare låta lärares kunskaper och erfarenheter utgöra grund för gemensam kunskapsutveckling och ny forskning.

FoU-programmet Programmering i ämnesundervisningen inleddes under höstterminen 2017 som en samverkan mellan fem huvudmän: Freinetskolan Hugin, Stockholm stad, Simrishamns kommun, Tyresö kommun och Åstorps kommun, Stockholms universitet och Ifous. Huvudmännen har medverkat med utvecklingsgrupper av lärare från totalt 15 grundskolor. Antalet personer har varierat något under åren, men totalt har 135 lärare, rektorer och ledare på huvudmannanivå medverkat.

En viktig förutsättning för att medverka i Ifous FoU-program är att huvudmannen involverar hela organisationen i arbetet, då vi vet från tidigare forskning och erfarenheter att det är en viktig faktor för att förändringsarbetet ska bli hållbart över tid. Konkret innebär det att chef på huvudmannens ledningsnivå, rektor och lärare alla deltar aktivt i FoU-programmet – såväl under gemensamma seminarier som mellan dessa tillfällen.

Varje skola utsåg en lokal processledare som ledde utvecklingsgruppens arbete på skolan. Stockholms stad och Tyresö kommun, som medverkat med flera utvecklingsgrupper på olika skolor, utsåg dessutom var sin central processledare på förvaltningsnivå

MÅL FÖR FOU-PROGRAMMET

Utvecklingsarbetet har medfört att:

1. arbetsmetoder har utvecklats för att kollegialt och i samarbete med rektor pröva, genomföra och följa upp inslag av programmering i undervisningen så att eleverna utvecklar kunskaper i programmering inom bland annat matematikämnet
2. undervisningen i programmering har en tydlig progression
3. beprövade och vetenskapligt förankrade undervisningsmetoder används kring programmering
4. utvecklingsartiklar har publicerats
5. skolan och huvudmannen har skapat förutsättningar och en beredskap för fortsatt forsknings- och utvecklingsarbete i takt med den tekniska och ämnesdidaktiska utvecklingen
6. huvudmannen har en plan för hur kunskaper och erfarenheter från FoU-programmet ska spridas till andra skolor

Forskningen har medfört att:

1. forskningsfrågor har synliggjorts och besvarats genom samarbete mellan forskare, processledare och lärare
2. artiklar har publicerats i vetenskapliga och populärvetenskapliga tidskrifter
3. forskningsmetoden för praktikhäna forskning har vidareutvecklats

som internt stöttat de olika grupperna och skolornas arbete. De centrala och lokala processledarna har utgjort en processledargrupp. Denna grupp har varit central i arbetet med att leda och följa de processer som pågått under de tre år som programmet pågått. Processledargruppen och forskargruppen har tillsammans och kontinuerligt diskuterat och arbetat med utvecklingen av undervisningen i programmering, samt diskuterat preliminära forskningsresultat. Problem och utmaningar har fångats upp och kunnat hanteras gemensamt, och erfarenheter har delats mellan processledarna, samt med forskarna och Ifous.

Utvecklingsgruppernas arbete präglades inledningsvis av stor oro och osäkerhet om vad de nya styrdokumenterna i praktiken skulle innebära för dem

som lärare, rektor och för deras elever. Inledningsvis var fokus på att arbeta med att inventera, undersöka och införskaffa lämpligt material samt lära sig mer om programmering genom olika workshops, utbildningsdagar och kurser. Undervisningsmodellen lesson studies introducerades och gav det kollegiala samarbetet struktur och fokus.

Programmet har finansierats gemensamt av de fem medverkande skolhuvudmännen och Stockholms universitet. Det har letts av en styrgrupp bestående av representanter för varje skolhuvudman, forskningsledare Jalal Nouri från Stockholms universitet samt Anette Jahnke och Karin Hermansson som övergripande projekt-/processledare respektive FoU-ansvarig på Ifous.

FEM HUVUDMÄN TOG TÄTEN

Lärare, rektorer och ledare på förvaltningsnivå från de fem huvudmännen har alla gjort var sin resa under drygt tre år. I detta kapitel tar vi del av under vilka förutsättningar de gick med i programmet. Varför valde de att medverka och med vilka förväntningar startades arbetet? Vilka tidigare erfarenheter fanns kring programmering och att genomföra FoU-arbete?

Huvudmännen beskriver även hur de genomfört, följt upp och utvärderat sitt utvecklingsarbete. Hur organisationen för deltagandet har utvecklats och hur arbetet har spridits, samt hur utvecklingsgrupperna har arbetat och bidragit till forskningsarbetet.

Resultatet av FoU-arbetet beskrivs och i vilken utsträckning man upplever att man har nått FoU-programmets syfte och mål. Utmaningar, möjligheter och eventuella oväntade resultat lyfts fram i huvudmännens berättelser. På vilket sätt elevernas lärande och utveckling påverkas diskuteras samt vad som återstår att fortsätta arbeta med kring programmering i ämnesundervisningen. Slutligen dras slutsatser om vilka nya kunskaper man erövrat och hur planerna ser ut för framtida undervisning om, i och med programmering i deras verksamheter.

FREINETSKOLAN HUGIN: BASNIVÅN HOS ELEVERNA HÖGRE OCH SPJUTSPETSARNA SKJUTER ÄN HÖGRE

*Teresa Sundström, lokal processledare
Freinetskolan Hugin*

Freinetskolan Hugin är belägen i Norrtälje och är en ekonomisk förening som drivs av ett personalkooperativ. Vi har cirka 250 elever från förskolan till årskurs 9. Vi har en rektor, samt en huvudman – styrelsen – som består av personal på skolan. Vår storlek har i många fall varit positiv, men också gett oss en del utmaningar i programmet.

Enheten hade ingen tidigare erfarenhet av att delta i FoU-program, dock av annat forskningsarbete. När förslaget om att medverka i Ifous FoU-program initialt presenterades för oss, så kändes det som rätt väg att gå. På vår enhet hade vi under en längre tid arbetat med digitala verktyg. I förskoleklass upp till årskurs 2 hade vi Ipad ”en-till-två” (två elever per Ipad) och i årskurserna 3–6 hade vi en-till-en (en elev per Ipad). I årskurserna 7–9 hade eleverna varsin MacBook. Infrastrukturen på skolan var väl utbyggd, vi hade goda tekniska förutsättningar och flera pedagoger hade goda kunskaper inom digitalt arbete.

Programmering var inte heller främmande för oss. Vi hade sedan tidigare haft en årlig temadag – Hugin Hackar, där eleverna fått delta i åldersblandade aktiviteter. En hel del material var redan inköpt, såsom olika appar och robotar samt en 3D-skrivare. Rektor hade under en längre tid haft en tanke om att få fler pedagoger intresserade av forskning inom skolområdet, och när programmering kom in i de uppdaterade styrdokumenterna så kändes detta FoU-program som vårt naturliga nästa steg.

Då vår skola arbetar tematiskt föll det sig också naturligt att väva in Ifous FoU-program i mer än bara matematiken. Vår utvecklingsgrupp bestod därför av lärare från olika stadier och inom olika ämnen. Att delta i programmet var frivilligt men intresset var stort. En tredjedel av berörda pedagoger (grundskola + fritidshemmet) valde att delta. Det

uppfattades som både utmanande och utvecklande att få delta i att beforska sin egen verksamhet.

Förkunskaperna i utvecklingsgruppen var blandade. Vissa hade knappt kommit i kontakt med programmering tidigare, och andra hade redan årliga stående projekt som de arbetade med.

Utmaningar som gruppen brottades med i ett tidigt skede var en rädsla för tidsåtgång samt konkurrensen med undervisning i övriga ämnen. Hur skulle man hitta tid för att dokumentera och hur mycket plats fick programmering ta från övriga ämnen? Det fanns även en osäkerhet kring innehållet, i synnerhet från dem som inte hade så stor tidigare erfarenhet av programmering.

De pedagoger som inte undervisade i matematik kände sig till en början ensamma i programmet. Det upplevdes svårt att fläta samman programmering med de ämnen de undervisade i. De som kommit längre inom programmering gick in med höga förväntningar om att få en inblick i andras verksamhet och dela erfarenheter.

GENOMFÖRANDE OCH UPPFÖLJNING

Programmet prioriterades högt från ledningshåll och utvecklingsgruppen hade fasta månatliga möten. Viljan att göra det som ålagts oss var hög och vi har haft ett tydligt mål och syfte med programmet hela tiden.

Vi inledde våra möten med att göra en genomlysning av deltagarnas egna förväntningar, farhågor och förhoppningar. Men också en behovsanalys för att se vad pedagogerna ville lära sig under programmets gång. Vår förhoppning var att ha mycket tid för vidareutbildning, både internt och externt. Vi har dock inte genomfört utbildningar i den mån vi

önskade. Dels beroende på en upplevd tidsbrist som konkurrerar med pågående verksamhet, men också för att programmering kändes som ett osäkert område vilket gjorde motståndet lite större till en början.

På mötena med utvecklingsgruppen har vi till stor del arbetat med uppdragen vi fått från Ifous och forskarna, exempelvis lesson studies, dokumentation samt planering av workshops att presentera på seminarierna. Vi har även haft tid till gemensamma lektionsplaneringar, diskussioner om progression samt utbildning och egen dokumentation av vår utveckling. Vi har gemensamt gått Skolverkets webbaserade kurs inom programmering och åkt till Applebutiken i Täby och fått en kurs i Swift Playground av dem. Gemensamt har vi tittat på olika verktyg som finns på skolan, planerat inköp och försökt lära oss tillsammans och av varandra, till exempel Micro:bit och Cozmo. Vi har även använt mötena till att medverka i planeringen av Hugin Hackar, vår årliga temadag, och under programmets gång sett Hugin Hackar förvandlas från en årlig happening till aktiviteter på en mer medveten nivå.

Inom utvecklingsgruppen har vi läst litteratur om programmering, diskuterat och inspirerat varandra och även peppat varandra när det har känts lite motigt och svårt. Ibland har vi sökt inspirationen inom vår egen verksamhet och ibland utifrån för att komma vidare. Eftersom flera pedagoger i gruppen var bekanta med programmering och i vissa fall kommit väldigt långt, så ansåg vi att inspirationen och drivkraften fanns inom själva gruppen. Processledaren valdes ut för att hålla i strukturen, driva arbetet åt rätt håll och för att sköta dokumentationen. De starka programmerarna fick i stället utrymme för att driva den verksamheten vidare.

Arbetet i utvecklingsgruppen har gått i vågor. Vi är vana vid kortare projekt, tematiska arbeten eller kortare studier – projekt där man har ett tydligt mål. Här var det svårare att se vad vi gav oss in på och hur det skulle bli, vilket ledde till en entusiastisk start som övergick i en mellanperiod där vi enbart gjorde det som ålades oss – för att avsluta med entusiasm igen.

Under programmets gång har vi sett både fördelar och nackdelar med att vara en mindre enhet. Eftersom programmet har fått så stort utrymme i verksamheten så har hela enheten blivit engagerad. Personalen peppar varandra och det finns en stark delningskultur. Däremot har utbytet med andra skolor inte varit lika starkt. Dels ligger vi lite avsides jämfört med storstadsskolorna och som enda enhet i kommunen har en naturlig samarbetspartner i programmet inte funnits.

Vi har bidragit till forskningen i hög grad. Åsa Chibás som är ansvarig för förskoleklassen har haft ett tätt samarbete med forskningsledaren Jalal Nouri där de forskat om didaktiska arbetssätt i yngre

åldrar. Enheten har även bidragit med en artikel (Chibás m.fl., 2018) och fler kommer publiceras inom kort. Vi har haft forskare på besök hos oss på Hugin för klassrumsobservationer vid ett flertal tillfällen och flera pedagoger har blivit intervjuade. Utöver det har vi bidragit med flera workshops på seminarierna samt två föreläsningar. Internt valde vi att sprida kunskapen delvis genom en allpersonalkonferens där utvecklingsgruppen presenterade sitt arbete.

RESULTAT

Diskussioner som förts i utvecklingsgruppen visar att vi anser oss i hög grad ha nått syfte och mål med programmet. Det som finns kvar att göra är att utvärdera vad eleverna fått med sig av programmet, genom enkäter och diagnoser, och att fortsätta att sprida programmet till övriga pedagoger på skolan. Framöver vill vi, genom vårt utbildningsbolag Hugin Edu, sprida vår kunskap vidare ut till Norrtälje kommun och övriga skolor genom bland annat workshops.

Vi har en ganska tydlig progressionstanke för undervisningen i programmering, men behöver vidareutveckla och specificera den baserat på programmet's forskningsresultat. Vi vill ha ett verktyg för att ta emot nya pedagoger och även nya elever som inte arbetat med programmering tidigare.

Vi ser att vi på enheten har strukturer som överlappar varandra och till viss del skapar förvirring, så vi behöver utveckla en samverkan och tydlighet mellan olika ansvarsområden, till exempel inköp. Det behövs även en inventering av det material vi har och ett lånesystem för att göra robotar och övrig teknik mer tillgänglig och levande i undervisningen.

Det pedagogerna uppfattat som utmanande under programmets gång har huvudsakligen varit hur man ska kombinera ämnesinnehållet med programmering så att det blir värdefullt använd tid. Delar av dokumentationen som legat till grund för forskningsarbetet har uppfattats som krävande, såsom inlämningsformatet på lesson studies. Det har även upplevts som svårt att få ihop tider för att genomföra lesson studies och att hitta bra ämnesområden att fokusera på. Men generellt sett har formatet lesson study upplevts som positivt genom att den fått lärarna att samarbeta och grundligt tänka igenom lektioner tillsammans för att nå ett bättre resultat. Vi har även sett positiva effekter av att vi valde att ha med lärare med olika ämnesfokus. Projektet som skapats har blivit stora och öppna ämnesintegrerade projekt vilket har uppskattats av eleverna.

Samarbetet mellan elever och kollegor har varit positivt. Både bland kollegor och elever finns några som är starkare inom ämnet som hjälpt övriga att komma vidare. De praktiska förutsättningarna har

varit mycket positiva, tekniken har funnits och har oftast fungerat. En viktig faktor är att vi haft ett stort stöd från ledningen och det har underlättat att hålla kvar engagemanget i programmet.

Flera pedagoger upplever att det varit roligt och nytt att ämnesintegrera inom programmering och komma i kontakt med tekniken på ett nytt sätt. Vi ser också att arbetet med lesson studies och programmering gett fler arbetslagöverskridande projekt, både mellan stadier och mellan fritidshemmet och skola. Fler pedagoger har börjat prata om programmering, ser det som en naturlig del av verksamheten och har uttryckt en önskan om vidareutbildning.

Engagemanget från eleverna har varit positivt, många har uppskattat programmering och flera har fått ett höjt självförtroende och hittat ett starkt intresse. Vissa har gått från att spela spel på rasterna till att i stället sitta och programmera. Utvecklingen hos eleverna är generellt positiv. Basnivån är högre och spjutspetsarna skjuter än högre. På gruppnivå är inställningen till programmering mer positiv än den var tidigare. Vi ser även vid stadiemyten att elevernas grundkompetens höjts.

LÄRDOMAR OCH FORTSATT UTVECKLING

I huvudsak ser vi den positiva effekten av att ha en platt organisation som kan fatta snabba beslut. Att ha en rektor som har prioriterat arbetet i FoU-programmet högt och sett till att avsätta tid för pedagogerna vid behov, har varit mycket värdefullt eftersom i synnerhet tid varit en bristvara för pedagogerna.

Vi ser att pedagogerna som deltagit i programmet har fått ett mer forskningsbaserat tänkande. De lektioner som vi har i programmering på enheten nuförtiden är i mycket högre grad medvetna med ett tydligt syfte och mål samt väl valda verktyg,

inte lustfyllda happenings som det lätt kunde bli inledningsvis. Spridningen inom enheten har generellt varit god, åtminstone sett ur elevsynpunkt. På mellanstadiet och högstadiet har vi ämneslärare vilket har lett till att alla elever fått ta del av programmering. De lågstadiepedagoger som deltagit i FoU-arbetet har arbetat med programmering i sina klasser, medan de pedagoger som inte deltagit har hamnat lite utanför med sina elevgrupper. Att utveckla samarbeten lågstadiegrupperna emellan är ytterligare något att arbeta vidare med. En tanke är att integrera programmering i våra temaarbeten.

I backspegeln ser vi att fritidshemmet borde ha varit med i programmet från början. Samarbetet mellan skola och fritidshemmet skapar en bättre fördelning av ämnesinnehåll och lustfyllda uppgifter där elever tillåts vara kreativa i längre projekt.

Det var bra att vi tog med lärare från olika ämnen trots en inledande skepsis. De pedagoger som arbetar i andra ämnen känner inte att programmet gett deras elever högre kunskaper inom deras ämnen, men ser det däremot som positivt att de själva fått kunskaper och bidragit till de ämnesintegrerade uppgifterna. Vi ser hur viktigt det är för elevernas utveckling att ha en balans mellan öppna och slutna uppgifter. Det vi behöver träna dem vidare i är felsökning som vi hoppas kan spilla över till andra ämnen.

Under de två första åren såg deltagarna i utvecklingsgruppen vissa svårigheter med FoU-programmet. När forskarna i sina resultat presenterade samma svårigheter blev det ett nyvaknade för deltagarna och de fick bekräftelse på det de upplevt. Fortfarande finns det vissa svårigheter med att väva in programmering i alla ämnen, men vi har fått större kunskaper om programmering och hur det kan nyttjas på ett värdefullt och lustfyllt sätt för både elever och pedagoger.

STOCKHOLM: FÖRÄLDRAR HÖR AV SIG – BARNEN BERÄTTAR OM PROGRAMMERING

*Per Anderhag, central processledare,
lektor FoU-enheten Stockholm stad*

I Stockholms stad går uppskattningsvis 66 000 elever i någon av de 146 kommunala grundskolorna. Staden är organiserad i sju grundskoleområden där varje område innefattar två stadsdelar med olika socioekonomiska förutsättningar. I FoU-programmet har åtta skolor från fem grundskoleområden deltagit, se Tabell 1.

Utbildningsförvaltningens FoU-enhet, som är en stödfunktion till stadens grund- och gymnasieavdelning, har haft en samordnande roll i FoU-programmet. Enhetens chef har representerat Stockholms stad i FoU-programmets styrgrupp och en person från enheten har fungerat som central processledare. Varje skola har haft en utsedd lokal processledare som har ansvarat för arbetet i utvecklingsgrupperna. Detta kapitel har som ambition att lyfta erfarenheter från Stockholms samtliga nivåer i organisationen, det vill säga processledare och utvecklingsgrupper,

rektorer samt central förvaltning. Notera att såväl huvudman som central förvaltning avser FoU-enheten. När texten hänvisar till ”skolor/skolan” bygger den på utsagor som processledarna och rektorena uttryckt mer eller mindre samstämmt.

FÖRUTSÄTTNINGAR NÄR VI STARTADE PROGRAMMET

Våren 2017 genomfördes en kvantitativ och kvalitativ kartläggning gällande omfattningen av programmering i stadens grundskolor (Ödalen, 2020). Mycket kortfattat visade denna på en stor variation i omfattningen av programmering i den ordinarie undervisningen mellan stadens grundskoleområden, skolor och årskurser. Detta var knappast förvånande eftersom de nya styrdokumenterna ännu

Tabell 1: Skolor i Stockholms stad vars lärare deltagit i FoU-programmet.

Skolnamn	Typ av skola	Antal elever	Antal deltagare i utvecklingsgruppen
Aspuddens skola	F-9	1019	7
Blommensbergsskolan	F-9	565	9
Bobergsskolan	F-6	218	5
Magelungsskolan	F-6	223	3
Sjöängsskolan	F-6	693	5
Skönstaholmsskolan	F-9	237	5
Trollbodaskolan	F-9	699	9
Västertorpsskolan	F-6	465	3

inte börjat gälla. Lite oväntat var att programmering framför allt var ett inslag i de högre årskurserna och att programmeringsinslag i princip var obefintlig i skolornas fritidshemsverksamhet. Våren 2017 var undervisning om eller med programmering något som vissa "eldsjälar" arbetade med på eget initiativ. I mycket liten utsträckning var programmeringsaktiviteter kopplade till något specifikt skolämne. Det som framkom ur kartläggningen utgjorde en övergripande bakgrund till varför central förvaltning valde att uppmantra skolor att delta i FoU-programmet. En förväntan från central förvaltning var därför att FoU-programmet skulle vara ett (1) professionsutvecklande sammanhang där (2) deltagande lärare och skolor fick möjlighet att utveckla ämnesundervisningen och (3) tillsammans med forskare producerade undervisningsrelevant kunskap.

När FoU-programmet startade var programmering ett nytt ämnesområde för samtliga utom en av de åtta lokala processledarna. Denna person hade erfarenheter av programmering och hade även använt det tidigare i sin undervisning, dock aldrig i relation till matematikämnet. Samtliga skolor, vilket både rektorer och processledare lyfter, hade liten kunskap om programmering i allmänhet och programmering som ett inslag i undervisningen i synnerhet. Skolorna hade mycket olika förutsättningar gällande hårdvara när programmet startade, någon skola hade i princip möjlighet till en-till-en-undervisning (en dator per elev) medan någon skola inte hade några enheter överhuvudtaget. Utmaningar i relation till detta var också något som diskuterades under det första året. Staden var mitt uppe i en mjuk- och hårdvaruupphandling och skolorna skulle besluta om inköp av enheter. För- och nackdelar med Ipad, PC och Chromebooks i relation till programmering diskuterades därför ingående. Första årets diskussioner kan sammanfattas som att de fokuserade på vad programmering är och kan vara i relation till ett ämne och vilka typer av prylar som bör användas i relation till detta.

Även om det varierar något hade skolorna tämligen likartade förväntningar på FoU-programmet. Man såg det som en möjlighet att arbeta med de nya skrivelserna i styrdokumentet och få tillfälle att arbeta mer systematiskt med programmering och hur det kan användas i relation till matematikämnet. Möjligheten att få tips och idéer från andra sågs också som en styrka med programmet. Både rektorer och processledare såg det också som ett medel för att börja prata om programmering och att få in det i skolans verksamhet generellt. Flera rektorer såg det även som ett innehåll i skolans övergripande digitaliseringsarbete, som för flera skolor var ett område man redan arbetade med. Samtidigt uttrycker flertalet att det inledningsvis var frustrerande då man inte riktigt var klar över

vad man förväntades göra och varför. Ingen, med något undantag, hade tidigare deltagit i forskningsprojekt och man hade inte heller några specifika förväntningar på forskningsinsatsen.

GENOMFÖRANDE OCH UPPFÖLJNING

Central förvaltning var inledningsvis otydlig gällande vad FoU-enheten respektive skolan skulle finansiera (förutom lärares tid och eventuella vikariekostnader vid lokala och nationella möten och seminarier stod skolorna för kostnaderna för att delta i utvecklingsseminarierna). Detta skapade problem och för någon skola innebar det även att man var tvungen att minska antalet personer i utvecklingsgruppen.

Tid och organisation för genomförande har varierat mellan skolorna, från ingen avsatt tid under den arbetsförlagda tiden till en timme i veckan. Samtliga processledare har kunnat delta i de nationella och lokala möten som anordnats. På några av skolorna hade antingen processledaren eller rektorn varit väldigt drivande i att skolan skulle gå med och hade därför också en idé om vad man vill få ut av FoU-programmet. Eventuellt på grund av detta fanns det på några skolor tidigt en organisation för hur arbetet skulle genomföras, till exempel formaliserade strukturer för arbetet i utvecklingsgruppen. På några skolor var det mer otydligt och någon processledare beskriver det som att hen fick uppdraget utan att riktigt veta vad det skulle innebära. Några beskriver också att de genomgående har haft liten möjlighet att faktiskt organisera ett arbete i utvecklingsgruppen, bland annat på grund av att det inte har allokerats tid för arbetet eller att deras ansvar och mandat har varit otydligt. Detta är något som även lyfts i rektorsgruppen.

Under FoU-programmets gång har flera av skolorna gjort organisatoriska förändringar, till exempel hur ofta man har möten och under hur lång tid. Bemanningen av utvecklingsgrupperna har också sett olika ut och även förändrats över tid. På några av skolorna har arbetet fokuserat på matematik- och teknikämnet vilket har återspeglats i utvecklingsgruppens sammansättning. Någon skola har fokuserat på ett specifikt stadium, medan andra har haft samtliga stadier representerade. Som antyds ovan har organisationen styrts av vad man velat uppnå i arbetet. För någon skola var det till exempel viktigt att arbeta fram en progression, och därför valde man att ha personer från flera stadier representerade i utvecklingsgruppen.

Skolorna uttrycker olika möjligheter och utmaningar kopplade till hur FoU-arbetet organiseras. Ett exempel kan vara utvecklingsgruppens storlek, som varierat mellan två personer i en skola till tio personer i en annan. En för liten grupp blir sårbar

och har ofta svårigheter att ge avtryck på den övriga verksamheten. I en stor grupp, å andra sidan, kan det vara svårt att hitta mötestider, planera och genomföra aktiviteter.

Relativt samstämmigt beskriver skolorna att arbetet i utvecklingsgrupperna successivt förändrades från att kartlägga och diskutera syften och mål, hård- och mjukvara till att mer handla om undervisning. Införandet av lesson study-metoden var viktigt för denna förskjutning i fokus.

Utöver utvecklingsgruppernas egna möten samt nationella möten och seminarier har lokala rektors- och processledarmöten genomförts. Syftet med de förra har framför allt varit att uppdatera, dela erfarenheter samt att fånga in önskemål och tankar till programmets styrgrupp. De lokala processledarmötena har haft delvis samma funktion men också att diskutera hur FoU-programmets övergripande mål kan konkretiseras i specifika aktiviteter. Dessa sammanhang har även varit betydelsefulla för att skapa en gemensam förståelse för programmering på de olika skolorna. Kontinuerlig uppföljning har framför allt handlat om övergripande aspekter som till exempel huruvida lektionsaktiviteter har fungerat som planerat eller om lesson study-metodiken har varit användbar. Undervisningens effekter på elevernas lärande har utvärderats inom ramen för de specifika lesson studies som genomförts. En kvantitativ kartläggning motsvarande den som genomfördes våren 2017 har genomförts hösten 2019.

I början av programmet fanns det ett behov av att få en gemensam förståelse för programmering. Som en konsekvens av detta arbetade skolornas utvecklingsgrupper dels med att definiera vad programmering i skolan kan vara och dels med specifika undervisningsaktiviteter. På vissa skolor har lärare deltagit i kurser, arrangerade av till exempel Skolverket, Stockholms universitet och Mediateket, där innehållet diskuterades i utvecklingsgrupperna. Efter introducerandet av lesson study-metodiken kom arbetet att fokusera på specifika undervisningsaktiviteter. Samtliga skolor har följt det upplägg som presenterades av forskarna. Man har således arbetat med att designa och skriva fram lektioner som sedan genomförts och utvärderats. Arbetet har skett iterativt och reviderade lektioner har ofta testats i nya klasser. I detta arbete har lärarna bland annat undersökt möjligheter och utmaningar vid programmering i visuella miljöer, geometriska figurer och vinklar, printhälsbestämning, dricksberäknare samt automatiska trafikljus i teknikämnet. Flertalet av de vanligaste verktygen och miljöerna använts, som: Bee-bot, Scratch, Scratch junior och Micro:bit. Någon skola har också använt sig av ett textbaserat programmeringsspråk (Python). Direkt samverkan mellan utvecklingsgrupperna och forskarna har varit sparsam.

RESULTAT

Under hösten 2019 genomfördes en uppföljning av den kartläggning som alla stadens grundskolor gjorde 2017 gällande omfattningen av programmering i den ordinarie undervisningen (Ödalen, 2020). Enkäten utgick, precis som 2017, från en fyrgradig skala där skolorna fick skatta huruvida: (1) inga (2) ett fåtal (3) ett flertal eller (4) en majoritet av eleverna i skolans olika årskurser (F-3, 4-6, 7-9 och fritidshemmet) programmerar i den ordinarie undervisningen. Enkäten visade följande skillnader i medelvärden för staden som helhet: för årskurserna F-3 ökade medelvärdet från 1,7 till 3; för årskurserna 4-6 från 1,8 till 3,2; för årskurserna 7-9 från 2,2 till 3,1, och för fritidsverksamheten från 1,3 till 2,5. Medelvärdet för skolorna som helhet, det vill säga samtliga årskurser, är 2,9. I tabell 1 presenteras resultatet från deltagande skolors självskattning. Inte oväntat har det, med något undantag, skett en ökning i omfattning av programmering i samtliga årskurser. Jämför man respektive skolas medelvärde har samtliga förutom en skola ökat minst ett steg. Sex av skolorna ligger över stadens medelvärde (2,9) och om materialet jämförs sammanslaget och årskursvis är de positiva förändringarna kraftigare hos de deltagande skolorna jämfört med de genomsnittliga förändringarna i staden (Ödalen, 2020). Det vill säga, givet läget 2017 har utvecklingen varit mer positiv på de deltagande skolorna jämfört med staden som helhet. Skillnaderna är också signifikanta (Ödalen, 2020).

Under programmet har skolorna genomfört ett antal lesson studies. I enlighet med metodens syfte har varje lesson study haft ett specifikt lärandemål och ambitionen har således varit att eleverna ska lära sig detta. Det finns inte utrymme här att presentera utfallet av varje aktivitet men de specifika mål som aktiviteterna har syftat mot har, också i enlighet med metoden, utvärderats. Dessa utvärderingar visar på tillfällena då aktiviteterna stödjer elevernas lärande i enlighet med vad som avsetts med dem, men det finns även exempel där lesson studyn inte haft den effekt som man hoppats på. Förändringar i elevernas förståelse för programmering syns även på andra sätt, till exempel hänvisar lärarna till ett ökat intresse och engagemang hos eleverna, förändrat språkbruk där introducerade begrepp används av eleverna i nya sammanhang samt att föräldrar hör av sig och berättar att deras barn berättar om programmering hemma.

Med något undantag hade deltagande lärare och rektorer väldigt liten kunskap om programmering när programmet startade. Samtliga lyfter att deras förståelse av programmering har förändrats under programmet. Då lärarna/skolorna har deltagit i andra programmeringssammanhang än FoU-programmet (kurser, mindre fortbildningsinsatser, själv-

Tabell 2: Deltagande skolors självskattning av omfattning av programmering i den ordinarie undervisningen.

Skola	2017					2019				
	F-3	4-6	7-9	Fritids	Medel	F-3	4-6	7-9	Fritids	Medel
Bobergs-skolan	1	1	-	1	1	3	3	-	4	3,3
Aspuddens skola	3	2	1	2	2	3	3	2	3	2,8
Blommens-bergsskolan	1	2	4	1	2	3	3	3	3	3,0
Magelungs-skolan	1	1	-	-	1	3	4	-	3	3,3
Sjöängs-skolan	1	1	1	1	1	4	4	4	4	4,0
Skönsta-holmsskolan	1	1	-	1	1	3	4	-	3	3,3
Trollbodasko-lan	1	1	1	1	1	4	4	4	4	4,0
Västertorps-skolan	1	1	-	1	1	3	4	-	3	3,3
Medel	1,3	1,3	1,8	1,1		3,0	3,3	3,3	3,0	

studier, etc.) kan kanske inte allt som presenteras nedan härledas till programmet enskilt. Hursomhelst, lärarna och rektorerna menar att deras kunskap om programmering som företeelse har ökat, det vill säga vad programmering är och vad det kan användas till. I relation till detta har lärarna mött nya begrepp, material och sammanhang som man i varierande grad har använt i sin undervisning (till exempel programmeringsmiljöer och begrepp som loop, funktion, variabel, etc.). Deltagarnas kunskap om programmeringens betydelse i samhället har ökat, kopplat till detta är också aspekter som man inom programmet har problematiserat, till exempel vem programmeraren är och vilken betydelse detta kan ha för den kod som produceras. Kunskapen om undervisning och programmering har förändrats, samtliga lyfter att de under programmet har haft möjlighet att diskutera och utveckla undervisningsaktiviteter. Slutligen lyfter samtliga lesson study-metodikerna som ett medel för att på ett mer systematiskt sätt förhålla sig till undervisningens effekter på elevernas lärande. Samtidigt som man uppskattat metoden ser man utmaningar i att använda den. Framför allt handlar det om den tid man anser att man behöver investera i framtagandet av de stöddokument som använts för planering och uppföljning.

LÄRDOMAR OCH FORTSATT UTVECKLING

Utöver det som beskrivits ovan har resultat från en lesson study presenterats på konferensen *Leda Lärande* (Norberg m.fl. 2019). Studien syftade till att eleverna genom tre stegvisa lektioner fick upptäcka sambandet mellan analog och digital pilprogrammering. Studiens resultat visade att eleverna utvecklade sin förmåga att skriva kod, samt att de fick en ökad insikt i varför det ibland är viktigt att felsöka. I denna process använde eleverna också önskade ämnesspecifika begrepp. Utfallet av övriga undervisningsinterventioner har analyserats av forskarna och presenteras i aggregerad form i kapitlet "Forskningsinsatsens genomförande och resultat". på sid 35. Erfarenheter och lärdomar från programmet kommer att utgöra en grund för respektive skolas fortsatta utveckling av sin programmeringsundervisning. Nätverket av processledare kommer att ha fortsatt kontakt och centralt anordnade möten kommer att anordnas tills vidare. Detta kommer således utgöra ett sammanhang för de deltagande skolorna att fortsätta diskussionen om undervisning och programmering.

SIMRISHAMN: ELEVERNA MER POSITIVA TILL PROGRAMMERING JU MER UNDERVISNING DE FÅR

Pia Svensson, rektor, Simrishamns kommun

Utifrån regeringens beslut att programmering skulle införas i matematik, valde Barn- och utbildningsförvaltningen tillsammans med Korsavadsskolans rektor och matematiklärare att skolan skulle delta i FoU-programmet *Programmering i ämnesundervisningen*. Barn- och utbildningsförvaltningen gav skolan ekonomiska förutsättningar för att delta.

Kommunen har endast en högstadieskola. Alla matematik- och tekniklärare har deltagit i programmet och inga andra stadier har involverats i programmet. Skolans utvecklingsgrupp hade inga tidigare erfarenheter av FoU-arbete. På Korsavadsskolan hade varje elev redan en egen Chromebook, sedan några år tillbaka. På skolan fanns en IT-utvecklare på 20 procent. Arbetet har underlättats genom att lärarna arbetar på samma skola i kommunen. Lärarna på skolan hade två års erfarenhet av att ha arbetat med Lego Mindstorms och Scratch inom teknikämnet innan Ifous FoU-program påbörjades.

Lärarna hade liten förståelse kring vad det innebar att delta i ett FoU-program. De hade uttryckt förhoppningar om att programmet skulle leda till att de praktiskt skulle lära sig att programmera. De förväntade sig att få färdiga lektionsupplägg att använda sig av i undervisningen. De önskade också att en plan för undervisningen i programmering, med en tydlig progression genom högstadiet, togs fram under arbetets gång i programmet.

GENOMFÖRANDE OCH UPPFÖLJNING

Arbetet inom FoU-programmet har haft en hög prioritet på skolan, vilket blivit synligt bland annat genom att matematik- och tekniklärarna hade två timmar mötestid avsatt för området varje vecka, lett av processledaren. Mötestiden inkluderade alla lärare som undervisar i matematik och teknik samt skolans IKT-pedagog.

Rektor skapade förutsättningar genom att frigöra tid för återkommande forum varje vecka, samt

gav utrymme i tjänstefördelning till processledaren för att ge stöd till matematiklärarna. Skolans IKT-pedagog utvecklade digitaliseringen av undervisningen och såg till att skolans digitala infrastruktur fungerade. IKT-pedagogen följde också utvecklingsgruppens arbete. Att gruppen möttes regelbundet och deras arbete prioriterades var en signal till hela skolan att detta var ett viktigt utvecklingsspår.

Det som varit utmanande är att alla matematiklärare deltagit på utvecklingsseminarierna samtidigt, då det i en liten kommun som Simrishamn är svårt att ordna vikarier. Eftersom vi under tre år har lagt fokus på matematikämnet och programmering har de naturorienterade ämnena inte fått samma utrymme på skolans konferenstid som tidigare. Det har gjort att de naturorienterade ämnena inte har utvecklats i den utsträckning vi önskat. Det beror på att det är samma lärare som undervisar i de naturorienterade ämnena som i matematikämnet.

Processledaren och lärarna har varit positiva och drivande i arbetet. Rektor har regelbundet varit med på mötena och tagit del i processen. Spridning i kommunen har skett genom att politiker, rektorer och gymnasieskolan har informerats om programmet och vad vi gör. Representanter för alla kommunala skolor har träffats vid några tillfällen och tillsammans arbetat fram en plan för undervisningen i programmering för alla stadier. Detta arbete ska leda till en röd tråd från lågstadiet till högstadiet. Eleverna ska ha en likvärdig kunskapsbas att stå på när de går från ett stadium till ett annat.

Matematiklärarna har informerat övriga lärare på skolan om FoU-programmets syfte, innehåll och arbetssättet kring programmering på en arbetsplatsträff. Rektor, processledare och de flesta lärare i utvecklingsgruppen har varit med under hela programmet. Två lärare har tillkommit och två har slutat.

Lärarna har påbörjat Skolverkets moduler om programmering och lärarna har tagit fram tre lektioner med hjälp av *lesson study*. Forskare inom

FoU-programmet har varit på skolan vid två tillfällen och observerat vårt arbete med programmering. Erfarenhetsutbyte har skett med lärare från Åstorps kommun som också deltar i FoU-programmet.

Forskning och utveckling, samt arbetet med programmering ur ett ämnesdidaktiskt perspektiv, ingår i skolans och därmed även förvaltningens systematiska kvalitetsarbete. På skolan finns en struktur för resultatanalys inom ämneslagen, och arbetet följs upp i ämneslagets systematiska utvärdering av undervisning och resultat. Resultaten delges rektor som i sin tur analyserar ämneslagets utvärderingar och kommunicerar dem till förvaltningen. Förvaltningen efterfrågar analyserade resultat och sammanställer sin egen kvalitetsrapport och rapporterar till nämnden. Utifrån vad som framkommer i dialog med skolorna om deras analyserade resultat tas prioriterade utvecklingsområden fram. Denna arbetsgång har vi också utgått ifrån när vi arbetat med för att följa upp och utvärdera arbetet med programmering inom FoU-programmet.

De resultat som kommit fram när vi gjorde intervjuer och en enkätundersökning var att pojkarna hade ett mycket större intresse för programmering än flickorna. Pojkarna skattade också sin egen förmåga högre och tyckte att programmering var viktigare och roligare än vad flickorna tyckte. Alla elever på skolan kände sig tryggare med programmering än tidigare.

RESULTAT

Lärarna har utvecklat didaktiska arbetssätt, uppgifter och innehåll för undervisning i programmering, samt planerat för fortsatt utveckling. En tydlig plan

för undervisningens progression har tagits fram inom programmering för alla stadier i kommunen. Lärarna har använt sig av lesson study som är en vetenskapligt förankrad metod. Lärarna har skrivit en utvecklingsartikel om arbetet som skett kring programmering under tre år (Mårtensson & Olsson, 2020).

Det lärarna har upplevt som mest utmanande var att lära sig själva programmerandet, att våga programmera med eleverna och att få det som en naturlig del av undervisningen. Lärarna har fått tekniken att fungera på ett ändamålsenligt sätt. Samarbetet i lärargruppen har fungerat bra och alla har hjälpts åt med att ta fram lektioner, observera och driva arbetet framåt. Lärarna upplever att eleverna har blivit mer positiva till att programmera ju mer undervisning de fått. Eleverna uttrycker att programmering är spännande och att de vill lära sig mer.

Lärdomar och fortsatt utveckling

Lärarna har lärt sig grundläggande programmering och planerar att fortsätta utveckla sina undervisningsmetoder i den delen av matematiken som innehåller programmering. De vill också fortsätta att använda lesson study som metod för att utveckla undervisningen i matematik men även i andra ämnen. Lärarna önskar också att arbeta vidare kollegialt för att förbättra kvaliteten i undervisningen inom programmering. Svårigheten att införa ett nytt ämnesområde medförde att utvecklingsgruppens arbete med programmering tog betydligt längre tid än vad vi hade förväntat oss. Det tog också tid från annat ämnesdidaktiskt utvecklingsarbete, som därför fått stå tillbaka under tre år.

Vi har dragit lärdomen att det är utvecklande att under en längre period arbeta med ett och samma område.

TYRESÖ: ELEVERNA HAR LÄRT SIG BEGREPP OCH HAR PRECIS SOM LÄRARNA NU ETT GEMENSAMT SPRÅK

*Sara Vikslund, central processledare,
IT-strateg, Tyresö kommun*

Tyresö kommun är belägen i södra Stockholm nära vatten och skog och det är två mil in till centrala Stockholm. I kommunen bor cirka 48 000 invånare, det finns 13 kommunala grundskolor i varierande storlek och dessa är belägna i olika socioekonomiska områden. Huvudmannen har tidigare deltagit i ett annat av Ifous FoU-program och hade därför kännedom om upplägget sedan innan.

I FoU-programmet *Programmering i ämnesundervisningen* har fyra av våra kommunala grundskolor deltagit. Två stycken F–9-skolor, en 4–9-skola samt en F–6-skola.

Således har vi haft fyra utvecklingsgrupper i gång under dessa tre år. De har till största delen varit intakta. Utvecklingsgruppernas konstellation har sett något olika ut mellan våra fyra enheter. Det är främst lärare inom matematik, de naturorienterade ämnen och teknik som har ingått, samt några övriga ämneslärare efter intresse. På vissa enheter har även fritidshemmet deltagit. Det har funnits skillnader i frivillighetsgraden i deltagandet. På en skola var det ett obligatorium att alla MA/NO/TK lärare skulle delta, medan det på andra skolor var utifrån intresse och engagemang, snarare än ämnesbehörighet. Utvecklingsgrupperna har bestått av fyra till 14 deltagare.

Från Barn- och utbildningsförvaltningen har kommunens IT-strateg haft det samordnande ansvaret i form av rollen som central processledare. Dennes roll har varit allt från att stötta i klassrummen för att kunna genomföra lesson studies till att vid behov sitta med i utvecklingsgrupperna på de planerade mötena. En viktig del i detta är att komma med utvecklingsförslag och dela goda exempel från våra andra utvecklingsgrupper. Den centrala rollen innebär även att ansvara för delning och spridning av resultatet till våra övriga skolor som inte deltar i programmet i kommunen.

Skolchefen har deltagit i styrgruppen och haft en viktig roll i att frigöra medel och resurser för att vi ska kunna få ut så mycket som möjligt av vårt deltagande i programmet. Skolchef och IT-strateg har även haft regelbundna möten för att stämma av att vi är på rätt väg och om något behöver justeras så har beslut fattats runt det i dessa möten. Korta beslutsvägar är att föredra.

Rektorerna har ansvarat för att organisera och frigöra tid för utvecklingsgruppsmöten samt att se till att de tekniska förutsättningarna funnits på plats på enheterna.

Förkunskaperna inom programmering innan programmet startade var begränsade för de flesta av våra deltagande lärare, utom i några enskilda fall där man hunnit arbeta längre tid med programmering. Som exempel på de som kommit lite längre kan nämnas en skola där vi har en teknikprofilklass.

Förväntningar fanns på att detta skulle vara en praktisk fortbildning inom programmering. Många var oroliga hur tiden skulle räcka till och hur man skulle kunna genomföra programmet på bästa möjliga sätt trots tidsbegränsningar. De flesta utvecklingsgrupper har haft mötestider om 1,5 timme men med lite olika intensitet. I perioder har det varit varannan vecka för vissa medan det ibland har varit längre uppehåll på grund av att verksamheten krävt annat fokus tillfälligt. Ambitionen har dock varit att utvecklingsgrupperna bör träffas minst en gång per månad.

De flesta var positiva till att få dela kunskaper och erfarenheter med andra för att få ett hum om ifall de var på rätt väg eller inte, samt för att få inspiration och uppslag på hur de praktiskt kan göra i klassrummet. En viss osäkerhet fanns kring hur man skulle kunna/hinna lära sig så pass mycket att man kunde undervisa med programmering som verktyg. Denna

oro fanns såväl bland högstadielärare som lärare i de yngre åren.

GENOMFÖRANDE OCH UPPFÖLJNING

Vi har i huvudsak sett till att kunna genomföra så många lesson studies som möjligt och att producera artiklar i samarbete med forskarna (Sjöberg m.fl., 2019, 2018; Chibas m.fl., 2018). Det har gjorts lesson studies för alla åldrar och förkunskaper, och dessa har innefattat allt från analog programmering till textprogrammering.

Några skolor har även samarbetat sinsemellan gällande lesson studies och det har gett ringar på vattnet i form av att andra skolor som inte deltagit aktivt i programmet velat pröva dessa.

Inplanerade möten har i så stor utsträckning som möjligt genomförts och vi har även haft flera besök i klassrummen av forskarna Jalal Nouri och Eva Norén. De har såväl observerat som intervjuat elever i årskurs 5 och 7. Detta har gett otroligt mycket mer värde då både lärare, rektorer, förvaltning och elever verkligen fått syn på vad de gör, vad det resulterar i och hur deras erfarenheter faktiskt blir till resultat i forskningen ”på riktigt”. Praktik och forskning hittar här varandra på ett naturligt sätt, vilket vi inte hade trott skulle kunna ge en sådan kvalitetshöjning som det faktiskt har gjort.

Eleverna har uttryckt att de upplever att det de gör när de programmerar känns viktigt och flera har kommenterat att det var intressant med fler vuxna i klassrummet. På frågan om vad det innebar svarade eleverna att de som inte var deras lärare hade andra frågor och tankar kring programmering, vilket var kul eftersom det gjorde att de behövde tänka på flera sätt och i andra vinklar.

Huvudmannen hade som ambition att utbilda alla lärare i programmering via den högskolekurs som erbjudits via olika lärosäten. Dock lades denna ned i den form vi önskat, nämligen att kursen skulle hållas ute hos oss i vår kommun. Några enstaka lärare i kommunen har dock gått kursen enskilt. De som gjorde det tyckte att nivån var för hög och att den inte kändes relevant för att undervisa i grundskolan.

Vi har deltagit med föreläsningar på utvecklingsseminarium och delat med oss av bland annat våra ämnesövergripande lesson studies. Huvudmannen har genomfört nätverk där representanter från alla skolor i hela kommunen deltagit. Dessa har innehållit diskussioner om vad som ska undervisas om och när, tekniska möjligheter och funderingar, progressionen inom programmering, begrepp, vad forskningen kommit fram till hittills och på slutet har vi kommit till diskussioner kring det datalogiska tänkandet.

Dessa träffar har varit mycket uppskattade av framför allt de skolor som inte deltar i programmet eftersom de kunnat följa och kopiera det våra utveck-

lingsgrupper har gjort. Dessutom kommer man inte ifrån att det är mer inspirerande, och inte minst mer trovärdigt, att lyssna på kollegor som precis som en själv kanske saknat förkunskaper i början, men ändå lyckats genomföra högkvalitativa lektioner inom programmering. Våga prova har varit vår devis.

Vi har även startat ett centralt Digitek, där allt material för programmering finns att låna för alla skolor. Man kan även ta med sig en grupp elever dit och arbeta på plats. Två tjänster på 20 procent har inrättats centralt. Dessa två delas mellan alla skolor för att ge praktisk hjälp med programmering i klassrummen. Det har varit mycket uppskattat av lärare som haft svårt att komma i gång eller behöver ny inspiration.

RESULTAT OCH SPRIDNING

Vi har sammanfattningsvis sett ett mycket positivt resultat i vår kommun, inte minst bland eleverna. De lesson studies som har genomförts har gett flera måhända oväntade bieffekter till exempel i form av att elever som ofta inte tar sig ton inom ämnet matematik fått ett helt annat talutrymme. Det har varit så pass spännande med programmering att eleverna har börjat diskutera med kamrater de annars inte talar lika ofta med. Problemlösningsförmågan har ökat markant hos flera elever enligt lärarna.

Lärarnas lesson studies har även hjälpt till att till viss del ändra synen på ämnet matematik. De allra flesta elever har upplevt våra lärares lesson studies som lustfyllda och de har längtat till matematiklektionerna – även de elever som i vanliga fall är neutralt eller negativt inställda till matematikämnet. Det vi kunnat se är att när vi arbetat ämnesövergripande så har det blivit lite mindre diskussioner kring just matematiken och mer kring det andra ämnet. Det hade vi inte riktigt kunna förutspå. Detta blev något av en besvikelse för lärarna i matematik, då tanken från början var att det var just matematik som skulle diskuteras mest.

Eleverna har lärt sig fler begrepp och har precis som lärarna fått ett mer professionellt och gemensamt språk när de talar om det arbete de gör, jämfört med i början av FoU-programmet. Lärarlagen har skapat en gemensam syn på vad vi gör, varför och hur, och inte minst en mer avslappnad inställning till programmering; ett mod att våga prova, prova igen och dela med kollegor.

Skolinspektionen har under året granskat huvudmannen och våra högstadieskolors undervisning inom programmering. Tyresö kommun fick grönt ljus på samtliga deltagande skolor och för huvudmannen (Skolinspektionen, 2019). Vi tror att FoU-programmet är en bidragande orsak till detta. Framför allt för att man under dessa tre år haft avsatt tid att diskutera enbart programmering på alla nivåer.

Vi inom kommunen har nätverk som sprider resultat och erfarenheter från programmet flera gånger per termin samt ger handledning till enskilda lärare, skolledare och arbetslag efter önskemål och behov. Vi har ett resurs- och utvecklingscentrum som har en heltidsanställd som kan ta emot för att genomföra praktiska workshops och visa hands-on hur olika verktyg fungerar samt låna ut material.

Vikten av central samordning är tydlig för hela kommunens sammantagna utveckling och likvärdighet inom programmering. Även en engagerad rektor ses som central för framgång. Där rektorsbyte förekommit har det varit svårare för lärarna att kunna driva programmet framåt. Ledning och styrning i dessa program är oerhört viktigt, så innan man tackar ja till att delta bör man vara säker att rektorerna har tid och möjlighet att organisera för deltagandet samt att lärarna deltar frivilligt. På skolor där deltagande varit obligatoriskt har vi inte sett lika tydliga konkreta resultat av programmet. Oron över att inte ha tid för programmet som fanns i början har på några skolor bekräftats. Man upplever att det hade behövts mer tid till att arbeta med sina lesson studies och även mer tid för analys och reflektion. En viss besvikelse finns även i form av att man gått in i tron att detta var en ren fortbildning inom programmering. Som lärdom ser vi sammanfattningsvis att det är några punkter som är avgörande från vår synvinkel för ett alltigenom lyckat deltagande i programmet:

- Tillräckligt mycket tid avsatt för lokala utvecklingsgrupper
- En processledare som har mandat i gruppen samt en central processledare som håller ihop hela huvudmannens arbete och ser synergier
- En rektor som organiserar för medverkan i god tid innan programstart
- Att deltagandet är frivilligt
- Rätt förståelse för FoU-programmets syfte och upplägg innan deltagande
- Tillgång till olika digitala verktyg avsedda för programmering

I början var osäkerheten bland många lärare stor då man upplevde viss stress kring att inte ha nog mycket kompetens för att kunna programmera med sina respektive klasser. Vi ser under det sista året i programmet att diskussionerna har vänt från att vara tekniska i form av vilka program ska vi använda till att faktiskt handla om undervisning. Didaktiken är nu i fokus och vi pratar om det datalogiska tänkandet snarare än om man ska ha Scratch eller Beebot för olika lektioner. Detta är en tankeomställning som har tagit tid, men som har lyft oss till nästa nivå, sett så här i efterhand.

För Tyresö kommun har medverkan i programmet som helhet bidragit till ett ökat intresse för programmering inom våra kommunala grundskolor på alla nivåer (elever, lärare, personal på fritidshemmet, skolledare och huvudman), en förståelse för vikten av det datalogiska tänkandet och didaktiken runt programmering. Det har även gett oss möjlighet att samverka mellan skolorna på ett nytt utforskande sätt där man inte vet några svar i förväg kopplat till att programmering var relativt nytt för oss för tre år sedan då vi trädde in i programmet.

Vi har även sammanfattningsvis fått en ökad förståelse och kunskap kring hur programmering kan användas i undervisningen, exempelvis som redovisningsform och i ämnesövergripande arbete.

Den tid vi spenderat på programmet hade dock kunnat ökas om förutsättningarna tillåtit, för ett ännu större genomslag. Med facit i hand efter tre år är vi nöjda med vad programmet lärt oss, kontra den tid vi har lagt ned, och tycker att programmet gav den utdelning vi förväntat oss som huvudman.

ÅSTORP: ALLA ELEVER I KOMMUNEN HAR PROVAT PÅ PROGRAMMERING

*Annika Hoppe, ordförande i styrgruppen,
förvaltningschef. Åstorps kommun*

Åstorps kommun ligger i det nordvästra hörnet av Skåne och har en befolkning på cirka 16 000 invånare. Kommunen har en stor andel unga människor i ålder 0–19 år jämfört med landets övriga kommuner. Åstorp organiserar förskola, grundskola, särskola samt kommunal vuxenutbildning. I Åstorp finns ingen gymnasieskola, vilket innebär att en stor andel av gymnasieeleverna går till gymnasieskolor i Helsingborg och Klippan.

Åstorps kommun har som mål att vara en expansiv kommun, där barn och unga är en prioriterad grupp i kommunens utveckling. Utbildningen är en viktig del av de ungas vardag och den blir allt viktigare i takt med att andelen kunskapsintensiva yrken ökar. Det är därför viktigt att rusta barn och unga med kunskaper och kompetenser för framtidens arbetsmarknad.

Utifrån bildningsnämndens vision, ”Vi öppnar dörrar för framtiden genom att våra barn och unga är väl förberedda, att på ett ansvarsfullt och demokratiskt sätt utveckla vårt samhälle. Vi erbjuder en kreativ och stimulerande lärmiljö i funktionella lokaler, där barn och elever är delaktiga i sitt lärande, har tilltro till sin förmåga och där 100 procent av eleverna minst når lägsta godtagbara kunskapsresultat”, bedrivs ett systematiskt utvecklingsarbete i syfte att säkerställa goda kunskapsresultat och personlig utveckling hos eleverna i kommunen.

Orsakerna till att Åstorp gick in i forsknings- och utvecklingsprogrammet *Programmering i ämnesundervisningen* hösten 2017 var flera. Dagens skola står hela tiden inför nya utmaningar och nya förutsättningar, vilka tydliggörs i nationella styrdokument. Då programmering i ämnesundervisningen blev ett obligatoriskt inslag i undervisningen behövde lärare och arbetslag utveckla kunskap och kompetens inom ämnet samt utveckla didaktiska arbetssätt, uppgifter och innehåll för

programmering i undervisningen. Utbildningen ska vila på vetenskaplig grund och beprövad erfarenhet och i FoU-programmet skulle pedagoger i skolan få möjlighet att utveckla sin undervisning i programmering genom kollegialt lärande i samverkan med forskning. Genom samarbetet mellan praktik och forskning öppnas det möjligheter för pedagogerna i programmet att få ett vetenskapligt perspektiv på sin praktik och genom ett organiserat kollegialt lärande inom kommunen och med övriga kommuners deltagare utveckla arbetet med programmering.

ORGANISATION

Programmering i ämnesundervisningen är det tredje FoU-programmet som Åstorps kommun deltar i. Från våra tidigare erfarenheter, *Inkluderande lärmiljöer 2012–2015* och *Undervisning i förskolan 2016–2019*, har vi lärt oss att för att uppnå optimal spridning och nå effekt ut i yttersta ledet, är det viktigt att alla enheter deltar i utvecklingsprogrammet. Inför FoU-programmet beslutades en organisation i vilken alla skolenheter deltar men även representanter för skolbibliotek och kommunens pedagogiska mediacentrum.

Kommunens deltagande leddes och styrdes från förvaltningens ledningsgrupp, där rektorer, elevhälsochef och kultur- och fritidschef ingick tillsammans med utvecklingsstrateger och förvaltningschef. En av rektorerna företrädde alla rektorer i FoU-programmet. Skolledarnas roll var generellt att leda arbetet på hemmaplan, skapa förutsättningar för och följa upp det pågående arbetet med programmering i ämnesundervisningen. Det fanns samtidigt en lokal processledare och en pedagog från *Pedagogiskt mediacentrum (PMC)*, som arbetade nära pedagogerna som deltog i programmet. Processle-

darens roll och uppgift var att delta i programmet gemensamma processledarmöten tillsammans med de övriga huvudmännen processledare och arbeta utifrån programmet uppdrag. Processledaren var också stöd vid beslut i den lokala processen. Processledaren ledde detta arbete tillsammans med pedagogen från PMC.

De deltagande pedagogerna representerade alla skolor i kommunen och var i dialog med skolleningen utvalda utifrån intresse och kompetens. Utöver pedagogerna ingick även en av kommunens skolbibliotekarier samt tidigare nämnd representant från kommunens pedagogiska mediacentrum. Syftet med denna organisation var att alla elever fick tillgång till den kompetens som finns som gemensam resurs i kommunen, att säkerställa likvärdigheten för alla kommunens elever.

FÖRUTSÄTTNINGAR NÄR VI STARTADE PROGRAMMET

Under det första året fanns det en arbetsstruktur där utvecklingsgruppen träffades var tredje vecka för att skapa hållbara strukturer i det kommunövergripande arbetet. Gruppen inventerade bland annat vilka erfarenheter och vilka kompetenser som fanns på respektive skola, vilka verktyg skolorna arbetade med samt vilka behov det fanns för kompetensutbildning och hur denna kunde organiseras. Utvecklingsgruppen besökte bland annat Apple, testade Swift Playground och Sphero samt genomförde olika workshops (Blue-bots, ScratchJr, och Exel) i syfte att stärka gruppens kompetens genom kollegialt lärande.

GENOMFÖRANDE OCH UPPFÖLJNING

Utifrån genomförd inventering och gjorda erfarenheter fattades det beslut inom bildningsförvaltningen, om vilka kompletteringar gällande programmeringsverktyg, det behövde investeras i. Varje skola skulle ha en uppsättning av programmeringsverktyg. Skolbibliotekarien köpte in böcker om programmering till alla skolenheter, materiel som lärare och elever har tillgång till.

Under det andra året av programmet fick gruppen en ny processledare. Arbetet utvecklades och det formulerades en tydlig uppdragsbeskrivning för gruppens interna arbete. Alla deltagare formulerade också individuella målsättningar, framtagna i dialog med respektive skollening. I syfte att stärka gruppens arbete fick alla pedagoger 20 procents tid avsatt i sin tjänst för att implementera programmering på sin skola samt att intensifiera arbetet inom utvecklingsgruppen och få utökade möjligheter till

samverkan och gemensam kompetensutveckling. Pedagogerna har genomfört programmeringslektioner, genom *lesson studies* utvärderat och följt upp arbetet och samtidigt spridit kunskaper och erfarenheter både till varandra och till kolleger genom ett kollegialt lärande. Alla deltagare har träffats kontinuerligt tillsammans med processledaren och pedagogen från PMC, för att tillsammans driva arbetet framåt.

De inom programmet genomförda *lesson studies* var tidskrävande men oerhört lärorika. I detta arbete fick pedagogerna syn på saker, vilka hade varit svåra att upptäcka utan denna grundliga observation. Forskare inom programmet har varit i kommunen och genomfört intervjuer med elever och genomfört egna observationer.

I det fortsatta arbetet på hemmaplan har utvecklingsgruppen efter hand delat upp sig i mindre arbetsgrupper och dessa har varit stadiindelade. Pedagogerna har hjälpt och stöttat varandra i implementeringsarbetet på skolorna, genomfört inspirationsworkshops m.m. Dialog med kolleger och kontinuerligt kollegialt lärande har präglat processen.

När det gäller uppföljning och utvärdering av arbetet i klassrummet med eleverna har pedagogerna använt ett utvärderingsverktyg, en digital dokumentation, som heter LoopMe. LoopMe ger en möjlighet till direkt utvärdering och analys av aktiviteterna i klassrummet. Resultatet har sammanställts på grupp-, enhets- och kommunnivå.

FoU-programmets syfte och Åstorps syfte

Det övergripande syftet för alla huvudmän i FoU-programmet var att utveckla didaktiska arbetssätt, uppgifter och innehåll för undervisning i programmering. Syftet med Åstorps deltagande var att utveckla det didaktiska arbetssättet gällande programmering i undervisningen samt att utveckla kunskap och kompetens inom ämnet för alla pedagoger i kommunen, i syfte att säkerställa likvärdigheten i utbildningen. Utöver detta har pedagogerna aktivt arbetat med *lesson studies* under programmet. Vi har också utvecklat arbetet med LoopMe under pågående program.

FoU-programmets mål och Åstorps kommunala mål

I följande tabeller redovisas i vilken utsträckning Åstorp har lyckats nå FoU-programmets mål respektive sina egna kommunala mål kring programmering.

Tabell 3: Nuläget i relation till de gemensamma målen i FoU-programmet.

Mål i FoU-programmet	Status, måluppfyllelse
Arbetsmetoder har utvecklats för att kollegialt och i samarbete med rektor pröva, genomföra och följa upp inslag av programmering i undervisningen, så att eleverna utvecklar kunskaper i programmering inom bland annat matematikämnet.	Delvis uppfyllt genom arbetet med lesson study. Deltagarna i utvecklingsgruppen har på sina respektive skolor i olika grad genomfört lektioner, workshops och utbildningar på sina skolor. Programmering har använts brett, i fler ämnen än i matematik. Dock finns det fortfarande utmaningar ute på skolorna bland kolleger och skolledare.
Undervisning i programmering har en tydlig progression.	Forskargruppen i programmet har arbetat fram en progressionsmodell, och detta kommer att göras även på kommunnivå.
Beprövade och vetenskapligt förankrade undervisningsmetoder används kring programmering.	Via lesson study och LoopMe har undervisningsmateriel arbetats fram och beprövats av pedagogerna själva.
Utvecklingsartiklar har publicerats.	I skrivande stund har en artikel (Jonasson m.fl., 2020) publicerats och fler kommer att publiceras under 2020.
Skolan och huvudmannen har skapat förutsättningar och en beredskap för fortsatt forsknings- och utvecklingsarbete i takt med den tekniska och ämnesdidaktiska utvecklingen.	I det fortsatta utvecklingsarbetet kommer utvecklingsgruppen i programmering att få nya uppdrag inom ramen för teknik- och ämnesdidaktisk utveckling, detta i ett övergripande kommunperspektiv men även på den enskilda skolenheten.
Huvudmannen har en plan för hur kunskaper och erfarenheter från FoU-programmet ska spridas till andra skolor.	I Åstorps kommun har alla skolor deltagit i FoU-programmet. En fördjupad kunskap och förståelse behöver dock spridas till all pedagogisk personal, på respektive enhet.

LÄRDOMAR OCH FORTSATT UTVECKLING

Lärdomar:

- Ett framgångsrikt införande av programmering i undervisningen kräver gemensam implementering, avsatta resurser och tydligt uppsatta och utvärderingsbara mål.
- Andra faktorer som påverkar är tekniken, lärares tid för kompetensutveckling samt kollegialt lärande.
- Anpassad struktur och organisation för arbetet med införandet av programmering.
- Införandet av programmering underlättas av att arbetslaget har en person som driver frågan och som besitter en större kompetens i programmering.

- Arbetet med att dokumentera och analysera sina lektioner med hjälp av lesson studies och digital dokumentation genom LoopMe har medfört ett fördjupat lärande och en större medvetenhet gällande didaktiken.

Fortsatt utveckling:

- Skapa organisation och i övrigt goda strukturella förutsättningar för fortsatt utvecklingsarbete.
- Avsätta personella resurser med avsatt tid för att driva arbetet framåt.
- Kompetensutvecklande insatser för alla pedagoger.
- Strategiska val av teknik.

Tabell 4: Nuläget i relation till Åstorps kommunala mål med att medverka i FoU-programmet.

Kommunalt mål för programmering	Status, måluppfyllelse
Samtliga skolor i Åstorp har testat och spridit olika arbetssätt kring programmering i teknik och matematik	På samtliga skolor i kommunen har utvecklingsgruppen spridit olika sätt att arbeta med programmering i undervisningen, inte bara i matematik och teknik utan i flera ämnen. Spridningen har skett genom lektionsbesök, workshops, utbildningar och genom att gruppens medlemmar funnits närvarande som stöd.
Utvecklingsgruppen har spridit lektionsexempel på sin skola och till ämneskolleger.	Det här målet har delvis uppnåtts. Skolorna och deras personal har haft olika sätt att hantera möjligheten att få ta del av lektionsexempel.
Åstorp har en tydlig progression i undervisningen av programmering i årkurs 1–9 med exempel på lektioner som alla lärare kan använda.	Åstorps kommun har delvis utarbetat en progressionsplan i programmering. Fortsatt arbete kommer att bedrivas under hösten 2020.
Samtliga elever skall ha provat programmering.	Målet är uppnått. Alla elever har kommit i kontakt med och provat på programmering.

- Synliggöra aktuella transfereffekter till exempel problemlösningsförmåga, logiskt tänkande och att arbeta strukturerat och kreativt och fundera kring hur vi kan stärka dessa i olika ämnen i syfte att öka elevernas lärande.
- Skapa en kommunal progressionsplan innehållande bland annat en arbetsbeskrivning och en uppgiftsbank, i syfte att garantera likvärdigheten i vad eleverna ska lära sig.

Vår programmeringsresa under tre år, med guider i egenskap av processledare och forskarstöd, närmar sig sitt slut och nu ska vi stå på egna ben. Den påbörjade resan fortsätter på lokal nivå och vi ritar nu upp vår egen karta och tar ut en tydlig färdriktning. Vi fortsätter vårt arbete med goda förutsättningar att nå uppsatta mål.

FEM HUVUDMÄNS UTVECKLINGSARBETE

Anette Jahnke, projekt- och processledare Ifous

Vilka effekter har FoU-programmet Programmering i ämnesundervisningen haft på verksamheten – från huvudmannanivå till lärarnivå – hos de fem deltagande huvudmännen? Vilka mönster kan urskiljas mellan de fem huvudmännens berättelser? I detta kapitel görs en analys av huvudmännens kapitel.

Analysen tar sin utgångspunkt i den metastudie som gjorts av Ifous sju avslutade FoU-program presenterade i *Ifous fokuserar rapporten: Effekter av Ifous FoU-program* (Lindgren, Jahnke & Norman, 2020). I rapporten lyfts fem områden fram som FoU-program visat sig påverka: Undervisningen, lärarprofessionalismen; Organisation; Planer, mål och rutiner samt Systematiskt kvalitetsarbete. Två områden synliggörs där effekterna är mer oklara, dels i vilken mån programmet har påverkat elevernas skoltillvaro och prestationer, och dels i vilken mån forskningsarbetet och utvecklingsarbetet har samverkat.

Övergripande ger en jämförelse mellan metastudiens resultat och en analys av de fem huvudmännens berättelser om arbetet i FoU-programmet *Programmering i ämnesundervisningen* att två av de fem utpekade områdena påverkats: lärarprofessionalismen och undervisningen. Fler beskrivningar ges på elevnivå och om hur forskning och utvecklingsarbete samverkat i FoU-programmet om programmering, i jämförelse med programmen som analyserades i metastudien.

LÄRARPROFESSIONALISMEN HAR STÄRKTS

I alla de fem huvudmännens berättelser återfinns exempel på hur lärares samarbete stärkts, både inom ämnesområden och mellan lärare i olika ämnen. Det ges exempel på hur lärare inom en huvudman samarbetat över utvecklingsgruppernas gränser, mellan skolor.

Tyresö: Det har även gett oss möjlighet att samverka mellan skolorna på ett nytt utforskande sätt där man inte vet några svar i förväg ..//..

Samarbetsformen mellan lärarna i form av lesson study har stärkt kvaliteten genom att aktiviteter och material systematiskt har prövats ut. Det vetenskapliga förhållningssättet till att utveckla sin undervisning har utvecklats.

Freinetskolan Hugin: De lektioner som vi har i programmering på enheten nuförtiden är i mycket högre grad medvetna med ett tydligt syfte och mål samt väl valda verktyg, inte lustfyllda happenings som det lätt kunde bli inledningsvis.

Vidare beskrivs att lärarna har utvecklat både individuell och ny gemensam kunskap och ett gemensamt språk kring programmering och undervisning i programmering.

Tyresö: Lärarlagen har skapat en gemensam syn gällande vad vi gör, varför och hur, och inte minst en mer avslappnad inställning till programmering, ett mod att våga prova, prova igen och dela med kollegor.

Kunskaperna har ökat om hur programmering kan användas inom och utanför skolan. Lärarna har utvecklat sina kunskaper och förståelse för programmering, datalogiskt tänkande och programmeringsdidaktik. Speciellt lyfts att det skett en förskjutning i lärarnas fokus från teknik till *didaktik*:

Stockholm: Relativt samstämmigt beskriver skolorna att arbetet i utvecklingsgrupperna successivt förändrades från att kartlägga och diskutera syften och mål, hård- och mjukvara till att mer handla om undervisning.

Tyresö: Didaktiken är nu i fokus och vi pratar om det datalogiska tänkandet snarare än om man ska ha Scratch eller Bee-bot för olika lektioner. Detta är en tankeomställning som har tagit tid, men som har lyft oss till nästa nivå, sett så här i efterhand.

De nya kunskaperna har fördjupats och fått vidare spridning genom att lärare har presenterat genomförda lesson studies på externa konferenser och publicerat forskningsartiklar och utvecklingsartiklar (Sjöberg m.fl., 2019, 2018; Chibás m.fl., 2018; Norberg m.fl. 2019; Mårtensson & Olsson, 2020; Jonasson m.fl., 2020)

UNDERVISNINGEN I PROGRAMMERING HAR SKETT OCH UTVECKLATS

Alla fem huvudmännen beskriver i sina berättelser att undervisningen om, med och i programmering har förändrats under programmets gång. De framarbetade lektionerna genom lesson studies lyfts fram. Programmering förekommer inom undervisningen i många olika ämnen, och i ämnesöverskridande projekt. Programmets mål om att utvecklat didaktiska arbetssätt, uppgifter och innehåll för undervisning i programmering verkar ha uppfyllts. Flera huvudmän beskriver att de har utvecklat, eller arbetar med att utveckla tydliga planer för progressionen i undervisningen i programmering. Detta arbete är nära kopplat till de forskningsresultat i form av en progressionsmodell som forskarna återkopplat kontinuerligt till programmets deltagare vid processledarmöten och utvecklingsseminarium.

ELEVER MÖTER PROGRAMMERING I UNDERVISNINGEN OCH FÅR ÖKAT INTRESSE

Ett första steg i att eleverna utvecklar kunskaper i och om programmering är att de överhuvudtaget möter programmering i den ordinarie undervisningen. Två huvudmän lyfter denna aspekt, dels har Åstorp under programmets gång uppnått ett kommunalt mål där nu alla elever i kommunen kommit i kontakt med och provat på programmering. I Stockholms stad har mätningar gjorts 2017 och 2019 av hur stor andel (1=inga, 2=fåtal, 3=flertal, 4=majoriteten) av skolans elever som programmerar inom ramen för ordinarie undervisning. Mätningen 2017 totalt i Stockholm visade att variationen var stor, och att inga eller ett fåtal elever i staden mötte programmering i sin undervisning. Efter programmets slut konstateras:

Stockholm: Sex av [deltagande] skolorna ligger över stadens medelvärde (2,9) och om materialet jämförs sammanslaget och årskursvis är de positiva förändringarna kraftigare hos de deltagande skolorna jämfört med de genomsnittliga förändringarna i staden (Ödalen, 2020). Det vill säga, givet läget 2017 har utvecklingen varit mer positiv på de deltagande skolorna jämfört med staden som helhet. Skillnaderna är också signifikanta (Ödalen, 2020).

Tyresö kommun (och tre av kommunens fyra deltagande skolorna i FoU-programmet) blev under programmets gång föremål för Skolinspektionens kvalitetsgranskning av *Undervisning med digitala verktyg – matematik och teknik i grundskolans årskurs 7–9* (Skolinspektionen, 2020). I Skolinspektionens beslut beskrivs FoU-programmet och Skolinspektionen fann att verksamheten inom det granskade området fungerade väl.

Det som mest frekvent nämns av huvudmännen som en effekt av att eleverna nu möter programmering i undervisningen är att elevernas inställning är mer positiv, att de visar ett större intresse och engagemang för programmering. Det ges också exempel på en mer positiv inställning till matematikämnet.

I vilken mån eleverna utvecklat kunskaper och/eller förmågor, och inom vad, varierar mellan berättelserna. Stockholm lyfter det faktum att de genomförda lesson studies har haft elevens lärande som fokus. I vissa fall har de planerade aktiviteterna verkligen medfört att eleverna lärt sig det som avsetts, i andra fall inte. Forskningsresultaten, som består av en av övergripande analys av lesson studies, blir därför ett mycket viktigt resultat för att kunna fortsätta utveckla aktiviteter som de facto ger resultat på elevers lärande.

Stockholm lyfter ett förändrat språkbruk bland eleverna, där introducerade begrepp används i nya sammanhang. Detta är något som även Tyresö uppmärksammar:

Tyresö: Eleverna har lärt sig fler begrepp och har precis som lärarna ett mer professionellt och gemensamt språk när de talar om det arbete de gör nu i slutet av programmet än i början.

Freinetskolan Hugin konstaterar att vid stadiebyte har elevernas grundkompetens höjts, att basnivån är högre och spjutspetsarna skjutit än högre. Samtidigt som man lyfter att det finns mer att följa upp kring elevernas ”deltagande” i programmet:

Freinetskolan Hugin: Det som finns kvar att göra är att utvärdera vad eleverna fått med sig av programmet, genom enkäter och diagnoser ...//...

ETT SYNLIGT ”OCH” I FOU-PROGRAMMET

Alla huvudmän berör i sina berättelser den forskningsinsats som har pågått parallellt med och i samverkan med det utvecklingsarbete som skett i utvecklingsgrupperna. Stockholm anger att den direkta samverkan mellan forskare och utvecklingsgruppernas deltagare har varit sparsam. Övriga huvudmän beskriver hur de bidragit till forsknings-

arbete genom att deltagare intervjuats och undervisningen har observerats av forskare. Att dokumentationen från de genomförda lesson studies varit bidrag till forskningsarbete berörs av de flesta huvudmän. Flera relaterar också till de preliminära forskningsresultat som deltagarna har tagit del av i form av till exempel en progressionsmodell. Freinet-skolan Hugin lyfter också att forskare och praktiker har sett samma saker:

Freinetskolan Hugin: Under de två första åren såg deltagarna i utvecklingsgruppen vissa svårigheter med FoU-programmet. När forskarna i sina resultat presenterade samma svårigheter blev det ett nyvaknade för deltagarna och de fick bekräftelse på det de upplevt.

Fortfarande finns det vissa svårigheter med att väva in programmering i alla ämnen, men vi har fått större kunskaper om programmering och hur det kan nyttjas på ett värdefullt och lustfyllt sätt för både elever och pedagoger.

Att synliggöra ”och:et”, med andra ord på vilka sätt forskningsarbete och utvecklingsarbete samverkar och interagerar, är en av de utmaningar som meta-studien av avslutade FoU-program uppmärksammar (Lindgren, Jahnke & Norman, 2020). I detta program är samverkan tydligare uttryckt av deltagande huvudmän än i tidigare avslutade FoU-program.

KUNSKAP OM OCH UTMANINGAR MED ATT GENOMFÖRA FOU-ARBETE

Alla berättelser innehåller olika typer av utmaningar, insikter eller lärdomar kring att som huvudman, skola eller utvecklingsgrupp genomföra ett FoU-program. Tre av de fem huvudmännen har tidigare genomfört ett FoU-program i Ifous regi. Åstorp och Tyresö lyfter upp denna erfarenhet, och det som utmärker deras berättelser är att de tidigt har en organisation och avsatta resurser, i form av tid och personal, för hur kunskaper och erfarenheter bildade i programmet ska komma fler elever och skolor till del i deras verksamhet.

Sammanfattande lärdomar och utmaningar får bilda avslutning av detta kapitel om huvudmännens arbete inom FoU-programmet.

Viktiga faktorer för genomförandet

- Gemensam implementering, anpassad struktur och organisation för införandet av programmering
- Avsatta resurser
- Tydligt uppsatta och utvärderingsbara mål

- Tillgång till teknik
- Att personer som driver arbetet i form av såväl lokal som central processledare har mandat och tid för uppdraget.
- Tillräckligt med tid så att utvecklingsgrupperna kan genomföra både utvecklingsarbetet och bidra till forskningsarbetet
- En modell för hur undervisningen kan utvecklas systematiskt, i detta fall lesson studies
- Att rektor har engagemang, tid och möjlighet att delta. Vid rektorsbyten har arbetet blivit svårare att genomföra. Att rektor organiserar upp arbetet för lärare innan programmet startar.
- Att deltagandet är frivilligt för lärare
- Deltagarna har förförståelse för FoU-programmets syfte – det är inte en kurs.
- Sätta samman utvecklingsgrupper på ett strategiskt sätt. Till exempel lyfter en huvudman upp att deras fritidshem borde ha varit med i programmet från början.
- Inse att FoU-arbete under en längre tid är utvecklande, men också att annat utvecklingsarbete kommer att stå tillbaka. Vara beredd på konsekvenserna av detta.

Utmaningar i genomförandet

- Avsaknaden av kompetens i programmering
- Avsaknaden av teknik
- Svårigheter med hur programmering ska kombineras med ämnesinnehåll då programmering är en del av olika skolämnen kursplaner
- Att frigöra gemensam tid för att kunna genomföra lesson studies, vilket inkluderar att flera lärare ska kunna befinna sig i samma klassrum samtidigt, och att tid avsätts för att dokumentera och utvärdera arbetet systematiskt.

FORSKNINGSINSATSENS GENOMFÖRANDE OCH RESULTAT

Jalal Nouri, docent, Stockholms universitet

INLEDNING

Denna forskningsrapport är ett resultat av Ifous FoU-program *Programmering i ämnesundervisningen 2017–2020* som utfördes mot bakgrunden att Sverige i mars 2017 tog beslutet att införa programmering i grundskolans läroplan inom matematik- och teknikämnet. Detta gjordes som i ett led att skapa förutsättningar för elever att fungera som medborgare i en tid när digitaliseringen förändrar samhället, arbetsmarknaden och vårt sätt att leva och vara (Skolverket, 2017). Införande av programmering inom matematikämnet har av Skolverket motiverats utifrån argumentet att programmeringsundervisning dels resulterar i en utveckling av datalogiskt tänkande och digital kompetens, och dels i en stärkt matematisk problemlösningsförmåga. Under hösten 2018 genomfördes läroplansförändringen och lärare i grund- och gymnasieskola påbörjade arbetet med programmering i undervisningen. Ett antal utmaningar, som är av vikt för att belysa bakgrunden och inramningen av detta forsknings- och utvecklingsprogram, var associerade med läroplansförändringen.

När den nya läroplanen implementerades saknade en stor majoritet av lärarna både ämnes- och didaktisk kunskap relaterat till programmering. Kompetensutvecklingsinsatser föregick inte läroplansförändringen utan erbjöds parallellt och då främst med fokus på ämneskunskaper kring programmering. Nu, två år senare, erbjuder lärosäten kurser i programmeringsdidaktik. Det fanns inte heller tidigare forskning inom programmeringsdidaktik på grund- och gymnasienivå som stöd i arbetet (Kjällander, Åkerfeldt, Petersen, 2016; Åkerfeldt, Kjällander & Selander, 2018). Det rådde med andra ord en stor kunskapsbrist på området, både hos lärare och forskare. Detta var – och är fortfarande – troligtvis den största utmaningen associerad med införandet av programmering. En annan utmaning som fanns redan från början och som lärare även i dag behöver förhålla sig är att programmering infördes inom redan existerande ämnen, i synnerhet matematik och teknik, vilket i

sig är relativt unikt internationellt sett. Med detta följde antagandet att programmering skulle främja exempelvis matematiskt lärande och problemlösning, utan en kunskapsbas som pekade ut hur detta skulle ske. Från lärares perspektiv uppstod frågor som: hur ska programmering integreras inom matematikämnet? Hur ska vi få plats med programmeringsinslag i ett redan etablerat ämne? Vilka förmågor och kunskaper utvecklar programmeringsundervisning? Hur ska vi tänka kring progression, inom årskursen och tvärs över årskurser? Och inte minst: varför ska vi egentligen göra detta?

Det är mot en sådan bakgrund som hela programets övergripande frågeställning fick nedanstående formulering, och att vi i forskargruppen kompletterade denna med en rad underfrågor:

- Vilka didaktiska arbetssätt, innehåll och uppgifter stödjer utveckling och kvalificering av kunskap hos elever i grundskolan när de arbetar med programmering?

Denna generella fråga följdes av ett antal underfrågor som:

- Hur bör programmeringsundervisning planeras inom och tvärs över årskurser med progression i åtanke?
- Vilka förmågor och kunskaper utvecklar elever när de arbetar med programmering?
- Vilka pedagogiska förändringar uppstår när programmering införs?
- Vilka utmaningar följer med integreringen av programmering inom befintliga ämnen?
- Vilka ämneskunskaper samt programmeringsdidaktiska kunskaper behöver lärare utveckla för att kunna arbeta med programmering i grund- och gymnasieskola på ett meningsfullt sätt?

Vi tog med andra ord avstamp i att besvara de mest grundläggande didaktiska frågorna: Vad ska undervisnas (*innehåll*)? Varför (*potentiell effekt*)? Hur (*didaktiska arbetssätt*)? När (*progression*)?

METODOLOGI

I det praktiktäna forskningsarbetet antog vi i forskargruppen två roller, som handledare och som forskare. Som handledare agerade vi stöd och bollplank för lärarna i deras planerings-, analys- och utvärderingsprocesser. Som forskare utförde vi specifika och fokuserade studier där våra egna forskningsfrågor var vägledande. Desto viktigare för det praktiktäna forskningsarbetet på programnivå, är att vi kontinuerligt samlade in data från klassrummen och utförde holistiska analyser för att extrahera generella insikter och lärdomar som låg som underlag för formativ återkoppling till lärarna, så att de kunde planera det fortsatta arbetet med stöd i data och på ett evidensbaserat sätt. Vi ansåg att det var av stor vikt att kontinuerligt fånga upp lärdomar och få spridning av dessa i hela programgruppen. Exempel på detta är att vi i de allra första studierna – mot bakgrund av bristen på tidigare forskning – tog avstamp i att fånga upp pionjrlärares erfarenheter av programmeringsundervisning, i termer av använda didaktiska arbetssätt och upplevda effekter av dessa. Insiktarna som dessa analyser skapade kommunicerades sedan ut till samtliga lärare i programmet. Ett annat exempel är de holistiska analyser av de inledande lesson studies som utfördes under programmets gång. Dessa analyser identifierade utmaningar och möjligheter med beprövade didaktiska arbetssätt, vilka kommunicerades ut till programdeltagarna inför planeringen av nästkommande lesson studies.

DATAINSAMLING OCH ANALYS

Under programmets gång utförde vi ett stort antal klassrumsobservationer samt intervjuer med lärare och elever. Totalt utfördes ett 50-tal klassrumsobservationer, lika många lärarintervjuer och närmare 70 elevintervjuer. Vi har också samlat in dokumentation från 90 lesson studies utförda av lärare. Dokumentationen var i form av skriftliga lesson study-planeringar, samt skriftliga utvärderingar av utförda lesson studies. Utöver detta har dokumentation samlats in från cirka 15 workshops (genomförda under processledarmöten och utvecklingsseminarier). Gällande analysförfarandet har tematisk innehållsanalys (Braun & Clarke, 2006) använts för att analysera intervjuer som genomförts samt lesson study-dokumentationen.

LÄRDOMAR

När vi nu drygt tre år efter programstarten tittar tillbaka för att identifiera vilka lärdomar som gjorts, konstaterar vi att de är många. Att beskriva alla dessa med detaljrikedom och utförlighet i detta kapitel är

dessvärre inte möjligt. För intresserade läsare hänvisar vi till de publicerade artiklar som kapitlet också refererar till. I det följande redogör vi i stället med breda penseldrag, och på ett sammanfattande vis, de i vårt tycke centrala insikter som forskningsarbetet resulterat i. Denna redogörelse lyfter följande teman: hur programmering förändrat skolpraktiker, de fördelar och nackdelar som följer med programmering, utmaningar associerade med införandet och utförandet av programmeringsundervisning, samt inte minst didaktiska arbetssätt och metoder med hänsyn till progression.

FÖRÄNDRADE PRAKTIKER OCH FÖRHÅLLNINGSSÄTT

Lärarnas experimentella, kollegiala och inkluderande arbetssätt

Med tanke på att programmering var – och än i dag är – ett relativt nytt fenomen i för- och grundskola, så innebar själva införandet av programmering i ämnesundervisningen en direkt förändring i lärarnas arbetssätt. Om lärare i andra ämnen arbetat med stöd av gedigen ämnes- och didaktisk kunskap samt utifrån etablerade didaktiska arbetssätt, så blev utgångspunkten när programmering infördes i stället att utgå ifrån avsaknaden av sådan kunskap och etablerade praktiker. Det låg med andra ord i förutsättningarnas natur att främja ett mer experimentellt, utforskande och vetenskapligt förhållningssätt, samt att öppna upp för beprövandet av nya praktiker som gick bortom de invanda. Detta kom under programmets gång till uttryck på många sätt. Vi kunde exempelvis observera hur det experimenterades med lärar- och elevcentrade arbetssätt, att lärare som tidigare utgick från mer lärarledda aktiviteter övergick till elevcentrerade och vice versa. Det elevcentrerade arbetssättet kom med tiden att bli det dominerande arbetssättet bland lärarna i programmet, och implicerade en lärarroll med ansvar för inramning och förberedelse av läraaktiviteter och stöttning i klassrummet när elever behövde hjälp med individuella projekt.

Vi kunde också observera att lärare i och utanför FoU-programmet till stor utsträckning tog stöd i kollegiala diskussioner kring arbetssätt som fördes på och tvärs över skolorna, men också i diverse internetbaserade forum där lärare i hela Sverige samlades för erfarenhetsutbyte. I kontrast mot andra kollektiva och mer lärarledda didaktiska diskussioner involverades också elever, föräldrar och annan skolpersonal i både planering och utförandet av programmeringsundervisning. Både våra egna observationer och de intervjuer vi gjorde med lärare gjorde det tydligt att många elever givits utrymme att inta rollen som en pedagogisk aktör i klassrum-

met, med kunskaper och kompetenser kring programmering som kom till nytta för både lärare och övriga elever. Bland dessa elever som fått en mer utmärkande pedagogisk roll, fanns många gånger sådana som tidigare inte varit lika inkluderade i undervisningen, på grund av exempelvis bristande studiemotivation. Detta var för oss och de intervjuade lärarna en mycket intressant utveckling. I intervjuer med flertalet lärare framgick också att föräldrar med programmeringsyrken, eller övrig skolpersonal med relevant kompetens, involverades i planeringen av undervisning, för framtagande av material eller som bollplank för lärarna. Sammantaget drar vi slutsatsen att införandet av programmeringsundervisning gjordes genom en historisk unik kollektiv ansträngning som agerade samlande och inkluderande för väldigt många.

NYA INSLAG FÖR ELEVER

Utöver ett stort antal observationer i klassrum utfördes också 50 elevintervjuer med syftet att bättre förstå hur elever uppfattar integrationen av programmering inom matematikämnet i synnerhet. För eleverna har programmering varit ett mycket positivt inslag i skolan. I själva verket har vi ännu inte stött på någon elev som betraktat programmering som något negativt och icke-önskvärt. Vi har i stället observerat och fått höra av eleverna att det nya inslaget är motiverande och engagerande. I flertalet intervjuer med både elever och lärare har vi fått höra att motivationen knutet till programmering i många fall varit så hög att den förändrat elevers skolbeteendet. Exempel på sådana förändringar som nämnts vid flertalet tillfällen är att många lärare bevittnat ett ökat engagemang hos elever som tidigare uppvisat ett svagt engagemang inom exempelvis matematikämnet där programmering införts, eller att elever avstått från att gå ut på rasten för att i stället sitta kvar med Scratch-projekt. Flertalet lärare har också erfarit att nya elever tagit plats i klassrummet – att programmering medfört inkludering av elever som tidigare av olika anledningar befunnit sig i klassrummets periferi. Många lärare har erfarit att elever med svårigheter och diagnoser av olika slag, exempelvis elever som uppvisat sociala svårigheter, eller elever som tidigare uppvisat bristande motivation, tillhört den mängd som antagit en mer betonande roll i klassrummet – ofta en pedagogisk roll.

Vi har också observerat att programmeringsundervisning blivit en arena för engagemang för elever med estetiska intressen och inte minst för de elever som på sin fritid engagerar sig i spelande ("gamers"), i hög utsträckning på grund av blockprogrammeringens betoning av spelskapande och multimodala konstruktioner (Nouri & Mozelius, 2018; Nouri, Zhang, Mannila & Norén, 2019; Zhang

& Nouri, 2019a; Sjöberg, Nouri, Sjöberg, Norén & Zhang, 2018). För den senare gruppen i synnerhet, men i viss utsträckning också för den förra, har programmeringsundervisningen utgjort en meningsfull brygga mellan elevernas fritidsintressen och skolpraktiker, och på så sätt förstärkt den elevnära undervisningen.

Intervjuer med elever i låg- och mellanstadiet har kastat ljus på ett antal anledningar till det starka engagemanget och motivationen som elever uppvisar. I en studie utförd för att fånga upp elevers uppfattningar av programmering inom matematikundervisningen (Ahmed, Nouri, Norén, & Zhang, 2019) fastslog vi att elever uppfattar att matematikämnet transformerats med programmeringsinslaget – att det fått starka inslag av praktiskt skapande och kreativt arbete, samt med experimenterande och lekfullt lärande (spelorienteringen).

Flertalet lärare i förskoleklass samt låg- och mellanstadiet har därtill påpekat att flickor uppvisat ett större intresse för teknik än tidigare som en följd av programmeringsundervisning. Flickors intresse för programmering tycks emellertid vara störst i dessa stadier och avtar kraftigt i högstadiet. I studien som utfördes i Simrishamn av lärare som deltog i FoU-programmet (Olsson & Mårtensson, 2020) visade resultaten att pojkar var signifikant mer intresserade av programmering än flickor, och att cirka 90 procent av flickorna i årskurs 8 inte kunde tänka sig att arbeta med programmering längre fram i livet. Precis som lärarna bakom den nämnda studien rekommenderar vi framtida forskning som undersöker anledningar till detta dystra resultat.

Ytterligare förändringar som vi observerat när vi följt programmeringsundervisning är en förstärkning av ämnesöverskridande undervisning. Under programmets gång har flertalet lesson studies utförts med en sådan orientering, exempelvis med kombinationer som matematik, programmering och samhällskunskap i mellanstadiet (Sjöberg, Risberg, Nouri & Norén, 2019). Vi har också kunnat notera, genom dokumentation, observationer och intervjuer med lärare på högstadiet, att programmering kommit att utgöra en möjliggörare för ämnesöverskridande undervisning i vilken ämnen som naturvetenskap, teknik och matematik sammanvävts på ett meningsfullt sätt.

EFFEKTER PÅ ELEVERS KUNSKAPER OCH KOMPETENSER

Effekterna av programmeringsundervisning kommer att behöva studeras närmare framöver, och det har inte primärt stått i fokus för den forskning som utförts i detta program. Genom ett rikt lesson study-underlag som inkluderar lärarutvärderingar, samt genom intervjuer med lärare och elever, har vi

dock kunnat skapa en preliminär förståelse av programmeringsundervisningens effekter i relation till elevers kunskaper och kompetenser. I generella drag kan dessa effekter indelas i två övergripande kategorier: *programmeringsrelaterade förmågor* (datalogiska förmågor) samt *generella förmågor*. En mer detaljerad redogörelse för dessa effekter går att återfinna i Nouri, Zhang, Mannila & Norén (2019) och Zhang & Nouri (2019a). Utan att här gå närmare in på de datalogiska förmågor som programmeringsundervisning utvecklar, kan vi här nämna att de är i linje med de förmågor som pekas ut i ramverket för datalogiska förmågor som presenterats av Brennan & Resnick (2012), och det utvidgade ramverk som vi själva presenterade i Zhang & Nouri (2019a). Vad beträffar de generella förmågorna så kunde vi identifiera fyra underkategorier av förmågor: kognitiva förmågor och attityder, språkliga förmågor, förmågor relaterade till kreativ problemlösning, samt samarbetsförmågor.

Enligt lärares utvärderingar utvecklade programmeringsundervisning i relation till de kognitiva förmågorna det logiska och abstrakta tänkandet hos elever samt förmågan att arbeta med hög noggrannhet (Nouri, Zhang, Mannila & Norén, 2019). Därtill betonades utvecklingen av produktiva attityder som exempelvis modet att göra misstag och testa sig fram, vilket är en naturlig process i programmeringssammanhang, samt ett utvecklat tålamod och uthållighet i arbetet med komplexa uppgifter. Fler-talet lärare påpekade att det utvecklade tålamodet och uthålligheten inneburit en positiv utveckling för lärandet inom flertalet ämnen, som exempelvis för matematikämnet där många elever annars tidigt gett upp när svårigheter uppstått.

I ljuset av språkliga förmågor visar lärares utvärderingar att elever genom upprepade övningar i att komprimera kod och vara noggrann med syntax stärkt förmågan att skriva koncist och exakt. Många pekar också på att elever utökat sin vokabulär genom programmeringsundervisning och då i synnerhet i termer av en teknisk vokabulär. Många lärare betonar också – i linje med de antaganden som gjorts gällande datalogiskt tänkande och programmeringsundervisning (Brennan & Resnick, 2012) – att eleverna utvecklar förmågor relaterade till kreativ problemlösning, inklusive förmågan att i samarbete med andra lösa problem och vid behov bistå med pedagogisk kommunikation (Nouri, Zhang, Mannila & Norén, 2019).

Vi har också kunnat observera, och fått bekräftat från lärares utvärderingar, att programmeringsundervisning – i synnerhet då blockprogrammering i lärmiljöer som Scratch använts – har bidragit till att stärka såväl elevers digitala kompetens som mer specifika förmågor, till exempel sådana som har med multimodalt skapande att göra. Det som är särskilt intressant i detta sammanhang är att ovan nämnda

förmågor, exempelvis kollaborativ problemlösning och multimodalt skapande, är centrala element i diskurserna kring 21st Century Skills och digital kompetens (Ananiadou & Claro, 2009; Griffin & Care, 2015; Ala-Mutka, 2011; Erstad, 2008). Till exempel, i det konceptuella ramverket presenterat av Ala-Mutka (2011) inkluderas i digital kompetens avancerade färdigheter för skapande av material, problemlösning, samarbete och innovation, det vill säga färdigheter som överensstämmer väl med de teman för färdigheter som vi identifierat i detta FoU-program. När vi tittar närmare på de specifika digitala kompetenser som nämns i litteraturen, hittar vi specifika färdigheten att kreativt remixa (utgå och förändra något existerande), något som i vår tematiska analys är en underkategori av temat kreativa problemlösningskunskaper. Erstad (2008) beskriver denna färdighetskategori som en ”väsentlig del av digital kompetens” som representerar en ”förändring i våra skolor i dag, från kunskapsutveckling baserad på fördefinierat innehåll i skolböcker och reproduktion av kunskap från läraren till en situation där studenter tar tillgängligt innehåll och skapar något nytt, något som inte är fördefinierat ” (s.178). Element som samarbete och uthållighet listas också som tillvägagångssätt som är väl integrerade i befintliga datalogiska ramverk.

Således drar vi slutsatsen att programmeringsutbildning i grundskola, när man överväger lärarnas bedömning av den, inte bara utvecklar datalogiska färdigheter som till stor del har varit det primära målet när man introducerar programmeringsplaner (Brennan & Resnick, 2012; Heintz, Mannila, Nygård, Parnes, & Regnell, 2015), men främjar också färdigheter och attityder av mer allmän karaktär som är starkt associerade med det nya århundradets färdigheter och digital kompetens / läskunnighet.

UTMANINGAR

I ovan nämndes en rad positiva effekter som följer av det nya programmeringsinslaget i skolan. Vi har emellertid under programmets gång identifierat ett antal utmaningar som lärare ställts inför i relation till uppdraget att implementera programmeringsundervisning. En av dessa utmaningar har att göra med lärares kunskaper och kompetenser vad beträffar programmering och programmeringsdidaktik. När programmet påbörjades 2018 fanns få möjligheter till kompetensutveckling inom ämnet i form av utbildningar och kurser. Det växte fram allteftersom. Till bakgrunden hör också att det från forskningshåll endast fanns knapphändig forskningslitteratur kring programmeringsundervisning för yngre barn och för grundskolenivå. Lärarna blev således tvungna att ta utgångspunkt i det okända och pröva sig fram, och på så sätt skapa erfarenheter och kunskaper – vilket

de gjorde utomordentligt bra. Som innovatörer av programmeringsundervisning tog de fram otaliga meningsfulla didaktiska upplägg som avancerade förståelsen och bidrog till grunden för en programmeringsdidaktik för förskoleklass och grundskola. Vi kunde emellertid observera, liksom många lärare i programmet, att lärarnas förståelse för programmeringens grundbegrepp och praktiker – för datalogiskt tänkande – satte en tydlig gräns för dels hur mycket de kunde avancera programmeringsundervisning och ta hänsyn till progression inom och tvärs över årskurser, dels för effekten på och utvecklingen av elevers kunskaper och förmågor. Analyser av lärares datalogiska kunskaper och färdigheter visade också att de lärare som hade en förståelse för centrala datalogiska begrepp och praktiker kände sig mer bekväma med att planera och utföra programmeringsundervisning (Zhang & Nouri, 2019b).

Vi kunde också genom våra observationer och intervjuer, samt analyser av den första omgången av lesson studies, börja se att det exempelvis inom matematikämnet var utmanande att få till didaktiska upplägg som kunde förbättra elevers matematiska kunskaper och färdigheter, i de fall då elevers och lärares datalogiska grunder var begränsade. Liknande resultat/utmaningar har identifierats i andra länder, exempelvis i utvärderingar av nationella projekt i England (Benton m.fl., 2017; Boylan m.fl., 2018; Åkerfeldt, Kjällander & Selander, 2018).

Utbildningsreformens grundantagande att programmering kan vara ett medel för att främja elevers matematiska kunskaper och problemlösningsförmågor – utan ett fokus på och en utveckling av datalogiskt tänkande hos elever och lärare – visade sig vara utmanande att realisera. Detta resulterade i ett perspektivskifte i FoU-programmet, så att programmering började betraktas som både medel och ändamål. I andra och tredje rundan av lesson studies inkluderade således de flesta lärarna ett explicit fokus på att utveckla elevers datalogiska förmågor med utgångspunkt i centrala datalogiska begrepp som exempelvis variabler, slingor och villkorssatser. På så sätt möjliggjordes att eleverna kunde arbeta med mer avancerade programmeringsprojekt, liksom att lärarna kunde börja tänka kring progression samt skapa bättre förutsättningar för att nyttja programmering som ett medel för begreppsutveckling inom andra ämnen, exempelvis inom matematiken.

En annan genomgående upplevd utmaning i programmet för både lärare och oss i forskargruppen har att göra med det sätt programmering skrivits in i läro- och kursplaner. Många upplever än i dag att de relativt korta skrivningarna om programmering generellt är svårtydda, och att de på grund av att de är formulerade relativt lika tvärs över stadier inte ger vägledning kring progression. Tittar vi exempelvis på det nya centrala innehållet för matematikämnet, kan vi för årskurs 1–3 utläsa att arbete ska ske med

”Hur entydiga stegvisa instruktioner kan konstrueras, beskrivas och följas som grund för programmering”; för åk 4–6 ”Hur algoritmer kan skapas och användas vid programmering. Programmering i visuella programmeringsmiljöer”; och för åk 7–9 ”Hur algoritmer kan skapas och användas vid programmering. Programmering i olika programmeringsmiljöer”.

En serie med stegvisa instruktioner kan betraktas som en algoritm. Det innebär i princip att ett och samma mål är beskrivet, nämligen ett fokus på hur algoritmer kan skapas och användas vid programmering. Därtill framgår att eleverna förväntas göra detta i olika programmeringsmiljöer. Dessa formuleringar har väckt en rad frågor och utmaningar för lärare och oss som forskare: Hur ska vi arbeta med algoritmer tvärs över stadierna? Med vilken progression och med vilka specifika mål i sikte? Hur ska vi få plats med programmering inom existerande ämnen som redan är välfyllda med innehåll? Räcker det med att vi utför en analog programmeringsaktivitet i årskurs 1–3 för att klara av målet? Hur omfattande ska vi arbeta med programmering egentligen? Hur viktigt eller oviktigt är detta nya inslag i undervisningen? Hur säkerställer vi likvärdighet för elever om lärare tolkar målen och omfattningskraven olika?

I ljuset av samtliga ovan nämnda utmaningar kan det vara av relevans att problematisera och förnya diskussionen kring huruvida programmering bör föras in i existerande skolämnen eller göras till ett eget ämne.

DIDAKTISKA ARBETSSÄTT

Ett treårigt FoU-program med så många involverade lärare som arbetar med att ta fram, implementera och utvärdera didaktiska arbetssätt genererar naturligt en stor uppsättning arbetssätt. Baserat på en tematisk analys av dokumentation från 90 lesson studies (inkluderat lärares utvärderingar), våra observationsanteckningar och intervjudata, presenteras i det följande ett urval beprövade didaktiska arbetssätt som visat sig fungera väl.

Över lag identifierade vi fem programmeringsverktyg som användes som utgångspunkt för att strukturera och förmedla programmeringsundervisning. Dessa är analog programmering, robot- och app-programmering, blockprogrammering, mikrokontroll-programmering och textprogrammering. Var och en av dessa verktyg har sin egen uppsättning egenskaper och användningsområden. Dessa användningsområden har i vår analys brutits ned till två övergripande områden: utveckling av datalogiskt tänkande och allmänna ändamål (se tabell 5).

Nedan presenteras konkreta exempel på dessa didaktiska arbetssätt.

Tabell 5: En översikt av didaktiska verktyg och användningsområden.

Didaktiskt verktyg	Datalogiska förmågor	Generella ändamål
Analog (åk F-9)	Algoritmiskt tänkande och stegvisa instruktioner Läsa och skriva kod Testning och felsökning Kollaborativ problemlösning	Bedömning av datalogiskt tänkande För ökad motivation och engagemang
Robot (åk F-3)	Algoritmiskt tänkande och stegvisa instruktioner Läsa och skriva kod Testning och felsökning Kollaborativ problemlösning Villkorssatser Slingor Kollaborativ problemlösning	För ökad motivation och engagemang För ämnesöverskridande arbete För introduktion av blockprogrammering
Appar för spelbaserat lärande (åk F-3)	Algoritmiskt tänkande och stegvisa instruktioner Läsa och skriva kod Testning och felsökning Kollaborativ problemlösning Villkorssatser Slingor Kollaborativ problemlösning	För ökad motivation och engagemang
Block (Scratch Jr åk F-3, Scratch eller miljöer som Code.org åk 4-9)	Samtliga datalogiska förmågor i Brennan & Resnicks ramverk (2012) och i det utvidgade ramverket av Zhang & Nouri (2019).	För ökad motivation och engagemang Som medel för ämneslärande För ämnesöverskridande arbete För multimodalt skapande För utveckling av designtänkande
Mikrokontroller (åk 4-9)	Samtliga datalogiska förmågor i Brennan & Resnicks ramverk (2012) och i det utvidgade ramverket av Zhang & Nouri (2019).	För ökad motivation och engagemang För lärande i matematik, naturvetenskap och teknik För teknikkonstruktion (maker culture) Som brygga mellan block- och textprogrammering
Text (åk 4-9)	Samtliga datalogiska förmågor i Brennan & Resnicks ramverk (2012) och i det utvidgade ramverket av Zhang & Nouri (2019).	För ökad motivation och engagemang Som medel för ämneslärande För ämnesöverskridande arbete För skapande av mer avancerade program (exempelvis simuleringar)

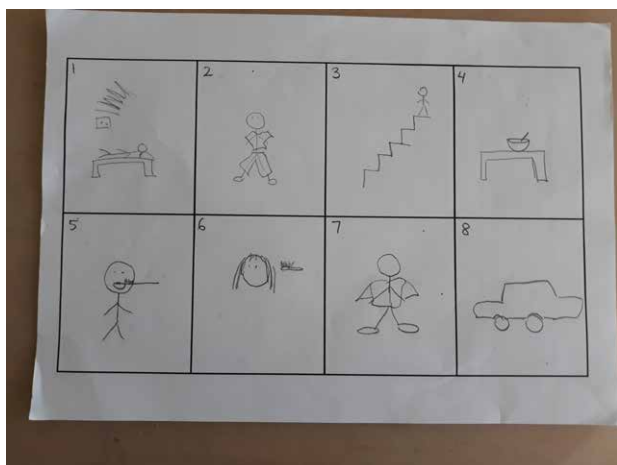
ANALOG PROGRAMMERING

Analog programmering innebär att tekniska verktyg och digitala medel inte används under programmeringsaktiviteten. Vi identifierade tre olika didaktiska arbetssätt för att analog programmering: 1) hela lektioner med analog programmering; 2) analog programmering som stöd för digital programmering;

samt 3) analog programmering som stöd för bedömning av datalogiska förmågor.

Hela lektionsmetoden användes huvudsakligen i de tidigaste stadierna (F-3) som den allra första introduktion till datalogiskt tänkande. Mer specifikt utvecklade lärarna dessa metoder för att göra eleverna bekanta med och göra dem intresserade av algoritmiskt tänkande (stegvisa instruktioner).

Figur 1: Analog programmering av en morgon-aktivitet representerad som en sekvens av åtta bilder.



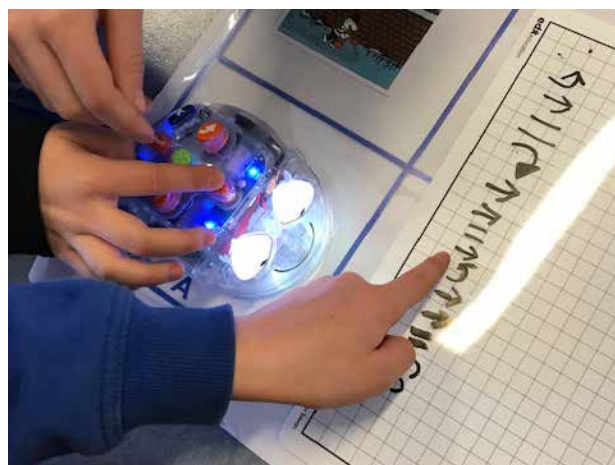
Detta gjordes så konkret som möjligt, inklusive till exempel kinestetiska inlärningsmetoder där eleverna programmerar varandra för att fysiskt röra sig på vissa sätt i eller utanför klassrummet, eller genom att koda sekvenser av instruktioner på papper med olika typer av representationer (teckning, bilder, text), eller för mer mogna elever, genom att använda abstrakta symboler av olika slag (till exempel navigationssymboler).

Figur 1 visar en sådan analog övning där eleverna fick i uppdrag att berätta om vad de hade gjort på morgonen innan de kom till skolan i form av en sekvens av bilder. I det här specifika fallet använde lärarna denna didaktiska metod för att anknyta till elevernas vardag. Genom att använda ritningar introducerades eleverna till att tänka i termer av stegvisa instruktioner och diskutera begreppet algoritmer, samt i att träna hur man skriver, läser och felsöker kod.

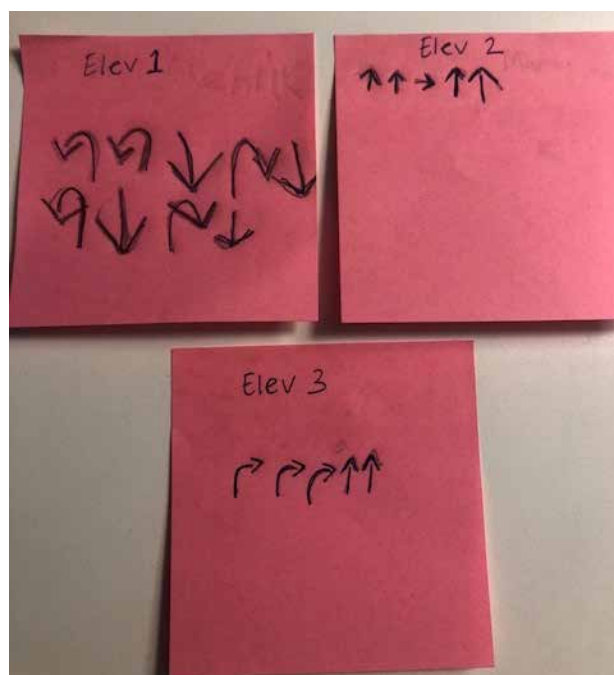
I den andra didaktiska metoden (använd på samtliga nivåer) nyttjades analog programmering som stöd för digitala programmeringsaktiviteter. I de flesta fall innebar detta att eleverna instruerades att koda på papper innan den faktiska kodningen av en robot eller i en digital programmeringsmiljö, för att stödja modellering av kod och problemlösning, samarbete och konversation kring koden samt felsökningspraktiker. I figur 2 ser vi hur elever i tredje klass skriver en sekvens av symboler som en Blue-Bot-robot är programmerad att utföra. I den specifika situationen testar och felsöker eleverna koden med stöd i analog programmering.

I det tredje didaktiska arbetssättet användes analoga medel för att stödja bedömning av elevers datalogiska färdigheter. Det vill säga, lärare utvecklade bedömningspraktiker i vilka elever skrev kod på papper som bedömdes av lärarna. Ett konkret exempel på detta demonstreras i figur 3 som visar

Figur 2: Analog programmering som stöd för digital programmering.



Figur 3: Exit ticketet med en sekvens av koder.



en exit ticketet som eleverna lämnar till läraren efter avslutad lektion.

ROBOTPROGRAMMERING

En annan vanlig didaktisk strategi i F-3-klassrum var att använda enklare robotar som Bee-Bots och Blue-Bots.

Lärandeaktiviteter som involverade sådana robotar skedde oftast direkt efter introduktion av programmering med analoga medel. Lärarutvärderingar, vår egen analys av lärardokumentationer samt

Figur 4: Kollaborativt ämnesöverskridande arbete med Blue-Bots.



observationer vi har gjort, har lett oss att identifiera ett antal didaktiska användningsområden för dessa typer av robotar, relaterade till utveckling av datalogiska förmågor och för mer allmänna ändamål. När det gäller färdighetsträning i datalogiskt tänkande är robotarna mycket lämpliga för att vidareutveckla eleverns algoritmiska tänkande (sekvenser och stegvisa instruktioner), förmåga att läsa och skriva kod, samt kollaborativa problemlösningsförmågor. I följande intervjuutdrag lyfter en lärare fram hur dessa robotar stöder elevernas problemlösningsförmågor och deras algoritmiska tänkande:

“Jag brukar göra det på två sätt. Antingen använder jag rutmattan som vi köpte med Bee-Bot, eller så konstruerar vi egna rutter på golvet med vit tejp. Sedan programmerar vi Bee-Bot för att korsa rutten. Ofta arbetar de parvis ... Jag tycker att Bee-Bots är utmärkta verktyg för att stödja elevernas logiska tänkande ... de tänker mer exakt och stegvis, och deras färdigheter för att lösa problem blir absolut bättre.”

I flera genomförda lektionsstudier kom lärarna också fram till att robotarna var mycket lämpliga för att bygga upp elevernas färdigheter och tänkande relaterade till testning och felsökningskod. Således verkar robotarna vara ett bra verktyg för att främja en central datalogisk praktik (testning och felsök-

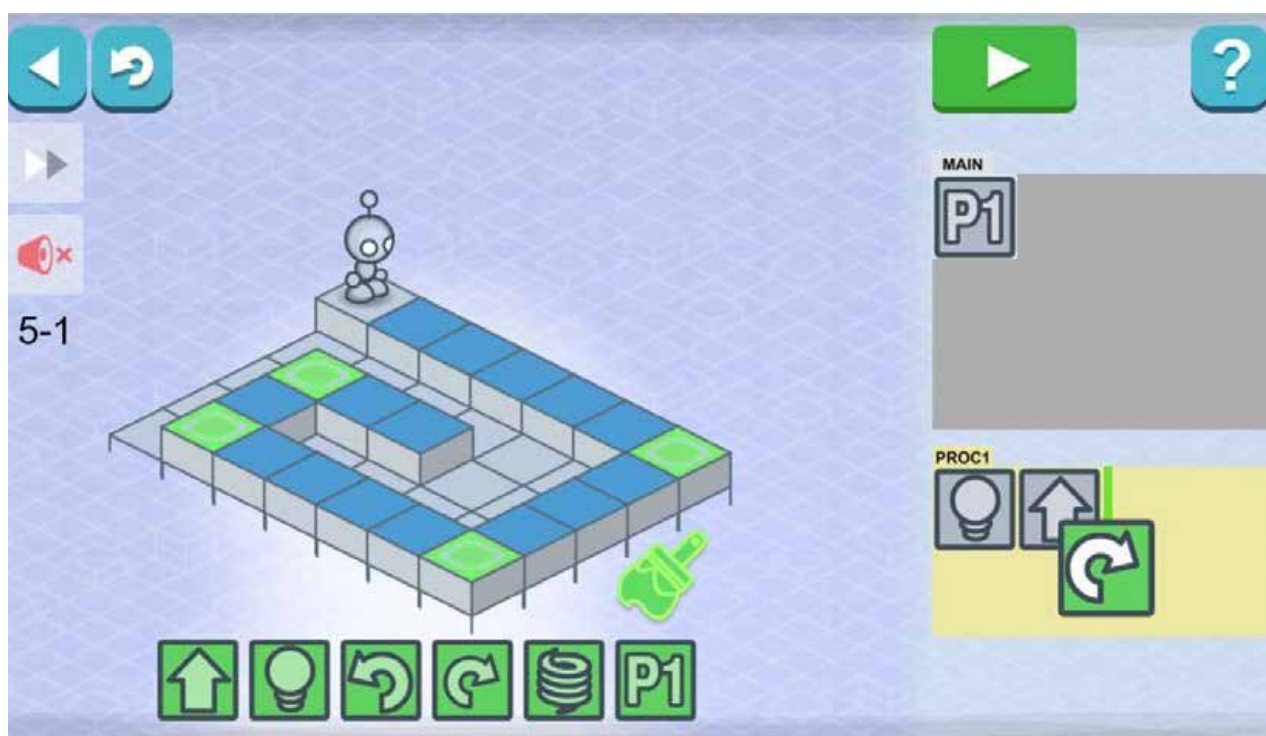
ning). Som exempel drog lärarna i en av lektionsutvärderingarna slutsatsen:

“Felsökningen blev mycket bättre under lektion 2 och 3 jämfört med den traditionella analoga lektionen med papper och penna. Eleverna var mycket mer engagerade och hade mer detaljerad diskussion när de hade Blue-Bots och kunde testa och felsöka ”

Vi identifierade tre huvudfördelar med att använda de enklare robotarna för att införa programmering. En av dessa fördelar är att både lärare och forskare i programmet ser att eleverna blir engagerade och motiverade när de lär sig att programmera genom konkreta robotar som de kan kontrollera. Detta betonas i princip i alla lektionsutvärderingar och intervjuer med lärare.

Ett annan fördel gällde ämnesöverskridande lärande. Lärarna i programmet har olika undervisningsämnen och har använt detta kreativt för att skapa meningsfulla ämnesöverskridande lektioner med robotar i fokus. I figur 4 ser vi ett exempel på en aktivitet som innehöll både programmering och matematik. I en deluppgift i den aktuella aktiviteten programmerade eleverna robotarna för att korsa en specifik rutt på rutnätet som innehöll geometriska figurer, och eleverna instruerades att identifiera och diskutera de geometriska figurernas namn och egenskaper.

Figur 5: Ett LightBot-pussel.



Som den tredje fördelen identifierade vi att andra typer av robotar, som Lego Wedo, kunde användas för att introducera blockprogrammering. Detta synliggjordes i en av de genomförda lesson studies, där lärare instruerade eleverna att bygga en Lego-robot och sedan programmera den med hjälp av blockspråk i stället för symbolprogrammering (navigering) som Blue-Bot och Bee-Bot. Denna metod är således ett exempel på hur robotar kan användas för att införa blockprogrammering. Enligt lärarna var det också ett mycket motiverande arbetssätt för eleverna: ”alla elever tyckte att det var kul”, och ”eleverna blir mer bekanta med blockprogrammering och med syftet och användningen av symboler”.

APPAR FÖR SPELBASERAT LÄRANDE

Ett annat sätt att lära sig grundläggande datalogiska begrepp och praktiker är genom användning av appar för mobil eller pekplattor. Appar som LightBot används ofta och var bland de populäraste valen för lärare. I huvudsak är LightBot ett spel som följer samma logiska struktur som tidigare beskrivna robotaktiviteter. Tanken med LightBot är att använda symboler för att skapa en rutt för en virtuell robot för att nå ett önskat mål på en rutnätstrukturerad karta (se figur 5). När eleverna arbetar med LightBot-appen tränar de vanligtvis datalogiska begrepp som algoritmiskt tänkande (sekvenser och stegvisa

instruktioner) och datalogiska praktiker som att läsa och skriva kod, testa och felsöka samt kollaborativ problemlösning. Men LightBot introducerar också programmeringskoncept som slingor och villkor, vilket Bee-Bot eller Blue-Bot inte gör.

BLOCKPROGRAMMERING

ScratchJr för de yngsta

Efter införandet av analog programmering och programmering via robotar introducerade lärarna i programmet ofta blockprogrammeringsaktiviteter. För de yngsta eleverna (F-3) gjordes detta huvudsakligen genom ScratchJr. I de observerade och analyserade lektioner där ScratchJr används är det vanligaste att eleverna skapar interaktiva berättelser och enklare spel. Genom detta fördjupar eleverna sin förståelse för datalogiska begrepp som algoritmer och stegvisa instruktioner (sekvenser), såväl som introduktion till begrepp som slingor, villkor och händelser. De stärks också ytterligare i färdigheter relaterade till datalogiska praktiker som att läsa / skriva / testa / felsöka kod, och introduceras till modularisering av kod (till exempel genom att koda enskilda delar av den interaktiva berättelsen och sedan sätta ihop allt i slutet). Vi har sett exempel på lärare som arbetar systematiskt och uttryckligen med alla ovan nämnda datalogiska färdigheter redan i förskoleklass (se fi-

Figur 6: En förskoleklasslärare förklarar begrepp som händelser och villkor.



gur 6), med framgångsrika resultat (eleverna är mottagliga och kan arbeta med begreppen och praxis).

Dessutom observerar vi, och lärarutvärderingen verifierar, att eleverna är mycket motiverade av det faktum att ScratchJr involverar många kreativa processer, särskilt när det gäller multimodalt skapande som involverar modaliteter som ljud, text, bilder och video. I en av lärarutvärderingarna förklarar de till exempel:

”De har nu möjlighet att förvandla sina tidigare erfarenheter och kunskaper på ett målriktat och kreativt sätt. Elementet att välja musik, visuella bakgrunder, olika figurer och andra effekter har en stor positiv effekt på elevernas skapande. Eleverna uttrycker stolthet för de saker de skapade.”

Vi observerar också att med möjligheterna till multimodalt skapande följer möjligheter för ämnesöverskridande lärande. Många av lektionerna som lärarna utvecklat i programmet involverar olika skolämnen som bild (ritning av bilder) och språk (röstinspelningar eller textbeskrivningar av berättelsen). Dessutom har vi observerat att designtänkande introduceras i samband med ScratchJr. Mer specifikt görs detta genom storyboards som lärare instruerar sina elever att skapa innan de börjar med själva kodningen (se figur 7). I de flesta fall har lärare infört denna typ av praktik för att möta en observerad utmaning, nämligen att elever tenderar att spendera

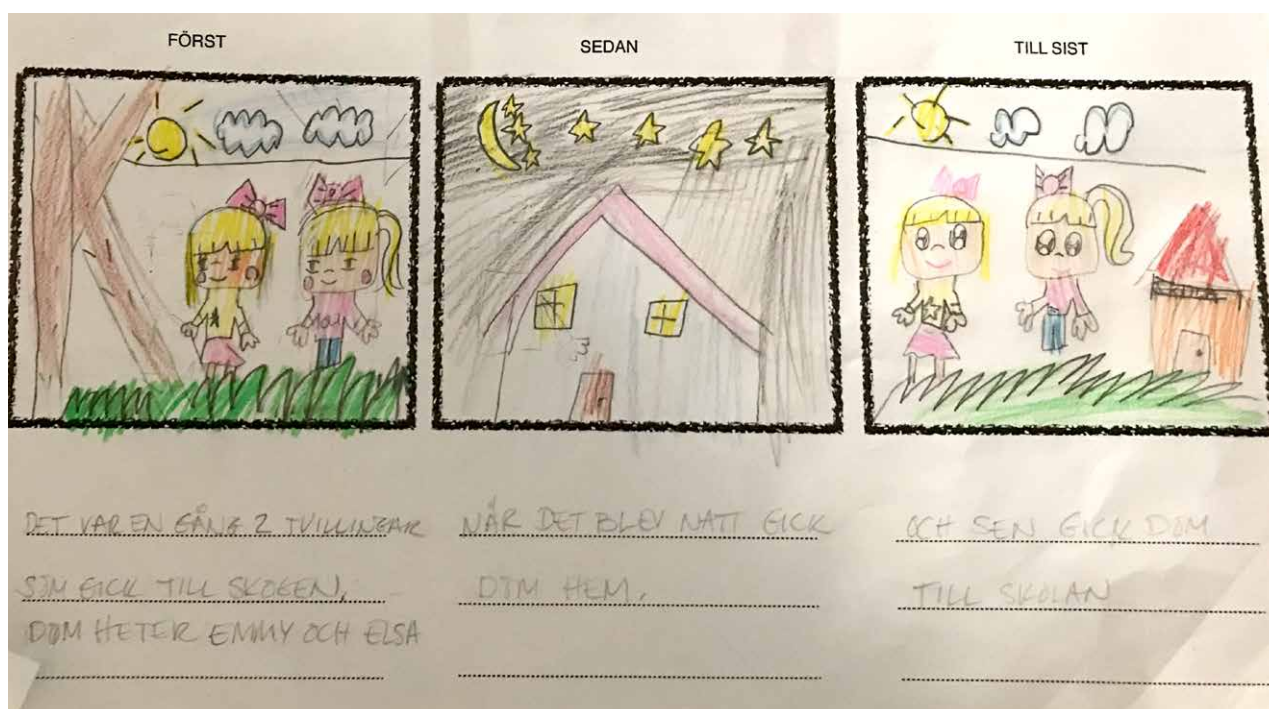
mycket tid med estetiska aspekter av berättelsen när de arbetar med ScratchJr, på bekostnad av att göra själva kodningen och lära sig datalogiska begrepp. Lärarna anger i sina utvärderingar att eleverna blir mycket mer fokuserade när de har en förberedd storyboard (en struktur) som utgångspunkt.

BLOCKPROGRAMMERING NÄSTA NIVÅ

För de äldre eleverna, från årskurs 3 till 9, gjordes blockprogrammering huvudsakligen genom standard Scratch, eller i vissa fall via online-miljöer som Code.org. De viktigaste skillnaderna mellan dessa två olika sätt att arbeta med blockspråk är att lektioner och aktiviteter är fördefinierade i onlinemiljöer som Code.org, medan Scratch saknar denna förutbestämda struktur, det vill säga användaren är fri att göra vad hen vill utan någon förutbestämd didaktisk struktur som avgränsar åtgärder som ska vidtas.

Våra observationer och analyser av genomförda lektionsstudier får oss att dra slutsatsen att en majoritet av Scratch-aktiviteter äger rum i årskurserna 4–6, men vissa också redan i klass 3 och en betydande del i årskurs 7–9. Som med ScratchJr utvecklade eleverna antingen spel eller interaktiva berättelser när de arbetade med standard Scratch. Till skillnad från ScratchJr ser vi dock att samtliga datalogiska begrepp och praktiker angivna av Brennan & Resnick (2012) kan arbetas med genom standard Scratch, vilket vi också visade genom den systema-

Figur 7: En storyboard gjord av en förskoleklass elev.



tiska översikt vi gjorde av internationell forskning med fokus på blockprogrammering i Scratch (Zhang & Nouri, 2019). Vi observerade också ett stort antal didaktiska arbetssätt. Vissa lärare planerar lektioner systematiskt med fokus på att utveckla specifika datalogiska färdigheter, medan en stor del gör det utan ett sådant uttryckligt fokus. Enligt våra undersökningar är detta delvis en reflektion av lärarnas datalogiska kompetens (se Zhang, Nouri och Norén, 2019), och delvis på grund av att de svenska läroplanerna inte innehåller några uttryckliga mål som är relaterade till datalogiska färdigheter.

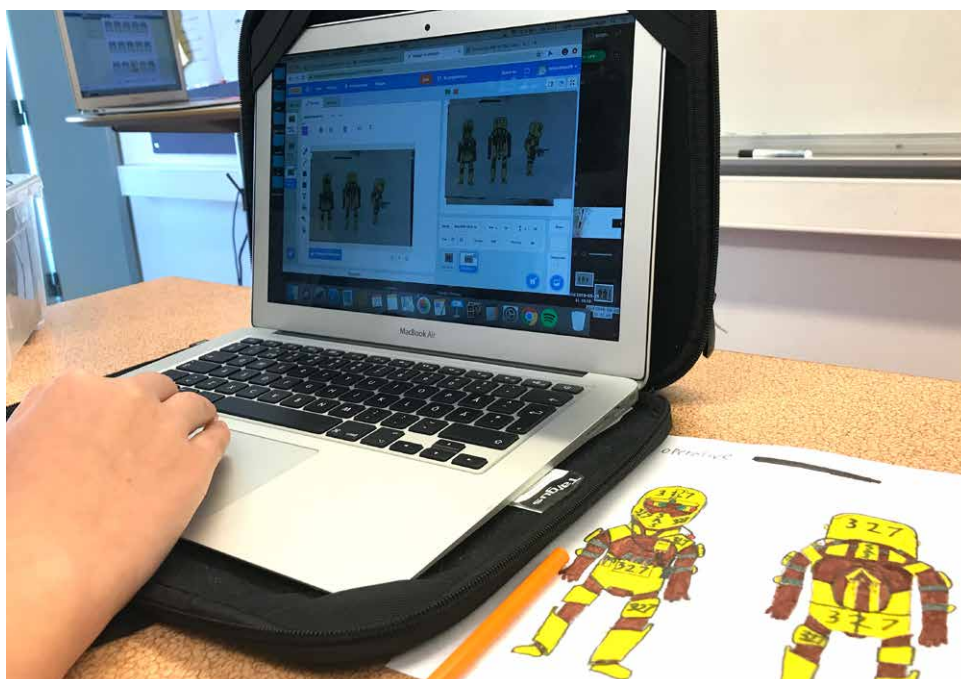
När det gäller mer allmänna fördelar är aktiviteterna med standard Scratch, i likhet med Scratch Jr, mycket engagerande för elever på alla nivåer och involverar i stor utsträckning kreativ design och multimodalt skapande. Vi observerar dock två skillnader: 1) designtänkande och multimodalt skapande sker på en mer avancerad nivå (se exempel i figur 8); och 2) elever är praktiskt taget uppdelade i två grupper när det gäller engagemang, såväl vad gäller det kreativa design- och multimodala arbetet som själva kodningen. Vi ser att skillnaderna i engagemang ökar med elevernas ålder. Även kön verkar vara en faktor eftersom flickor tenderar att spendera mycket mer tid med det estetiska arbetet och värdera det högre (bekräftat av observationer och intervjuer), medan pojkar primärt intresserar sig för kodning, i synnerhet när spel ska konstrueras. Ytterligare undersökningar behövs dock vad avser eventuella könsskillnader. Vi kan emellertid dra slutsatsen att

Scratch – som ett medium för att uttrycka estetiska värden – verkar vara mycket motiverande för estetiskt orienterade studenter oavsett kön.

Vidare kan vi, baserat på våra iakttagelser, dra slutsatsen att Scratch ofta används för ett ämnesöverskridande lärande som involverar flertalet skolämnen, och att lärare och elever uppfattar dessa aktiviteter som meningsfulla och engagerande. En stor del av Scratch-aktiviteterna görs också med särskilt fokus på enskilda skolämnen, oftast matematik då många på grund av läroplanens formuleringar antagit att programmering ska ses som ett instrument för elevers matematiska lärande. Baserat på våra observationer samt analys av lesson studies och lärarutvärderingar kan vi dra ett par slutsatser gällande användning av Scratch för att lära sig matematik. En första slutsats är att vi ser att Scratch kan stödja inläring av matematik på meningsfulla sätt. Till exempel genom att öka elevens motivation att lära sig matematik, vilket många lärare betonar, eller genom att introducera och tillämpa matematiska begrepp på ett kontextuellt meningsfullt sätt. Ett exempel på detta är att koordinatsystem ganska enkelt introduceras och förstås av små barn genom Scratch som till en viss utsträckning kräver att användaren navigerar i koordinatsystem. Andra exempel inkluderar användning av begrepp som variabler och funktioner som elever naturligt stöter på i Scratch.

Vi noterar dock också att lärare upprepade gånger i utvärderingar anger, vilket vi också observerade i

Figur 8: Multimodalt arbete med programmering av spel i årskurs 7.



klassrummen, att matematisk inlärnin g inte äger rum i den utsträckning lärarna hade hoppats på. Till exempel genomfördes flera lesson studies med kvasiexperimentell forskningsdesign som involverade kontroll- och experimentgrupper med för- och eftertester, i vilka eleverna lärde sig matematiska begrepp genom Scratch (eller ScratchJr) eller genom användning av mer traditionella metoder. Alla dessa studier drog slutsatsen att de traditionella metoderna (ofta papper och penna) var effektivare när det gäller att lära sig de matematiska begrepp / färdigheter som bedömts. Intervjuer med lärare och analys av mer kvalitativa icke-experimentella lektionsstudier bekräftar att många lärare kämpar för att använda programmering för matematisk inlärnin g. Enligt våra observationer beror detta på att elever och en del lärare inte har tillräckliga datalogiska kunskaper som krävs för att göra programmering till ett instrument för matematisk inlärnin g, vilket tycks botten a i två orsaker. För det första: i utvärderingar av lesson studies, samt i diskussioner med lärare och i samband med våra klassrumsobservationer, pekar lärare upprepade gånger på att eleverna möter svårigheter när det gäller den faktiska kodningen och att mycket tid ägnas åt att hantera dessa utmaningar på bekostnad av det matematiska innehållet och lärandet. Detta uttrycks tydligt i en av lärarutvärderingarna:

”Programmeringen gjorde uppgiften svår för eleverna. De blev distraherade och fastnade i programmeringsrelaterade frågor. De verkar

se uppgifterna [matematik och programmering] som separata och använde inte programmering som stöd. Deras resultat var sämre än eleverna i kontrollgruppen ”

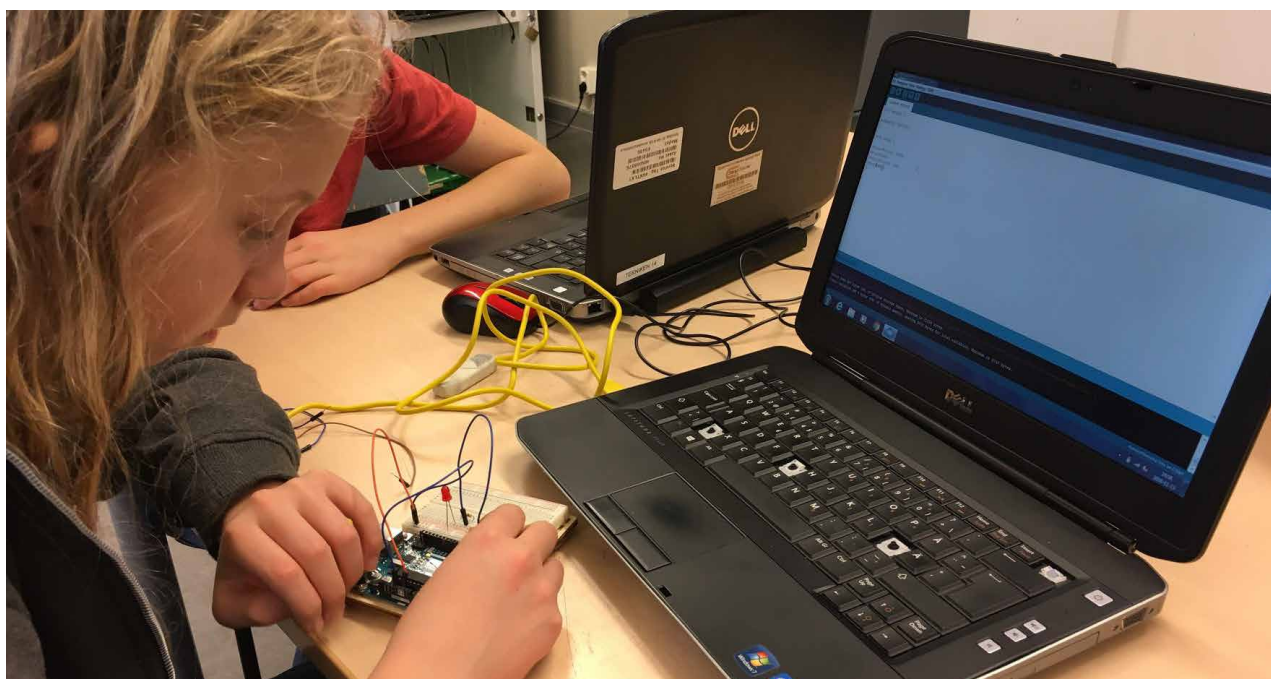
För det andra observerade vi att matematiklärare i programmet över tid alltmer uttryckligen planerar och utvärderar lektioner med mål som fokuserar på datalogiska begrepp. På förfrågan, och bedömt utifrån deras utvärderingar, förklarar de att de har identifierat behovet av att först fokusera på de datalogiska begreppen för att skapa bättre förutsättningar för matematiskt lärande.

PROGRAMMERING MED MIKROKONTROLLER

Mikrokontroller som Micro:bit eller Arduino är exempel på ett annat distinkt didaktiskt verktyg som lärarna i FoU-programmet använt sig av när de arbetat med programmering i skolan. Dessa mikrokontroller är intressanta för lärare och elever dels för att ett antal sensorer är integrerade i styrenheterna som kan användas för olika ändamål, och dels för att de kan programmeras både med hjälp av block- och textspråk. Det går också att växla mellan block- och textkod för jämförelse mellan språken.

I FoU-programmet har mikrokontroller endast använts i klass 4 och uppåt. I de flesta fall har endast ett fåtal datalogiska begrepp använts och uttryckligen fokuserats, såsom variabler, loopar, villkor och funktioner (abstraktion), men vi har också observe-

Figur 9: En makerspace-aktivitet med mikrokontroller och elektronik.



rat lärare och elever som arbetar med samtliga begrepp, praktiker och perspektiv i olika elevprojekt. Således kan vi dra slutsatsen att alla datalogiska färdigheter i Brennan & Resnicks (2012) och Nouri & Zhangs (2019) ramverk kan utvecklas genom användning av mikrokontroller. Ur vårt perspektiv är den mest intressanta användningen av dessa mikrokontroller förmodligen av mer allmän natur än specifikt relaterat till datalogiskt tänkande. Vi observerar, och lärare verifierar, att dessa mikrokontroller stöder meningsfullt arbete med sensorer och annan elektronik, som bidrar till en tillverkarkultur (maker culture) vilket är mycket engagerande för elever. Till exempel, i ett av klassrummen som observerades i årskurs 7, hade läraren utformat ett makerspace där allehanda elektroniska och tekniska resurser gjordes tillgängliga för eleverna. Den didaktiska metoden som användes var ganska fri och öppen. Läraren hade en enkel instruktion: bygg vad du vill med hjälp av mikrokontroller och de tillgängliga resurserna i skolan. Lärarens roll blev att stötta eleverna när de modellerade problemet och lösningen samt att hjälpa till vid nedbrytning av problem och lösning i delar. Eleverna uppmuntrades också att använda online-resurser för att lära sig att koda specifika delar och slutligen montera hela lösningen.

Således är de allmänna fördelarna, som både vi och lärarna identifierade, att mikrokontrollerna kan användas för att öka elevernas engagemang och därmed också användas för lärande som involverar ämnen som matematik, fysik, elektronik och för att förstå hur teknik (hårdvara och programvara) kan

konstrueras. Läraren som utformade makerspaceklassrummet uppgav detta tydligt enligt följande:

”Att arbeta med mikrokontroller som Arduino ger en fördel för interaktion med fysiska prylar som motorer, lampor, servor etc. Min erfarenhet är att eleverna är lättare att motivera för den typen av programmeringsprojekt. Det som också är bra är överflödet av webbplatser där eleverna kan kopiera kod som sedan kan modifieras och användas i egna projekt ”

Dessutom har vi observerat att mikrokontroller utgör en bra brygga mellan block- och textprogrammering. Många lärare introducerar textprogrammering för eleverna genom att först låta dem koda mikrokontroller med blockspråk. Programvaran som medföljer dessa mikrokontroller gör det möjligt för eleverna att översätta blockkoden till text för jämförelse.

TEXTPROGRAMMERING

Programmering med textbaserade språk som Python eller Javascript (som är de vanligaste i programmet) skedde huvudsakligen i årskurs 7–9, men också i viss utsträckning i årskurs 4–6. I de flesta fall instruerades eleverna att remixa kodmallar som läraren tillhandahöll, alternativt fanns tillgängliga på Internet, eller som gick att följa via online-handledning av kodningsprocessen. Vi kunde observera att det var relativt ovanligt att lärare utgick från lä-

rarledda aktiviteter med fokuserade genomgångar kring de grundläggande programmeringskoncepten och strukturerna, för att sedan tillhandahålla övningar och uppgifter där eleverna får instruktioner om att koda något från grunden. Vi har med andra ord endast observerat några enstaka fall där lärare systematiskt har arbetat med att utveckla elevernas datalogiska färdigheter på bredden. Vissa specifika datalogiska praktiker blev mer fokuserade än andra, som till exempel remixing och testning och felsökning. Vi tror att detta dels kan förklaras av lärarnas ovana med textprogrammering och av deras begränsade färdigheter med de datalogiska begreppen och praktiker, och dels av att de svenska läroplanerna inte innehåller några uttryckliga mål som är relaterade till datalogiskt tänkande.

I stället ägnade lärarna en större del av den begränsade tid de hade till förfogande åt att försöka göra programmering till ett instrument för ämnesinlärning (främst för matematikinlärning) – i enlighet med läroplanens förväntningar. Som i fallet med blockprogrammering resulterade många av dessa försök till både önskade och framgångsrika resultat. För att ge ett tydligt exempel: eleverna i ett av de observerade klassrummen använde textprogrammering som en plattform för att göra matematiska simuleringar, något som är svårt att göra på annat sätt. Mer specifikt arbetade de med det matematiska begreppet "sannolikhet" och kodade program som simulerade miljontals tärningskast. För att göra detta behövde de utveckla en förståelse för datalogiska begrepp som variabler, data, villkor, loopar, funktioner och så vidare.

Ytterligare andra allmänna fördelar var att textprogrammering visade sig vara ett bra verktyg för elever som är särskilt intresserade av programmering. Eftersom textprogrammering kan användas för att skapa mer avancerade program (och spel), stöder det en progression av datalogiska färdigheter. I programmet har vi upprepade gånger observerat att lärare från årskurs 4 till 9 erbjuder textprogrammering till de mer avancerade eleverna som vill göra mer än vad som är möjligt att göra i blockspråk. Resultaten har varit imponerande. Med stöd av lärare, kamrater, online-gemenskaper och övriga tillgängliga resurser har dessa elever utvecklat fullt fungerande

erande spel med grafik och all förväntad spelmekanik – och programmen har oftast krävt utveckling av den matematiska förståelsen.

Progressionsmodell för programmeringsundervisning i förskoleklass och grundskola

En fråga som vi forskare och programmets deltagande lärare sökt svar på under hela programmets gång har varit hur lärare kan planera undervisningen inom årskursen och tvärs över årskurser med hänsyn till progression. Det treåriga programmet och det stora antalet beprövade och utvärderade lektioner, i kombination med ett i övrigt rikt dataunderlag, har möjliggjort framtagandet av en progressionsmodell som består med vägledning i denna fråga. Modellen är framtagen baserat på en analys av samtliga utförda och utvärderade lesson studies (90 stycken), observationsanteckningar, intervjudata, och inte minst genom att den vid flertalet tillfällen under programmets gång presenterats för programmets deltagande lärare som i workshops diskuterat och reflekterat kring modellen och bistått med feedback. På så sätt är den starkt förankrad i och validerad av lärares vetenskapligt beprövade och utvärderade praktiker – och den är ett resultat av de kollektiva ansträngningar som gjorts i detta treåriga FoU-program. För en mer detaljerad beskrivning av modellen, se (Nouri & Zhang, 2021).

Progressionsmodellen (se figur 10) består av tre dimensioner, datalogiskt tänkande (DT), digital kompetens och programmeringsverktyg, samt fyra årskursnivåer.

Vi ser att modellen kan ha två övergripande användningsområden. Det huvudsakliga användningsområdet är att den kan användas av lärare och skolor vid planering av programmeringsundervisning tvärs över årskursnivåer. Modellen vägleder kring vilka programmeringsverktyg som kan vara relevanta på de olika nivåerna och vilka datalogiska kunskaper och färdigheter som kan fokuseras. Därtill tänker vi oss att modellen kan informera framtida revideringar av läro- och kursplaner, om inte annat utgöra ett erfarenhetsbaserat inspel bland andra.

Tabell 6: Progressionsmodell för programmeringsundervisning i förskoleklass och grundskola.

Datalogiskt Tänkande (DT)	DT nivå 1 (F)	DT nivå 2 (åk 1–3)	DT nivå 3 (åk 4–6)	DT nivå 4 (åk 7–9)
	<p>Introducera DT begrepp:</p> <ul style="list-style-type: none"> • Sekvenser, stegvisa instruktioner <p>Introducera DT praktiker:</p> <ul style="list-style-type: none"> • Testa och felsöka • Inkrementellt och iterativt arbete <p>Introduktion av DT perspektiv:</p> <ul style="list-style-type: none"> • Kollaborativ problemlösning 	<p>Introducera DT begrepp:</p> <ul style="list-style-type: none"> • Villkor • Slingor <p>Introducera DT perspektiv:</p> <ul style="list-style-type: none"> • Gestaltning • Kritiskt perspektiv <p>Fördjupning DT begrepp:</p> <ul style="list-style-type: none"> • Sekvenser, stegvisa instruktioner) <p>Fördjupning DT praktiker:</p> <ul style="list-style-type: none"> • Testa och felsöka • Inkrementellt och iterativt arbete <p>Fördjupning DT perspektiv:</p> <ul style="list-style-type: none"> • Kollaborativ problemlösning 	<p>Introducera DT begrepp:</p> <ul style="list-style-type: none"> • Händelser • Parallellism • Operatorer • Data <p>Introducera DT praktiker:</p> <ul style="list-style-type: none"> • Abstraktion • Modularisering • Återanvända och remixa <p>Fördjupning DT begrepp:</p> <ul style="list-style-type: none"> • Villkor • Slingor <p>Fördjupning DT perspektiv:</p> <ul style="list-style-type: none"> • Gestaltning • Kritiskt perspektiv 	<p>Fördjupning och tillämpning av samtliga DT begrepp, praktiker och perspektiv</p>
Digital kompetens		Design och multimodalt skapande	Design och multimodalt skapande	Design och multimodalt skapande
Programmeringsverktyg	Analog, enkla robotar, Appar	Analog, Appar, Robotar och Block	Block och Mikrokontroller	Block, Text. Mikrokontroller

UTVECKLADE LEKTIONER MED HJÄLP AV LESSON STUDIES

Anette Jahnke, projekt- och processledare Ifous

Inom ramen för programmet har lärarna i sina utvecklingsgrupper genomfört 90 lesson studies under hösten 2018 samt våren och hösten 2019. Som inspiration och som stöd i arbetet har artikeln *Learning to learn from teaching: a first-hand account of lesson study in Japan* (Bradley m.fl., 2014) använts. Utifrån artikeln, och samarbete mellan forskare, processledargruppen och Ifous process- & projektledare togs tre stöddokument fram. Dokumenten, som ligger som bilaga 1, 2 och 3, tjänade flera syften. Planeringsmallen (bilaga 1) och debriefing-mallen (bilaga 2) stödde det lokala genomförandet med att planera, genomföra och utvärdera lektionen. Presentations-

mallen i form av en PowerPoint-mall (bilaga 3) gav stöd åt lärare så att de på de gemensamma utvecklingsseminarierna kunde delge varandra sina erfarenheter vid så kallade bordpresentationer (lärare presenterade för andra deltagare vid ett runt bord). På detta sätt blev gjorda erfarenheter kollegialt delade och i viss mån granskade. Slutligen syftade stöddokumentet till att förse forskningsgruppen med data att analysera, det vill säga att alla dokument delgavs forskargruppen.

I detta kapitel ges exempel på tre genomförda lesson studies från tre stadier baserade på lärarnas redovisade PowerPoint-presentationer.

HITTA STRATEGIER FÖR MATEMATISK PROBLEMLÖSNING MED HJÄLP AV SCRATCH.JR I ÅRSKURS 3

Lektion utvecklad av: Teresa Jörgensen, Lena Abbing, och Åsa Chibás, Freinetskolan Hugin

Lärarna hade sett att eleverna hade svårt för matematisk problemlösning och att tänka steg för steg. Syftet med lektionen som utvecklades var att låta programmeringen hjälpa eleverna i detta. Genom att programmera "lästalet" i matematik steg för steg var idén att detta skulle stödja eleverna att "ta sig in" i problemet för att lättare kunna lösa det. Det centrala innehållet i kursplanen för matematik som lektionen syftade att behandla var: 1) strategier för matematisk problemlösning i enkla situationer; 2) matematisk formulering av frågeställningar utifrån enkla vardagliga situationer.

Lektionen utvecklades av tre lärare och prövades ut vid tre tillfällen i tre olika grupper av elever. Mellan de olika tillfällena gjordes justeringar av upplägget. Tillfälle 1 och 2 genomfördes av en lärare i två halvklasser i årskurs 3, och tillfälle 3 genomfördes av en annan lärare i helklass i årskurs 2. I årskurs 3 användes den halva av klassen som inte fick den utprovade lektionen som kontrollgrupp. Detta för att

jämföra resultaten mellan gruppen som programmerade problemlösningen med kontrollgruppen som löste samma problem, fast enbart med hjälp av penna och papper. Båda grupperna fick göra samma eftertest med enbart penna och papper.

Efter att ha genomfört lektionen första gången insåg lärarna att eleverna behövde kunna Scratch-Jr ordentligt för att inte fastna i programmeringen. Programmeringen gjorde uppgiften svårare för eleverna. De blev distraherade av programmeringen och fastnade i programmeringsfrågor. De verkade se uppgifterna som separata, och använde inte programmeringen som stöd. Eftertestet visade att gruppen som programmerade fick sämre resultat än kontrollgruppen.

Inför att genomföra lektionen en andra gång lade lärarna därför till en inledande lektion där de introducerade ScratchJr. Efter att därefter ha genomfört lektionen en andra gång insåg lärarna att eleverna behövde mer stöd i arbetsmetoden kring problemlös-

ning och att den valda uppgiften i högre grad behövde passa ScratchJr. Eleverna hade tyckte uppgifterna var för svåra, de fastnade på programmeringsdetaljer och frustrationsnivån blev hög för många. Eftertestet visade att gruppen som programmerade fick sämre resultat än kontrollgruppen, precis som efter första tillfället då lektionen genomfördes.

Inför att genomföra lektionen en tredje gång lade lärarna därför till ytterligare en lektion där de visade en modell för hur man kan tänka när man använder programmering för att lösa ett matematiskt problem. Lektionens genomförande justerades också inför den tredje utprovningen genom att den valda uppgiften stod på tavlan och var bättre anpassad för ScratchJr, samt att eleverna fick ett papper efter programmeringen för att visa svaren och hur de tänkt. Under lektionen läste läraren upp problemet och hjälpte vid behov elever med läsningen. Nu blev resultatet att 80 procent av klassen kom fram till rätt svar och huvudsakligen korrekt redovisning, och många elever upplevde att programmeringen hjälpt dem.

Som kontrollgrupp användes vid det tredje tillfället klasser ur årskurs 3 som fick lösa samma problem med penna och papper. Där var det cirka 50 procent som kom fram till rätt svar och huvudsakligen rätt redovisning. Kunskapsmässigt sett bedömdes elevgrupperna vara ungefär lika starka. Därför anser lärarna att i den lektion som slutligen utarbetades hjälpte programmeringen eleverna att tänka stegvis och använda programmeringen som verktyg, och att de därmed nått sitt syfte och mål med lektionen.

Att tillsammans genom lesson study-modellen utveckla en lektion gav lärarna möjlighet att finjustera och tänka igenom vad det var som var fel med deras ursprungliga tanke och hitta en väg som blev bra, i stället för att ge upp efter det första misslyckade försöket.

Här följer en översiktlig beskrivning av den utprovade lektionen

Förutsättningar för lektionen:

- Matematik i årskurs 2–3
- Matematiskt problem som passar för ScratchJr.
- Ipads en-till-en, penna och papper.
- Tid: 60 minuter
- Hel- eller halvklass
- Förkunskaper: Eleverna behöver kunna grunderna i ScratchJr samt ha fått en modell för problemlösning i ScratchJr. Eleverna behöver få två lektioner i förväg, dels om ScratchJr och dels om problemlösning i matematik.
- Läraren behöver kunna grunderna i ScratchJr.

Hur lärare och elever genomför lektionen:

- Den didaktiska metod som används är block-programmering.
- Lektionen inleds med en genomgång där läraren kort repeterar hur man löser problem i matematik och hur man använder ScratchJr, samt förklarar att de ska använda ScratchJr för att lösa ett matematiskt problem.
- Eleverna arbetar därefter individuellt med att programmera steg för steg på varsin Ipad. De använder sig av sin programmering för att hitta svaret och redovisa detta på papper, samt visa hur de har tänkt för att lösa problemet.
- Läraren går runt och stöttar eleverna vid behov, huvudsakligen i programmeringen men även med det matematiska.
- Eleverna lämnar in lösningen i pappersform där läraren ser om de kan beskriva hur de kom fram till svaren.

SCRATCH SOM ÄMNESÖVERGRIPANDE REDOVISNINGSVERTYG I ÅRSKURS 4–6

Lektion utvecklad av: Caroline Sandberg, Emil Mattsson, Helena Terje och Pia Lethin, Åstorps kommun

Syftet med den utvecklade lektionen var att visa hur ett ämnesövergripande arbetssätt i programmering skulle kunna se ut. Idén var att visa hur programmering skulle kunna användas som en redovisningsform i alla ämnen.

Lärarna ville undersöka om eleverna kunde lära sig att förstå innebörden av och använda villkoret *om* i programmering. Lärare ville ge eleverna möjlighet att lära och uppvisa detta genom att de programmerade en presentation av sig själva, med

villkoret *om*, där de presenterar sig själva både på svenska och engelska.

Målen för elevens lärande var att eleven skulle kunna:

- följa givna instruktioner
- använda villkoret *om*
- förstå och använda operatorerna *och*, *lika med* och *större än*
- lösa problem
- delge varandra sina kunskaper

Lektionens mål och syfte hade kopplingar till syfte, centralt innehåll och kunskapskrav i teknik, matematik, svenska och engelska. Till exempel, i tekniska undervisningen ”syfta till att eleverna utvecklar sitt tekniska kunnande och sin tekniska medvetenhet så att de kan orientera sig och agera i en teknikintensiv värld” och ”bidra till att eleverna utvecklar intresse för teknik och förmåga att ta sig an tekniska utmaningar på ett medvetet och innovativt sätt”. I matematikämnet ska undervisningen ”bidra till att eleverna utvecklar kunskaper för att kunna formulera och lösa problem” och ges ”möjlighet att utveckla kunskaper i att använda digitala verktyg och programmering för att kunna undersöka problemställningar”.

Lektionen har designats av fyra mellanstadielärare med varierande kunskaper om Scratch. Lektionen är en av en rad lektioner i ett ämnesövergripande projekt. Lektionens prövades ut vid två tillfällen, först i en klass i årskurs 6 och därefter i en klass i årskurs 5. Vid genomförandet använde lärarna observationer, direkt feedback och ”exit ticket” som utvärderingsmetoder. Vid de två tillfällena genomförde en lärare lektionen medan övriga observerade.

Efter att ha genomfört lektionen första gången i en klass i årskurs 6 insåg lärarna att det krävdes ett väldigt högt tempo för att alla planerade moment skulle hinnas med. I förberedelsen inför utprovningen av lektionen första gången fick eleverna skapa bakgrunder i Scratch fritt, vilket skapade bekymmer vid programmeringen.

Inför att genomföra lektionen en andra gång, nu i årskurs 5, justerades genomförandet: 1) de två sista momenten uteslöts så att lektionen skulle få mer andrum; 2) elever och lärare hade samma Scratch-bakgrunder; 3) lektionen avslutades med en ”exit ticket”.

Vid utvärderingen av det andra tillfället konstaterade lärarna att eleverna hade lyckats skapa en presentation med fungerande knappar där bakgrunden ändrades. Lektionen hade en tydlig struktur och tydliga mål. Detta gjorde att eleverna visste vad som förväntades av dem och de blev mycket inspirerade att ta sig an uppgifterna. Eleverna var också medvetna om att lektionen var en del av ett större ämnesövergripande projekt. Programmeringen som eleverna skulle genomföra var på en nivå som alla kunde klara. Lektionen hade en tydlig stegvis arbetsgång med visuella genomgångar och praktiska aktiviteter. Detta gjorde att eleverna inte tappade fokus, utan var delaktiga, stöttade och hjälpte varandra.

Eleverna hade lärt sig:

- villkoret *om*
- att använda operatorerna *och*, *lika med* och *större än*
- lösa problem
- skapa en presentation genom kod

Syfte och mål med lektionen hade därmed uppnåtts.

Utmaningen med att genomföra lesson study anser lärarna är att ha tid att utvärdera lektionen tillsammans efteråt, vilket är viktigt för kvaliteten i arbetet. Speciellt svårt blir det då man som i detta fall är fyra lärare från fyra olika skolor. Vid observationer i klassrummet är det en fördel om de observerande lärarna befinner sig i olika delar av klassrummet.

Här följer en översiktlig beskrivning av den utprovade lektionen

Förutsättningar för lektionen:

- Elever i årskurs 5 eller 6.
- Ämnesövergripande arbete i svenska, engelska, teknik och bild. Eleverna ska skapa en presentation i Scratch, där svenska och engelska är valbara språk.
- Eleverna har tillgång till en egen chromebook.
- Lektionstid: 60 minuter.
- Elevgrupp 22–24 elever.
- Förkunskaper hos eleverna: De har arbetat i verktyget Scratch tidigare, kan grunderna och har arbetat problembaserat i Scratch tidigare. Eleverna är vana vid att stötta varandra i sitt lärande. Eleverna har skapat de bakgrunder och sprajter som behövs till lektionen vid tidigare tillfälle.

Hur man genomför lektionen:

- Eleverna programmerar i Scratch med hjälp av blockprogrammering. De använder blocken: *loop*, *villkor* och *operator*.
- Genomgång av villkoret *om*, där dess funktion visas analogt.
- Kollektiv genomgång med hjälp av Smartboard där koden visualiseras steg för steg och varje block förklaras. Eleverna följer efter på sina enheter.
- Eleverna arbetar enskilt med att lösa det problem som ges.
- Gemensam genomgång på Smartboard där lösningen på problemet visas.
- Avslutar med en ”exit ticket” där eleven ska förklara villkoret *om*.
- Läraren rör sig i klassrummet och ger enskild återkoppling till eleverna under tiden som de arbetar.

Använda resurser:

- Film om villkor: <https://urskola.se/Produkter/204630-Programmera-mera-uppdrag-Villkor>
- Kod i Scratch: <https://scratch.mit.edu/projects/325104797/>
- Lathund i programmering <https://tinyurl.com/y5s4wmzo>

LÄRA SIG VARIABLER MED HJÄLP AV LÄROMEDLET "RÄKNA MED KOD" I ÅRSKURS 7

Lektion utvecklad av: Jonas Wennsten, Anna Wiklund, Gudrun Andersson, Robert Hegstedt och Rosmarie Sjöberg, Strandskolan Tyresö kommun

Lärarnas övergripande syfte och mål med lektionen var att eleverna skulle förstå och kunna använda sig av variabler inom programmering. Målen med lektionen var att eleverna skulle både utveckla generell förståelse för begreppet *variabel*, men även lära sig hur man kan använda sig av variabler inom programmering. Med kunskap om detta var syftet att de skulle kunna skriva program som gör beräkningar.

Lektionens innehåll kopplades till kursplanen i matematik som anger att undervisningen ska ge eleverna "möjligheter att utveckla kunskaper i att använda digitala verktyg och programmering för att kunna undersöka problemställningar och matematiska begrepp, göra beräkningar och för att presentera och tolka data." Lektionens innehåll adresserade också det centrala innehållet i kursplanen i matematik för årskurs 7–9:

- Hur algoritmer kan skapas och användas vid programmering
- Programmering i olika programmeringsmiljöer.

Lärarna arbetade med läromedlet *Räkna med kod – programmering i matematik* (Reuterswärd & Lovén, 2018). Den lektion som prövades ut var lektion 3 i boken, "Input – att mata in text och tal". Lektionen prövades ut vid två tillfällen i två olika klasser av samma lärare. Under lektionerna observerade två lärare lektionen, och eleverna fick göra en så kallad "exit ticket". Mellan lektionerna hade lärarna ett möte där de diskuterade hur lektionen gått och vad som kunde förbättras. Till det andra tillfället gjordes ett par ändringar av lektionens upplägg och innehåll.

Genom att göra en exit ticket kunde lärarna mer samlat få en bild av om lektionens syfte uppnåts. De såg det som många forskningsstudier visat, nämligen att begreppet variabel är svårt för många elever. De elever som hade en god förståelse och stort intresse för programmering sedan tidigare hade mycket enklare att lösa uppgifterna än de som inte hade det. Arbetet med att skriva exakt rätt var svårt för en del elever. Några gav upp fort. De flesta hade dock god uthållighet och ville göra rätt. De hjälpte varandra mycket.

Många elever såg aldrig riktigt "nyttan" med en variabel och använde den inte heller när de skulle skriva sina exit tickets. Eleverna kan behöva mycket mer tid och mer övning för att se att variabeln hjälper dem i koden. Då kan de kanske lättare förstå hur och när det är lämpligt att använda variabler. Bokens upplägg upplevde lärarna ändå som bra, trots

att ganska många elever inte ännu lärt sig variabler. De hann inte arbeta med så många uppgifter eftersom mycket tid gick åt till att få alla tecken rätt.

Lärarna drar slutsatsen att syftet och målet för lektionen delvis uppfylldes, men konstaterar att eleverna behöver mer träning och undervisning för att de ska bli förtrogna med variabler.

Lärarna fann fem olika elevsvar på exit ticket: *Skriv ett program som upprepar texten "Jonas är bäst!" 5 gånger efter varandra. Programmet ska använda sig av en variabel.*

Elevsvar 1 bedömdes ha lägst kvalitet, och elevsvar 5 högsta kvalitet.

- Elevsvar 1: `print (5*"Jonas är bäst")`
Eleven har löst uppgiften men har inte använt sig av en variabel.
- Elevsvar 2: `namn = input("Vem är bäst?")`
`print (5*namn)`
Eleven har löst uppgiften med input och en variabel, det är inte säkert att programmet löser uppgiften. Det bygger på att eleven skriver rätt sak i input.
- Elevsvar 3: `Upprepa = 5 print("Jonas är bäst"*5)`
Eleven har löst uppgiften korrekt med en variabel, men variabeln är talet som ska upprepas i stället för texten.
- Elevsvar 4: `namn = ("Jonas är bäst!")`
`print (5*namn)`
Eleven har löst uppgiften korrekt med en variabel
- Elevsvar 5 `namn = ("Jonas är bäst! ")`
`print (5*namn)`
Eleven har gjort en snyggare lösning eftersom det blir ett mellanrum mellan meningarna.

Planeringen av lektionen krävde mycket tid av flera orsaker. En orsak var bristande programmeringskunskaper i Python hos nästan alla inblandade lärare. Att matcha genomgångar och vad som är viktigt att ta upp mot lektionens syfte blev därför en utmaning. Lärarna hade dock god hjälp av varandra och mötena utvecklade lärarnas kunskaper i programmering.

Det gjordes inget förkunskapstest som kollade vad eleverna redan kunde. Efteråt uppmärksammade lä-

rarna behovet av ett sådant för kunna göra en bättre analys av vad eleverna lärt sig.

Här följer en översiktlig beskrivning av den utprovade lektionen

Förutsättningar för lektionen:

- Årskurs 7
- Ämne: Matematik
- Eleven har tillgång till egen Chromebook. Eleverna programmerar i Google Colaboratory, textbaserad programmering i språket Python.
- Läromedlet *Räkna med kod – programmering i matematik*.
- Tid: 1 lektion
- Antal elever: 25
- Förkunskaper hos eleverna: Stor vana vid blockprogrammering
- Svårighetsgrad: Medel
- Förkunskaper hos läraren: Inledande låg men behövde lära sig en hel del inför lektionen.

Hur man genomför lektionen:

- Läraren inleder med att förklara begreppet variabel utifrån hur man räknar ut arean på en kvadrat.
- Läraren visar och eleverna härmar en startupp-gift framme på projektorn. De får göra program som skriver ut namn, ålder om fem år och längd som omvandlas från meter till centimeter. Alla program innehåller variabler. Under genomgången betonar läraren små men viktiga saker som att man använder punkt i stället för komma vid decimaltal, att man måste skriva exakt likadant när man anropar en variabel som när man angav den (till exempel går inte variabeln "namn" att anropa som "Namn").

- Eleverna arbetar med några uppgifter i läroboken där de tränar på att skriva program med variabler. De sitter i grupper och tar hjälp av varandra och lärarna i rummet.
- Under tiden när eleverna arbetar med uppgifterna i boken hinner läraren uppmärksamma vad som fungerar bra och vad som är svårt.
- Eleverna gör en exit ticket med tre frågor:
 1. Vad kommer programmet att visa när det körs? `a = 3 print(1000 * a)`
 2. Skriv ett program som upprepar texten "Jonas är bäst!" 5 gånger efter varandra. Programmet ska använda sig av en variabel.
 3. Skriv ett program som frågar vem som är bäst och sedan upprepar vem som är bäst 5 gånger. Programmet ska innehålla en variabel.

LÄRARES UTVECKLINGSARBETE I ARTIKLAR

Anette Jahnke, projekt- och processledare Ifous

Inom FoU-programmet har deltagande lärare publicerat två utvecklingsartiklar i Skolportens tidskrifts-serie Leda & Lära. I skrivande stund är ytterligare tre utvecklingsartiklar om programmering inskickade för granskning och publicering. I detta kapitel återges artiklarnas sammanfattningar och hänvisar med länkar till fortsatt läsning av hela artiklarna på nätet. För de artiklar som inte ännu blivit publicerade så uppmanas läsaren att hålla utkik på FoU-programmets sida på Ifous webb: <https://www.ifous.se/programmering-i-amnesundervisningen/>

SKOLAN TAPPAR TJEJERNAS INTRESSE FÖR PROGRAMMERING

Leda & Lära nummer 4/2020: *Programmering i matematikundervisningen. En undersökning av elevers inställning till programmeringsinslag på matematiken, med fokus på genus.*

Vi har i denna undersökning tittat på hur elever i årskurs åtta uppfattar programmeringsinslag på matematiken. Vi har klart kunnat fastställa att det finns en markant skillnad i pojkars respektive flickors inställning till området. Pojkar uppger, i större utsträckning än flickor, att de tycker programmeringen är rolig, att de lär sig på lektionerna och framför allt att de kan tänka sig att jobba med programmering längre fram i livet.

Läs hela artikeln här:

<https://www.skolporten.se/app/uploads/2020/08/leda-och-lara-nummer-4-2020.pdf>

Författare är: *Ola Olsson* är lärare i matematik, NO-ämnen och teknik, i årskurs 7–9. Han arbetar på Korsavadsskolan i Simrishamn. *Fredrik Mårtensson* är lärare i matematik, biologi samt idrott och hälsa, i årskurs 7–9. Han arbetar på Korsavadsskolan i Simrishamn.

KUNSKAPSSPRIDNING I PROGRAMMERING MED SKILDA RESULTAT

Leda & Lära nummer 5/2020: *Implementering av programmering i ämnesundervisningen genom kollegialt lärande. En fallstudie från arbetet i en utvecklingsgrupp inom FoU-programmet Programmering i ämnesundervisningen*

Genom revisionen av styrdokumentet för programmering 2018 har det uppstått oro och kritik då många lärare anser sig sakna kompetens för att undervisa i programmering. För att höja kompetensen har Åstorps kommun utsett en särskild grupp av lärare från olika skolor i kommunen vilka har deltagit i ett forsknings- och utvecklingsprogram (FoU). Ett av målen har varit att sprida dessa kunskaper till respektive skola. I denna studie utforskar vi genom enkäter och skriftliga intervjuer, vilka effekter denna satsning har haft på deltagare i utvecklingsgruppen och övriga kollegor som inte varit med i programmet genom att jämföra dessa svar. Våra resultat visar att lärare i utvecklingsgruppen upplever sig ha fått större kompetens genom kollegialt lärande. Däremot ser vi att lärare utanför utvecklingsgruppen inte upplever sig fått nya kunskaper i samma utsträckning, vilket tyder på att spridningen av kompetensen inte fått det resultat som var tänkt.

Läs hela artikeln här:

<https://www.skolporten.se/app/uploads/2020/10/leda-och-lara-5-20202.pdf>

Författarna är MA/NO/TK-lärare i Åstorps kommunala skolor: *Julius Jonasson* är lärare i åk 7–9 på Björnekullaskolan. *Malin Midander* är speciallärare i grundsärskolan. *Caroline Sandberg* är lärare i åk 4–6 på Haganässkolan. *Helena Terje* är lärare i åk 4–6 på Tingdalsskolan.

VILKEN EFFEKT KAN PROGRAMMERINGS- UNDERVISNING HA PÅ ELEVERS LÄRANDE?

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 diskussionsunderlag för utvecklingen av programmeringsundervisning på i sin skola eller i sin kommun. Studien är gjord i en mellanstor kommun där lärare från årskurs ett till och med årskurs nio systematiskt har dokumenterat sina lektioner i programmering. De främsta effekterna i studien visar bland annat 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.

Hela artikeln kommer att kunna nås via:

<https://www.ifous.se/programmering-i-amnesundervisningen/>

Författare är: *Helén Viebke* är utvecklingsstrateg och arbetar på Barn- och utbildningsförvaltningen i Åstorp.

ATT FRÄMJA DATALOGISKT TÄNKANDE I FÖRSKOLEKLASS – BARN KAN MER ÄN VI TROR

I många länder världen runt har digitalisering och programmering införts i läroplaner för förskola och grundskola. Under senare tid har flera studier om undervisning i programmering i olika sammanhang genomförts. Dessa studier har oftast haft fokus på elever från sju år och äldre och det finns få studier som rapporterar om införandet av programmering för yngre barn i förskola och förskoleklass. Denna klyfta hanterar vi i denna studie som fokuserar på programmering och utveckling av elevers datalogiska tänkande i förskoleklass. År 2017 startades ett forsknings- och utvecklingsprogram av Ifous tillsammans med fem huvudmän, och en forskargrupp vid Stockholms universitet. Fokus var på utveckling av didaktiska modeller för programmering i utbildningen från förskola till och med grundskolan. I detta projekt använde lärare och forskare lesson study-metodik som ett vetenskapligt instrument för planering, genomförande och utvärdering av programmeringslektioner. Denna artikel presenterar

hur en förskolelärare arbetade med programmering för att främja datalogiskt tänkande hos elever i förskoleklass under en period av tre år och den erfarenhet som hon förvärvat. I artikeln drar vi slutsatsen att förskoleklasselever med systematisk och tankeväckande didaktisk modellering fullt ut kan utveckla ett antal grundläggande datalogiska färdigheter.

Hela artikeln kommer att kunna nås via:

<https://www.ifous.se/programmering-i-amnesundervisningen/>

Författare är: *Åsa Chibás*, lärare Freinetskolan Hugin, *Jalal Nouri*, docent, Stockholms universitet, *Eva Norén*, docent, Stockholms universitet, *Lechen Zhang*, doktorand, Stockholms universitet

PROGRAMLÖSNING MED SCRATCHJR

På Freinetskolan Hugin har vi i många år sett att brister i läsförståelse försvårar för elever att lösa textuppgifter i matematik. Vår teori har varit att detta delvis beror på elevernas tempo i läsningen. Vi har uppmuntrat dem att läsa långsammare för att ta sig an problemet stegvis.

Vi valde att introducera programmering i den matematiska undervisningen för att låta eleverna programmera textuppgifterna i matematik, i en förhoppning att det skulle sänka deras tempo och låra dem att ta sig an problemet mening för mening.

Resultaten var nedslående. De elever som programmerade klarade sig sämre än kontrollgruppen. Inte nog med att de fick sämre måluppfyllelse, de reagerade dessutom med stress. Vi fick modifiera uppgifterna i hög grad och anpassa dem till de faktiska förutsättningarna i programmet, ord för ord, ge eleverna mer programmeringskunskaper samt öva på liknande uppgifter för att ge eleverna tydliga exempel. På slutet såg vi att eleverna som programmerade fick bättre resultat än kontrollgruppen.

Trots att resultaten till slut blev positiva så ser vi inte att denna metod är användbar i lågstadiet. Uppgifterna och exempel behöver vara så specifika för att eleverna ska lyckas, och vi är tveksamma till om det kommer att hjälpa dem att ta sig an andra uppgifter i ett annat sammanhang. Däremot tror vi att metoden eventuellt kan vara användbar i äldre åldrar.

Hela artikeln kommer att kunna nås via:

<https://www.ifous.se/programmering-i-amnesundervisningen/>

Författare är: *Lena Abbing*, rektor Freinetskolan Hugin och *Teresa Sundström*, lärare Freinetskolan Hugin.

SLUTSATSER FRÅN UTVÄRDERINGEN

Lejla Mundzic, Emerga Institute

Emerga Institute har utvärderat Ifous forsknings- och utvecklingsprogram *Programmering i ämnesundervisningen* som har pågått i tre år mellan år 2017 och 2020. Uppdraget har varit en processutvärdering som innebar att Emerga fortlöpande har samlat in och analyserat data och återkopplat resultat till programledningen och programmets styrgrupp. Syftet med processutvärderingen har varit att:

- 1 stödja och styra programmet mot uppställda mål;
- 2 undersöka vilka effekter programmet har haft på lärare, skolan och huvudmannen som helhet.

Utvärderingens upplägg har anpassats till programmetts olika stadier. Löpande resultat har rapporterats varje år. I slutrapporten presenteras fördjupade analyser av utvärderingens resultat som består av det tredje årets datainsamling. Utvärderarna har använt sig av både kvantitativa och kvalitativa metoder för datainsamling och analys, till exempel webbenkäter, fokusgrupper samt workshops för presentation av resultat. Den primära målgruppen har varit lärare, skolledare, processledare och styrgruppsrepresentanter.

Den övergripande slutsatsen är att FoU-programmet *Programmering i ämnesundervisningen* har varit uppskattat och framgångsrikt. Programmet har haft en tydlig strävan att nå de övergripande uppsatta mål som har definierats initialt och resultat tyder på att målen i stor uträkning har nåtts.

Majoriteten av de som har medverkat i Emergas datainsamling anser att de tack vare medverkan i programmet har utvecklat arbetsmetoder för att tillsammans med andra kollegor och rektorn pröva, genomföra och följa upp inslag av programmering i undervisningen. Respondenterna anser att det kollegiala lärandet har ökat och utvecklats och upplever att rektorn har visat stöd och varit engagerad i utvecklingsgruppens arbete. Respondenterna anser att de har fått ökad kunskap om programmering och de för resonemang kring hur programmering kan integreras i ämnesundervisningen till exempel matematik och språk.

Det är svårt att dra generella slutsatser kring programmets inverkan på elevers lärande och kunskapsutveckling inom programmering, men flera

lärare och styrgruppsrepresentanter för intressanta resonemang och menar att de ser att någonting har hänt bland de elever vars lärare har medverkat i programmet. Det respondenterna fortfarande ser som en utmaning är att följa upp undervisningen i programmering. Nästan hälften av respondenterna i enkäten uppger att de endast i liten utsträckning har arbetsmetoder för att följa upp undervisningen.

Majoriteten av respondenterna upplever att de har utvecklats i sin profession tack vare medverkan i programmet och att det kollegiala lärandet har stärkts och utvecklats. Lärarna kan se en tydlig progression inom programmeringsundervisningen och många ger uttryck för att deras arbete har professionaliserats och utvecklats. Det finns en större medvetenhet kring planering, genomförande och även uppföljning även om den sistnämnda fortfarande uppfattas som svår. Lärarna anser att de i större utsträckning använder vetenskapliga och beprövade metoder i programmeringsundervisningen och kan anpassa dessa till olika situationer som uppstår. Många lärare menar också att de ser ett ökat intresse för programmering bland eleverna – i synnerhet bland tjejer.

Respondenterna anser att de under programmetts gång har skapat en struktur inom skolan för att fortsätta att driva forsknings- och utvecklingsarbete i takt med den tekniska och ämnesdidaktiska utvecklingen. Det som är viktigt att komma ihåg att huvudmännen har gått in i programmet med helt olika förutsättningar. Medan en aktör har gått in med en skola har en annan huvudman gått in med många deltagare från olika skolor. Respondenterna anser att de skolor som har klarat av att hålla i och hålla ut har skapat en struktur för att fortsätta att utveckla undervisning i programmering. Hos samtliga fem huvudmän finns det en plan för hur lärdomar från programmet kan spridas vidare till fler skolor och fler lärare. Huruvida det faktiskt kommer att ske en spridning som leder till implementering av ett nytt arbetssätt inom programmeringsundervisning återstår att se. Nästan alla respondenter på ledande befattning anser att programmet har stärkt dem i deras ledarskap. Det har skapats förutsättningar för att fortsätta att driva utvecklingsarbete och använda lärdomar från detta program i andra typer av insatser som kräver en tydlig struktur och ett tydligt ledarskap.

LITTERATURLISTA

- Ahmed, G., Nouri, J., Norén, E. & Zhang, L. (2019). Students perceptions of programming in primary school. WiPSCE 2019, ACM, Glasgow.
- Ahmed, G., Nouri, J., Zhang, L. & Norén, E. (2020). Didactic methods of integrating programming in mathematics in primary school: findings from a Swedish national project. In Proceedings of SIGCSE' 20: 51th ACM Technical Symposium on Computer Science Education March 11–14, 2020 Portland, OR, USA. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/1234567890>
- Ala-Mutka, K. (2011). *Mapping digital competence: Towards a conceptual understanding*. Sevilla: Institute for Prospective Technological Studies.
- Ananiadou, K. & Claro, M. (2009). *21st century skills and competences for new millennium learners in OECD Countries*. OECD Education Working Papers, No. 41. Paris: OECD Publishing.
- Benton, L., Hoyles, C., Kalas, I., & Noss, R. (2017). Bridging Primary Programming and Mathematics: some findings of design research in England. *Digital Experiences in Mathematics Education*, 3(2), 115–138
- Boylan, Mark, Sean Demack, Claire Wolstenholme, John Reidy, and Sarah Reaney. (2018). *ScratchMaths: evaluation report and executive summary*. Hämtad den 30/9 2020 från: https://educationendowmentfoundation.org.uk/public/files/Projects/Evaluation_Reports/ScratchMaths.pdf
- Braun, V. & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2), 77–101.
- Bradley A. E., Genevieve G-E, (2014). Learning to learn from teaching: a first-hand account of lesson study in Japan. *International Journal for Lesson and Learning Studies*, 3(2), s. 170–191.
- 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).
- Chibás, Å., Nouri, J., Norén, E., Zhang, L., & Sjöberg, C. (2018). Didactical strategies and challenges when teaching programming in pre-school. EDULEARN18 Proceedings, s. 3345–3350. Läs abstract och kontakta författarna för fulltext.
- Erstad, O. (2008). The digital literacy in schools: The Norwegian experience. *Italian Journal of Technology*, 16(1), 4.
- Griffin, P., & Care, E. (2015). *Assessment and teaching of 21st century skills – Methods and approach*. Dordrecht, Heidelberg, New York, London: Springer.
- Heintz, F., Mannila, L., Nygård, K., Parnes, P., & Regnell, B. (2015). Computing at School in Sweden – Experiences from Introducing Computer Science within Existing Subjects. 8th International Conference on Informatics in Schools: Situation, Evolution, and Perspectives I *Informatics in Schools. Curricula, Competences, and Competitions /Lecture Notes in Computer Science and General Issues*, 9378, 118–130.
- Ifous. (2020). *Handbok – att medverka i Ifous FoU-program*. Stockholm: Ifous.
- Jonasson, J., Midander, M., Sandberg, C., & Terje, H. (2020). Implementering av programmering i ämnesundervisningen genom kollegialt lärande. En fallstudie från arbetet i en utvecklingsgrupp inom FoU-programmet: Programmering i ämnesundervisningen. *Leda och lärande*, 5. Tillgänglig <https://www.skolporten.se/app/uploads/2020/10/leda-och-lara-5-20202.pdf>
- Kjällander, Susanne; Åkerfeldt, Anna & Petersen, Petra (2016). *Översikt avseende forskning och erfarenheter kring programmering i förskola och grundskola*. Hämtad från http://omvarld.blogg.skolverket.se/wpcontent/uploads/sites/2/2016/06/oversikt_programmering_i_skolan.pdf

- Lindgren, L., Jahnke, A. & Norman, S. (2020). *Effekter av Ifous FoU-program*. Stockholm: Ifous.
- Mozelius, P. & Nouri, J. (2018). Factors to consider when using learning games for learning programming in K-9 education. 12th European Conference on Games Based Learning.
- Mårtensson, F., & Olsson, O. (2020). Programmering i matematikundervisningen. En undersökning av elevers inställning till programmeringsinslag på matematiken, med fokus på genus. *Leda & Lära*, 4. Tillgänglig <https://www.skolporten.se/app/uploads/2020/08/leda-och-lara-nummer-4-2020.pdf>
- Norberg, E., Mårdh, M., Enbom, S. & Karlsson, C. (2019, september). *Från analog till digital programmering – en lesson study i tre delar*. Presenterad på Leda Lärande, Utbildningsförvaltningen, Stockholm, 26 september 2019.
- Nouri, J. & Mozelius, P. (2018). A framework for evaluating and orchestrating game-based learning for development of computational thinking. In International Conference on Education and New Learning Technologies.
- Nouri, J., Norén, E. & Skog, K. (2018). Didactical strategies employed by teachers when teaching programming in K-9 education. The 12th annual International Technology, Education and Development Conference.
- Nouri, J., Norén, E., & Skog, K. (2018). Didactical strategies employed by teachers when teaching programming in K-9 education. INTED2018: Proceedings, s. 7983–7989. Läs abstract och kontakta författarna för fulltext.
- Nouri, J., Norén, E., & Skog, K. (2018). Learning programming by playing and coding games in K-9. The 12th annual International Technology, Education and Development Conference, Valencia, Spain. 5–7 april. Läs abstract och kontakta författarna för fulltext.
- Nouri, J., Zhang, L., Mannila, L., & Norén, E. (2020). Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. *Education Inquiry*, 11(1), 1–17.
- Nouri & Zhang (2021). A progression model for programming education in compulsory school (skickad för granskning till journal).
- Olsson, O & Mårtensson, F. (2020). Programmering i matematikundervisningen. En undersökning av elevers inställning till programmeringsinslag på matematiken, med fokus på genus. Konferensen Leda lärande, Stockholm. 4/2020.
- Reuterswärd, E. & Lovén, K. (2018). *Räkna med kod: programmering i matematik*. (Första upplagan). Stockholm: Sanoma utbildning.
- Sjöberg, C., Nouri, J., Sjöberg, R., Norén, E., & Zhang L. (2018). Teaching and learning mathematics in primary school through scratch. EDULEARN18 Proceedings, s. 5625–5632. Läs abstract och kontakta författarna för fulltext.
- Sjöberg, T. Risberg, J. Nouri, E. Norén, L. Zhang. (2019). A lesson study on programming as an instrument to learn mathematics and social science in primary school. 13th annual International Technology, Education and Development Conference.
- Skolinspektionen. (2019). *Beslut efter kvalitetsgranskning av Undervisning med digitala verktyg – matematik och teknik i grundskolans årskurs 7–9 i Tyresö kommun. Dnr 400-2018:6938*. Stockholm: Skolinspektionen.
- Zhang, L., & Nouri, J. (2019a). A systematic review of learning computational thinking through Scratch in K-9. *Computers & Education*, 141, 103607.
- Zhang, L. & Nouri, J. (2019b). Assessing K-9 Teachers' Computational Thinking Skills through a Computational Thinking Test. Handbook of Research on Integrating Computer Science and Computational Thinking in K-12 Education. IGI Global. Doi: 10.4018/978-1-7998-1479-5
- Åkerfeldt, A., Kjällander, S. & Selander, S. (2018). *Programmering – introduktion till digital kompetens i grundskolan*. Stockholm: Liber.
- Zhang L, Nouri J, and Rolandsson L. (2020). Progression of Computational Thinking skills in Swedish compulsory schools with block-based programming. In Proceedings of Twenty-second Australasian Computing Education Conference (ACE'2020). ACM, Melbourne, Australia, 10 pages. <https://doi.org/10.1145/1234567890>
- Zhang, L., Nouri, J. & Norén, E. (2018). Reviewing teaching approaches for programming through Scratch in compulsory education. International Conference on Technology in Education.

Zhang, L. & Nouri, J. (2019). Assessing K-9 Teachers' Computational Thinking Skills through a Computational Thinking Test. Handbook of Research on Integrating Computer Science and Computational Thinking in K-12 Education. IGI Global. Doi: 10.4018/978-1-7998-1479-5

Ödalen, J. (2020). Omfattningen av programmering i undervisningen i Stockholms stads grundskolor. En jämförelse mellan 2017 och 2019. *Uppföljningsenhetens rapportserie*, 2020:01.

BILAGA 1: PLANERINGSMALL INFÖR PRÖVNING AV LEKTION

Elevgrupp och ämnen:

Generellt och övergripande mål med lektionen:

Förutsättningar och förkunskapskrav:

Lektionsaktivitet	Förväntade elevreaktioner och planerade lärarresponser och stöd	Mål och syfte för momentet	Metoder för utvärdering	Utvärdering
<p>Beskriv detaljerat delmomenten som lektionen består av och vad lärare och elever ska göra under dessa.</p> <p>Försök att dela upp lektionen i så många delmoment ni kan med tillhörande specifika delmål, i stället för att sammanslå till ett mindre antal moment som siktar in sig på mer allmänna mål.</p> <p>Detaljplanering och detaljriktighet är viktigt.</p> <p>Ange uppskattat tidsomfång för varje del.</p>	<p>Hur kan eleverna tänkas reagera under de olika delarna/momenten? Förväntade beteenden?</p> <p>Vilka svårigheter kan komma upp för olika elever?</p> <p>Hur ska läraren respondera med stöd?</p>	<p>Vilka är målen och syftena för varje enskilt delmoment? Var så specifika och konkreta som möjligt. Måluppfyllelse bör kunna utvärderas/bedömas.</p>	<p>Vilka metoder och tekniker använder ni för att utvärdera huruvida målen i delmomenten är uppfyllda?</p> <p>Vad för slags evidens och tecken för lärande ska ni samla in och basera utvärderingen på?</p> <p>Hur ska ni samla in evidensen/data? Videoinspelningar? Ljudinspelningar? Observationsanteckningar? Intervjuer? Diagnoser? Kombinationer?</p>	<p>Denna fylls i efter utförd lektion. Utvärdera varje delmoment på ett evidensbaserat sätt. Vad fungerade väl, vad fungerade mindre väl och baserat på vilka indikationer/evidens?</p>
Delmoment 1.				
Delmoment 2.				

BILAGA 2: DEBRIEFINGMALL EFTER PRÖVNING AV LEKTION

RIKTLINJER FÖR DEBRIEFING EFTER UTFÖRD LEKTION

Primärt mål: att på ett evidensbaserat sätt utvärdera och förbättra delmomenten och helheten.

Tidsåtgång: cirka 2–2,5 timmar. Ta god tid på er.

Exempel på relevanta frågor:

- Vad fungerade väl och vad fungerade mindre väl med delmomenten och med helheten?
- Hur effektiva var lärarens strategier och arbetssätt?
- Vilken effekt skapades av de valda arbetssätten och strategierna i termer av elevers motivation, beteende och lärande?
- Vilka svårigheter upplevdes av lärare och elever?
- Vad var för enkelt för eleverna?
- Vad var lyckat och gick bra för eleverna?
- Var målen relevanta? Hur kan de justeras?
- Hur väl fungerade lärresurserna? (digitala och analoga resurser)
- Hur väl hanterades individuella skillnader mellan elever?
- Hur var kvalitén i elevernas arbeten?
- Vad behöver vi lägga mer eller mindre tid på?
- Hur kunde ni ha samlat in data och utvärderat lektionen bättre?
- Vad bör tas bort, läggas till eller förbättras?
- Hur förbättrar vi lektionen? Fokus på varje delmoment och helheten.
- Argumentera och diskutera detaljerat med fokus på det specifika likväl som på det generella med utgångspunkt i evidens (insamlat data, observationer, diagnoser, konversationer, etc.)

REVIDERA ER PLAN FÖR ER LESSON STUDY.

BILAGA 3: PRESENTATIONSMALL

Följande sidor fanns med i den PowerPoint-mall som använts.

1. Titel: Lära sig xxx med hjälp av yyy i årskurs z, Maja Majasson, Amira Nehex. Gladaskuttskolan 201x.

2. Innehåll i presentationen

- Syfte & mål med lektionen
- Metod – framtagning av lektionen genom lesson study
- Resultat – beskrivning av lektionen och lektionens effekter
- Diskussion om metod och resultat
- Referenser
- Tack och kontaktuppgifter

3. Syfte

Beskriv varför har ni tagit fram denna lektion

4. Målen för elevens lärande

Beskriva vilka kunskaper lektionen avser att utveckla hos eleverna.

Koppla till kursplanens/ernas syfte, förmågor och/eller centralt innehåll.

5. Metod

Här följer en beskrivning av hur lektionen är framtagen med hjälp av lesson study

Hur är lektionen framtagen och utprovad?

Här synliggör ni den process som ligger bakom det ni senare påstår är ert resultat – här ser jag som läsare om jag kan lita på det ni påstår – är detta en väl utprovad lektion? Verkar det troligt att om jag genomför den så kommer även mina elever lära sig

Beskriv hur ni arbetat med att genomföra lesson study, t.ex.

Hur prövades lektionen ut? Flera gånger?

Av olika lärare? Samma lärare? Olika elever? Klasser?

På vilka sätt utvärderade ni lektionen?

Auskultationer? Observationer? Exit-ticket?

Tester? Enkäter? De-briefing möten?

På vilka sätt utvärderade ni att de mål ni hade uppnåddes? På vilka sätt ändrade ni upplägget succesivt?

6. Resultat

Här följer en beskrivning av lektionen som är utprovad och vad lärare kan förvänta sig att eleverna lär sig

Förutsättningar för lektionen

Årskurs?

Inom ramen för vilket ämne eller ämnen?

Teknik?

Annat material?

Tid?

Antal elever?

Förkunskaper hos eleverna?

Svårighetsgrad/nivå – är det t.ex. en introduktionsaktivitet? Förkunskaper hos läraren?

7. Hur man genomför lektionen

Hur undervisar läraren i klassrummet?

Vad gör läraren? Vad gör eleverna?

I beskrivningen ange vilken didaktisk strategi som används. Med didaktisk strategi menas: analog, robot, block, textbaserad, eller kombinationer av dessa. Om ni funnit en ny didaktisk strategi beskriv detta.

I beskrivningen ange också vilka didaktiska arbetssätt som används. Har ni funnit ett nytt arbetssätt – beskriv detta. Med arbetssätt menas:

- Elever arbetar i grupper
- I början arbetar de med befintlig kod. Reproducera, modifiera, utveckla.
- Läraren har korta kollektiva genomgångar kring blocken och programmeringsverktygen (villkorssatser, slingor, etc.). Ger övningar på dessa specifikt. Går också igenom kodkonstruktion och hur man kan tänka för att få till olika effekter.
- Elever hjälper varandra. Lärare uppmuntrar en sådan kultur.
- Skapa spel och interaktiva berättelser/presentationer

8. Hur man följer upp elevernas lärande

Beskriv på vilka sätt och när läraren följer upp vad elevernas lär sig. Vad gör läraren? Vad gör eleverna? När?

9. Effekter på elevernas kunskaper

Vilka effekter får lektionen på elevernas lärande? Vilka kunskaper utvecklar eleverna? Får lektionen några oönskade effekter? Vilka? I vilken utsträckning nås syfte och mål med lektionen

10. Diskussion kring metoden

– lesson study

Lyft upp några utmanande eller kritiska delar när ni genomförde metoden, lesson study. Något som skedde som kan ha påverkat resultatet? Något i arbete med lesson study som ni ska göra bättre nästa gång?

11. Diskussion kring resultatet


Varför blev det som det blev? Varför lärde sig eleverna det som de lärde sig? Eller varför lärde sig det inte det ni tänkt? Har ni några förklaringar? Finns det någon litteratur eller forskning som kan förklara? När ni nu ser tillbaka till lektionsupplägget – hur skulle ni kunna utveckla detta vidare? Hur kan det eleverna lärde sig användas av en lärare när hen planerar framöver? Var det något som förvånade er?

12. Referenser och/eller bilagor

Om ni hänvisar till något tidigare kan ange det här

13. Tack och kontakt

Har ni frågor eller kommentarer kontakta oss på maja.majasson@huvudman.se



Ifous – Innovation, forskning och utveckling i skola och förskola

Ifous är ett fristående forskningsinstitut som bedriver forsknings- och utvecklingsarbete (FoU) inom skolområdet i samarbete med skolhuvudmän och lärosäten. Verksamheten utgår från medlemmarnas behov och syftar till att bidra till skolutveckling på vetenskaplig grund. I dag har Ifous drygt 140 medlemmar, både kommunala och fristående skolhuvudmän.

Läs mer om vårt arbete på www.ifous.se