



Universiteit  
Leiden  
The Netherlands

# Opleiding Informatica en Economie

Analyse van de kennis en opvattingen van docenten in  
het Nederlands basis- en middelbaar onderwijs (K12)  
over de vakintegratie van computational thinking

Lucas de Bruin

Supervisors:

F. Hermans & S. Yeni

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)

[www.liacs.leidenuniv.nl](http://www.liacs.leidenuniv.nl)

10/07/20

## Samenvatting

De toenemende technologisering, informatisering, globalisering en individualisering heeft geleid tot discussies over hoe het toekomstige onderwijs er uit moet gaan zien. Centraal stond de vraag welke vaardigheden voor leerlingen van belang zijn om te kunnen functioneren in deze snel veranderende maatschappij. Als antwoord hierop, identificeerde het SLO een aantal 21e-eeuwse vaardigheden waaronder computational thinking, de vaardigheid die binnen deze scriptie centraal staat. Veel scholen in Nederland zijn inmiddels begonnen met het integreren van computational thinking in hun curriculum. Echter, de uitdaging is nu het uitzoeken hoe dit op een effectieve manier gedaan kan worden. De moeilijkheid hiervan is de integratie van computational thinking in de bestaande schoolvakken buiten informatica. Momenteel voeren het Radboud Universiteit en Universiteit Leiden een onderzoeksproject hiervoor uit met als doel het ontwikkelen van richtlijnen voor docenten om computational thinking te onderwijzen. Om vast te stellen wat de huidige situatie van de integratie is, zijn er interviews gedaan bij twaalf docenten in het primair en voortgezet onderwijs. Deze scriptie beschrijft de resultaten van de interviews. De interviews zijn volgens het Pedagogical Content Knowledge (PCK) model geanalyseerd met het programma 'ATLAS.ti'. De resultaten geven aan de hand van het PCK model een beeld van welke ideeën docenten hebben ten aanzien van het onderwijzen van computational thinking en welke vakdidactische aspecten zij op het moment bij het lesgeven toepassen. Daarnaast geeft de scriptie een beeld van wat docenten al doen in de lessen, wat ze willen doen en waar docenten momenteel tegenaan lopen om dit te doen. Uit de analyse blijkt dat er veel uiteenlopende visies zijn op computational thinking en de wijze waarop het geïntegreerd kan worden en dat docenten wel bereid zijn tot de integratie, maar nog tegen een aantal obstakels lopen.

# Voorwoord

-

# Inhoudsopgave

<b>1</b>	<b>Introductie</b>	<b>1</b>
1.1	Aanleiding . . . . .	1
1.2	Probleemstelling . . . . .	1
1.3	Doelstellingen & relevantie . . . . .	2
1.4	Thesis overzicht . . . . .	3
<b>2</b>	<b>Theoretisch kader</b>	<b>3</b>
<b>3</b>	<b>Methode</b>	<b>3</b>
3.1	Onderzoeksopzet . . . . .	3
3.2	Context . . . . .	4
3.3	Data-analyse . . . . .	6
<b>4</b>	<b>Resultaten</b>	<b>10</b>
4.1	PCK . . . . .	10
4.1.1	CK . . . . .	10
4.1.2	M1 - Begrip van leerdoelen bij het onderwijzen van computational thinking .	11
4.1.3	M2 - Begrip van de (benodigde) kennis van en omtrent leerlingen . . . . .	14
4.1.4	M3 - Begrip van onderwijsaanpak bij het onderwijzen van computational thinking . . . . .	18
4.1.5	M4 - Begrip van toetsing . . . . .	23
4.2	Interconnecties . . . . .	23
4.3	De praktijk . . . . .	24
4.4	De opvattingen . . . . .	24
4.5	De verwachtingen . . . . .	25
<b>5</b>	<b>Conclusie</b>	<b>26</b>
5.1	Het beeld van de docenten . . . . .	26
5.2	De huidige staat van de integratie en diens opvattingen en verwachtingen . . . . .	26
5.3	De verwachtingen . . . . .	27

<b>6</b>	<b>Discussie</b>	<b>27</b>
6.1	Reflectie op resultaten . . . . .	27
6.2	Vervolgonderzoek . . . . .	27
	<b>References</b>	<b>27</b>
<b>A</b>	<b>Interview Protocol</b>	<b>27</b>
<b>B</b>	<b>Content Representation (CoRe)</b>	<b>27</b>
<b>C</b>	<b>Huidige algemene leerlijn</b>	<b>27</b>

# 1 Introductie

## 1.1 Aanleiding

De toenemende technologisering, informatisering, globalisering en individualisering heeft geleid tot discussies over hoe het toekomstige onderwijs er uit moet gaan zien. Centraal staat de vraag welke vaardigheden voor leerlingen van belang zijn om te kunnen functioneren in deze snel veranderende maatschappij. Denk hierbij aan kennis over omgaan met privacy, social media en nepnieuws. Om deze vaardigheden te identificeren zijn er internationaal onderzoeken uitgevoerd, zoals in 2006 door de Europese Unie en in 2008 door het OECD. Om deze vaardigheden te identificeren heeft het SLO, het nationaal expertisecentrum leerplanontwikkeling, een literatuurstudie uitgevoerd om voor het Nederlandse onderwijs deze benodigde kennis en vaardigheden vast te stellen. Deze literatuurstudie resulteerde in de belangrijkste vaardigheden voor de 21e eeuw volgens het SLO.

Aanleiding voor deze literatuurstudie was de vraag van het ministerie van Onderwijs, Cultuur en Wetenschap om het concept 'digitale geletterdheid' te verduidelijken en een onderwijskundige invulling te identificeren. Deze vraag ontstond door een rapport van het KNAW in 2012 dat constateerde dat de digitale kennis en vaardigheden in het onderwijs nog niet voldoende onderwezen werden, met name op het gebied van 'digitale geletterdheid'. Als antwoord hierop, identificeerde het SLO digitale geletterdheid als een paraplueterm voor de 21e-eeuwse vaardigheden informatievaardigheden, ICT-basisvaardigheden, mediawijsheid en computational thinking, waarvan laatstgenoemde centraal zal staan binnen dit onderzoek.

Computational thinking is dus een essentieel onderdeel van 'digitale geletterdheid'. Het krijgt wetenschappelijk veel onderbouwing als belangrijke vaardigheid (Wing, 2006; Barr, Harrison & Conery, 2011) en de vaardigheid sluit nauw aan bij het leren van programmeren op de middelbare school. Veel scholen in Nederland en het buitenland zijn daarom begonnen met het integreren van computational thinking in hun curriculum.

Echter, de uitdaging is nu het uitzoeken hoe dit op een effectieve manier gedaan kan worden.

## 1.2 Probleemstelling

Nu nationaal het maatschappelijk belang van computational thinking duidelijk is, richt onderwijsonderzoek zich op de vragen hoe computational thinking onderwezen kan worden en hoe computational thinking in het huidige onderwijscurriculum geïntegreerd kan worden. Het is van belang om alle schoolvakken bij de integratie van computational thinking te betrekken, zodat alle leerlingen deze vaardigheid kunnen leren. De moeilijkheid hiervan is de integratie van computational thinking met de verschillende schoolvakken. De wetenschap heeft hier nog geen duidelijk antwoord op. Er wordt veel onderzoek gedaan naar bepaalde strategieën en modellen om hieraan bij te dragen. Echter, huidig onderzoek richt zich met name op het vak informatica, terwijl het van belang is ook de andere schoolvakken bij deze integratie te betrekken, zodat alle leerlingen in het Nederlands onderwijssysteem computational thinking vaardigheden leren. Er is dus nog weinig kennis over het maken van deze zogenoemde 'transfer' van computational thinking naar andere schoolvakken, zoals economie. Het gaat hierbij ook om het voorbereiden van de docenten zelf. Docenten moeten kennis

hebben van de middelen en manieren om de leerlingen computational thinking bij te brengen. Ze moeten worden voorbereid en worden geholpen om de leerdoelen eigen te maken, passend bij hun stijl en vak om computational thinking te kunnen onderwijzen.

### 1.3 Doelstellingen & relevantie

Naar aanleiding van bovenstaand probleem is in februari 2019 een langlopend onderzoeksproject gestart naar het integreren van computational thinking in bestaande schoolvakken. Dit onderzoeksproject loopt tot juli 2022 en betreft een samenwerking tussen het Radboud Universiteit en Universiteit Leiden, gefinancierd door het Nationaal Regieorgaan Onderwijsonderzoek (NRO). Het onderzoeksproject richt zich op het ontwikkelen van bepaalde richtlijnen en methodes om docenten te ondersteunen bij het onderwijzen en integreren van computational thinking.

Het onderzoek bevindt zich nog in de oriëntatiefase. Deze oriëntatiefase focust zich op het identificeren van de huidige staat van de integratie van computational thinking in het Nederlands onderwijs. Voor het identificeren zijn interviews afgenomen om de Pedagogical Content Knowledge (PCK) van docenten te bepalen. De constructie van PCK kan worden gebruikt om de opvattingen en kennis van docenten in kaart te brengen over het onderwijzen van allerlei onderwerpen, zoals modelleren in informatica om dit in kaart te brengen is tijdens de interviews gevraagd wat voor hun de betekenis is van computational thinking, wat hun intentie is bij het onderwijzen van computational thinking (de leerdoelen) en wat hun ideeën zijn van: de (benodigde) kennis van en omtrent leerlingen, de manieren om computational thinking les te geven (hun onderwijsaanpak) en het begrip te toetsen. Deze scriptie beschrijft de resultaten van deze interviews.

De focus van deze scriptie ligt daarbij op het in kaart brengen van de kennis en opvattingen van docenten over het onderwijzen van computational thinking. Deze inventarisatie is heel breed en identificeert waar docenten op het moment aan denken en wat ze benoemen als het gaat om het integreren van computational thinking in hun vak. Het doel van deze scriptie is het schetsen van het beeld dat docenten op het moment hebben bij het onderwijzen van computational thinking. Het beeld wordt gecreëerd door het beschrijven van de huidige vakdidactiek van docenten (het begrip), de huidige staat van de integratie (de praktijk), de redenen voor deze huidige staat en wat zij verwachten in de toekomst van de integratie (de opvattingen). Het doel is niet het reflecteren op de correctheid van de opvattingen en de kennis van docenten. Door het beeld van docenten eerst individueel te benaderen, kan daarna een aantal overeenkomsten in de individuele PCK van de docenten worden gevonden. Deze typificatie kan samen met de redeneringen en verwachtingen van de docenten gebruikt worden om docenten beter te begeleiden bij het integreren van computational thinking. Gerelateerd aan het onderzoeksproject kan het geschetste beeld een bijdrage leveren aan het ontwikkelen van workshops voor de docenten. Tevens biedt het beeld een nulmeting om de effectiviteit van deze workshops te bepalen. De ontwikkelingen binnen de PCK van de docenten kan namelijk gemonitord worden door na de workshops opnieuw een identificatie uit te voeren.

Om de doelstellingen van deze studie te behalen zal in conclusie de volgende hoofdvraag beantwoord worden:

“Wat valt op aan een inventarisatie van de huidige situatie van de integratie van computational thinking in het Nederlands onderwijs?”

De deelvragen die hiervoor beantwoord zullen worden, zijn:

- Wat is het huidige beeld van docenten bij het onderwijzen van computational thinking?
- Hoe integreren docenten op het momenten computational thinking?
- Wat zijn de onderliggende opvattingen en verwachtingen van docenten bij het integreren van computational thinking?

## 1.4 Thesis overzicht

Het theoretisch kader bestaat uit drie delen. Het eerste deel beschrijft de verschillende interpretaties van de definitie en vaardigheden van computational thinking. Het tweede deel betreft een uiteenzetting van een framework om computational thinking in het curriculum van verschillende schoolvakken te integreren. Hierbij wordt elk onderdeel van het framework globaal toegelicht, behalve de onderdelen die in de interviews specifiek zijn genoemd. Deze genoemde onderdelen worden ter ondersteuning van de resultaten van het onderzoek uitgebreider beschreven. Vervolgens wordt in het derde deel de huidige staat van onderzoek naar computational thinking integratie in het algemeen en in combinatie met de benoemde onderdelen beschreven. Met het laatste deel voorziet het kader hopelijk genoeg informatie om samen met de interviews een aantal ontwerpprincipes te concluderen.

TODO

## 2 Theoretisch kader

TODO

## 3 Methode

In dit hoofdstuk wordt uiteengezet welke methode is gebruikt om de hoofdvraag en de deelvragen te beantwoorden. In paragraaf 3.1 wordt eerst uitgelegd hoe de interviews zijn opgezet en uitgevoerd. Vervolgens wordt in paragraaf 3.2 de gegevens van de participanten en de relevante context van de desbetreffende scholen uiteengezet. Tot slot wordt in paragraaf 3.3 toegelicht hoe de interviews zijn geanalyseerd en hoe de data is gekoppeld aan de modellen om uiteindelijk in hoofdstuk 4 als resultaten te kunnen presenteren.

### 3.1 Onderzoeksopzet

Deze scriptie maakt gebruik van transcripties van twaalf semi-gestructureerde interviews. De interviews zijn uitgevoerd door twee andere onderzoekers, waarbij het doel van de interviews was



om de PCK van de docenten te construeren. Appendix A bevat het protocol van deze interviews. De interviews duurde gemiddeld een uur. Bij aanvang van elk interview werd gevraagd naar de persoonlijke en de studieachtergrond van de docent. Daarna zijn de interviews aan de hand van de rest van de vragen van het protocol afgenomen. De vragen van het protocol zijn gebaseerd op het Content Representation questionnaire (CoRe). De onderdelen van de CoRe questionnaire sluiten aan bij de vier PCK-elementen (M1 tot M4) zoals beschreven in het theoretisch kader, zie tabel 2. De CoRe questionnaire is uitgebreid met vragen over de technologische integratie van de docenten, die aansluiten bij de onderwijsaanpak van docenten (M3). Het doel van de vraag is om de identificatie van de PCK van de docenten completer te maken. De eerste PCK-gerelateerde vraag richtte zich op het gebruik van de computer of technologie in de les van de docent, daarna werd de koppeling met computational thinking gemaakt. Alle docenten hebben een toestemmingsformulier ingevuld, waarin zij toestemming geven voor de audio-opname en aangeven dat de deelname aan het onderzoek vrijwillig is.

## 3.2 Context

Tabel 1 toont de gegevens van de participanten. Zes docenten hebben minder dan tien jaar ervaring, vijf docenten tussen de tien en twintig jaar en één docent meer dan 30 jaar ervaring. De docenten zijn werkzaam op zeven verschillende scholen.

Tabel 2 toont de verdeling van de docenten op de verschillende scholen. Het vak informatica wordt op de middelbare scholen 1 en 6 niet gegeven. School 5 geeft alleen in de bovenbouw informatica. Scholen 1, 5 en 6 zijn technasium scholen. Op een technasium staat het vak onderzoek & ontwerpen (O&O) centraal. O&O is een erkend eindexamenvak voor havo en vwo. Bij het vak werken leerlingen in teamverband aan ‘actuele bètatechnische opdrachten uit de praktijk’. Andere kenmerken van het technasium is het geven van projectmatig onderwijs vanuit een activerende didactiek, de rol van de docent is coachend en leert met leerlingen. De projecten zijn in samenwerking met het bedrijfsleven. Daarnaast staat talentontwikkeling centraal en moeten de scholen in bezit zijn van een technasiumwerkplaats. Daarom dat alle drie de scholen een E-lab hebben. Een E-lab is een aparte ruimte met verschillende soorten technologische apparatuur, zoals 3D printers of Arduino’s zodat de leerlingen bijvoorbeeld kunnen leren programmeren. Van het lokaal maken leerlingen gebruik om met de verschillende technologieën te leren werken en opdrachten voor bepaalde vakken een creatieve invulling te geven. Eenzelfde soort ruimte met technologische apparatuur heeft school 9 ook.

School 4 verzorgt sinds twee jaar Scratch-lessen, in periodes van drie weken achter elkaar voor de groepen 5 tot en met 8, onderwezen door een externe expert. De externe expert heeft een opleiding tot applicatie-ontwikkelaar gevolgd en heeft zelf Scratch-lessen ontwikkeld. Bij het interview met docent 4 is deze externe expert ook aan het woord geweest. Voor het onderzoek zijn de antwoorden van de externe expert ook meegenomen in de analyse. Er is geen onderscheid gemaakt in de visie van de externe expert of de docent zelf. Aangezien het onderzoek een beeld probeert te vormen van wat de docenten op het moment hebben bij het onderwijzen van computational thinking, zullen de ideeën van de externe expert ook het beeld van docent 4 vormen. In het interview met docent 4 en de externe expert zijn er ook geen momenten waarin zij het niet met elkaar eens zijn. De reden hiervoor is dat de externe expert is geholpen door docent 4 bij het maken van de lessen. Hierdoor

Naam docent	M/V	PO/VO	Leerjaar	Vakgebied	Jaren ervaring	Relevante achtergronden
Docent 1	M	VO	klas 1, 2, 4 en 5	Aardrijkskunde	32	Bestuurslid van de vakvereniging informatica en digitale geletterdheid en oud informatica docent
Docent 2	M	PO	groep 6 en 8	-	13	Deelprojectleider 'digitale geletterdheid' van O2G2. Geeft workshops op gebied van mediawijsheid en computational thinking
Docent 3	M	VO	klas 3 en bovenbouw	Natuurkunde	2	Twee jaar werkzaam geweest als software engineer
Docent 4	M	PO	groep 7	-	8	ICT-coördinator en deelprojectleider 'digitale geletterdheid' van O2G2.
Docent 5	M	VO	klas 1 en 2	Moderne techniek	12	
Docent 6	V	VO	klas 3, 4 en 5	O&O en kunst (tekenen)	7	
Docent 7	V	VO	onderbouw	Nederlands	4	
Docent 8	M	VO	klas 2 en 3	Wiskunde	12	Twee jaar informatica gestudeerd
Docent 9	V	PO	groep 7 en 8	-	7	Deelprojectleider 'digitale geletterdheid' van O2G2.
Docent 10	M	VO	alle klassen	Biologie	17	
Docent 11	M	VO	klas 1, 2 en 3	Science15	13	
Docent 12	M	VO	klas 2, 4, 5 en 6	Engels	2	

Tabel 1: Gegevens docenten

Scholen	VO/PO	Docenten
School 1	VO	1
School 2	PO	2
School 3	VO	3
School 4	PO	4
School 5	VO	5, 7, 8 en 11
School 6	VO	6, 10 en 12
School 7	VO	9

Tabel 2: Spreiding docenten op scholen

is het aannemelijk dat de PCK van docent 4 niet sterk is beïnvloed door de kennis en opvattingen van de externe expert.

School 7 houdt voor groep zes tot acht twee keer per jaar projecten voor een aantal weken (tussen 6-8) waarbij er vier keuzemogelijkheden zijn, waaronder toekomstkunde. Bij toekomstkunde worden onderwerpen van digitale geletterdheid onderwezen. Daarnaast is school 7 een TOM-school. TOM staat voor ‘Teamonderwijs Op Maat’. Bij deze vorm van onderwijs heeft elke docent een eigen specialisatie en wisselen de leerlingen van docent in plaats van dat elke docent een eigen klas lesgeeft. TOM-scholen kenmerken zich door een specifieke focus op zelfstandigheid bij de leerlingen, die actief, ontdekkend en samenwerkend leren door betekenisvolle opdrachten te maken. De computer wordt ingezet als leerbron en communicatiemiddel. De rol van de leraar is begeleidend en coachend. De specialisatie van docent 9 op de school is op het moment digitale geletterdheid.

School 2 maakt gebruik van weektaken waarin ook taken staan die bijdragen aan het leren van computational thinking. Voorbeelden van taken zijn het programmeren van de functionaliteiten van een Arduino of lessen in Code.org maken.

### 3.3 Data-analyse

De interviews zijn door middel van een kwalitatieve inductieve analyse geanalyseerd, met Magnusson’s (1999) PCK-elementen als hoofdcategorieën (Barendsen & Henze, 2017). De analyse van de interviews zijn in drie fases uitgevoerd. Deze drie fases bestonden uit open coderen, axiaal coderen en selectief coderen. De fases zijn gedefinieerd door Strauss en Corbin (1990) voor het uitvoeren van een ‘grounded theory’ analyse. Bij de grounded theory ligt de focus op het inductief genereren van nieuwe theoretische ideeën of hypothesen uit de gegevens, in tegenstelling tot het testen van vooraf bepaalde theorieën. Aangezien deze nieuwe theorieën uit de data ‘ontstaan’ en dus door de data worden ondersteund, zijn ze, zogezegd, gegrond (grounded) (Gibbs, 2007). De inductieve aanpak van de grounded theory past bij het exploratieve karakter van dit onderzoek.

De interviews zijn door middel van een kwalitatieve inductieve analyse geanalyseerd, met Magnusson’s (1999) PCK-elementen als hoofdcategorieën (Barendsen & Henze, 2017). De analyse van de interviews zijn in drie fases uitgevoerd. Deze drie fases bestonden uit open coderen, axiaal coderen en selectief coderen. De fases zijn gedefinieerd door Strauss en Corbin (1990) voor het uitvoeren

van een ‘grounded theory’ analyse. Bij de grounded theory ligt de focus op het inductief genereren van nieuwe theoretische ideeën of hypothesen uit de gegevens, in tegenstelling tot het testen van vooraf bepaalde theorieën. Aangezien deze nieuwe theorieën uit de data ‘ontstaan’ en dus door de data worden ondersteund, zijn ze, zagezegd, gegrond (grounded) (Gibbs, 2007). De inductieve aanpak van de grounded theory past bij het exploratieve karakter van dit onderzoek.

- M1: docent praat over doelen
- M2: docent praat over leerlingen of over klassen met betrekking tot de uitvoering van de doelen
- M3: docent praat over waarmee of hoe die doelen behaald kunnen worden
- M4: docent praat over de toetsing van het behalen van de doelen

Bij het lezen van het eerste interview bleek al gauw dat de hoofdcategorieën niet voldoende de context van de docenten kon classificeren. Daarom is er voor gekozen om hier een aparte hoofdcategorie voor te maken, genaamd *integratie*. Doordat bij vrijwel elke interview de betekenis van computational thinking en de bijbehorende aspecten uitvoerig werden besproken, werd hiervoor ook een aparte hoofdcategorie gemaakt.

Na het volledig lezen van het eerste interview werden de gefilterde tekstfragmenten per hoofdcategorie opnieuw doorlopen. Bij elk tekstfragment werd vervolgens een label geformuleerd die zo dicht mogelijk bij de antwoorden van de docent lag (in vivo). Door de in vivo benadering, resulteerde het open coderen per hoofdcategorie in een brede identificatie van ideeën en opvattingen.

Deze losse labels werden in een tweede fase, axiaal coderen, onder de loep genomen en daar waar overeenkomsten waren samengevoegd tot een nieuw label (Gibbs, 2007). Als er labels waren met verschillende visies op hetzelfde onderwerp werd hiervoor een subcategorie binnen de hoofdcategorieën aangemaakt. De pre-analyse zorgde voor een aantal algemene subcategorieën en twee toevoegingen aan de hoofdcategorieën van het codeerschema, namelijk:

- CK: docent praat over definitie of aspecten van computational thinking
- Integratie: docent praat over verwachtingen, eisen en redeneringen over de huidige staat en toekomst van het integreren van computational thinking, in hun vak / in het onderwijs

Daarna zijn de rest van de interviews open gecodeerd. Per interview werd telkens de tekst in fragmenten verdeeld en gekoppeld aan de hoofdcategorieën. Per hoofdcategorie werden de labels vervolgens in vivo geformuleerd. Soms was gelijk duidelijk dat een deel van het tekstfragment aansloot bij al een gecodeerd item in de subcategorie en werden deze meteen aan elkaar gekoppeld. Na het open coderen is elke hoofdcategorie weer onder de loep genomen en werden de labels geordend, gecategoriseerd en samengevoegd in ATLAS.ti. Bij het maken van de subcategorieën werd de loskoppeling van de rest als het belangrijkste gezien. De reden hiervoor is dat juist de cohesie van de verschillende subcategorieën van de PCK-elementen aantoont hoe sterk de PCK van de docenten is. Het uiteindelijke resultaat hiervan is een codeerschema waarin M1 is opgesplitst in 42 verschillende doelen met twee grote subcategorieën voor programmeren en het werken met technologie. M2 bestaat uit de subcategorieën voor de labels die de volgende onderdelen beschrijven:

- Leerling karakteristieken, zoals ‘creatief’ met een subonderdeel voor de aspecten waar leerlingen moeite mee hebben zoals ‘moeite met abstractie’
- Leerlingen vs. werken met de computer
- Niveauverschillen door... en niveauverschillen niet door... met bijbehorende redenen
- Wanneer beginnen? En wanneer niet beginnen? Als de docent een expliciet een klas noemde.
- Wat hebben leerlingen nodig? En wat hebben leerlingen niet nodig?

Deze wanneer wel / niet beginnen subcategorie is met name interessant om bij het construeren van de PCK te kunnen zien in hoeverre de docent begrip heeft voor het inzetten van een ander PCK-element, zoals een bepaald doel of bepaalde onderwijsaanpak in een bepaalde klas. Voor deze categorie is overigens wel gekozen om slechts één klas te labelen als de docent meerdere klassen voor de integratie benoemde. Er is gekozen voor de laagste klas, want met het oog op het maken van een doorlopende leerlijn is het relevanter om te weten wanneer de integratie op z'n vroegst kan beginnen.

M3 bestaat uit drie subcategorieën voor de labels die de rol van de docent, een technologie of een algemene leeractiviteit beschrijven. M4 bestaat uit subcategorieën voor toetsingen om het begrip of de progressie van een leerling te meten en de attitudes ten aanzien van deze toetsingen. CK bevat voor alle genoemde definities de in vivo labels en de labels of een docent een computational thinking aspect wel of niet herkend. Integratie is heel breed opgezet en bevat de volgende subcategorieën:

- Eisen aan de integratie / visie van de integratie
- Kenniswerving / waar komen lessen vandaan?
- Huidige staat van integratie op de scholen, bestaande uit onder andere de mentale status van de docent over de integratie en labels over de integratie in praktijk, zoals huidige initiatieven
- Redenen / oorzaken waarom stappen tot integratie nog niet zijn ondernomen / moeizaam gaat, vanuit de docent zelf
- Redenen / oorzaken waarom stappen tot integratie zijn ondernomen, vanuit de docent zelf
- Vakintegratie: bevat alle tekstfragmenten waarin een vak werd benoemd
- Toekomst: bevat alle tekstfragmenten die over de toekomst gaan en dus op het moment nog niet gebeuren

Nadat het codeerschema volledig was verwerkt in ATLAS.ti, startte de fase van het selectief coderen. In deze fase werd naar verbanden en relaties tussen de labels in de subcategorieën gezocht om theorieën te formuleren (Gibbs, 2007). Deze theorieën kunnen uiteindelijk worden gebruikt om de deelvragen te beantwoorden.

Voor het beschrijven van de CK is elke label met betrekking tot de definitie geanalyseerd aan de hand van de vier elementen van de definitie, zie hoofdstuk 2. Per definitie werd bekeken in hoeverre

een element terugkwam in het antwoord en zo is uiteindelijk het overzicht gecreëerd. Voor het beschrijven van het PCK-element M1 (begrip van leerdoelen bij het onderwijzen van computational thinking) is ervoor gekozen om de labels binnen deze categorie onder te verdelen in de taxonomie van Bloom, zoals toegelicht in het theoretisch kader. Hierbij moet vermeld worden dat de vaardigheden programmeren en werken met technologie ingedeeld zijn als cognitief waarbij het ‘toepassen’ van de vaardigheid samenvalt met het ‘oefenen’ van de vaardigheid in de psychomotorische taxonomie van Dave. Voor M3 is ervoor gekozen om de verschillende leeractiviteiten onder te verdelen in het SAMR model. Doordat het SAMR model het mogelijk maakt om de leeractiviteiten te koppelen aan de taxonomie van Bloom, is het makkelijker om later de interconnectie tussen M1-M3 te beschrijven. Er is voor alle overige labels die geen direct verband hadden met een praktische leeractiviteit gebruik gemaakt van een eerdere indeling van Voogt et al. (2013) Het beschrijven van de didactische kennis gaat aan de hand van de beschreven onderwijseigenschappen van Voogt (2013). De indeling beschrijft verschillende onderwijseigenschappen bestaand uit onderdelen voor de karakteristieken van het curriculum, de klasorganisatie, de rol van de docent en de sturing bij de onderwijsactiviteit. In deze indeling kon dus ook de rol van de docent verwerkt worden.

Voor de technologische kennis worden het totaal aantal gebruikte technologieën per docent beschreven. De technologieën zijn ingedeeld volgens de categorieën:

- Websites, apps en sociale media (Shazam, YouTube, etc.)
- Simulatie software (Minecraft, Coach, etc.)
- Grafische programmeeromgevingen (Scratch, Micro:bit, etc.)
- Sensoren en camera’s
- Ontwerp- en productietechnologieën (Microsoft Office, green screen, 3D printer, etc.)
- Robots (Arduino, Bee Bot, etc.)
- Programmeertalen (Python, R, etc.)

Deze indeling is gebaseerd op een eerder gemaakte indeling van Mannila (2014). Deze indeling is vervolgens uitgebreid met de toevoeging van de apps bij de websites en sociale media, de ontwerptechnologieën bij de productietechnologieën en de categorie sensoren en camera’s. Typische apparaten als de computer, laptops, tablets en de digibord zijn buiten beschouwing gelaten. De vraag is namelijk wat de docenten weten wat ze kunnen met die typische technologieën. Er is nog een aparte indeling gemaakt voor de materialen waarbij niet direct sprake is van een technologie of software. Deze materialen kunnen ook voor unplugged leeractiviteiten worden gebruikt en zijn op basis van die mogelijkheid ingedeeld.

Tot slot is ter completering van M3 een vergelijking gemaakt tussen de kennis van de soorten activiteiten en de kennis van de soorten technologie. Bij deze vergelijking is alleen gekeken naar een beschreven toepassing van een technologie of programmeertaal. Benoemingen als ‘programmeren’ of ‘computer’ zijn buiten beschouwing gelaten. Voorbeelden van toepassingen zijn Python bij programmeren of PowerPoint op de computer.

Door de vele verschillende opvattingen en redeneringen van docenten leidde het axiaal coderen tot veel samenvoegingen. Daarom zijn alle interviews ter afronding van deze fase nog een keer volledig gelezen en werd gecontroleerd of de labels nog consistent waren met de tekstfragmenten. Daarbij zijn ook de vakintegratie en toekomst labels extra gecontroleerd en daar waar nog miste aan de tekstfragmenten toegevoegd. De toekomst en vakintegratie labels waren namelijk nodig om in de resultaten de huidige activiteiten van de docenten met betrekking tot de integratie te beschrijven. ATLAS.ti biedt de mogelijkheid om alle tekstfragmenten die de label voor de toekomst bevatten eruit te filteren. Door deze filter konden de labels vervolgens nader worden geanalyseerd, ter beantwoording van de vraag hoe docenten op het moment computational thinking in hun vakken integreren. Voor het beschrijven van de PCK van de docenten zijn deze toekomst tekstfragmenten wel gebruikt, want het kunnen benoemen van een activiteit in de toekomst betekent dat de docent de kennis bezit maar het door de context wellicht niet kan uitvoeren.

De rest van de resultaten konden allemaal vrij direct aan de hand van de categorieën van het axiaal coderen beschreven worden.

## 4 Resultaten

### 4.1 PCK

#### 4.1.1 CK

	Docenten												Totaal
	1	2	3	4	5	6	7	8	9	10	11	12	
Het inzetten van denkprocessen	●	●	●	●	●	●	○	○	○	●	○	○	7
Voor het oplossen van problemen	●	●	○	●	○	●	○	●	○	●	○	○	6
En het zodanig formuleren hiervan	●	○	○	○	○	●	○	●	○	●	○	○	4
Dat mensen en/of computers de oplossingen kunnen	●	●	○	●	○	●	●	○	○	●	○	○	6
Totaal	4	3	1	3	1	4	1	2	0	4	0	0	23

Figuur 1: In hoeverre docenten de onderdelen van computational thinking kunnen benoemen.

Figuur 1 weergeeft welke onderdelen de docenten benoemen als zij een definitie gaven van computational thinking. Voor het inzetten van denkprocessen benoemen docenten 2, 3, 4 en 6 alleen algoritmisch denken. Docenten 1, 5, en 10 trekken de denkprocessen breder en zien het respectievelijk als ‘technical thinking’, een vorm van logisch denken en het denken als een computer. De docenten die daarbij het probleemoplossend aspect benoemen, betrekken daar in twee gevallen alleen de technologie bij of in één geval alleen het formuleren van de oplossingen. Dat laatste geval, docent 8, is interessant. De definitie die hij geeft luidt: ‘alles moet geautomatiseerd worden, of dat nou met robots is of met mensen, je wil een soort van standaard routine hebben om iets op te lossen, dat je daar onderscheid in kan maken wanneer iets wel vaste routine moet zijn of niet.’ De docent benoemt expliciet het oplossen van een probleem met het maken van een standaard routine, het formuleren.

Hij noemt niet de uitvoering van de routines en niet het maken van de routines door middel van denkprocessen. Het herkennen van problemen ziet hij wel, maar dat valt buiten de definitie en betreft geen denkproces om problemen om te lossen. Docenten 1, 6 en 10 zijn in staat een volledige definitie te geven. Docent 7 ziet het als ‘alles waarbij je ICT voor en mee kan gebruiken.’

Na het zien van de definitie en de betekenissen van de aspecten van computational thinking, bespraken de docenten of zij de aspecten herkenden bij hun vak. De resultaten hiervan zijn te zien in figuur 2.

	Docenten												Totaal
	1	2	3	4	5	6	7	8	9	10	11	12	
Abstractie	●		●		●	●	●	●	●	●		○	8
Algoritmisch denken	●		●	●	●	●	●	●	●	●		○	9
Decompositie	●		○		●	●	○	●	●	●	●	●	9
Evaluatie			●		●	●	●	●	●	●		●	8
Generalisatie	●		○		●	●	●	●	●	●		○	7
Totaal	4	0	5	1	5	5	6	5	6	5	1	5	48

Figuur 2: Het herkennen van CT aspecten in het schoolvak van de docent.

De lege plekken betekenen dat er niks over is gezegd tijdens de interviews. De witte bolletjes staan voor een expliciete afwijzing van herkenning van de docent. Bijvoorbeeld, voor evaluatie benoemt geen enkele docent expliciet dat hij dit aspect niet herkend binnen het vak. Bij de grijze bolletjes herkennen de docenten dus wel de aspecten.
























In het algemeen vindt docent 7 het direct koppelen van de computational thinking aspecten aan haar vak moeilijk en vraagt net zoals docent 9 om meer context voor het begrijpen van de aspecten. Docent 7 ziet ook dat ze veel aan evaluatie doen, omdat de leerlingen na elke opdracht een reflectie moeten schrijven. Docent 9 heeft moeite met het maken van een onderscheid tussen de aspecten, docent 3 had tot voor het onderzoek nog niet eerder van het begrip computational thinking gehoord en docent 11 zegt het begrip niks en vindt de aspecten voor haar te abstract.

De docenten lichten niet duidelijk toe waarom zij de aspecten niet herkennen binnen hun vak, vaak komt het doordat zij er geen voorbeeld bij kunnen verzinnen. Alleen voor decompositie legt docent 3 uit dat hij dit aspect bij natuurkunde niet ziet, want *‘we werken toch een beetje hap-stap van onderwerp naar onderwerp, van toets naar toets.’* Docent 9 herkent decompositie niet direct bij een bestaand vak op school, maar weet het wel toe te passen voor het oplossen van een ruzie: *‘eigenlijk is het ook decompositie, opdelen in stukjes, hoe gaan we dit nou oplossen, wat is de eerste stap, waar willen we naartoe?’*

#### 4.1.2 M1 - Begrip van leerdoelen bij het onderwijzen van computational thinking

In figuur 3 is te zien wat de geïnterviewde docenten willen bereiken bij het onderwijzen van computational thinking.



	Docenten												Totaal
	1	2	3	4	5	6	7	8	9	10	11	12	
Affectieve doelen - organiseren	○	○	●	○	●	●	○	○	○	○	○	○	 3
Affectieve doelen - waarderen	○	○	○	○	●	●	○	○	●	●	○	○	 4
Algoritmisch denken - toepassen	○	○	○	●	○	○	○	○	●	○	○	○	 2
Grafieken en modellen - begrijpen	●	○	○	○	○	○	○	○	○	●	○	○	 2
ICT basisvaardigheden - onthouden	○	○	○	●	○	○	●	●	○	○	○	●	 4
Informatievaardigheden - analyseren	●	○	○	○	○	○	●	○	○	●	○	○	 3
Informatievaardigheden - onthouden	○	○	○	○	○	○	●	●	○	○	○	○	 2
Informatievaardigheden - toepassen	●	○	○	○	○	○	○	○	○	●	○	○	 2
Leven in digitale samenleving - analyseren	●	●	○	○	○	●	●	●	●	○	●	●	 8
Leven in digitale samenleving - begrijpen	●	●	●	○	○	●	●	○	○	●	○	○	 6
Leven in digitale samenleving - toepassen	●	●	●	○	○	●	●	●	●	○	●	●	 9
Manier van werken - evalueren	○	○	○	○	○	○	○	○	●	○	○	○	 1
Motorische doelen - automatiseren (ontwikkelen)	●	○	○	○	○	●	○	●	○	○	●	○	 4
Motorische doelen - combineren	○	○	○	●	●	○	○	○	○	○	●	○	 3
Motorische doelen - imiteren	○	○	●	○	○	○	○	○	○	○	○	○	 1
Motorische doelen - oefenen	○	●	○	○	●	○	○	○	●	○	○	○	 3
Programmeren - begrijpen	●	●	○	●	○	●	○	●	●	○	○	○	 6
Programmeren - creëren	●	●	●	●	●	●	○	○	○	●	●	●	 9
Programmeren - toepassen	○	○	○	●	○	○	○	●	●	●	●	○	 5
Werken met technologie - analyseren	●	●	○	○	○	●	●	○	○	●	●	○	 6
Werken met technologie - creëren	●	●	○	●	●	●	○	○	○	●	●	○	 7
Werken met technologie - evalueren	●	●	●	●	●	●	○	●	○	●	●	○	 9
Werken met technologie - toepassen	●	●	○	●	●	●	●	●	●	○	●	○	 9
Totaal	13	10	6	9	8	12	8	9	9	10	10	4	108

Figuur 3: De benoemde leerdoelen

### *Cognitieve doelen*

Docenten benoemen cognitieve doelen met betrekking tot programmeren, werken met technologie, grafieken en modellen, ICT basisvaardigheden, informatievaardigheden, leven in digitale samenleving, manier van werken en algoritmisch denken. Binnen deze gebieden benoemen docenten doelen op verschillende denkniveaus. Het overzicht hiervan is te zien in tabel 3.

Het begrijpen van programmeren gaat over het kennen van de basisonderdelen van programmeren zoals algoritmes, herhaling en variabelen (docenten 2, 4, 8 en 9), maar ook het snappen van ‘het letterlijk nemen’ van de computer (docenten 1, 6 en 8). *“Die systematiek leren van iets precies doen, dat vind ik wel belangrijk dat ze dat snappen hoe dat werkt het”*, aldus docent 6. Het toepassen van programmeren gaat over het goed inzetten van de concepten van programmeren, docent 8 zegt hierover: *“ik wil dus if then, en if-then-else, vind het verschil maar uit, aan het eind van de les*

	Onthouden	Begrijpen	Toepassen	Analyseren	Evalueren	Creëren
Programmeren		6	5			9
Algoritmisch denken			2			
Leven in digitale samenleving		6	9	8		
Informatievaardigheden	2			3		
ICT basisvaardigheden	4					
Manier van werken					1	
Grafieken en modellen		2				
Werken met technologie			9	6	9	7

Tabel 3: Overzicht van leerdoelen op verschillende denkniveaus

*moet jij me vertellen wanneer gebruik je if-then, en wanneer gebruik je if-then-else*” en als het toepassen samengaat met het doel om iets te maken dan is er binnen de taxonomie van Bloom sprake van creëren. Het creëren gaat het meest gepaard met het programmeren van eenvoudige programmaatjes of opdrachtjes (docenten 2, 4, 6, 10 en 11), van een robot (docenten 2, 4, 5, 6 en 11) of van een database of website (docent 12). Docent 11 vat dit allemaal samen met: *“Idealiter dat iedereen in ieder geval bezig kan met programmeren. De een zal het op een andere manier nodig hebben dan de ander, liever in beeldverwerking of filmverwerking gaan, ander wil een robot leren programmeren, anderen die wil gewoon een goed stuk [code] schrijven, duidelijk”*. Docenten 1, 3, 11 en 12 hebben hierbij nog het doel om vakgerelateerde onderdelen te programmeren. Docenten 1 en 3 betrekken hierbij het maken van grafieken en modellen voor aardrijkskunde en natuurkunde. Docent 11 betreft het maken van een thermosmeter of snelheidsmeter voor science en docent 12 voor een *“programmaatje”* om literatuur relevanter te maken.

Voor de manier van werken formuleert docent 9 het evalueren als: *“Ze hebben het heel vaak over het product zeg maar, en niet zozeer over welke stappen heb je eigenlijk gelopen. Daar zijn we wel veel mee bezig.”*

### ***Affectieve doelen***

De affectieve doelen zijn onderverdeeld in organiseren en waarderen. Voor het organiseren willen drie docenten met computational thinking zelfvertrouwen geven aan de leerlingen. Docent 6 pleit voor het ontwikkelen van openheid en nieuwsgierigheid: *“Openheid vind ik wel belangrijk, dat ze open zijn in klas een en het niet eng vinden, vind ik heel belangrijk. [...] En dat wij die nieuwsgierigheid ook een beetje proberen aan te wakkeren van ze, dit is een mogelijkheid, en dat het leuk is om ermee te gaan, mee te gaan klooiën eigenlijk.”* Het aanwakkeren is ook een affectief doel, maar valt onder waarderen. Voor het waarderen benoemen docenten doelen als leerlingen enthousiast maken (docenten 5, 6 en 9) en eigen voorkeuren ontwikkelen voor mening en identiteitsvorming (docenten 6, 9 en 10).

### ***Motorische doelen***

Met computational thinking willen docenten vaardigheden als logisch nadenken (docenten 1 en 8), probleemoplossend vermogen (docent 8) en zelfstandig werken (docent 6 en 11) bij de leerlingen automatiseren (ontwikkelen).

Docent 3 wil met computational thinking het werken als programmeur imiteren, want *“ze bedenken*

*niet alles zelf wat ze programmeren, maar veel leen je van anderen, en dat ga je dan aanpassen naar je eigen situatie, waarbij je nog heel veel fouten kunt maken, en dat vond ik leuk om ze [de leerlingen] een keer te laten ervaren.”*

Bij het integreren van computational thinking benoemen docenten ook doelen om te oefenen met leren doorzetten (docent 9), omgaan met teleurstellingen (docent 5) en samenwerken (docenten 2, 5 en 9).

Het combineren binnen de motorische doelen gaat over het gebruiken van computational thinking om vakgerelateerde opdrachten te maken. Docent 5 wil bijvoorbeeld *“dat ze die machines weten te gebruiken zodat ze het kunnen toepassen met het maken van een spelbord voor M&M [Mens en Maatschappij], of voor Science dat ze bezig kunnen met robotica”*. Op de basisschool impliceert docent 4 dit doel bij het beschrijven dat *“kinderen werken daar [laptops] ook op, ze maken daar bepaalde lessen op, bijvoorbeeld van spelling, rekenen, en ze gebruiken die ook voor begrijpend lezen.”* Bij het onderwijzen van computational thinking is het dus hier de bedoeling dat leerlingen dankzij de denkprocessen achter computational thinking (onderdeel één van de definitie) de opdrachten van de lessen oplossen (onderdeel twee van de definitie) en deze oplossingen invoeren (onderdeel drie van de definitie) op de computer (onderdeel vier van de definitie).

#### **4.1.3 M2 - Begrip van de (benodigde) kennis van en omtrent leerlingen**

Voor het onderwijzen van computational thinking verwachten de docenten op verschillende gebieden rekening te moeten houden met een aantal karakteristieken van de leerlingen. Voor de les zelf verwachten docenten 2, 5 en 7 dat leerlingen creatief zijn. Docent 8 verwacht dat de leerlingen vragen gaan stellen als *“Hoezo dan?”* en *“Is dat wel zo?”*, terwijl docent 9 verwacht dat het kunnen beantwoorden van deze vragen niet uitmaakt: *“Maar ze weten ook dat ik ook niet alles weet. En dat vinden ze helemaal niet erg, dat vinden ze juist wel leuk”*. Docenten 2, 3, 5, 6, 8, 9 en 10 verwachten dat leerlingen enige mate van zelfredzaamheid bezitten, waarbij de leerlingen zelf met oplossingen kunnen komen en, indien de opdracht niet lukt, dat de leerlingen zelfstandig naar de docent toekomen voor hulp. Docent 4 en 7 verwachten juist dat leerlingen op de basisschool en in de brugklas nog niet zelfstandig genoeg kunnen werken.

Voor het ontwerpen van de lessen verwacht docent 9 rekening te moeten houden met dat leerlingen zintuiglijk intuïtief ingesteld zijn, dus *“alles moet aanraakbaar zijn, verschuifbaar zijn, en ze moeten zelf het gevoel hebben dat ze controle hebben over die systemen door fysiek handelen.”* Docenten verwachten dat leerlingen veel verschillende voorkeuren hebben. Er worden voorkeuren benoemt voor een bepaalde technologie (docent 6 en 10) en voor de manier van werken zoals liever alleen werken of samenwerken (docent 9) of het liever maken van theoretische of praktische opdrachten (docenten 8, 9 en 11).

Op het gebied van ervaring verwacht docent 12 dat leerlingen voldoende wiskundige kennis bezitten. Docenten 1 en 4 verwachten dat leerlingen kunnen werken met de computer, terwijl docenten 3 en 10 een gebrek aan ervaring met het werken op de computer verwachten.

Andere docenten verwachten deze gebreken niet bij alle leerlingen, maar wel bij sommige. Daarom dat docenten verwachten dat er bij het onderwijzen van computational thinking niveauverschillen aanwezig zullen zijn of gedurende de lessen zullen ontstaan. Docent 6 benoemt hierbij dat hij dit

ook ziet gebeuren ondanks dat leerlingen gelijke begeleidingsmogelijkheden hebben gehad. Figuur 4 toont de genoemde oorzaken voor de niveauverschillen.

	Docenten												Totaal
	1	2	3	4	5	6	7	8	9	10	11	12	
Concentratie	○	○	○	○	○	○	○	●	○	○	○	○	1
Didactische houding van docent	○	○	○	○	●	○	○	○	○	○	○	○	1
Extra motivatie bij vak	●	○	●	○	●	●	○	○	○	○	○	○	4
Geslacht	○	○	○	○	●	○	○	●	○	●	●	○	4
Bezit van bepaald inzicht	●	●	○	○	○	○	●	●	●	○	●	○	6
Leeftijd	○	○	○	●	○	○	○	○	●	○	○	○	2
Meer ervaring met technologie	●	●	○	●	●	●	●	○	●	●	●	○	9
Meer affiniteit met technologie	●	●	○	○	●	●	○	●	●	●	●	○	8
Totaal	4	3	1	2	5	3	2	4	4	3	4	0	35

Figuur 4: Oorzaken voor de niveauverschillen.

De verschillen in ervaring met technologie zien docenten 1, 6, 7 en 9 ontstaan door persoonlijke interesse van de leerlingen. Docenten 10 en 4 leggen de oorzaak bij hoeveel aandacht de (docent van de) basisschool heeft besteed aan het werken met technologie. Hieraan voegt docent 4 toe dat ook de thuissituatie van de leerlingen van belang is om rekening mee te houden: *“Anderen mogen [thuis] twee uur achter de computer zitten of tablet, dan hebben wij te maken met verschillende soorten kinderen.”* Docenten 2 en 9 twijfelen over of ervaring alle niveauverschillen kan verklaren of dat het bezitten van een bepaald inzicht ervoor zorgt dat leerlingen met even veel ervaring alsnog boven de rest uitstijgen. Docent 2 verklaart hierover: *“dan zijn ze zo fanatiek en dan kunnen ze meer bereiken dan de zogeheten hele slimme leerlingen. Dus we zien niet eens zo heel veel verschil [door bezit van bepaald inzicht]. Misschien wel in het kunnen oplossen van bepaalde wat complexere problemen, dat slimme leerlingen daar natuurlijk wat eerder uit zullen komen”*. Docent 8 verwacht juist niet dat de niveauverschillen ontstaan door geslacht en door meer ervaring van de basisschool. Over dat laatste licht hij het volgende toe: *“ik merk dat op de basisschool meer voor de leuk programmeren is [...] En zelfs met dat leuke als voorkennis merk ik dat het weinig verschil maakt in de derde klas als ik dit [werken met technologie] doe”*.

De docenten verwachten dat een aantal elementen van de integratie van computational thinking succesvol zijn bij de leerlingen. Docenten 1, 4, 5, 6, 8, 9 en 11 benoemen dat het werken met een technologie de leerlingen motiveert en enthousiast maakt. Hierbij benoemen docent 4 en 9 dat het fijn is dat het praktisch is. Volgens docenten 9 en 10 is het praktische aspect daarnaast succesvol, omdat leerlingen het interessant vinden als ze zich kunnen inleven en de vrijheid fijn vinden om zelf met het praktische aspect aan de slag te gaan. Dat laatste ziet docent 6 ook en die benoemt: *“Er zijn ook heel veel kinderen die dat niet willen en hun eigen dingetjes willen maken, tenminste dat zie ik in drie [vo] wel, die willen lekker hun eigen ding doen”*. Docent 1 benoemt dat leerlingen de uitdaging van het zelf moeten gaan oplossen interessant vinden.

Op de vraag waar leerlingen moeite mee zullen hebben bij de integratie van computational thinking, benoemen docenten een aantal botsende karakteristieken van leerlingen. Docenten 1, 9 en 11 verwachten dat leerlingen chagrijnig worden, snel opgeven of onrustig worden als opdrachten niet lukken. Docent 8 voegt hieraan toe dat hij denkt dat leerlingen moeite hebben met zelf de fout vinden. Door het missen van deze zelfstandigheid verwachten docenten 6 en 10 dat leerlingen in de brugklas moeite hebben met grote en vrije opdrachten. Docenten 1, 3 en 6 benoemen dat de technische aspecten van het werken met een technologie moeilijk is voor leerlingen, bijvoorbeeld omdat de leerlingen het eng vinden of de functionaliteiten niet snappen. Docenten 1, 2, 3, 6 en 9 verwachten dat leerlingen het werken met een programmeertaal moeilijk vinden, omdat leerlingen het letterlijk nemen niet goed begrijpen. Docent 1 zegt hierover: *“nou, daar zijn ze weken mee bezig voordat de pop op het beeldscherm heen en weer loopt.”*

Docent 5 voorziet problemen bij leerlingen met betrekking tot het volgen van de stappen van een handleiding. Docenten 1 en 10 verwachten dat leerlingen moeite hebben bij het werken met modellen.

Figuur 5 toont expliciet genoemde verwachtingen van docenten waarmee leerlingen moeite zullen hebben met betrekking tot de computational thinking aspecten.

	Docenten							Totaal
	1	3	4	5	8	9	11	
Abstractie	●	●				●	●	4
Algoritmisch denken			●		●			2
Decompositie						●	●	2
Evaluatie				●		●	●	3
Generalisatie					●	●		2
Totaal	1	1	1	1	2	4	3	13

Figuur 5: Verwachtingen van docenten waar leerlingen moeite mee zullen hebben.

De redenen hiervoor zijn niet duidelijk te achterhalen. Voor de moeite met abstractie benoemt docent 9 dat leerlingen geneigd zijn om naar de details te gaan. Docent 3 betreft hier de programmeertaal bij en verwacht dat leerlingen moeite hebben met de abstractie van de regels code en het begrijpen wat het doet zonder het daadwerkelijke resultaat te zien. Docent 4 benoemt als oorzaak dat leerlingen niet gewend zijn om algoritmisch te denken. Bij generalisatie beschrijft docent 8 de moeite bij leerlingen als volgt: *“dat ze die oplosmethode [voor de opdracht] niet los durven te laten. Ze denken dat dat dan meestal de oplossing is voor veel problemen [...] daar gaan ze wel wat in veranderen, maar het lukt niet, ze lopen vast, maar ze durven dan niet iets heel anders, die stap, dat vinden ze echt heel lastig.”* Een andere oorzaak voor de moeite met generalisatie vinden

docent 9 en 11 in het feit dat sommige leerlingen het lastig vinden om de theorie of de oplossing van een opdracht in een nieuw context te plaatsen. Een aantal docenten benoemen nog expliciet dat leerlingen met bepaalde aspecten geen moeite zullen hebben. Volgens docent 1 en 9 hebben leerlingen algoritmisch denken snel door. Docent 11 voegt toe dat leerlingen met interesse geen moeite hebben met evaluatie, want *“die willen maar een ding, dat het werkt en maakt niet uit hoe, als het maar gaat werken”*. Bij wiskunde hoeft docent 8 het aspect decompositie niet uit te leggen, want *“dat weten ze”*.

Figuur 6 toont de verwachtingen van docenten over wat leerlingen nodig hebben om computational thinking te integreren in hun les.

Wat hebben leerlingen nodig?	Docenten												Totaal
	1	2	3	4	5	6	7	8	9	10	11	12	
Aansluiting bij leerling karakteristieken	●	●	●	○	○	●	○	○	○	●	○	○	5
Computerkennis	○	○	●	●	○	○	●	●	○	○	○	●	5
Praktische elementen	●	○	●	○	○	●	●	○	●	○	○	○	5
Programmeervaardigheden	●	○	○	○	○	○	○	●	○	○	○	○	2
Specifieke leerling karakteristieken	●	●	●	●	○	●	●	○	○	○	●	○	7
Technologische / computer vaardigheden	●	○	○	○	○	○	●	○	○	○	○	○	2
Ervaring met algoritmisch denken	○	○	○	○	○	○	○	●	○	○	○	○	1
Vakkennis	○	○	●	○	○	○	○	○	○	○	○	●	2
Totaal	5	2	5	2	0	3	4	3	1	1	1	2	29

Figuur 6: Wat hebben leerlingen nodig?

De specifieke leerling karakteristieken die worden benoemd zijn concentratie (docenten 3 en 4), zelfstandig werken (docent 7), begrijpend lezen (docent 2), zelfvertrouwen (docenten 3, 6 en 11), creativiteit (docenten 1, 2 en 11) en enthousiasme (docenten 1, 2 en 3). Hierbij verwachten docenten enige aansluiting bij de leerling karakteristieken, zoals gebruiksvriendelijke technologieën (docent 6), regelmatige succeservaringen (docenten 3 en 6) en uitdagingen voor de leerlingen zodat het niet saai wordt (docenten 1, 2 en 10). Praktische elementen bestaan uit het nodig hebben van tastbare resultaten (docenten 1, 3, 6 en 9) en voorbeelden van wat goed en slecht is (docent 7). Docent 1 vindt bijvoorbeeld met name de combinatie van tastbare resultaten en passende uitdagingen van belang: *“je moet precies iets vinden waar leerlingen zeggen van, dit heb ik gemaakt, dit kan ik, en net op het vlak dat ze dat kunnen”*. Voor de vakkennis legt docent 3 uit dat de leerlingen kennis moeten hebben van de natuurkunde van het vorige jaar en docent 12 verwacht dat leerlingen bekend zijn met het onderwerp waar de docent computational thinking bij aan wil laten sluiten.

Tot slot, toont figuur 7 een overkoepelend resultaat om te identificeren in hoeverre docenten kennis hebben van de leerlingen. Het figuur is gebaseerd op antwoorden van docenten waarbij zij expliciet aangaven wanneer de integratie bij een klas wel of niet mogelijk was, of wanneer zij al zelf aan een vorm van integratie deden bij een bepaalde klas. Hiermee tonen zij aan dat ze kennis hebben van de leerlingen, weten wat voor hen wel of niet werkt en daardoor een completer PCK hebben.

Het figuur laat zien dat er een duidelijke splitsing is in de mate waarin docenten kunnen aangeven wanneer de integratie mogelijk is.

	Docenten												Totaal
	1	2	3	4	5	6	7	8	9	10	11	12	
PO groep 1	○	●	○	●	○	○	○	○	○	○	○	○	2
PO groep 2	○	○	○	●	○	○	○	○	○	○	○	○	1
PO groep 3	○	●	○	●	○	○	○	○	○	○	○	○	1
PO groep 4	○	○	○	●	○	○	○	○	○	○	○	○	1
PO groep 5	○	●	○	●	○	○	○	○	○	○	○	○	2
PO groep 6	○	○	○	●	○	○	○	○	○	○	○	○	1
PO groep 7	●	●	○	●	○	○	○	○	●	○	○	○	4
PO groep 8	●	●	○	○	○	○	○	○	●	○	○	○	3
VO klas 1	●	○	●	○	●	●	●	○	○	●	●	○	7
VO klas 2	●	○	○	○	●	○	○	○	○	○	○	○	1
VO klas 3	○	○	○	○	●	●	○	●	○	○	○	○	3
VO klas 4	○	○	●	○	○	○	○	○	●	●	●	○	3
VO klas 5	●	○	●	○	○	●	○	○	●	●	●	●	4
VO klas 6	●	○	○	○	○	○	○	○	●	●	●	○	3
<b>Totaal</b>	<b>6</b>	<b>5</b>	<b>3</b>	<b>7</b>	<b>3</b>	<b>3</b>	<b>1</b>	<b>1</b>	<b>5</b>	<b>4</b>	<b>4</b>	<b>1</b>	<b>43</b>

Figuur 7: Wanneer integratie mogelijk is.

De toepassing van deze kennis omtrent het starten van de integratie in het basis- en middelbaar onderwijs, zal later in paragraaf 4.3 van de resultaten worden besproken.

#### 4.1.4 M3 - Begrip van onderwijsaanpak bij het onderwijzen van computational thinking

In deze paragraaf wordt de didactische en technologische kennis van de docenten beschreven. Figuur 8 toont alle genoemde technologieën bij de interviews.

Figuur 9 toont alle soorten leeractiviteiten die zijn benoemd bij de interviews.

	Docenten												Totaal	
	1	2	3	4	5	6	7	8	9	10	11	12		
Websites, apps en sociale media	○	●	○	●	○	○	●	○	○	○	○	○	○	3
Simulatie software	●	○	●	○	○	●	○	○	○	○	●	○	○	4
Grafische programmeeromgevingen	●	●	○	●	○	●	○	○	●	●	●	○	7	
Sensoren en camera's	●	○	○	●	●	●	○	●	○	○	●	●	7	
Ontwerp- en productietechnologieën	●	○	○	●	●	●	●	●	○	●	●	●	9	
Robots	●	●	○	●	●	●	○	●	●	●	●	○	9	
Programmeertalen	●	●	●	●	●	○	○	○	●	●	●	●	10	
Totaal	6	4	2	6	4	6	2	3	3	5	5	3	49	

Figuur 8: Kennis van soorten technologie voor de integratie van computational thinking.

	Docenten												Totaal
	1	2	3	4	5	6	7	8	9	10	11	12	
Unplugged	●	●	○	●	●	●	●	○	●	●	●	○	9
Substitutie	●	●	○	●	●	●	●	●	●	●	●	○	10
Augmentatie	●	●	●	●	●	●	●	●	●	●	●	●	12
Modificatie	●	●	●	○	○	○	○	●	○	●	●	○	6
Herdefinitie	●	●	○	●	●	●	○	○	●	●	●	○	8
Totaal	5	5	2	4	4	4	3	3	4	5	5	1	45

Figuur 9: De soorten leeractiviteiten

Zeven docenten benoemen het spelenderwijs leren werken met een technologie als substitutie activiteit (docenten 2, 5, 6, 8, 9, 10 en 11). Het spelenderwijs werken met een technologie gaat op basis van een aantal opdrachten die de leerlingen moeten volgen en focust zich puur op het kennismaken met de technologie. Docent 6 licht hierover toe: *“we zeggen gewoon spelenderwijs, ga maar kijken wat je ermee kan. Dus heel open. Het mag ook mislukken, [...], gewoon om er kennis mee te maken.”* Alle zeven docenten gaan hiermee verder en benoemen het zelf kunnen maken van functionaliteiten als augmentatie leeractiviteit. Een voorbeeld hiervan is het programmeren van de bewegingen van een Arduino of het lampje van een micro:bit. Docenten 1, 3, 10 en 11 benoemen ook leeractiviteiten met betrekking tot het maken en editen van video's. Docent 12 denkt na over het combineren van technologie aan de Engelse literatuur en benoemt hierbij als voorbeeld het maken van een database. Docent 3 verwoordt een activiteit waarbij leerlingen middels een handleiding de activiteiten van professionele programmeurs uitvoeren.

Voor de augmentatie is een ander veel genoemde activiteit het maken van grafieken of modellen (docenten 1, 3, 5, 10 en 12). De toelichting hierbij is goed verwoord door docent 3: *“Dat zijn dingen waar je met de hand niet meer aan kan rekenen voor een vak, waarbij een computer, op zich met*



dezelfde regels die wij wel met de hand gebruiken maar heel veel tegelijk kan uitrekenen, waardoor de computer heel nuttig is.”

Docent 3 is zich ook bewust van een modificatie activiteit waarbij leerlingen de regels code al krijgen, maar de leerling moet bepaalde getallen aanpassen “zodat het echt goed bij elkaar past”. Dit is dan ook de meest voorkomende activiteit bij modificatie: het uitvoeren van een vorm van data-analyse aanleiding van het verwerken van de data (docent 1, 3, 8, 10 en 11).

Docent 3 benoemt het gebruik van een programmeertaal voor het maken van modellen als augmentatie activiteit en daarbij aansluitend de modificatie activiteit om met de programmeertaal de data achter de modellen te analyseren. Docent 8 ziet deze zelfde activiteiten maar beschrijft voor de stap naar augmentatie eerst nog de substitutie activiteit van het programmeren van een robot om het programmeren zelf onder de knie te krijgen.

De herdefinitie richt zich momenteel met name op het creëren van kleine vakgerelateerde producten met de functionaliteiten van de technologieën (docent 2, 4, 5, 6, 9, 10 en 11). Docent beschrijft de essentie hiervan als volgt: “Heel vaak is er een praktische noodzaak, en dan daarnaar toe werken [...] we hadden in een keer allemaal dingen, en wat gaan we ermee doen in plaats van wat hebben we nodig.” Het gevolg hiervan is dat de onderwijsaanpak bij deze docenten veelal een vast patroon volgt. “je pakt inderdaad een microbit erbij, en dat je met de kinderen gaat bespreken van, wat is het eigenlijk voor een computertje, wat kan je er allemaal mee en dat je een groepje een taak geeft, van wat zouden jullie ermee willen gaan doen? Oke, jullie willen een dobbelsteen programmeren, oke. Ga maar uitzoeken, ontdekken van hoe je dat zou moeten kunnen doen.” Deze verwoording van docent 2 schetst het gebruik van de technologie die loopt van substitutie (begrijpen van microbit) langs augmentatie (toepassen van microbit) naar de herdefinitie (creëren met microbit). Deze aanpak is terug te zien bij docenten 4, 5, 6 en 9.

Docent 7 is voor de leeractiviteiten puur gericht op haar eigen vak. Ze laat leerlingen op de computer een literaire mindmap te maken als substitutie activiteit en gebruikt verschillende apps en websites voor het uitleggen van de lesstof en waar leerlingen opdrachten op kunnen maken.

Figuur 10 toont de curriculum karakteristieken die de docenten benoemen bij de verschillende leeractiviteiten.

	Docenten												Totaal
	1	2	3	4	5	6	7	8	9	10	11	12	
Betekenisvolle en/of authentieke en/of gesitueerde contexten/instructie	●	●	●	●	●	●	●	○	●	●	●	●	11
Complex/probleemoplossend	●	●	●	○	○	○	○	●	●	●	○	●	7
Oefenen en/of memoriseren (drill and practice)	●	●	●	●	●	●	●	○	●	●	●	●	11
Begeleid ontdekkend leren	○	●	●	●	●	●	●	●	●	●	●	●	11
Totaal	3	4	4	3	3	3	3	2	4	4	3	4	40

Figuur 10: De curriculum karakteristieken die de docenten benoemen bij de verschillende leeractiviteiten.

Vrijwel alle docenten, op docent 8 na, benoemen leeractiviteiten die gebaseerd zijn op oefenen en/of memoriseren. Deze leeractiviteit is niet toegepast en gericht op reproductie door nadoen en/of namaken. Docenten 1, 2, 3, 8, 9, 10 en 12 bouwen op het oefenen voort en bieden de leerlingen een probleemoplossende leeractiviteit aan. Bij het oplossen kan het probleem ook van praktische aard zijn en moeten de leerlingen door te handelen de leeractiviteit uitvoeren (docenten 1, 3, 5, 6, 7, 10, 11 en 12). Deze leeractiviteiten worden voorzien van een context vanuit hun eigen vak (docenten 2, 3, 4, 7, 10, 11 en 12) of vanuit een persoonlijke context (docenten 4 en 10). Docent 9 beschrijft de wijze waarop zij gesitueerde instructie geeft als volgt: *“het is niet dat we uitgaan van digitale geletterdheid en daar bedenken we een thema omheen, maar als het [technologie] erbij past dan pakken we het er wel bij”*. Op docent 1 na, zien alle docenten dat de leerlingen vanuit een onderzoekende houding zelf moeten uitzoeken, ontdekken en de oplossing moeten uitvoeren. Docenten 3, 5, 6, 7, 8, en 12 bieden hierbij wel nog een handleiding aan om de leerlingen te begeleiden in het ontdekkend leren. Docent 8 koppelt bij het bieden van een handleiding zijn rol als docent om de leerlingen te helpen het leerproces door te komen en licht toe:

*Dat ontdekken ze zelf, dat bouw je op. In het begin bied je het eerst aan, en dan laat je ze het wat veranderen [...] en met de handleiding van wanneer wel wanneer niet, waar wel en waar niet.*

*N: En dat bouwt zich op, jij onderwijst dat, hoe zie jij jouw rol als docent daarin?*

*K: Bij de mbots ben ik eigenlijk helemaal geen docent, want dat is ontdekken en leren. Je bent leraar in begeleiden, dus niet de reguliere docent die ik bij wiskunde ben, zeg maar. Maar daar ga je eigenlijk met z'n allen een probleem tacklen, en ik hoef niet per se het antwoord al te weten, want het belangrijkste is dat ik leerlingen help het leerproces te doorlopen, om erachter te komen wat de oplossing is. [...] Ik kijk gewoon naar het leerproces en ik ga kijken welke stappen ze hebben overgeslagen en welke niet. Of het verstandig is of niet.*

Docent 8 beschrijft hier een toepassing van alle mogelijke rollen bij het onderwijzen. Figuur 11 toont het schematisch overzicht van de benoemde rollen van alle docenten.

	Docenten												Totaal	
	1	2	3	4	5	6	7	8	9	10	11	12		
De docent stimuleert leerlingen, geeft advies en suggesties, stelt vragen	●	○	●	○	●	●	●	●	●	●	●	●	●	10
De docent controleert, registreert en beoordeelt de voortgang van de leerling	○	○	●	○	●	●	○	●	●	●	●	○	7	
De docent draagt kennis over	●	●	○	●	○	○	●	●	○	●	●	○	7	
De docent geeft instructie gericht op het ondersteunen en aanleren van specifieke kennis en vaardigheden	●	○	●	●	●	●	●	●	●	●	●	●	11	
Totaal	3	1	3	2	3	3	3	4	3	4	4	2	35	

Figuur 11: De benoemde rollen van de docenten.

Docenten 2 en 4 focussen zich op het kunnen overdragen van kennis door eerst een korte uitleg te geven van de theorie, zodat leerlingen daar zelf mee verder kunnen waardoor de andere rollen minder van belang zijn. Docent 4 ziet dan wel dat in groep 1 de leerlingen veel leerkrachtsturing nodig hebben om na de uitleg er zelf mee aan de slag te gaan. Docenten 3, 5, 6 en 9 passen dezelfde integrale didactische aanpak als docenten 8, 10 en 11 toe, maar benoemen geen rol voor het

overdragen van kennis. De reden hiervoor kan zijn dat docenten 3, 5 en 6 ook gebruiken maken van een handleiding voor het overdragen van de kennis en hierdoor het tijdens de les minder van belang vinden om in de rol van onderwijzer hier meer tijd aan te besteden. Docent 9 lijkt de overdracht van kennis vanuit praktische noodzaak over te slaan door een persoonlijke voorkeur voor het zelf proberen ongeacht kennis van het onderwerp te hebben, met een focus op de rol als docent om feedback te geven bij het ontdekkend leren van de leerlingen. Docent 12 beschrijft een leeractiviteit waarbij zijn rol is om eerst de leerlingen een instructie te geven om vanuit de context van het vak een praktische opdracht te maken en als deze niet lukt bij de leerlingen om vragen te stellen vanuit die context, zodat de fouten in de opdracht beter begrepen kunnen worden. De rol van de leerlingen hierbij is om dus een opdracht uit te voeren en daarbij een goed eindproduct op te leveren. Deze vorm van rollen sluit ook aan bij de rollen die docent 4 en 7 beschrijven bij hun leeractiviteiten. De verschillen in rollen zijn weergegeven in figuur 12. Om de rollen van de leerlingen te bepalen is voor alle leeractiviteiten gekeken welke rollen de leerlingen daarbij uitvoeren.

	Docenten												Totaal
	1	2	3	4	5	6	7	8	9	10	11	12	
Onderzoeker (leerlingen doen zelf onderzoek, zoeken naar informatie)	●	●	●	○	○	○	○	●	○	●	●	○	6
Beoordelaar (leerlingen beoordelen zichzelf of elkaar)	○	○	○	○	○	●	○	○	●	●	○	○	3
Uitvoerder (leerlingen voeren de opdracht uit)	●	●	●	●	●	●	●	●	●	●	●	●	12
Toehoorder (leerlingen luisteren naar de instructie)	●	○	○	○	●	○	○	○	○	●	○	○	3
Creator / constructeur (leerlingen produceren zelf, zijn creatief)	●	●	●	●	●	●	●	○	●	●	●	●	11
Totaal	4	3	3	2	3	3	2	2	3	5	3	2	35

Figuur 12: De rollen van leerlingen bij leeractiviteiten.

Opvallend is dat in elke leeractiviteit een uitvoerend aspect zit. Figuur 13 toont de sturing van de onderwijsactiviteiten. Hier is in terug te zien dat meeste onderwijsactiviteiten door de lerende worden bepaald. Dit houdt in dat de leeractiviteiten aangeboden worden met veel keuzevrijheid. Docenten 6, 7 9 en 10 bieden binnen de leeractiviteiten keuzemogelijkheden, en net zoals alle andere docenten bieden zij de leeractiviteiten in het algemeen aan als differentiatie of als ondersteuning.

	Docenten												Totaal
	1	2	3	4	5	6	7	8	9	10	11	12	
Leeractiviteiten bepaald door lerende	●	●	●	○	○	●	●	○	●	●	●	○	8
Leeractiviteiten bepaald door docent (of didactisch medium)	●	○	○	○	○	○	○	○	●	○	●	○	3
Snelheid bepaald door docent (of didactisch medium)	○	○	○	○	○	○	○	●	○	○	●	○	2
Totaal	2	1	1	0	0	1	1	1	2	1	3	0	13

Figuur 13: De sturing van de onderwijsactiviteiten.

Docenten 8 en 11 kiezen ervoor om de leeractiviteiten vanaf nul te beginnen. Docent 11 gaat, net als docent 9, met deze start vanaf nul eerst bekijken wat er al aan kennis aanwezig is, om vervolgens aan de hand daarvan de lessen aan het niveau van de leerlingen aan te passen.

Als we de kennis van de soorten activiteiten gaan vergelijken met de kennis van de soorten technologie, valt op dat alle docenten een combinatie van een activiteit en technologie kunnen benoemen.

#### 4.1.5 M4 - Begrip van toetsing

In deze paragraaf wordt beschreven welke manieren de docenten weten om het begrip van de leerlingen bij het werken met computational thinking tijdens en na de les te toetsen.

Vrijwel alle docenten, behalve 1, 4 en 7, kijken naar het eindproduct om het begrip van de leerlingen te toetsen. Dit houdt in dat bij de toetsing wordt gekeken of de functionaliteiten werken. Docenten 9 en 11 kijken verder naar de code van de functionaliteiten om te zien hoeveel de leerlingen zelf hebben gedaan. Docent 12 beschrijft hoe het eindproduct getoetst kan worden, door te kijken of de handleiding goed is gevolgd. Het behalen van de oefenopdrachten is voor docenten 2 en 4 ook een manier om te toetsen.

Als het gaat om tijdens de les het begrip van de leerlingen te toetsen, beschrijven alle docenten, op docent 1, 2 en 7 na, eenzelfde soort aanpak om te zien of leerlingen het begrijpen. Bij deze aanpak loopt de docent tijdens de les rond, praat met de leerlingen en schat op basis van hun reacties in of de leerlingen het begrijpen. Docenten 6, 8 en 9 kennen daarnaast de reacties van de leerlingen die aantonen dat zij het begrijpen zonder ernaar te vragen, zoals vragen stellen die buiten de opdracht liggen of een grote mate van concentratie op de opdracht. Deze concentratie trekt docent 1 breder en geeft te kennen dat aan de hand van een onrustige, niet geconcentreerde klas ook tijdens de les getoetst kan worden of de opdracht gesnapt wordt.

Acht docenten weten manieren te benoemen om het begrip te meten, bestaande uit een eindverslag op het eindproduct (docent 1, 6 en 7), een quiz (docent 7), een mindmap (docent 9), een checklist van vaardigheden (docent 6) of een eindevaluatie waarbij leerlingen aan de docent het geleerde of de fouten in een programma moeten uitleggen (docenten 8, 10 en 11). Docenten 10 en 11 zouden zelf de fouten in het programma zetten, maar docent 3 beschrijft een manier om bij een afsluitende toets de fouten van de leerlingen in het project te gebruiken als toetsvragen. Hij raadt dit aan, want *“er waren heel veel types fouten die we erin konden stoppen die ze als het goed is hadden gezien tijdens het doen van het project.”*

Docent 9 geeft aan van toetsen weinig te weten, omdat *“dit zijn meer denkvaardigheden, dat moet je ook dan eerst weer kunnen meten, en dat vind ik lastig”*. Samen met de docenten 1, 2, 4, 6 en 7 benoemt zij dan ook dat leerdoelen met betrekking tot computational thinking op het moment nog niet gemeten of geëvalueerd worden.

## 4.2 Interconnecties

In deze paragraaf worden twee vergelijkingen tussen de onderdelen van de PCK beschreven. In het eerste onderdeel zullen de leerdoelen (M1) met het begrip van de leerlingen (M2) vergeleken. Deze vergelijking is interessant om te achterhalen welke ideeën docenten hebben over het inzetten van de leerdoelen bij de bepaalde klassen. Het tweede onderdeel betreft een vergelijking tussen

de leerdoelen en de onderwijsaanpak, om te kunnen zien welke ideeën docenten hebben om de leerdoelen te onderwijzen.

Met uitzondering van docent 7 kunnen alle docenten met hun begrip van de leerlingen bepalen wanneer een leerdoel in een klas toegepast kan worden. Met name weten de docenten wanneer het mogelijk is om te werken met technologie (docenten 1, 2, 4, 5, 6, 10 en 11) of om te gaan leren programmeren (docenten 2, 3, 4, 6, 8, 10, 11 en 12). Opvallend is dat voor de doelen omtrent het leven in een digitale samenleving slechts drie docenten (3, 6 en 12) weten te benoemen wanneer dat zou moeten beginnen. Hetzelfde geldt voor de affectieve doelen waar twee docenten (5 en 6) een aansluiting kunnen benoemen. De kennis van docenten voor leerdoelen in de onderbouw is het grootst. In deze klas weten docenten ook wanneer bepaalde leerdoelen niet kunnen. De leerlingen zouden moeite hebben met het vormgeven van informatie (docent 10) en de analytische kant van het werken met technologieën (docenten 5 en 11). Docent 9 maakt duidelijk dat het begrijpen van de basisonderdelen van programmeren een te hoge verwachting is in groep 7 en 8.

De docenten zijn niet goed in staat om de leeractiviteiten te koppelen aan bepaalde leerdoelen. In de interviews komt veel voor dat een docent aan het begin een aantal doelen bij het onderwijzen van computational thinking uiteenzet en vervolgens de verschillende activiteiten en de materialen op de scholen uitleggen, maar hierbij geen directe terugkoppeling meer maakt. Bij een nadere inspectie blijken docenten zijn 1, 7 en 10 hier opvallend erg in tekort te schieten, terwijl docenten 3, 8 en 12 vrij consistent blijven in het benoemen van activiteiten met bijpassende doelen.

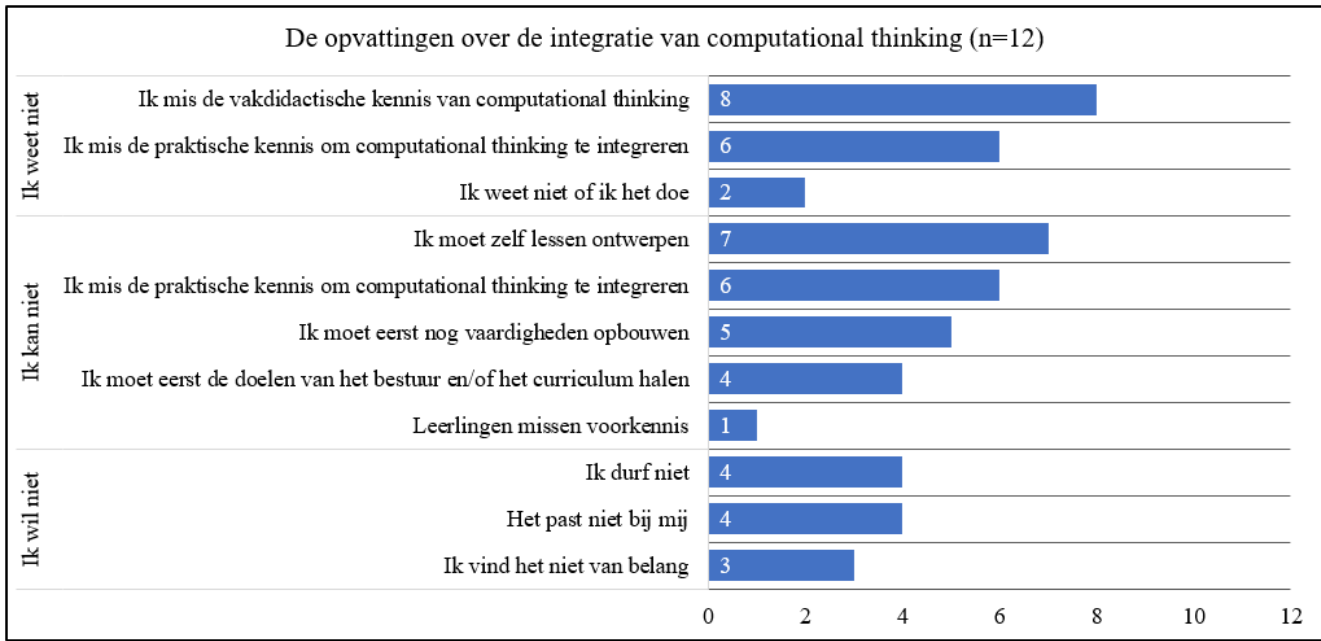
De overige opties voor interconnecties bleken geen duidelijke resultaten te geven. Het combineren van M4 met M1, M2, M3 gaven geen opvallende resultaten, omdat de docenten een vrij gemeenschappelijke aanpak hebben bij de toetsing tijdens de lessen. Bij het vergelijken van M2 met M3 bleken de meeste docenten de leerling karakteristieken in acht te nemen en formuleerde aan de hand daarvan rol als docent. Gezien de rollen van de docenten erg verschillen onderling, bleek deze analyse ook onvoldoende resultaten te geven.

### 4.3 De praktijk

Computational thinking wordt op het moment in 24 lessen van negen schoolvakken geïntegreerd. In Appendix C staat het overzicht van deze lessen. Dertien lessen zijn voor de eerste klas van het voortgezet onderwijs. De doelen zijn zeer uiteenlopend en er valt op het moment nog geen duidelijke leerlijn in te ontdekken. Het betreffen vaak losstaande activiteiten om een keertje kennis te maken met programmeren of het werken met technologie. In slechts twee lessen is er een specifiek doel voor het kunnen toepassen van computational thinking om vakgerelateerde problemen op te lossen. Zeven lessen hebben eigenlijk onderliggend doel aan de les. Het gebruik van de technologie of de denkprocessen van computational thinking zijn toevallig nodig bij de lessen en werden benoemd nadat de docent kennis had genomen van de betekenis van computational thinking.

### 4.4 De opvattingen

Figuur 14 toont de verschillende opvattingen van docenten over de integratie van computational thinking.



Figuur 14: De opvattingen van de docenten over de integratie van computational thinking

## 4.5 De verwachtingen

De docenten denken graag mee met de integratie van computational thinking. Hoewel het werkelijke aantal activiteiten op het moment laag is, weten zes docenten nog zeker vijftien activiteiten voor het onderwijzen van computational thinking te verzinnen. Bij deze activiteiten werden tien activiteiten al gekoppeld aan specifieke leerdoelen. Echter, om dit mogelijk te maken verwachten zij dat computational thinking ook wordt toegevoegd aan het huidige curriculum van de vakken (docenten 2, 4, 5, 8 en 9), wat naast het ontbreken van geschoolde docenten (docenten 2, 4, 7, 9 en 11) gezien wordt als grootste organisatorische struikelblok. Binnen deze reorganisatie wordt benadrukt dat het succesvol kunnen invoeren van computational thinking ook ligt bij de docenten zelf. De docenten verwachten dat alle docenten een gelijke aandacht aan de integratie besteden (docenten 2, 4, 5 en 7) en zich actief inzetten voor het opbouwen van vaardigheden (docenten 3, 4 en 7). Als de docenten hierbij meer tijd krijgen voor het maken van lesmateriaal (docenten 4, 5, 7 en 8) met de ruimte in de les om de materialen toe te passen (docenten 2 en 9), dan zien zij zeker een mogelijkheid om computational thinking te integreren (docenten 1, 4, 6, 7, 8, 11 en 12) met een expliciete wens voor de integratie van technologie (docenten 1, 3, 4, 5, 6, 7, 10, 12). Van de technologieën wordt verwacht dat die wel werken (docenten 1, 7, 8 en 9) en op school beschikbaar zijn (docenten 3, 4 en 7). Docenten 6 en 10 maken nadrukkelijk kenbaar dat het nodig is om de traditionele opvatting over docentschap bij andere docenten moet worden losgelaten om computational thinking succesvol te onderwijzen. Hierover concludeert docent 6 de verwachtingen met het volgende: *“het is een rolverandering, helemaal goed, helemaal prima, niks mis mee. Ik zeg heel vaak tegen leerlingen, dat vind ik een hele goede vraag, maar ik weet het antwoord er niet op. Zullen we samen op zoek gaan?”*

## 5 Conclusie

### 5.1 Het beeld van de docenten

Docenten hebben nog zeker geen volledig beeld bij computational thinking. Veel docenten kunnen nog niet benoemen wat leerlingen nodig hebben. Het kunnen benoemen van de verwachte moeite van de leerlingen bij de computational thinking aspecten is ook matig. Overigens weten de docenten wel veel mogelijke activiteiten te benoemen ten aanzien van het integreren van een technologie, maar ze missen daarbij de aansluiting met hun vak. Er blijkt onderscheid gemaakt te kunnen worden in drie soorten docenten. De eerste groep integreert computational thinking uit eigen belang, ‘omdat het moet’ voor de leerlingen, zonder direct verband met hun vak. Deze docenten kenmerken zich door de modificatie activiteit in hun redeneringen over te slaan en de leerlingen zelf aan de slag te laten gaan met een technologisch product. Hierbij is er vaak geen sprake van een specifieke toetsing. Ook is de gelijkwaardige didactiek en de stimulerende rol van de docent kenmerkend. De tweede groep kenmerkt zich door het zelf op zoek gaan naar technologieën om hun vak mee uit te breiden. Deze docenten maken veel gebruik van modificatie activiteiten en simulatie software. De laatste groep docenten zetten computational thinking alleen in als het een aanvulling is op hun vak. Veelal kenmerken deze docenten zich door niet veel leerdoelen voor het werken met een technologie te hebben. Het beeld dat deze docenten van computational thinking hebben is heel breed georiënteerd op de technologie zelf. Als losse vaardigheid zien zij het nog niet en bij deze groep is het onbewust gebruiken van computational thinking het grootst. Het beeld van de docenten kenmerkt zich tenslotte door een visie waarbij de leerling een uitvoerende rol heeft bij de activiteiten en de het toetsen aan de hand gaan van de eindproducten van deze uitvoeringen. Tijdens het uitvoeren hanteren de meeste docenten een vaste aanpak: rondlopen, vragen stellen en aan de hand van de reactie van de leerling een instructie geven.

### 5.2 De huidige staat van de integratie en diens opvattingen en verwachtingen

Het missen van de kennis om een aansluiting te maken tussen de leerdoelen en de leeractiviteiten blijkt ook uit het verkregen overzicht van de lessen van de docenten. Bij nadere inspectie worden slechts 24 lessen gegeven. De grootste oorzaak hiervan blijkt te liggen in het feit dat de integratie op het moment onbewust gaat. Door deze onbewuste integratie zijn de aantal lessen nog schaars, maar de bereidheid is er. De reden dat de activiteiten nog schaars zijn komt veelal door het ontbreken van kennis, zoals ook blijkt uit de PCK van de docenten.

### 5.3 De verwachtingen

## 6 Discussie

### 6.1 Reflectie op resultaten

### 6.2 Vervolgonderzoek

## Referenties

### A

#### Interview Protocol

TODO

### B

#### Content Representation (CoRe)

	Belangrijke ideeën en concepten			
	idee A	idee B	idee C	idee D
1. Wat wil je dat de leerlingen over dit idee leren?				
2. Waarom is het voor de leerlingen belangrijk om dit te weten?				
3. Wat weet jij nog meer over dit idee (maar wat de leerlingen nog niet hoeven te weten)?				
4. Problemen en beperkingen verbonden met het onderwijzen van dit idee				
5. Kennis over het denken van de leerlingen die jouw onderwijzen van dit idee beïnvloeden				
6. Andere factoren die jouw onderwijs van dit idee beïnvloeden				
7. Onderwijsaanpak (en de redenen om die te gebruiken bij dit idee)				
8. Specifieke manieren om achter het begrip of de verwarring van de leerlingen rond dit idee te komen				

### C

#### Huidige algemene leerlijn

Zie bijlage pdf.