



AUTHENTIEKE TOETSING IN HET ONTWERPONDERWIJS MET SCRUM

Een praktijkstudie naar formatieve toetsing van programmeerconcepten bij een authentieke ontwerpopdracht in een Scrum-gebaseerd lesraamwerk.



A.P. DEKKER, S1190156
Docent en student informatica
Betreft: Onderzoek van Onderwijs (10 ECTS)

Kampen, 18 augustus 2021

Voor u ligt het verslag van het onderzoek dat gedaan is naar de toepassing van formatieve toetsing van algoritmische concepten in een Scrum-gebaseerde leeromgeving. Dit onderzoek is gedaan in het kader van de opleiding Educatie en communicatie in de bètawetenschappen (ECB) aan de Universiteit Twente, voor het vak informatica.

Dit onderzoek is begeleid door de volgende personen:

dr. L.E.I. Breymann	ELAN, Universiteit Twente
W.J.H. Nijhuis	ELAN, Universiteit Twente

Naast mijn begeleiders, wil ik ook prof. dr. E. Barendsen bedanken. Allen hebben zij mij gesteund en geholpen met het opzetten en afronden van het onderzoek. Ook spreek ik mijn dank uit naar alle leerlingen die aan het onderzoek hebben deelgenomen en naar mijn vrouw, voor haar meedenken en geduld.

Adriaan Dekker

Samenvatting

In het kader van een groter onderzoek naar “Formatieve toetsing van begripsontwikkeling in ontwerponderwijs” is een verkennend onderzoek gedaan naar een raamwerk voor een lessenserie op basis van Scrum voor het uitvoeren van een authentieke ontwerpdracht. Daarbij stond centraal dat docenten de individuele conceptuele voortgang van leerlingen konden monitoren. Hiervoor heb ik kenmerken beschreven op basis van de ontwerpgerichte leerstrategieën Learning by Design™ en Design-Based Science. Aan de hand van deze kenmerken heb ik een raamwerk voor een lessenserie ontwikkeld. Dat heb ik vervolgens getest door binnen dat raamwerk 11 4-vwo-leerlingen een ontwerpdracht uit te laten voeren met LEGO™ Mindstorms™. Dit onderzoek bevestigt het vermoeden dat het mogelijk is om individuele conceptuele vorming bij leerlingen te meten tijdens het uitvoeren van een ontwerpdracht binnen het ontwikkelde raamwerk op basis van Scrum. Ook zijn er enkele verbeterpunten geformuleerd. Zodoende geeft dit onderzoek aanleiding tot interessante vervolgonderzoeken.

Inhoud

Samenvatting	1
Inhoud	3
Definities	5
1. Inleiding.....	7
1.1. Doel en hoofdvraag.....	8
2. Theoretisch kader	9
2.1. Onderzoekend leren	9
2.2. Scrum	11
2.3. Leren programmeren	13
2.4. Taxonomie.....	15
3. Onderzoeksvragen	17
4. Methode	19
4.1. Procedure.....	19
4.2. Context.....	20
4.3. Instrumenten	21
4.4. Analyse.....	21
5. Resultaten	23
5.1. Fase 1: Kenmerken beschrijven (deelvraag 1)	23
5.2. Fase 2: Raamwerk ontwikkelen (deelvraag 2)	25
5.3. Fase 3: Raamwerk evalueren (deelvraag 3)	32
6. Conclusie en Discussie	35
6.1. Beantwoording van de onderzoeksvraag	35
6.2. Beperkingen	36
6.3. Aanbevelingen	37
Referenties.....	41
Bijlagen.....	43
A : Werken met Scrum in de les.....	43
B : Competenties-enquête	50
C : Observatiematrix programmeerconcepten	51
D : Opdrachtbeschrijving.....	53
E : Sprint planning formulier	54

Definities

In dit hoofdstuk leg ik uit hoe de termen die voor meerdere uitleg vatbaar zijn in het verslag geïnterpreteerd dienen te worden.

Concepten en contexten

Dit onderzoek sluit aan bij de recente ontwikkeling van het concept-contextonderwijs bij de bètavakken op Nederlandse scholen. Het begrip “concept” wordt daarbij gebruikt om vakinhoudelijk begrippen die leerlingen moeten leren aan te duiden. De omgeving waarin de concepten aangeleerd en toegepast worden, wordt “context” genoemd. Contexten worden ingezet met het doel om concepten een herkenbare betekenis te geven.

Conceptuele vorming/conceptualisatie

In het kader van dit onderzoek wordt met “conceptuele vorming” het proces bedoeld waarin leerlingen zich een bepaald concept eigen maken. Deze term heb ik gereserveerd om aan te duiden hoe ver leerlingen gevorderd zijn in het correct kunnen beschrijven en toepassen van een bepaald concept. Hiervoor wordt in andere onderzoeken ook de term “begripsontwikkeling” of “conceptualisatie” gebruikt. Als deze termen in dit verslag gebruikt worden, hebben ze dezelfde betekenis als conceptuele vorming.

Ontwerponderwijs

Ontwerponderwijs is een vorm van onderwijs waarbij geleerd wordt door zelf bezig te zijn met het toepassen van theoretische kennis in een bijpassende context door een artefact te ontwerpen. Naast het bevorderen van conceptuele vorming, levert deze onderwijsvorm ook op dat leerlingen leren samenwerken en praktijkopdrachten leren aanpakken.

Formatieve en summatieve toetsing

Formatieve toetsing moet gezien worden in relatie tot summatieve toetsing. Summatieve toetsing is toetsing waarvan het doel is om te bepalen of leerlingen een bepaald aantal vooraf opgestelde doelen gehaald hebben en of dat voldoende is om verder te mogen naar bijvoorbeeld een volgend leerjaar. Formatieve toetsing heeft daarentegen als doel om de voortgang van de leerlingen tussentijds te meten om inzicht te krijgen in het leerproces, zodat dat bijgestuurd kan worden indien nodig.

Authentiek

Het woord authentiek gebruik ik voor de manier waarop leerlingen concepten aanleren. Als de context van de opdracht aansluit bij de context waarin in de beroepspraktijk dezelfde concepten toegepast moeten worden, noem ik het een authentieke opdracht en een authentieke context. Als in een dergelijke context wordt bepaald of een leerling een bepaald niveau heeft bereikt op een manier die past binnen de context, noem ik dit authentieke toetsing. De toetsing moet natuurlijk en realistisch voelen voor de leerlingen, zodat de authentieke context niet aangetast wordt.

Artefact

In dit verslag heb ik het over een “*artefact*” dat ontworpen of ontwikkeld wordt. Hiermee bedoel ik een product dat het resultaat is van een creatief proces. Dat kan een programma zijn of een stuk hardware, maar ook een ontwerp of prototype in 2D of 3D.

1. Inleiding

In de nieuwe examenprogramma's voor bètavakken wordt bijzondere aandacht geschonken aan concepten in contexten: situaties uit het persoonlijk leven, de samenleving, en de wereld van beroepen, wetenschap en technologie. Het plaatsen van concepten in dergelijke contexten zou het leren van concepten versterken en verankeren (Pieters, Marsman, & Taminiau, 2015).

Een manier om onderwijs vorm te geven waarbij concepten in contexten geplaatst worden is ontwerponderwijs. Dit is een manier van lesgeven waarbij geleerd wordt door iets (een artefact) te ontwerpen en te maken. Een onderwijsmethode die hierbij aansluit, is Design Based Learning (DBL). Daarbij worden volgens Doppelt et al. (Doppelt, Mehalik, Schunn, Silk, & Krynski, 2008) nieuwe concepten opgedaan dankzij activiteiten die zij karakteriseren als "hands-on and heads-in": leerlingen moeten zoeken naar verbeteringen ("heads-in") die ze direct toepassen op hun artefact ("hands-on"). Tijdens het testen van hun nieuwe ontwerp moeten ze zoeken naar een volgende verbetering, waardoor een cyclisch proces ontstaat. Volgens Doppelt et al. levert deze manier van werken in de volgende drie voordelen op:

1. Het motiveert leerlingen omdat ze de toepassing van de leerstof inzien;
2. De leerling bepaalt hoe hij leert, passend bij zijn eigen leerstijl;
3. Door samen te werken worden leerlingen breder gevormd.

Onder leiding van Technische Universiteit Delft en Radboud Universiteit is op 1 januari 2016 een onderzoek gestart naar ontwerponderwijs bij scheikunde en informatica dat gefinancierd is door NRO. Daaraan wil ik graag een bijdrage leveren. De nadruk van dit NRO-onderzoek ligt op het toepassen van formatieve toetsing tijdens het ontwerpproces, zoals ook uit de titel blijkt: "Formatieve toetsing van begripontwikkeling in ontwerponderwijs". Het doel ervan is om "ontwerpgericht lesmateriaal te ontwikkelen met aansluitend toetsmateriaal dat authentiek is voor ontwerpsituaties". De vakken scheikunde en informatica werken in dit onderzoek samen, omdat het vermoeden bestond dat beide vakgebieden veel van elkaar kunnen leren. De reden daarvoor is dat beide vakken sterk leunen op abstracte concepten. Toch is de manier waarop deze vakken onderwezen worden heel verschillend. Het vak informatica is namelijk doortrokken van ontwerp opdrachten, waarbij leerlingen vaardigheden opdoen door het maken van een werkend artefact, terwijl scheikunde meestal meer theoretisch benaderd wordt.

Het onderwerp van bovengenoemd onderzoek sprak mij erg aan, omdat ik het tijdens de ontwerp opdrachten bij informatica erg lastig vind om duidelijk te krijgen waar leerlingen tegenaan lopen. Om dit te achterhalen is er een middel nodig waarmee getoetst kan worden welke concepten ze al beheersen. Het liefst zou ik dit doen op een manier die de context van de ontwerp opdracht niet minder authentiek maakt voor de leerlingen. Ook zou ik hier graag een oplossing voor ontwikkelen die toepasbaar is voor meerdere onderwerpen en niet maar voor één lessenserie. Daarom wilde ik een raamwerk ontwikkelen voor lesmodules waarbij het mogelijk is om conceptuele kennis waar te nemen bij leerlingen.

Voor het raamwerk wilde ik gebruikmaken van de ontwikkelmethode Scrum. Daarvoor had ik twee redenen. Ten eerste wordt de methode veel toegepast bij ICT-projecten in het bedrijfsleven. Door Scrum aan te passen voor gebruik in het onderwijs, maken leerlingen kennis met iets wat ze waarschijnlijk bij hun studie of werk ook tegen zullen komen. Bovendien werken ze zo in een context die bijdraagt aan de authenticiteit van de opdracht. Daarmee bedoel ik dat een opdracht waarheidsgetrouw en echt overkomt op de leerlingen. Dat moet de motivatie ten goede komen. Motivatie is vooral in het programmeeronderwijs een sterke indicator gebleken voor het succesvol

leren programmeren (Tritrakan, Kidrakarn, & Asanok, 2016). Daarbij is van groot belang of de context waarin leerlingen oefenen motiverend werkt (Van Merriënboer & Paas, 1990).

Ten tweede wordt bij Scrum het ontwerpproces vormgegeven volgens een vast patroon. Het werken met “rituelen” wordt door Kolodner et al. (2003) aangedragen als goede manier om conceptuele vorming waar te nemen. Er worden bij Scrum bijvoorbeeld daily stand-ups gehouden en sprints worden op een vaste manier gestart en afgesloten. Zulke momenten zijn erop gericht om de voortgang van het ontwerp inzichtelijk te maken. Het zijn mogelijk ook gelegenheden om inzicht te krijgen in de conceptuele vorming van leerlingen. Door onderzoek te doen naar een werkvorm die gebaseerd is op Scrum en authentieke toetsing van conceptuele vorming mogelijk maakt, wordt beoogd een bijdrage te leveren aan het eerdergenoemde NRO-project.

1.1. Doel en hoofdvraag

Het doel van dit onderzoek is om Scrum aan te passen om het te gebruiken als raamwerk voor een lessenserie, waarbij het voor de docent mogelijk is om conceptuele kennis waar te nemen bij individuele leerlingen. In de lessenserie wordt daarbij gewerkt aan een ontwerpopdracht, waarvan de context niet verstoord mag worden bij het waarnemen van de conceptuele kennis, zoals wel gebeurt als er een toets afgenomen zou worden.

De bijdrage die dit onderzoek aan het NRO-project levert, haakt vooral in op de eerste deelvraag van het project: *“Welke deelproducten uit het ontwerpproces lenen zich voor conceptuele feedback? Hoe kunnen deelproducten zodanig worden gespecificeerd dat ze maximale informatie over begripsontwikkeling opleveren?”*. Aan de hand van de Scrum-methode zijn er deelproducten gespecificeerd die hieraan voldoen. Bij het ontwerpen van het bijbehorende materiaal, is ook aandacht besteed aan de tweede deelvraag van het project: *“Op welke manier (wanneer, door wie, bij welk type ontwerpproblemen, ...) kunnen deze toetsinstrumenten worden ingezet zodat ze een beeld geven van individuele ontwikkeling?”*.

Binnen deze kaders is in het uitgevoerde onderzoek gezocht naar een antwoord op de vraag:

Hoe kan Scrum aangepast worden tot raamwerk voor een lessenserie om conceptuele vorming van individuele leerlingen waar te nemen tijdens het uitvoeren van een authentieke ontwerpopdracht voor informatica in de bovenbouw van het vwo?

Deze hoofdvraag deel ik op in drie deelvragen. Voordat ik die beschrijf, ga ik eerst in op de theorie die ik gebruikt heb voor dit onderzoek. Daarna beschrijf ik mijn deelvragen en de fasering van het onderzoek.

2. Theoretisch kader

In dit hoofdstuk wordt een basis vanuit de literatuur gelegd voor het opstellen van de ontwerprichtlijnen in de eerste fase. Deze richtlijnen zijn gebaseerd op lesmethodieken die vallen onder de noemer “onderzoekend leren”. Daarvan worden twee voorbeelden hieronder behandeld, namelijk Learning by Design™ van J. Kolodner et al en Design-Based Science van D. Fortus et al. Vervolgens wordt de werkwijze van Scrum uitgelegd. Er wordt afgesloten met een beknopte weergave van het te onderzoeken vakconcept: “programmeerconcepten”.

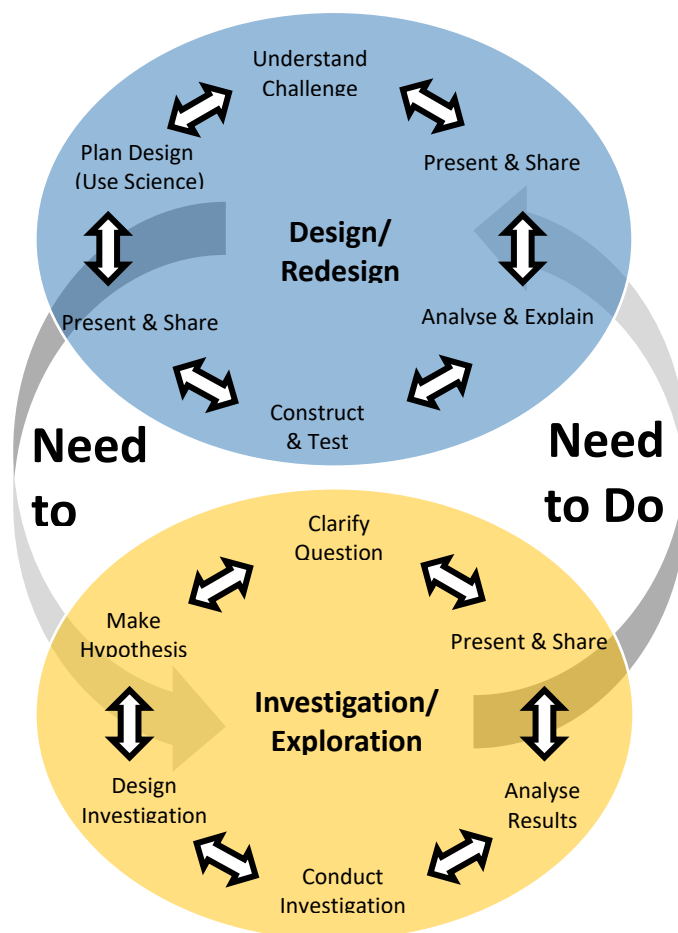
2.1. Onderzoekend leren

Traditioneel onderwijs wordt gezien als academisch van aard, en dien ten gevolge niet aansluitend op de weerbarstige werkelijkheid, aangezien daarin geen “beste oplossing” bestaat en er vaak informatie mist. Hierdoor krijgen leerlingen te weinig ervaring om hun kennis in de praktijk te kunnen toepassen (Fortus, Dersheimer, Krajcik, Marx, & Mamlok-Naaman, 2004; Kolodner et al., 2003). Naar aanleiding van deze conclusie zijn er verschillende onderwijsmodellen ontwikkeld waarin leerlingen de leerstof in een bepaalde context toepassen. Twee voorbeelden hiervan zijn Learning by Design™ en Design-Based Science. Beiden zullen hieronder worden uitgelegd, omdat aspecten hiervan toegepast zijn bij het opstellen van de ontwerprichtlijnen. Deze twee zijn geselecteerd, omdat ze erop gericht zijn om leerlingen *tijdens het ontwerpen* wetenschappelijke kennis op te laten doen. Veel andere programma’s zijn ontworpen om deze twee zaken elkaar op te laten volgen, hetzij door te beginnen met het maken van een ontwerp, hetzij door te beginnen met het leren van de benodigde theorie (Fortus et al., 2004).

2.1.1. Learning by Design™

Learning by Design™ (LBD) is een project-gebaseerde onderzoeksaanpak voor de exacte vakken (Kolodner et al., 2003). De wortels van deze onderwijsmethode liggen bij de constructivistische leerstrategieën “case-based reasoning” (CBR) (Kolodner, 1992) en “problem-based learning” (PBL) (Barrows, 1985). Het doel van LBD is om een leeromgeving te scheppen die geschikt is voor het leren kennen van wetenschappelijke concepten en vaardigheden en hun toepasbaarheid, waarbij ondertussen ook vaardigheden worden opgedaan op cognitief, sociaal, wetenschappelijk en communicatief gebied.

De basis van LBD is het herontwerpen en verbeteren van een basis-ontwerp. De leerlingen beginnen met het volgen van een stappenplan om een werkend product te maken. Vervolgens gaan ze onderzoeken hoe het ontwerp verbeterd kan worden, waarbij ze vuistregels proberen te ontdekken. Aan de hand van



Figuur 1 - schematische weergave van de werkwijze van LBD

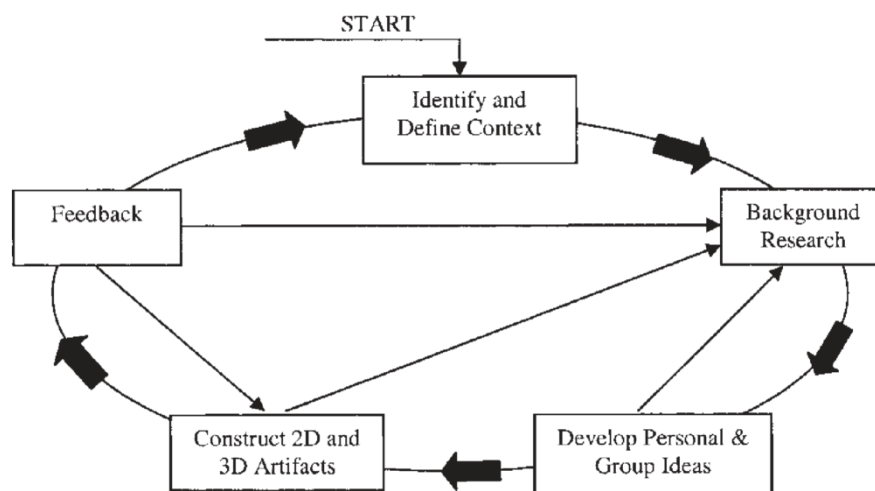
deze vuistregels wordt het ontwerp aangepast. Daarna wordt het weer getest om de cyclus opnieuw te beginnen met vuistregels opschrijven.

Het iteratieve proces van ontwerpen, testen en aanpassen, wordt aangevuld door een iteratief proces van onderzoeken. Deze twee processen zijn hiernaast weergegeven als twee cycli waartussen gewisseld wordt. De iteraties zijn steeds bedoeld om een beter begrip te krijgen van de leerstof en om steeds meer op een wetenschappelijke manier te werken. Dat houdt in dat het hoofddoel van een module dus niet “het best mogelijke ontwerp” is, zoals door leerlingen en docenten vaak wel werd gedacht.

Kolodner et al. (2003) ontdekte dat het gebruik van rituelen en manieren om te herinneren wat de volgende stap is, ervoor zorgt dat zowel de docent als de leerlingen makkelijker hun draai vinden in het LBD-proces. Het effect was dat de stappen die gezet moesten worden als relevanter werden ervaren. Ook bleef er meer tijd over voor leerlingen en docent om bezig te zijn met de belangrijke zaken: experimenteren en vuistregels opstellen. Er worden twee soorten rituelen onderscheiden: groepjes-rituelen en klassenrituelen. Het draait er bij beiden om dat het snel duidelijk is wat er precies gedaan moet worden. Dat wordt bereikt door bijvoorbeeld een schema te maken van de activiteiten, zoals in Figuur 1 te zien is. Daar is bijvoorbeeld te zien welke sub-activiteiten er horen bij het groepjesritueel van het ontwerpen van een experiment (Design). Andere rituelen worden gestuurd door middel van een stappenplan op een stencil dat ingevuld moet worden. Een voorbeeld van een klassenritueel is het opschrijven van vuistregels op het bord en erover discussiëren als resultaten van een experiment van een groepje er niet mee overeenkomen.

2.1.2. Design-Based Science

Design-Based Science is een onderwijskundige methode waarin het doel om een artefact te ontwerpen de context schept voor alle onderwijsactiviteiten (Fortus et al., 2004). Het doel van deze aanpak is “wetenschappelijke kennis en probleemoplossend vermogen te kweken door artefacten te laten ontwerpen”. Daarbij vermelden Fortus et al. nadrukkelijk dat het hierbij niet gaat om het toepassen van traditioneel aangeleerde theorieën. Ontwerpen wordt gezien als het vehikel voor het opdoen van nieuwe wetenschappelijke kennis.



Figuur 2 - schematische weergave van de werkwijze van DBS

Er wordt bij Design-Based Science gewerkt volgens een onderwijscyclus. Deze wordt op een poster opgehangen zodat iedere les de leerlingen kunnen zien hoe het ontwerpproces verloopt. De docent legt iedere les ook uit waar de activiteiten die uitgevoerd worden thuishoren in de cyclus. Alle onderdelen zullen hieronder behandeld worden.

2.1.2.1. Context identificeren en definiëren

Contexten kunnen leerlingen helpen om abstracte begrippen te begrijpen. Meerdere contexten worden gebruikt, omdat de kans dan groter is dat leerlingen de opgedane kennis toe kunnen passen in nieuwe contexten.

2.1.2.2. Achtergrondonderzoek

Het doel van achtergrondonderzoek is om leerlingen benodigde kennis op te laten doen voor het project. Hiervoor kunnen meerdere vormen van onderzoek gebruikt worden, waaronder het bestuderen van aangeboden lesmateriaal en het raadplegen van zelf gevonden bronnen. De docent heeft hierbij vanzelfsprekend een controlerende functie.

2.1.2.3. Ideeën ontwikkelen

Op basis van de opgedane basiskennis moeten de leerlingen individueel een plan maken voor een artefact of een verbetering van een artefact. In groepjes van vier personen worden deze plannen naast elkaar gelegd en wordt er beslist over wat er uiteindelijk gedaan zal worden. Ook wordt vastgesteld op welke manier het artefact getest zal worden.

2.1.2.4. Artefact maken

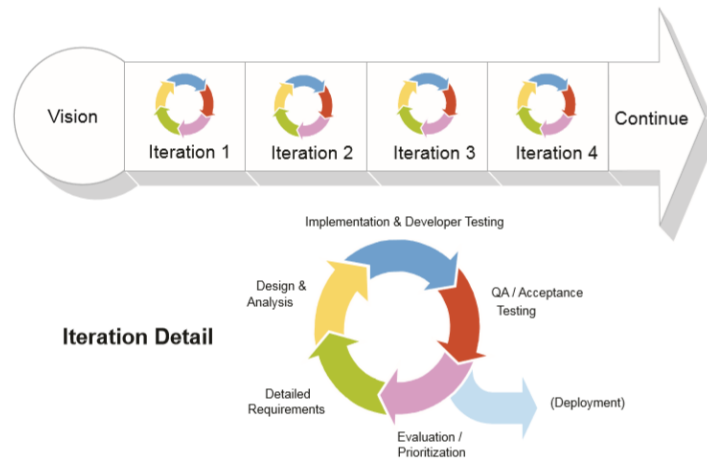
Het uitvoeren van het opgestelde plan gebeurt in tweetallen. Op deze manier wordt per groepje van vier leerlingen gewerkt aan twee varianten van een artefact. De resultaten hiervan worden getest en bediscussieerd. Het beste wordt gekozen door het groepje als het nieuwe “groepsresultaat”.

2.1.2.5. Feedback, formatieve toetsing

Van alle gemaakte keuzes moet uiteindelijk een uitleg gegeven worden aan medeleerlingen en de docent. Beiden kunnen hierop feedback geven. Op deze manier wordt opgedane kennis gedeeld met medeleerlingen en heeft de docent de mogelijkheid om de conceptuele ontwikkeling te meten.

2.2. Scrum

Scrum is een ontwikkelmethode waarin zelforganiserende teams incrementeel een product ontwikkelen. Het is een framework dat bestaat uit verschillende rollen, bijeenkomsten en artefacten (benodigde onderdelen). Een project wordt begonnen met een globaal geformuleerd doel (visie). Daaruit vloeit een initiële lijst met eisen. Als die is opgesteld, wordt zo snel mogelijk begonnen met het ontwikkelen van een minimale versie van het beoogde resultaat. Dat wordt gedaan in iteraties van twee weken of 30 dagen die “Sprints” worden genoemd (Figuur 3). In Scrum is niet vastgelegd op welke manier dat precies moet gebeuren. Het team is zelf verantwoordelijk voor het vormgeven van de samenwerking binnen het gegeven framework. De genoemde basisonderdelen van Scrum worden hieronder behandeld, te beginnen bij de rollen. Daarna volgen de bijeenkomsten en tot slot worden de gebruikte artefacten beschreven.



Figuur 3 - Opdeling van het Scrum-ontwikkelp proces in Sprints (James, 2010)

2.2.1. Rollen

Het contact met de opdrachtgever(s) en andere betrokkenen verloopt altijd via de **Product Owner**. Deze persoon is de eindverantwoordelijke voor de kwaliteit van het product. Het **Team** is zelfsturend en bestaat uit 5-9 leden met verschillende expertises. Een goed verloop van het Scrum-proces wordt bewaakt door de **Scrum Master** en zorgt ervoor dat alle bijbehorende artefacten zichtbaar zijn. Deze persoon bewaakt de tijd op bijeenkomsten en moedigt goede werkmethoden aan.

2.2.2. Artefacten

Om het werk dat gedaan wordt overzichtelijk weer te geven, wordt gebruikgemaakt van verschillende lijsten. De lijst waarop toekomstige functionaliteit op bijgehouden wordt, heet de **Product Backlog**. Alles op deze lijst mag bediscussieerd of aangevuld worden door alle belanghebbenden en het team. De Product Owner zorgt ervoor dat de lijst gesorteerd is op prioriteit en dat bovenaan alleen eisen staan die duidelijk en behapbaar zijn voor het team. Deze lijst bestaat uit **Product Backlog Items (PBI's)**. Die beschrijven een eis voor het product vanuit het oogpunt van de eindgebruiker. Daarom worden ze vaak geschreven in de vorm van een **User Story**. Daarbij gaat het expliciet nog niet om hoe het gerealiseerd kan worden.

Committed Backlog Items	Tasks Not Started	Tasks In Progress	Tasks Completed

Figuur 4 - Sprint Backlog met PBI's en ST's

Aan het begin van een Sprint wordt bepaald welke PBI's geïmplementeerd zullen worden. Deze worden verplaatst van de Product Backlog naar de **Sprint Backlog**. Deze worden vertaald naar concrete taken: **Sprint Tasks (ST's)**. Van deze taken wordt bijgehouden of er nog mee begonnen moet worden, dat er al aan gewerkt wordt of dat ze klaar zijn. Sprint Tasks gaan over alle taken die gedaan moeten worden om de nieuwe functionaliteit te realiseren, van brainstormen en ontwerpen tot testen en documenteren.

2.2.3. Bijeenkomsten

Er zijn vijf verschillende bijeenkomsten bij Scrum, waarvan er vier op vaste momenten in een Sprint plaatsvinden. Die zullen we eerst in chronologische volgorde behandelen, gevolgd door de vijfde bijeenkomst. Een Sprint begint altijd met de **Sprint Planning**, waarvoor afhankelijk van de Sprintlengte 4 tot 8 uur gerekend wordt. Tijdens die bijeenkomst worden er PBI's gekozen om uit te voeren en vertaald in ST's. Deze lijst hoeft niet volledig te zijn en kan in overleg op ieder moment tijdens de Sprint aangevuld of aangepast worden. Deze lijst is het uitgangspunt van wat er op een dag gedaan wordt door het team.

Iedere dag begint het team daarom bij de Sprint Backlog met de **Daily Scrum**. Dat is een bijeenkomst van 15 minuten waarin ieder teamlid drie vragen beantwoordt:

1. Wat heb ik sinds de vorige keer gedaan?
2. Wat ga ik vandaag doen?
3. Welke belemmeringen ervaar ik?

Zo wordt transparantie in het team gecreëerd en kunnen teamleden elkaar beter ondersteunen.

Aan het einde van iedere sprint wordt er tijdens de **Sprint Review** een demonstratie gegeven aan alle belanghebbenden. De Product Owner bepaalt op dit moment welke PBI's voldoende zijn afgerond. De Scrum Master helpt bij het formuleren van nieuwe PBI's aan de hand van de feedback van de belanghebbenden en de Product Owner.

Direct daarna wordt de **Sprint Retrospective** gehouden om terug te kijken op de samenwerking. Bij deze bijeenkomst zijn geen externen aanwezig, zodat het team de problemen die er zijn in een zo veilig mogelijke omgeving kan bespreken en proberen op te lossen.

De PBI's worden concreter beschreven tijdens de **Backlog Refinement**. Verder wordt overlegd met het team hoeveel werk ze ongeveer zijn (op schaal S-M-L-XL). Ook worden te grote PBI's opgesplitst, zodat er een genuanceerdere prioriteit gelegd kan worden op functionaliteit met de hoogste business value.

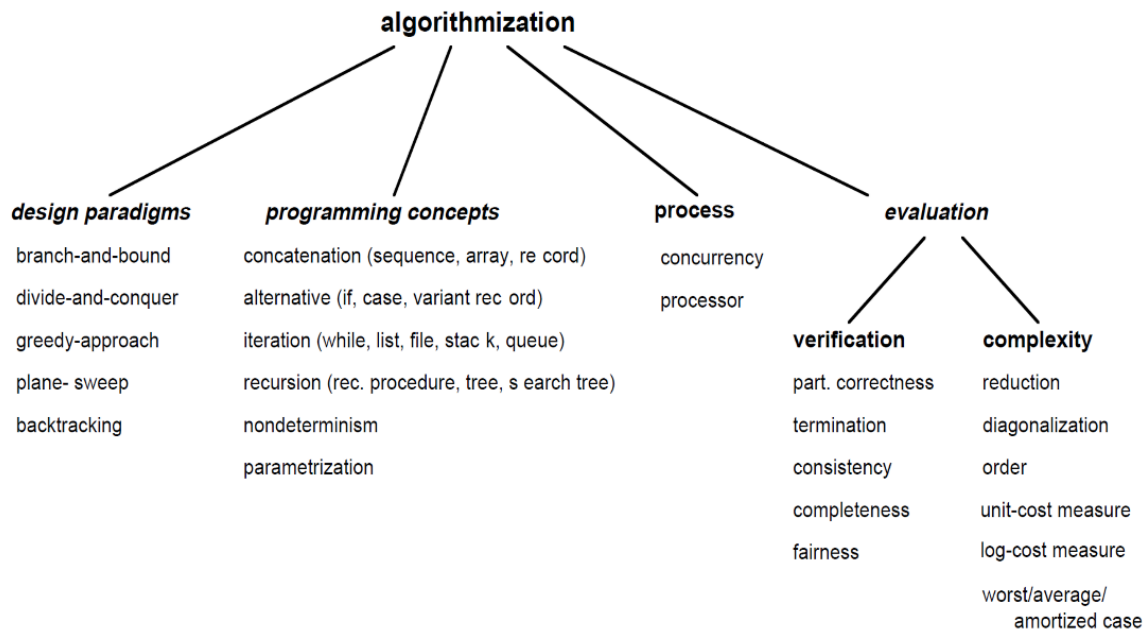
2.3. Leren programmeren

Het is algemeen bekend dat leren programmeren als erg moeilijk wordt ervaren door zowel leerlingen als hun docenten (Tritrakan et al., 2016). De complexiteit ligt erin dat het niet voldoende is om kennis van de concepten op te doen, maar dat er vaardigheden getraind moeten worden om een probleem te analyseren en op te lossen (Tritrakan et al., 2016). Daar is vooral veel tijd voor nodig. Het duurt ongeveer 10 jaar om een expert-programmeur te worden (Winslow, 1996). Winslow concludeerde in hetzelfde onderzoek dat beginners enkel beschikken over oppervlakkige kennis, gebrekkige mentale modellen, relevante kennis niet kunnen toepassen en "regel voor regel" programmeren in plaats van in "brokjes" of structuren. Het is dus goed om niet te hoge eisen te stellen aan de opbrengst van programmeerlessen in het voortgezet onderwijs.

In deze paragraaf wordt beschreven welke kennis en vaardigheden aandacht moeten krijgen in een programmeermodule. Daarvoor zullen we eerst stilstaan bij programmeerconcepten en programmeertaalconstructies die daarbij centraal staan. Daarna zullen we de cruciale vaardigheden om te kunnen programmeren behandelen. Ten slotte zullen we zien op welke manier programmeerlessen het beste vormgegeven kunnen worden in de context van het ontwerponderwijs.

2.3.1. Concepten

De concepten voor het informatica-vakgebied zijn door Schwill in kaart gebracht. Hij heeft deze ingedeeld in drie categorieën: algoritmiek, gestructureerde ontleding en taal (Schwill, 1994). Deze categorieën heeft hij onderverdeeld in meerdere subcategorieën. In het uitgevoerde onderzoek heb ik me beperkt tot algoritmiek. Deze categorie is door Schwill opgedeeld in vier subcategorieën: ontwerp-paradigma's, programmeerconcepten, processen en evaluatie. Een schematische weergave



Figuur 5 - algoritmiek: concepten in subcategorieën

hiervan staat in Figuur 5.

Deze concepten komen indirect ook terug in het nieuwe examenprogramma voor informatica dat in 2019 ingevoerd is. Daarin speelt programmeren een belangrijke rol (Schmidt, 2018a). Dat wordt het meest expliciet benoemd in domein D (*Examenprogramma informatica havo/vwo*, 2016):

De kandidaat kan, voor een gegeven doelstelling, programma-componenten ontwikkelen in een imperatieve programmeertaal, daarbij programmeertaalconstructies gebruiken die abstractie ondersteunen, en programma-componenten zodanig structureren dat ze door anderen gemakkelijk te begrijpen en te evalueren zijn.

In de handreiking die door SLO is opgesteld bij het nieuwe examenprogramma, wordt dit concreet uitgewerkt in vier programmeertaalconstructies: (Schmidt, 2018b)

- aanduidingen voor data-objecten (variabelen en constanten) en toewijzing om waarden aan variabelen toe te kennen
- zogenaamde controlestructuren: taalconstructies voor
 - opeenvolging
 - keuze
 - herhaling
- procedures (methoden, subroutines, ...) voor stukken programma die iets doen
- functies, al dan niet met parameters, voor stukken programma die een waarde opleveren

2.3.2. Vaardigheden

De vaardigheden die nodig zijn om te programmeren, worden door Nuutila et al. samengevat in twee termen: probleemoplossing en ontwerpen. Deze twee activiteiten wisselen elkaar constant af en zijn bij het programmeren met elkaar verbonden op een manier die in andere vakgebieden niet voorkomt (Nuutila, Törmä, Kinnunen, & Malmi, 2008).

Concreet moeten leerlingen gebruik kunnen maken van een ontwikkelomgeving (integrated development environment: IDE) en op een incrementele manier een programma ontwikkelen (Bennedsen & Caspersen, 2008). De stappen die daarbij achtereenvolgens gezet moeten worden, zijn: specificeren, implementeren, testen en fouten opsporen (Nuutila et al., 2008).

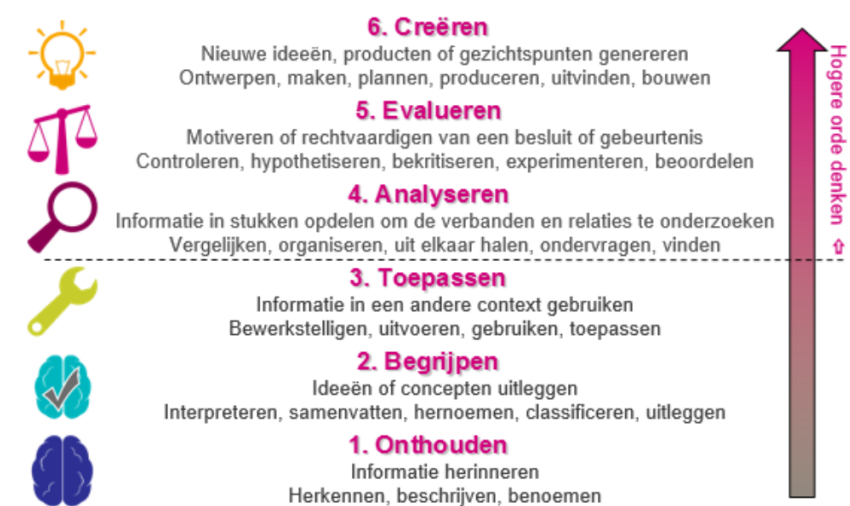
2.3.3. Ontwerponderwijs

Het ontwerponderwijs leent zich ervoor om concepten in verschillende toepassingsvelden te leren (Fortus et al., 2004). Een veelgebruikte manier om ontwerponderwijs vorm te geven, is Problem Based Learning (PBL). Learning by Design is daar bijvoorbeeld ook op gebaseerd. Nuutila et al. geven aan dat studenten dankzij Problem Based Learning (PBL) vroeg in de door hen ontwikkelde cursus in staat waren om een conceptueel model op te stellen van een programma (Nuutila et al., 2008). Zij maakten in hun cursus gebruik van tutoren, die primair de rol van *domein-expert* moest vervullen. Dat moest niet alleen passief, maar soms ook actief. Als een tutor merkte dat studenten de mist ingingen, moest hij vragen stellen zoals “Hoe weet je dat ...?”, “Kan je ook andere factoren noemen naast ...?” of “Wat zijn de oorzaken/onderdelen/eigenschappen van ...?”. Zulke vragen bleken voldoende te zijn om studenten weer goed op weg te helpen. Daarnaast moest de tutor ook het groepsproces faciliteren. Hij moet volgens hen echter niet te actief zijn, omdat de studenten dan weer het gevoel zouden krijgen dat hij verantwoordelijk is voor hun leerproces.

2.4. Taxonomie

2.4.1. Bloom

Om conceptuele vorming van leerlingen te noteren, wordt vaak gebruikgemaakt van de taxonomie van Bloom (Ministerie van OCW, 2017). Daarin wordt onderscheid gemaakt tussen zes verschillende denkniveaus: Onthouden, Begrijpen, Toepassen, Analyseren, Evalueren en Creëren. De eerste drie gaan over het verwerken van direct aangeboden informatie en worden gerekend tot het “lagere orde denken”. De andere drie worden gerekend tot het “hogere orde denken”.



Figuur 6 - taxonomie van Bloom (Ministerie van OCW, 2017).

2.4.2. SOLO

Uit onderzoek is echter gebleken dat de taxonomie van Bloom onvoldoende inzicht geeft in de mate waarin leerlingen kunnen programmeren (Koldenhof, Jeurig, & Ruth, 2011). Daarvoor is de volgende redenering gevolgd:

Leren programmeren gebeurt in een context, bestaande uit voorbeelden, toepassingen, en opgaven. Het eerste niveau in de Bloom-taxonomie kan niet bereikt worden zonder een context waarin vaardigheden uit de hogere Bloom-niveaus gebruikt worden. Bij het bestuderen van het leren programmeren is het daarom belangrijk te bepalen waar en hoe de context (voorbeelden, toepassingen, opgaven) terugkomt. De 4C/ID-methode en de SOLO-taxonomie besteden expliciet aandacht aan de context van het leren.

Koldenhof et al. stellen in hetzelfde onderzoek dat de SOLO-taxonomie beter toepasbaar is dan Blooms taxonomie voor het beoordelen van het niveau waarop een leerling kan programmeren. Deze taxonomie bestaat uit vijf categorieën die de complexiteit aangeven van de context waarin de opdracht gedaan moet worden:

1. Prestructural: een eenvoudige, over-gesimplificeerde benadering waarbij alleen feitelijke informatie voldoet.
2. Unistructural: een enkelvoudige benadering waarbij hoofdzakelijk één onderdeel of aspect gebruikt of benadrukt wordt.
3. Multistructural: een meervoudige benadering waarbij meerdere onderdelen of aspecten gebruikt of herkend moeten worden. Deze hoeven nog niet aan elkaar gerelateerd te worden en het is niet nodig om het totaalplaatje te zien.
4. Relational: een relationele, holistische benadering waarbij onderliggende verbanden gezien worden. De relevantie van verschillende onderdelen ten opzichte van het geheel wordt doorzien en gewaardeerd.
5. Extended abstract: een gegeneraliseerde benadering waarbij de context wordt gezien als een voorbeeld van een generiek idee.

2.4.3. Tweedimensionale taxonomie: SOLO × Bloom

Meerbaum-Salant et al. concluderen zelfs dat er een nieuwe taxonomie nodig was, omdat programmeeropdrachten moeilijk te categoriseren waren in de bestaande taxonomieën (Meerbaum-Salant, Armoni, & Ben-Ari, 2013). Daarom hebben ze zelf een taxonomie opgesteld met twee dimensies. De ene dimensie gaat over de complexiteit van de context van de opdracht (de programmacode). Hiervoor zijn drie categorieën uit de SOLO-taxonomie gebruikt: eenvoudig, meervoudig en relationeel. Deze categorieën worden onderverdeeld in drie stukken op basis van Bloom: begrijpen, toepassen en creëren. Deze hebben ze specifiek in de context van programmeren als volgt uitgelegd:

- Begrijpen: het vermogen om concepten en programmeertaalconstructies samen te vatten, uit te leggen, toe te lichten, te classificeren en te vergelijken
- Toepassen: het vermogen om programma's en algoritmes uit te voeren, te traceren en hun doel te herkennen.
- Creëren: het vermogen om programma's en algoritmes te plannen en te produceren.

In het onderzoek dat Meerbaum-Salant et al. gedaan hebben, bleek deze taxonomie niet alle opdrachten waterdicht te kunnen ordenen qua complexiteit. Soms scoorden leerlingen beter voor opdrachten in hogere categorieën dan dat ze scoorden in lagere categorieën. Ze vermoedden dat dat te maken kan hebben met het soort opdracht dat de leerling moest doen en welke concepten en programmeertaalconstructies betrokken waren bij de opdracht. Verder gaven ze ook aan dat het niveau op de SOLO-schaal meer invloed heeft op de complexiteit van de opdracht.

3. Onderzoeksvragen

Het doel van dit onderzoek was om te ontdekken hoe Scrum gebruikt kan worden in de klas, zodat de conceptuele ontwikkeling van leerlingen tijdens hun bezigheden waargenomen kan worden. Specifieker is er gezocht naar een antwoord op de volgende vraag:

Hoe kan Scrum aangepast worden tot raamwerk voor een lessenserie om conceptuele vorming van individuele leerlingen waar te kunnen nemen tijdens het uitvoeren van een authentieke ontwerpopdracht voor informatica in de bovenbouw van het vwo?

Om deze vraag te beantwoorden, is er een antwoord gezocht op drie deelvragen:

1. *Aan welke kenmerken moet een lesmodule voldoen om leerlingen tijdens een ontwerpopdracht waarneembaar conceptuele kennis op te laten doen?*
 - a. *Hoe kunnen leerlingen tijdens een ontwerpopdracht conceptuele kennis opdoen?*
 - b. *Hoe kan conceptuele kennis van individuele leerlingen zichtbaar gemaakt worden op een manier die naadloos aansluit bij het uitvoeren van de ontwerpopdracht?*
2. *Hoe kan Scrum aangepast worden tot organisatorisch raamwerk voor een lesmodule die voldoet aan de gevonden kenmerken?*
 - a. *Hoe kan Scrum vertaald worden naar verschillende lestypen, rekening houdend met de gevonden kenmerken?*
 - b. *Hoe kan de docent tijdens de lessen de individuele conceptuele kennis van leerlingen structureel en overzichtelijk bijhouden?*
 - c. *Hoe kunnen de gevonden kenmerken toegepast worden in een lessenserie binnen het raamwerk?*
3. *In welke mate biedt de ontwikkelde Scrum-variant de mogelijkheid om conceptuele kennis van individuele leerlingen waar te nemen tijdens een ontwerpopdracht waarbij robots geprogrammeerd moeten worden.*
 - a. *Op welke momenten kan de docent conceptuele kennis bij individuele leerlingen waarnemen?*
 - b. *In welke mate ondersteunen de ontwikkelde materialen het waarnemen van conceptuele kennis van individuele leerlingen?*

Deze drie deelvragen heb ik achtereenvolgens beantwoord. Het onderzoek heb ik ingedeeld in drie fasen: voor iedere deelvraag een fase. In de *eerste fase* heb ik gezocht naar kenmerken van twee raamwerken voor ontwerponderwijs die als doel hebben om leerlingen conceptuele kennis op te laten doen tijdens het uitvoeren van een ontwerpopdracht. Dit betreft Learning by Design™ van Kolodner et al. (2003) en Design-Based Science van Fortus et al. (2004).

In de *tweede fase* heb ik een raamwerk voor een lessenserie ontwikkeld op basis van Scrum met behulp van de gevonden kenmerken. Daarvoor heb ik een reader geschreven. Daarbij heb ik zo min mogelijk willen afwijken van de principes die door de bedenkers daarvan zijn opgesteld in het Agile Manifesto (Schwaber & Beedle, 2002). Bij alle aanpassingen en toevoegingen is steeds in de gaten gehouden dat het op een natuurlijke (authentieke) manier in het ontwerpproces paste. Traditionele toetsing was daarom bijvoorbeeld geen optie. Verder heb ik ook een sjabloon ontwikkeld waarmee een docent conceptuele vorming kan waarnemen terwijl leerlingen een ontwerpopdracht uitvoeren. Hier heb ik ook een concrete toepassing van gemaakt om in het ontwikkelde raamwerk te kunnen testen. Daarvoor heb ik een opdrachtbeschrijving gemaakt en het observatie-sjabloon ingevuld met programmeerconcepten.

In de *derde fase* heb ik een lessenserie uitgevoerd met het in fase 2 ontwikkelde materiaal. Deze lessenserie heb ik uitgevoerd om te testen of leerlingen op deze manier inderdaad conceptuele kennis op konden doen waarbij de docent dat ondertussen ook waar kan nemen.

4. Methode

In dit hoofdstuk beschrijf ik hoe ik het onderzoek opgezet en uitgevoerd heb. Daarvoor beschrijf ik eerst de procedure. Daarin leg ik uit in welke drie fasen ik het onderzoek aangepakt heb. Daarna geef ik de context weer waarin ik het onderzoek heb uitgevoerd. Vervolgens leg ik uit waarop ik heb gelet tijdens dit onderzoek. Ten slotte beschrijf ik de manier waarop ik de gevonden resultaten heb geanalyseerd.

4.1. Procedure

Ik heb het onderzoek in drie fasen opgedeeld. Deze drie fasen zijn direct afgeleid van de drie deelvragen van dit onderzoek. Ik beschrijf ze nog eens kort. In de eerste fase heb ik gezocht naar kenmerken waaraan een lessenserie in het ontwerponderwijs moet voldoen. Daarbij heb ik er vooral op gelet hoe conceptuele vorming van individuele leerlingen kan blijken. In de tweede fase heb ik deze gebruikt bij het ontwikkelen van materiaal voor een raamwerk op basis van Scrum voor een lessenserie. Om de opdracht zo authentiek mogelijk te laten blijven, heb ik Scrum zo veel mogelijk intact gehouden. In de derde fase heb ik een concrete opdracht binnen dit raamwerk geschreven en deze getest, om te bepalen of het raamwerk inderdaad geschikt was om een ontwerpopdracht aan leerlingen te geven terwijl de docent tijd heeft om individuele conceptualisatie waar te nemen bij de leerlingen.

4.1.1. Fase 1: Kenmerken beschrijven (deelvraag 1)

Ik ben begonnen met een literatuuronderzoek om te ontdekken welke kenmerken een lessenserie heeft waarin leerlingen conceptuele kennis opdoen door het uitvoeren van een ontwerpopdracht. Hiervoor heb ik gebruikgemaakt van papers die ik heb gekregen van een onderzoeker die meewerkt aan het NRO-onderzoek waarbij dit onderzoek aansluit. Verder heb ik gezocht naar literatuur in de databases van ERIC en PsycINFO. De relevante informatie die daaruit voortkwam heeft u kunnen lezen in het theoretisch kader. Hieruit heb ik kenmerken afgeleid. Deze kunt u vinden in het hoofdstuk "Resultaten". Daarbij heb ik gelet op twee aspecten: (1) hoe leerlingen kennis opdoen en (2) hoe de opgedane kennis zichtbaar kan worden voor de docent.

4.1.2. Fase 2: Raamwerk ontwikkelen (deelvraag 2)

Aan de hand van de ontwikkelmethode Scrum heb ik beschreven hoe leerlingen een ontwerpopdracht uit moeten voeren. Daarbij heb ik erop gelet dat de manier waarop de leerlingen werken aansluit bij de kenmerken die ik in fase 1 gevonden heb. Zoals gezegd wilde ik zo min mogelijk afwijken van de manier waarop Scrum in het bedrijfsleven wordt gebruikt. Daarom heb ik de vaste bijeenkomsten van Scrum zo waarheidsgetrouw mogelijk vertaald naar verschillende lestypen. Daarbij heb ik erop gelet wanneer de gevonden kenmerken van toepassing waren in het ontwerpproces. Zo ben ik gekomen tot een uitleg van Scrum voor leerlingen en de toepassing ervan in een lessenserie.

Op basis van de literatuur vermoedde ik dat tijdens een lessenserie waarin leerlingen op die manier zouden werken, de docent genoeg tijd zou hebben om conceptualisatie te meten. Daarom heb ik beschreven welke manier mij daarvoor het meest geschikt leek. Dit heeft een sjabloon opgeleverd dat ingevuld kan worden met de concepten die geleerd moeten worden tijdens de ontwerpopdracht. Het doel hiervan is om docenten houvast te geven en een middel te bieden om conceptualisatie consistent te meten.

Ten slotte heb ik in deze fase ook een concrete toepassing van het raamwerk gemaakt, zodat ik het kon evalueren door een try-out in mijn eigen lespraktijk. Daarvoor heb ik een ontwerpopdracht geschreven die de leerlingen uit moesten voeren volgens de ontwikkelde Scrumvariant. Op het opdrachtblad heb ik vooral duidelijk gemaakt wat de context was van de opdracht, zodat leerlingen

deze als authentiek zouden ervaren. Voor de docent heb ik het sjabloon ingevuld met voorbeeldvragen naar de concepten die leerlingen nodig hadden bij het uitvoeren van de opdracht.

4.1.3. Fase 3: Raamwerk evalueren (deelvraag 3)

Ik heb als docent en onderzoeker de lessenserie zelf uitgevoerd. Deze begon ik met een uitleg en had ik verder opgedeeld in drie sprints. Terwijl leerlingen de opdracht uitvoerden, heb ik gemeten van welke conceptuele kennis ze blijk gaven. Ondertussen heb ik opvallendheden genoteerd ten behoeve van de evaluatie van de werkvorm. Met enige regelmaat heb ik bijvoorbeeld genoteerd hoe leerlingen bezig waren, elkaar hielpen en hoe ze reageerden op de vragen die ik ze stelde. Zo heb ik het ontwikkelde raamwerk in de praktijk geëvalueerd.

4.2. Context

Het onderzoek heb ik in mijn eigen lessen uitgevoerd onder 10 jongens en 1 meisje in 4 vwo. De leerlingen waren allemaal tussen de 14 en 16 jaar oud. Deze groep heb ik ingedeeld in twee teams van 5 en 6 personen. Deze teams werkten intern in tweetallen, waarbij het team van 5 personen werkte in een tweetal en een drietal.

De informatica-kennis van de leerlingen was nog erg beperkt, aangezien het hun eerste jaar was dat ze het vak volgden. Drie van de leerlingen gaven wel aan al enige ervaring met websites maken te hebben. Een andere leerling had naar eigen zeggen al eens zelf een computer gebouwd. Voorafgaand aan het onderzoek had ik alleen een module over hardware gegeven, waarin onder andere behandeld was hoe binair rekenen en logische schakelingen werkten. Leerlingen hadden dus nog niet eerder geprogrammeerd.

De lessenserie duurde 21 lessen van 45 minuten. De lessen zijn gegeven in een computerlokaal met een smartboard en 32 leerling-pc's. Per tweetal was er een LEGO Mindstorms EV3 Smart Brick beschikbaar, met alle sensoren uit de LEGO Mindstorms NXT Education set. Ook waren er enkele tientallen dozen met LEGO-onderdelen beschikbaar om de robots mee te bouwen. Grotendeels bestond deze verzameling uit onderdelen uit de Education sets. Hiermee waren in een eerdere lessenserie door een andere groep leerlingen al robots gebouwd. Ik heb de leerlingen toegestaan om daar gebruik van te maken.

Verder heb ik mijn leerlingen ook digitaal materiaal aangereikt. Dit betreft uitleg van de gebruikte programmeertaal en -omgeving en een manier om de eisen en taken in bij te houden. Als programmeertaal en -omgeving heb ik gekozen voor het door Microsoft ontwikkelde Small Basic (www.smallbasic.com). Op de bijbehorende site is een Engelse handleiding te vinden waarin de IDE en de taalconstructies uitgelegd worden. Deze IDE is niet specifiek bedoeld om Mindstorms-robots programmeren, maar om eenvoudige programma's te schrijven in een opdrachtprompt- of grafische omgeving. Om Mindstorms te kunnen programmeren, is de extensie EV3Basic (www.ev3basic.com) nodig. Op de daarbij behorende site is de documentatie van alle beschikbare functies te vinden, aangevuld met voorbeeldprogramma's. Op die site wordt voor uitleg van de programmeerconcepten en -taalconstructies verwezen naar de site van Small Basic.

Om de leerlingen eisen en taken bij te laten houden, heb ik gekozen voor de WebApp Trello (trello.com). Op die site is na het aanmaken van een account op een eenvoudige manier bij te houden welke eisen er gesteld worden aan het te ontwikkelen artefact en welke taken daarvoor concreet uitgevoerd moeten worden. Daarbij kan ook aangegeven worden wie welke taak moet doen en of er al aan een taak gewerkt wordt of dat deze al afgerond is.

4.3. Instrumenten

In de eerste fase van het onderzoek ben ik uitgegaan van de theorie. Die heb ik in de tweede fase gebruikt om Scrum toe te passen als basis voor een lessenserie. In de derde fase heb ik zelf metingen gedaan. De instrumenten die ik daarvoor gebruikt heb, beschrijf ik in deze paragraaf.

Ik wilde bepalen in welke mate leerlingen waarneembaar conceptuele kennis opdeden tijdens een lessenserie die ingericht was volgens mijn ontwikkelde raamwerk. Dit onderzoeksdoel komt overeen met het doel dat ik had met het raamwerk voor een lessenserie dat ik in fase 2 ontwikkelde. Dat moest de docent namelijk de mogelijkheid geven om bij te houden welke conceptuele kennis de leerlingen opdeden. Daarom heb ik geen extra instrument ontwikkeld naast het sjabloon dat ik in fase 2 heb ontwikkeld en in fase 3 heb ingevuld. Dit instrument leek mij voldoende om te kunnen bepalen of de docent conceptuele kennis waar kon nemen.

Het gebruikte instrument beschrijf ik nog uitgebreider in het hoofdstuk "Resultaten". Voor nu beperk ik mij tot de kern van het instrument. Die bestaat uit een aantal vragen die de docent in de les aan zijn leerlingen kan stellen om erachter te komen in hoeverre ze de programmeerconcepten beheersen. Vervolgens kan hij op het instrument zelf noteren wat hij concludeert uit de reactie van de leerlingen.

4.4. Analyse

Na het testen van het materiaal heb ik de verzamelde gegevens geanalyseerd. Dit heb ik gedaan aan de hand van het aantal waargenomen concepten en de manier waarop de leerlingen in de les werkten. Op basis hiervan heb ik aandachts- en verbeterpunten voorgesteld voor het ontwikkelde lesmateriaal, waarmee het beter mogelijk zou moeten worden om conceptuele ontwikkeling waar te nemen.

5. Resultaten

In dit hoofdstuk is voor iedere fase te lezen hoe die verlopen is en wat de resultaten waren. In de eerste fase heb ik een literatuuronderzoek gedaan naar kenmerken voor een lessenserie-raamwerk dat het mogelijk maakt om individuele conceptuele vorming waar te nemen. In de daaropvolgende fase heb ik materiaal ontwikkeld dat aan de gevonden kenmerken voldoet. In de laatste fase heb ik het ontwikkelde materiaal getest in de lespraktijk om te valideren of het raamwerk het inderdaad mogelijk maakt om individuele conceptuele vorming waar te nemen.

5.1. Fase 1: Kenmerken beschrijven (deelvraag 1)

Om kenmerken te beschrijven heb ik gezocht naar nuttige papers in de databases van ERIC en PsycInfo. Daarbij heb ik geen papers gevonden die direct betrekking hadden op projectgebaseerd informatica-onderwijs. Daarom heb ik ervoor gekozen om papers te gebruiken die projectgebaseerde onderwijsmethodieken beschrijven die gebruikt zijn bij lessenseries over natuurkunde, namelijk Learning by Design (LBD) (Kolodner et al., 2003) en Design Based Science (DBS) (Fortus et al., 2004). De focus lag bij beiden voornamelijk op het meten van de uiteindelijke leeropbrengst, terwijl ik zocht naar geschikte tussentijdse momenten om individuele conceptualisatie te meten. Hoewel de onderzoekers wel aandacht gaven aan het tussentijds feedback geven, stonden er in de papers geen expliciete tips of kenmerken. Daarom heb ik de onderzoeken zelf geanalyseerd om er kenmerken uit af te leiden. Hieronder beschrijf ik deze, opgedeeld in twee categorieën die ik heb gebaseerd op deelvraag 1: conceptuele kennis opdoen (1a) en conceptuele kennis waarnemen (1b).

5.1.1. Conceptuele kennis opdoen (deelvraag 1a)

De deelvraag die ik in deze sectie wil beantwoorden, luidt: *Hoe kunnen leerlingen tijdens een ontwerp opdracht conceptuele kennis opdoen?* Daarvoor beschrijf ik hieronder kenmerken die ik heb afgeleid uit LBD en DBS.

K1 Beperk de tijd die leerlingen aan bouwen moeten besteden.

Kolodner (2003) stelt dat leerlingen meer tijd en aandacht overhouden voor de details van het ontwerp waar het meeste van geleerd kan worden door de tijd te beperken die ze nodig hebben om te bouwen. Bij LBD wordt hier invulling aan gegeven door een herontwerp-benadering: leerlingen krijgen daarbij de bouw instructie voor een redelijk werkend artefact. Vervolgens moeten ze bepalen welke eigenschappen er veranderd moeten worden. Door deze verbeteringen te generaliseren en toe te passen, moeten leerlingen natuurkundige principes ontdekken.

K2 Maak het gebruik van concepten expliciet voor de leerlingen.

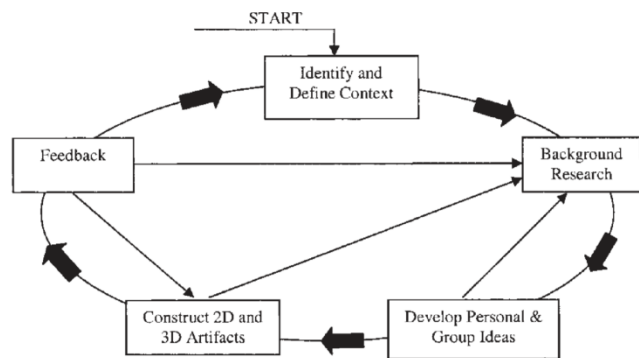
Kolodner et al. (2003) ontdekten in hun onderzoek dat het nodig is om het gebruik van concepten expliciet te maken. Daarom lieten ze leerlingen vuistregels opschrijven die ze door middel van experimenten moesten bewijzen of ontkrachten. Doordat ze zo expliciet bezig zijn met de concepten die ze toepassen, kunnen ze die ook in andere contexten toepassen. Bij LBD krijgen de ontdekte concepten blijvend aandacht door ze iedere les op het schoolbord weer te geven.

K3 Begin met een ontwerpspecificatie.

Om de leerlingen helderheid te verschaffen over wat ze moeten bereiken, wordt bij DBS (Fortus et al., 2004) begonnen met een ontwerpspecificatie. Daarin staan de eisen waaraan het artefact moet voldoen. De manier waarop het beschreven doel bereikt moet worden hoeft nog niet duidelijk te zijn, omdat dat stap voor stap bepaald wordt in DBS. Op deze manier ligt de nadruk op *hoe* de benodigde concepten werken. Deze manier van werken sluit goed aan bij het doel van ontwerponderwijs. Volgens Fortus et al. onderscheidt dat zich namelijk van veel andere vormen van onderwijs, omdat daarbij meestal vragen naar *wat* en *waarom* centraal staan (Fortus et al., 2004).

K4 Gebruik een overzicht om leerlingen duidelijk te maken welke stap ze moeten nemen.

Het doel van DBS (Fortus et al., 2004) is niet om leerlingen te leren ontwerpen, maar om ontwerpen te gebruiken om conceptuele kennis op te doen. Daarom wordt bij DBS gebruikgemaakt van een schematische weergave van de cyclus van activiteiten die doorlopen moet worden om iets te ontwerpen, zoals weergegeven in Figuur 7 (zie Figuur 2 voor een grotere weergave). Deze was bij hun onderzoek iedere les goed zichtbaar op een poster voorin het lokaal. De docent gaf aan hoe de activiteiten van de les daarbij pasten. Daardoor konden leerlingen zich focussen op het opdoen van conceptuele kennis.



Figuur 7 - schematische weergave van de werkwijze van DBS

We hebben nu gezien op welke manieren er bevorderd kan worden dat leerlingen conceptuele kennis opdoen. In hoofdzaak zijn dat drie dingen. Ten eerste moet de aandacht van leerlingen gericht worden op het gebruik van concepten en niet op bouwen. Ten tweede moet duidelijk zijn welke stappen ze moeten zetten in het ontwerpproces en ten slotte moeten leerlingen weten aan welke eisen hun artefact moet voldoen.

5.1.2. Conceptuele kennis waarnemen (deelvraag 1b)

De deelvraag die ik in deze sectie wil beantwoorden, luidt: *Hoe kan conceptuele kennis van individuele leerlingen zichtbaar gemaakt worden op een manier die naadloos aansluit bij het uitvoeren van de ontwerp opdracht?* Daarvoor beschrijf ik hieronder kenmerken die ik heb afgeleid uit LBD en DBS.

K5 Neem op vaste momenten de tijd voor reflectie op het gedane werk.

LBD (Kolodner et al., 2003) kent vaste momenten om concepten tot leerlingen door te laten dringen. Daar wordt bijvoorbeeld vorm aan gegeven door groepjes hun voortgang te laten presenteren aan de klas, waarna een bespreking volgt waarin medeleerlingen de werkwijze bevragen en duidelijk proberen te krijgen waarom bepaalde keuzes gemaakt zijn. Zulke momenten hebben dus een dubbel doel: de leerlingen doen conceptuele vorming op én de docent kan de opgedane kennis waarnemen. Omdat de nadruk in mijn onderzoek op het tweede ligt, heb ik ervoor gekozen om dit kenmerk in te delen onder "Conceptuele kennis waarnemen".

K6 Laat de leerlingen regelmatig testen of hun artefact voldoet aan de eisen.

Dit kenmerk is ook afgeleid van LBD (Kolodner et al., 2003). Het doel van dit kenmerk is om leerlingen te laten inzien wat ze nog niet helemaal begrijpen en wat ze al wel begrijpen. Bovendien is een testmoment geschikt voor de docent om feedback te geven, waardoor het duidelijk is voor de leerlingen wat er goed is aan hun ontwerp en waar ze nog naar moeten kijken om het artefact te verbeteren.

K7 Maak gebruik van formulieren die leerlingen in moeten vullen.

Om de leerlingen te stimuleren om op een gestructureerde manier te werken aan een oplossing voor een probleem, kunnen volgens Kolodner et al. (2003) formulieren een uitkomst bieden. Dit wordt bij LBD gedaan door voor verschillende stappen van het ontwerpproces een stencil aan de leerlingen te geven. Door dat stencil in te vullen worden leerlingen bij de hand genomen om te werken volgens dezelfde structuur die een professional zou hanteren. Daardoor hoeven ze niet steeds na te denken of op te zoeken op welke manier ze een probleem aan moeten pakken. Zo kan er meer lestijd besteed worden aan conceptuele vorming door het uitvoeren van ontwerpactiviteiten.

K8 Splits teams op in tweetallen als ze een ontwerp uit gaan werken.

DBS (Fortus et al., 2004) werkt met groepjes van vier leerlingen. Om ervoor te zorgen dat toch iedere leerling betrokken is bij het ontwerpproces, worden groepjes in DBS opgesplitst in tweetallen. Bovendien geeft dit meer inzicht in de mate waarin leerlingen begrijpen waar ze mee bezig zijn. Omdat ieder team wel aan hetzelfde model werkt, wordt uiteindelijk per team maar één model klassikaal besproken en beoordeeld. Het voordeel daarvan is tweeledig: leerlingen leren om samen te werken en bij het presenteren van artefacten wordt er lestijd bespaard.

Om conceptuele kennis waar te nemen voldoen de onderzochte lesmethodieken aan drie kenmerken, die als volgt samengevat kunnen worden. Maak gebruik van vaste momenten om te reflecteren en te testen. Help leerlingen bij het uitvoeren hiervan door hulpmiddelen aan te reiken. Laat ze samenwerken in tweetallen om de betrokkenheid van alle leerlingen te stimuleren.

5.2. Fase 2: Raamwerk ontwikkelen (deelvraag 2)

In deze paragraaf leest u hoe ik de materialen ontwikkeld heb voor een lessenserie in een raamwerk op basis van Scrum. Als eerste presenteer ik hoe ik de benodigde materialen ontwikkeld heb voor het raamwerk waarmee Scrum in de les gebruikt kan worden. Daarbij beschrijf ik hoe ik de kenmerken toepas die ik in fase 1 gevonden heb. Vervolgens beschrijf en onderbouw ik op welke momenten de docent conceptualisatie zal observeren en op welke manier hij dat zal doen. Ten slotte leg ik uit hoe ik een ontwerpopdracht geschreven heb die past binnen het ontwikkelde raamwerk om dat in de praktijk te evalueren. Dit koppel ik ook weer aan de gevonden kenmerken die erop van toepassing zijn.

5.2.1. Lestypen op basis van Scrum (deelvraag 2a)

Hieronder leg ik uit hoe ik Scrum vertaald heb naar een aantal verschillende lestypen. Om de toepassing van Scrum duidelijk te maken, beschrijf ik eerst de aspecten van Scrum die ik daarbij als leidraad genomen heb. Het belangrijkste principe daarvan is dat er gewerkt wordt in sprints. Dat zijn perioden met een vaste lengte waarin een artefact verder ontwikkeld wordt om aan vooraf opgestelde eisen te voldoen. Deze sprints verlopen volgens een vaste structuur, die ik overgenomen heb in mijn raamwerk.

Een sprint wordt begonnen met een Sprint Planning Meeting, waarin de teamleden bepalen welke eisen er de komende sprint op het programma staan. Iedere dag in een sprint wordt begonnen met een Daily Scrum: een bijeenkomst om de individuele vorderingen en belemmeringen in het team uit te wisselen. Een sprint wordt afgesloten met een Sprint Review en een Sprint Retrospective. De Review is een demonstratiemoment waarin aan alle belanghebbenden getoond wordt wat er in de afgelopen sprint bereikt is. De Retrospective is een meeting waar alleen teamleden bij aanwezig mogen zijn en is bedoeld om de samenwerking in de volgende sprint waar nodig te verbeteren.

Deze structuur heb ik naar lestypen vertaald door een sprint uit 4 tot 6 lessen te laten bestaan. Iedere les wordt er gewerkt aan de eisen die door het team gekozen zijn. Het kiezen van eisen moet gebeuren in de eerste les van de sprint. In de laatste les van de sprint geven de teams een demonstratie aan de docent en de andere teams, waarna de overige tijd gebruikt wordt om terug te blikken op de samenwerking en verbeterpunten op te stellen voor de volgende sprint. Meer details over de lestypen geef ik in het vervolg van deze paragraaf.

5.2.1.1. Werken met Scrum in de les

Ik heb een document geschreven waarin ik de Scrum-lestypen uitleg (bijlage A). Daarin geef ik eerst een kort overzicht van hoe Scrum normaal gesproken werkt. Vervolgens beschrijf ik daar de toepassing van voor een lessenserie. Hieronder geef ik de hoofdlijn van dit document weer.

In het raamwerk heb ik gebruikgemaakt van vijf verschillende lestypen, waarvan er vier zijn afgeleid van de Scrum-rituelen, zoals ik hierboven beschreven heb. De eerste les in van een lessenserie is niet afgeleid van Scrum. Daarin legt de docent uit wat de opdracht is en hoe ze die met Scrum moeten uitvoeren. De rest van de lessenserie is opgedeeld in drie sprints. De eerste sprint begint met een startles, alle volgende sprints beginnen steeds met een planles. Daarna volgen drie tot vier werklessen. Een sprint wordt afgesloten met een demoles. Hieronder heb ik alle gebruikte lestypen beschreven en toegelicht.

5.2.1.1.1. Uitlegles

Voor er begonnen kan worden met het project, moet de docent eerst uitleggen wat de werkwijze bij het project zal zijn. Een belangrijk onderdeel daarvan is dat Scrum uitgelegd wordt. Er worden twee varianten belicht: zoals het in het bedrijfsleven gebruikt wordt en hoe het in de klas gebruikt gaat worden. Om de leerlingen te overtuigen van de authenticiteit van Scrum, laat de docent hen een video-opname van een professionele Scrum-training zien waarin duidelijk uitgelegd wordt hoe Scrum door bedrijven gebruikt moet worden en wat daar het nut van is.

Deze les hoort dus niet bij de Scrum-methode zelf, maar is vergelijkbaar met de training die werknemers van bedrijven krijgen voordat ze met Scrum gaan werken. Dit lestype hoort dus bij de voorbereiding op het project. Tijdens lessen van dit type zal de docent niet proberen om conceptuele ontwikkeling waar te nemen, maar legt hij het fundament om dit mogelijk te maken.

5.2.1.1.2. Startles

De eerste les van de lessenserie wordt gevuld met het vormen van groepjes en het uitvoeren van een eerste Daily Standup. Als er dan nog tijd over is, gaan de leerlingen aan de slag met taken die door de docent op Trello klaargezet zijn. Deze les is vooral bedoeld om kennis te maken met Scrum, daarom zal de focus van de docent voornamelijk liggen op het goed laten uitvoeren van de daarbij behorende rituelen. Als daarna nog tijd over is, vullen de leerlingen die met de Sprint Planning, zoals die ook in de Planles wordt gehouden.

De teams worden door de docent gevormd. Het doel hiervan is dat er teams ontstaan met personen die elkaar aanvullen door verschillende kwaliteiten die ze hebben. De docent maakt daarvoor gebruik van formulieren. Iedere leerling krijgt een blaadje waarop hij vijf kwaliteiten een cijfer toekent van 1 tot 5 waarmee hij aangeeft hoe goed hij erin is (bijlage B). Ieder cijfer mag hij maar een keer toekennen, om zo de leerling te verplichten een duidelijke rangschikking te maken. Als ze hun formulier ingevuld hebben, vouwen ze het gedeelte waarop hun naam staat naar achteren, zodat alleen de cijfers nog zichtbaar zijn. De kwaliteiten die zo gerangschikt moeten worden, zijn:

- Als ik iets doe, doe ik het goed.
- Ik vind overleggen belangrijk.
- Ik kan goed beslissingen nemen.
- Ik kan goed “out-of-the-box” denken.
- Ik houd overzicht bij complexe problemen.

De docent kiest Scrum Masters uit die hebben aangegeven overleggen erg belangrijk te vinden. Hij kiest er zoveel als er groepjes gevormd moeten worden. De Scrum Masters moeten vervolgens formulieren kiezen waarop de naam van de leerlingen niet zichtbaar is, en zo een team samenstellen waarin alle kwaliteiten vertegenwoordigd zijn. Verder moet de Scrum Master in de gaten houden dat het Scrum-proces goed verloopt. Dit houdt voornamelijk in dat hij ervoor zorgt dat iedere les begonnen wordt met de Daily Standup. Verder moet hij er ook voor zorgen dat aan het begin van een

sprint de opgestelde taken in Trello gezet worden en dat er aan het einde van een sprint een demonstratie gehouden kan worden.

De teams delen zich vervolgens zelf op in tweetallen en bepalen wie de Product Owner wordt. Deze leerling moet in de gaten houden dat de belangrijkste eisen bovenaan de Product Backlog staan in Trello. De bovenste eisen worden namelijk aan het begin van een nieuwe sprint gekozen om te vertalen naar taken.

Tijdens deze les heeft de docent nog geen tijd om conceptuele ontwikkeling te meten, aangezien de docent zich vooral moet richten op het begeleiden van het Scrum-proces. Naar mate deze begeleiding minder noodzakelijk wordt, zal er meer ruimte zijn om conceptuele ontwikkeling te observeren.

5.2.1.1.3. Planles

Iedere volgende sprint wordt begonnen met een planles in plaats van een startles. In een planles wordt de Sprint Planning Meeting gehouden. Leerlingen bepalen welke eisen ze de komende sprint willen halen. Daarvoor worden er Product Backlog Items op de Sprint Backlog gezet waarvan de benodigde tijd ingeschat wordt. Aan de hand van deze PBI's worden taken opgesteld. Deze taken worden verdeeld over de teamleden. De docent ondersteunt hierbij en controleert de haalbaarheid van de geplande sprint. Na het plannen verloopt de les verder zoals een werkles.

Tijdens lessen van dit type kunnen er tijdens het opstellen van taken vermoedelijk al indicaties gezien worden van conceptuele ontwikkeling. Dat kan door te kijken welke taken er opgesteld worden voor het realiseren van eisen. Dat kan op twee manieren. De eerste optie is om te luisteren welke ideeën de leerlingen opperen over welke taken er uitgevoerd moeten worden en hoe die uitgevoerd moeten worden. Dit kan echter niet bij alle groepjes tegelijk. Een andere indicator is het resultaat van de planfase, omdat de opgestelde taken een direct gevolg is van de gezamenlijke conceptuele ontwikkeling, maar ditmaal van een groepje in zijn geheel. Op individueel niveau kunnen hier dus nog geen conclusies aan verbonden worden. Daarom kan de docent dit gebruiken als aanleiding om leerlingen vragen te stellen in de werklessen, zoals hieronder beschreven.

5.2.1.1.4. Werkles

Een werkles is de meest voorkomende les. Hierin worden de opgestelde taken uitgevoerd die tijdens de Sprint Planning zijn opgesteld. Iedere les wordt begonnen met een Daily Standup. De leerlingen vertellen elkaar hoe ver ze zijn, welke problemen ze ervaren en waaraan ze verder werken. Deze bijeenkomst moet kort gehouden worden. Daarom is het niet mogelijk voor de docent om alle teams te monitoren tijdens deze bijeenkomst. Het resultaat van de bijeenkomst is een bijgewerkt Scrum-bord op Trello. Aan de hand daarvan kan de docent bepalen welke concepten leerlingen waarschijnlijk zullen gebruiken tijdens het vervolg van de les. Aanvullend kan de docent aan de Scrum Master vragen hoe het gaat met het project. Op deze manier is indirect informatie te krijgen over de conceptuele ontwikkelingen binnen het team: de Scrum Master kan vertellen tegen welke problemen bepaalde teamleden aanlopen.

De overige tijd in deze lessen lijkt het meest geschikt om conceptuele ontwikkeling te meten. Daarvoor kan de docent als vorm van begeleiding vragen wat een leerling maakt en waarom hij het zo doet. Aan de hand hun bijdrage aan het artefact en de mondelinge toelichting kan de docent noteren hoe ver de leerling is qua conceptuele vorming. Het vragen om toelichting is daarvoor noodzakelijk, om uit te sluiten dat het "toevallig werkend knip- en plakwerk" is. De docent maakt hierbij gebruik van het observatiemateriaal dat verderop uitgelegd wordt. Dit is materiaal dat niet ontwikkeld is op basis van de Scrum-methode, maar specifiek om concepten te kunnen waarnemen in een onderwijsomgeving.

5.2.1.1.5. Demoles

Aan het einde van iedere sprint demonstreren leerlingen wat ze hebben bereikt, zoals dat bij Scrum gebeurt tijdens de Sprint Review. Daarbij moeten ze onderbouwen waarom ze hun artefact zo hebben gemaakt en welke concepten ze daarbij toegepast hebben. Dat gebeurt door de robot te laten doen wat hij kan, aangevuld met een mondelinge toelichting.

De demonstratie kan ook informatie geven over de ontwikkeling van conceptuele kennis van leerlingen, hoewel dit een bijeenkomst is die in teamverband gehouden wordt. Tijdens deze bijeenkomst kan de docent allereerst noteren op welke zaken hij kan letten bij de teamleden aan de hand van het geboekte resultaat. Individuele vorderingen zullen voornamelijk meetbaar zijn bij hen die het woord voeren. Door gericht vragen te stellen aan individuele leerlingen is ook meer inzicht te krijgen in de individuele vorderingen. Hierbij moet echter wel rekening gehouden worden met de beperkte tijd die beschikbaar is voor de bijeenkomst.

Na de demonstraties wordt er een Sprint Retrospective gehouden. Deze bijeenkomst is bedoeld om een team te laten reflecteren op hoe de samenwerking verliep, om afspraken te maken voor een betere volgende sprint. Hierbij wordt de docent zo min mogelijk betrokken, zodat de groepjes interne frustraties zelf kunnen uitpraten. Als dit niet lukt, kan de docent natuurlijk wel ingeschakeld worden. Dit onderdeel van de les leent zich er in ieder geval niet voor om conceptuele ontwikkeling waar te nemen, omdat het daar allereerst niet om gaat en ook omdat de docent zo min mogelijk betrokken wordt bij dit onderdeel.

5.2.1.2. *Kenmerken toegepast op Scrum in de les*

In deze paragraaf beschrijf ik hoe de gevonden kenmerken toegepast zijn in het ontwikkelde raamwerk. Kenmerk 1 en 3 behandel ik hier niet, omdat die niet van toepassing zijn op dit gedeelte van het ontwikkelde materiaal maar alleen op de ontwerpopdracht. Die behandel ik verderop in dit verslag.

K2 Maak het gebruik van concepten expliciet voor de leerlingen.

Het gebruik van concepten wordt in het ontwikkelde raamwerk expliciet gemaakt doordat de docent ernaar vraagt tijdens de werklessen.

K4 Gebruik een overzicht om leerlingen duidelijk te maken welke stap ze moeten nemen.

Een dubbelzijdig A4'tje moest leerlingen houvast bieden bij het werken volgens Scrum. Daarop was van ieder lestype een tabel te vinden met een tijdsindeling en wat er gedaan moest worden. Dit overzicht is een samenvatting van de uitgebreidere uitleg van Scrum en de toepassing daarvan in de les en is te vinden op de laatste pagina van de Scrum-reader (bijlage A).

K5 Neem op vaste momenten de tijd voor reflectie op het gedane werk.

Deze ontwerprichtlijn sluit naadloos aan bij het Scrum-werkproces, omdat er twee vaste momenten voor zijn voor reflectie op uitgevoerde taken. Aan het begin van iedere les moeten leerlingen namelijk samen overleggen wat ze hebben gedaan en waar ze verder mee gaan en aan het einde van iedere sprint is er een Sprint Review om het gemaakte artefact te presenteren.

K6 Laat de leerlingen regelmatig testen of hun artefact voldoet aan de eisen.

Tijdens de Sprint Planning moeten de leerlingen ook test-taken opstellen. De bedoeling van deze taken is om van tevoren duidelijk te hebben hoe een ander tweetal uit het team het resultaat kon testen. Op dergelijke momenten kunnen de tweetallen met elkaar meedenken en heeft ook de docent de mogelijkheid om feedback te geven. Deze momenten leken mij daarvoor in het bijzonder geschikt, omdat testmomenten niet centraal gecoördineerd waren, zodat de docent iedere les bij meerdere tests kon kijken en helpen.

K7 Maak gebruik van formulieren die leerlingen in moeten vullen.

Om leerlingen structuur te bieden, is er in de webapp Trello voor ieder team een bord aangemaakt met zes lijsten: Product Backlog, Sprint Backlog, Te doen, mee bezig, gedaan en problemen. De eerste vijf lijsten komen overeen met de lijsten die bij Scrum gebruikt worden. De laatste lijst is er om de problemen te noteren waarvan blijkt tijdens de Daily Standup dat ze er zelf niet uitkomen. Op de Scrum-lijsten moeten de leerlingen bijhouden waar ze mee bezig zijn en welke eisen er later nog gerealiseerd kunnen worden in het ontwerp van hun artefact.

K8 Splits teams op in tweetallen als ze een ontwerp uit gaan werken.

In tegenstelling tot DBS wordt er een groeps grootte van circa 6 in plaats van 4 gehanteerd. Dit aantal is afgeleid van Scrum, waar teams uit 7±2 personen bestaan. Een team moest zich wel opsplitsen in tweetallen. Per twee leerlingen moest er aan een eigen verbetering van het artefact gewerkt worden. Daardoor kon ieder team onderling de taken verdelen over de tweetallen, zodat er parallel gewerkt kon worden aan eisen van de opdrachtbeschrijving, zoals dat bij Scrum in het echt ook gedaan zou worden.

5.2.2. Conceptualisatie observeren (deelvraag 2b)

In deze paragraaf presenteer ik hoe ik een instrument heb ontwikkeld dat een docent kan gebruiken om individuele conceptuele kennis van leerlingen waar te nemen. Ik begin met de momenten waarop ik verwachtte dat conceptualisatie te meten is. Vervolgens beschrijf ik het hulpmiddel dat op de geselecteerde momenten ingezet kan worden. Ten slotte beschrijf ik de schaal waarop het beheersingsniveau van een leerling aangegeven kan worden.

5.2.2.1. Momenten om conceptualisatie waar te nemen bij Scrum

Het was mijn verwachting dat de vaste bijeenkomsten die bij Scrum horen niet erg geschikt zouden zijn om individuele vorming van leerlingen te meten, omdat dit momenten zijn dat leerlingen in groepsverband bezig zijn. Bovendien is het de bedoeling om dan de voortgang van het project te bespreken zonder op details in te gaan. Het enige moment dat draait om individuele voortgang is de Daily Standup, hoewel ook daar geen ruimte is om details te bespreken. Deze bijeenkomst dient namelijk erg kort gehouden te worden. Bovendien moet deze bijeenkomst onvermijdelijk aan het begin van de les gehouden worden, waardoor alle groepjes hier tegelijk mee bezig zijn. Daardoor is de docent niet in staat om bij alle leerlingen tijdens hetzelfde ritueel de conceptuele vorming waar te nemen. Het resultaat van de bijeenkomst, namelijk een lijst met problemen en een bijgewerkt Scrum-bord, kan vermoedelijk wel behulpzaam zijn.

De momenten die mij het meest kansrijk leken om inzicht te krijgen in de individuele conceptuele ontwikkeling, zijn tijdens het uitvoeren van de ST's. Door dan vragen te stellen aan leerlingen, verwachtte ik inzicht te krijgen in de individuele conceptuele vorming van leerlingen. Daarom heb ik een observatiehulpmiddel geschreven dat de docent helpt om vragen te stellen en te noteren welke conceptualisatie hij daarbij waarnam.

5.2.2.2. Ontwerp van het observatiehulpmiddel

Het observatiehulpmiddel (bijlage C) is bedoeld om de docent overzicht en houvast te geven bij het waarnemen van conceptuele vorming van individuele leerlingen. Ik heb hiervoor een dubbelzijdig A4'tje gemaakt. Op de ene kant staat een tabel waarin de docent zijn waarnemingen kan bijhouden. Daarvoor noteert hij in de linker kolom de namen van zijn leerlingen. Daarachter staat per concept een kolom waarin hij een letter kan omcirkelen die aangeeft in welke mate een leerling dat concept beheerst.

Om ervoor te zorgen dat metingen consistent zijn met elkaar, heb ik algemene en concrete vragen opgesteld als handvatten om bepaalde niveaus te controleren. De algemene vragen staan boven de in te vullen tabel op het observatieformulier en zijn voor alle concepten te gebruiken (bijvoorbeeld “Hoe werkt dit?”). Op de achterzijde van het formulier heb ik ruimte gelaten om concrete vragen op te schrijven. Om het raamwerk te gebruiken moeten op het observatiehulpmiddel dus nog concepten en concrete vragen opgeschreven worden.

5.2.2.3. Beheersingsniveaus om te noteren

De docent kan zoals gezegd voor iedere leerling aangeven in welke mate die een bepaald concept beheerst. Daarvoor heb ik een schaal met drie niveaus gekozen. Deze heb ik afgeleid van de taxonomie van Bloom. Ik heb niet gekozen voor de SOLO-taxonomie, omdat ik inschatte dat de opdrachten die de leerlingen moesten uitvoeren in de categorie “unistructural” (enkelvoudige benadering) vallen. Verder heb ik geconcludeerd uit het onderzoek van Meerbaum-Salant (2013) dat de taxonomie van Bloom binnen één SOLO-niveau goed toe te passen moet zijn.

Bij dit onderzoek heb ik mij beperkt tot het onderste deel van de taxonomie van Bloom, omdat de uitdaging me dan al groot genoeg leek voor de vierdeklassers. Daarbij heb ik de volgende beschrijvingen en algemene vragen opgesteld en op het observatiehulpmiddel gezet:

- **Kennen/benoemen:** Wat is dit?
- **Begrijpen/voorspellen:** Hoe werkt dit?
- **Toepassen/maken:** Waarom gebruik je dit?

De eerste letters van bovenstaande elementen staan in de kolom van ieder concept achter de naam van iedere leerling. Door een letter te omcirkelen kan de docent aangeven dat een leerling heeft laten zien desbetreffend concept op dat niveau te beheersen.

5.2.3. Concrete toepassing van het raamwerk met een ontwerpopdracht (deelvraag 2c)

Om het ontwikkelde raamwerk te evalueren, heb ik ook een opdrachtbeschrijving gemaakt die erbij paste. Daarbij heb ik rekening gehouden met de gevonden kenmerken voor conceptuele vorming in het ontwerponderwijs. Verder heb ik ook de concepten beschreven die tijdens het uitvoeren van deze opdracht nodig waren. Deze heb ik ingevuld in het ontwikkelde sjabloon. Dit is te vinden in bijlage C.

5.2.3.1. Opdrachtbeschrijving

In de opdrachtbeschrijving (bijlage D) heb ik de context voor de opdracht geschetst. De gekozen context is een haven terrein. Een bedrijf wil dat er heftruck-robots ontwikkeld worden die daar zelfstandig containers kunnen verplaatsen. Zo krijgen leerlingen het gevoel met een realistische, authentieke opdracht bezig te zijn. Dat geeft ze het gevoel dat het relevant is waar ze mee bezig zijn, wat resulteert in een hogere motivatie.

Onder de inleidende beschrijving van de context staan in dit document de eisen waaraan de robot moet voldoen. Bij iedere eis staat een tekstuele toelichting, om de eis te verduidelijken en in de context te plaatsen van de opdracht. Daarom wordt bijvoorbeeld uitgelegd waarom containers op kleur gesorteerd moeten worden: iedere kleur heeft een andere bestemming.

5.2.3.2. Concepten voor het observatiehulpmiddel

De concepten die ik gekozen heb om te observeren, vallen allemaal onder de noemer van programmeerconcepten, zoals beschreven door Schwill (1994). De concepten die hij daaronder geschaard heeft, zijn:

- Opeenvolging
- Alternatieven
- Herhaling
- Recursie
- Non-determinisme
- Parametrisatie

Omdat deze concepten niet allemaal geschikt zijn om in één keer aan te leren, laat staan aan onervaren programmeurs, heb ik daaruit drie concepten geselecteerd om te meten: opeenvolging, alternatieven en herhaling. Van deze concepten heb ik concrete toepassingen genomen die ik bij leerlingen toegepast wilde zien worden. De volgende lijst is daarvan het resultaat:

- Commentaar (opeenvolging)
- Variabelen (opeenvolging)
- Procedures (opeenvolging)
- Objecten (opeenvolging)
- Keuzes (alternatieven)
- Herhalingen (herhaling)

Het concept van opeenvolging heb ik opgesplitst in vier onderdelen die ik afzonderlijk wilde meten bij leerlingen, omdat dit de concrete bouwstenen van programmeertalen zijn waar leerlingen gebruik van maken bij deze lessenserie. Aan deze lijst van concepten heb ik nog een item toegevoegd om recht te doen aan het robot-scenario:

- In- en uitvoer

Op de achterzijde van het observatiehulpmiddel heb ik concrete vragen over de programmeerconcepten opgesteld. Deze heb ik gegroepeerd per concept. Achter iedere vraag heb ik aangegeven welke mate van beheersing van het concept erbij hoort.

5.2.3.3. Kenmerken toegepast in de ontwerpopdracht

In deze paragraaf beschrijf ik hoe de in fase 1 gevonden kenmerken toegepast zijn in de geschreven ontwerpopdracht (bijlage D).

K1 Beperk de tijd die leerlingen aan bouwen moeten besteden.

Aan het begin van de lessenserie heb ik aan de leerlingen een bouwstructie gegeven voor een basis-robot. De robot kon rijden en was alleen voorzien van een afstandssensor. Daardoor konden leerlingen direct beginnen met het schrijven van een programma dat de robot een stukje liet rijden. Voor andere eisen waren nog andere sensoren nodig, die met redelijk weinig moeite op de robot gemonteerd konden worden. Daardoor was de tijd die leerlingen nodig hadden om aan de robot te bouwen beperkt.

K2 Maak het gebruik van concepten expliciet voor de leerlingen.

Bij de ontwerpopdracht werken leerlingen met materiaal dat opgebouwd is aan de hand van voorbeelden van hoe je bepaalde sensoren kunt gebruiken. Daarin worden de concepten wel gebruikt, maar niet expliciet genoemd. Dat gebeurt wel in de uitleg van de programmeertaal, die ik hen ook aangereikt heb. Om het gebruik van deze concepten te expliciteren, vraagt de docent ernaar tijdens de werklessen. Dit doet hij aan de hand van de vragen op het observatiehulpmiddel.

K3 Begin met een ontwerpspecificatie.

Naast een bouwstructie kregen de leerlingen aan het begin ook per team een opdrachtbeschrijving. Daarop stonden alle eisen vermeld waaraan de robot moest voldoen. Bij iedere eis was ook vermeld hoeveel punten het op zou leveren om de eis te implementeren. De eisen waren geordend van eenvoudig naar ingewikkeld. Dit moest duidelijk maken wat er uiteindelijk van de leerlingen verwacht werd.

5.3. Fase 3: Raamwerk evalueren (deelvraag 3)

Bij het evalueren van het raamwerk heb ik op twee dingen gelet, zoals weergegeven in de subvragen van deelvraag 3. Ten eerste heb ik erop gelet wat de momenten zijn waarop leerlingen blijken te geven van conceptuele vorming (deelvraag 3a). Het tweede is of de gebruikte hulpmiddelen voldoende ondersteuning bieden om dat bij alle leerlingen te meten (deelvraag 3b). Ten slotte zijn mij ook overige obstakels opgevallen. Deze drie zaken beschrijf ik hieronder.

5.3.1. Momenten (deelvraag 3a)

Hieronder leest u wanneer en in welke mate er tijdens het uitvoeren van de lessenserie conceptuele vorming van individuele leerlingen gemeten kon worden. Dit heb ik beschreven aan de hand van de verschillende lestypen waarmee gewerkt is tijdens de lessenserie.

5.3.1.1. Uitlegles

In de uitlegles zijn leerlingen nog niet bezig geweest met vakinhoudelijke zaken. Daarom zijn er tijdens deze les geen metingen verricht voor dit onderzoek.

5.3.1.2. Startles

De les was voor de docent volledig gevuld met het vormen van groepjes en geven van extra instructie aan de Scrum Masters. Zodoende is in deze les, zoals verwacht, geen conceptuele vorming gemeten.

5.3.1.3. Planles

De planlessen waren voor de docent het meest intensief wat betreft het begeleiden van het Scrum-proces. Dit kwam er vermoedelijk door dat er maar weinig zulke lessen waren, zodat leerlingen geen routine ontwikkelden in wat ze daarin moesten doen. Zodoende kwam de docent er zelfs niet aan toe om te letten op de indicaties van conceptualisatie die gezien zouden kunnen worden in de planfase. Daarbij kwam nog dat de leerlingen graag zo snel mogelijk verder wilden werken aan hun robot en daarom met zo min mogelijk overleg iemand lieten beslissen welke taken er verder uitgevoerd moesten worden. De bijgewerkte Scrum-boards waren wel bruikbaar als leidraad voor vragen in volgende werklessen.

5.3.1.4. Werkles

Meestal had ik als docent voldoende gelegenheid om de vragen die leerlingen hadden te beantwoorden. Bijna altijd was daarnaast nog tijd genoeg om leerlingen die geen vragen hadden zelf te vragen om uit te leggen waarmee ze bezig waren. Ik merkte dat leerlingen er niet vreemd van opkeken wanneer ik een vraag stelde over het programma waaraan ze werkten. Op deze manier heb ik met behulp van het observatiehulpmiddel tijdens de werklessen vaak kunnen meten hoe ver de conceptuele vorming van leerlingen gevorderd was. Daarbij bleken de leerlingen wel moeite te hebben met het beantwoorden van vragen zoals "Hoe komt het dat het zo werkt?" of "Waarom staan deze regels code hier?". Dergelijke vragen werden meestal heel algemeen beantwoord, zonder belangrijke concepten te benoemen. Ook als ik doorvroeg naar specifieke onderdelen van de code konden leerlingen niet uitleggen waarom ze het zo gedaan hadden. Vaak was de reden: "Omdat het voorbeeld het ook zo doet."

Helaas was bijna iedere les een kleine helft van de leerlingen bezig met het bouwen van de robot, terwijl de bedoeling was dat leerlingen zoveel mogelijk tijd zouden programmeren. Sommige teams hadden de taken voor het project als volgt verdeeld tussen de tweetallen. Een tweetal zou programmeren en het andere tweetal zou de robot bouwen. Het bleek voor mij erg moeilijk te zijn om de leerlingen tot een andere taakverdeling te laten komen. Vooral de leerlingen die de robot zouden bouwen leken een reden te hebben om niet te willen programmeren, of het bouwen van de robot zo minutieus te willen doen dat er geen tijd meer was voor programmeren.

5.3.1.5. Demoles

Tijdens de demonstraties vroeg de docent aan de leerlingen hoe ze ervoor hadden gezorgd dat de robot kon doen wat hij deed. Daarbij heb ik expliciet gevraagd om concepten te noemen die ik tijdens de werklessen uitgelegd had. Regelmatig kwam het voor dat er maar een enkeling antwoord wist te geven. Waarschijnlijk heeft dat ermee te maken dat de leerlingen per groepje de robot van één tweetal gekozen hadden en daarop ook alleen het programma van dat tweetal had staan. Op deze momenten was dus, zoals verwacht (zie 5.2.1.1.5), de conceptuele vorming van slechts een paar leerlingen waar te nemen.

Tijdens de demonstraties bleek dus dat de samenwerking tussen de tweetallen in een team niet verliep zoals ik dat had bedoeld. In de laatste les voor een demonstratie hadden ze de code van beide tweetallen moeten samenvoegen tot een geheel. Het bleek echter dat de tweetallen van een team vaak bezig waren met dezelfde taken of dat ze niet inzagen hoe de code van twee programma's samengevoegd kon worden tot een gecombineerd programma zonder dat het stuk ging.

5.3.2. Hulpmiddelen (deelvraag 3b)

In dit onderdeel beschrijf ik hoe alle gebruikte materialen ervaren zijn tijdens de lessenserie. Dat betreft allereerst de ontwikkelde materialen: de scrum-uitleg, het observatiehulpmiddel en de opdrachtbeschrijving. Daarnaast is er ook gebruikgemaakt van online uitlegmateriaal van Small Basic (www.smallbasic.com) en EV3Basic (www.ev3basic.com). De ervaringen met al deze vijf hulpmiddelen wordt hieronder achtereenvolgens behandeld.

5.3.2.1. Scrum-uitleg

Het werken volgens de Scrum-methode was voor alle leerlingen nieuw. Daardoor was het nodig dat ik de leerlingen hierbij aan het begin van de lessenserie intensief begeleidde, ondanks het overzicht van alle lestypen achteraan de Scrum-uitleg en de extra uitleg die de Scrum Masters hadden gehad.

De planlessen waren het meest intensief voor mij, omdat er daar maar drie van waren. Daardoor hadden de leerlingen het bij de volgende planles al niet meer zo helder hoe een planles ook alweer in zijn werk ging. Een concreet probleem dat vaak voorkwam, was dat leerlingen de eisen (PBI's) niet vertaalden naar taken (ST's), maar PBI's op de te-doenlijst zetten. Leerlingen leken de noodzaak er niet van in te zien om van tevoren te bepalen welke taken er precies gedaan moesten worden om een eis te realiseren. Daardoor was ik veel tijd kwijt om leerlingen toch ST's op te laten stellen.

Hoewel er evenveel demonstratielessen waren, was daar het probleem niet zo groot. Het maakte veel verschil dat die lessen klassikaal waren en niet in teams, waardoor de docent steeds expliciet kon zeggen wat er moest gebeuren.

De tussenliggende werklessen gingen geleidelijk steeds beter. Ik wees er gedurende de eerste sprint iedere les op dat de leerlingen moesten beginnen met een Daily Standup. Daarna was dat er wel aardig ingeslepen en begonnen de leerlingen iedere les stevast met het pakken van de robot met wat extra LEGO™-onderdelen en het houden van een Daily Standup. Een enkele keer was het nog nodig dat ik de leerlingen eraan herinnerde, maar het kwam ook voor dat ik tussen de lessen in gesprek was met een leerling en dat de leerlingen van 4 vwo ondertussen aan de slag gingen met de opdracht, nog voordat de les officieel begon, zonder de Daily Standup over te slaan.

5.3.2.2. Observatiehulpmiddel

Het observatiehulpmiddel maakte het voor mij erg makkelijk om bij te houden in hoeverre een leerling een concept al beheerste. De vragen die ik op de achterkant van het blad had genoteerd heb ik daarbij nauwelijks gebruikt. Dat komt vooral omdat ik ze zelf ook opgesteld had, waardoor ik al wist welke vragen ik kon stellen om te vragen naar conceptuele kennis. Het is wel opmerkelijk dat geen enkele

leerling verder kwam dan het laagste niveau bij veel concepten. Dat wil zeggen dat ze de concepten wel konden benoemen, maar niet duidelijk konden maken hoe ze werkten of toegepast moesten worden.

5.3.2.3. Opdrachtbeschrijving

De leerlingen hadden weinig aansporing nodig om aan de slag te gaan met de opdracht. Uit de manier waarop ze ermee bezig waren bleek dat deze manier van werken ze wel aansprak. Zodra ze binnenkwamen, pakten ze de LEGO, startten ze de benodigde programma's en zochten ze de benodigde websites op. Regelmatig kwam het voor dat enkele leerlingen individueel of in tweetallen zelfs buiten de reguliere lessen om verder kwamen werken aan de opdracht.

Een probleem dat zich aandeed, was de constructie van de robots zelf. Hoewel ze al begonnen met een bouw instructie voor een eenvoudige basisrobot, kostte het nog veel tijd om deze aan te passen aan de eisen van de opdracht en aan de wensen van de leerlingen. Voornamelijk het monteren van sensoren en het bouwen van een hef-mechanisme voor de containers kostte veel tijd. Daarnaast besteedden sommige leerlingen ook veel aandacht aan het uiterlijk van de robot, hoewel dat niet van belang was voor de opdracht. Uiteindelijk ging er dus veel meer tijd zitten in het bouwen van de robots dan dat ik van tevoren ingeschat had, waardoor leerlingen minder lang bezig waren met het leren en toepassen van programmeerconcepten dan dat ik gehoopt had.

5.3.2.4. Uitlegmateriaal Small Basic en EV3Basic

Helaas bleek tijdens de werklessen dat de leerlingen de concepten in eerste instantie niet wisten te benoemen (eerste niveau: kennen/benoemen). Dat lukte ze ook niet nadat ik ze gewezen had op de uitleg op www.smallbasic.com. Daarom heb ik de concepten zelf expliciet benoemd en aangewezen in de voorbeeldprogramma's die ze op www.ev3basic.com vonden. Dat deed ik als een tweetal het nodig bleek te hebben in het programma dat ze wilden schrijven. Zo zorgde ik ervoor dat het leerproces vraaggestuurd bleef.

Vaak bleken leerlingen ervan uit te gaan dat de voorbeelden eenvoudig achter elkaar te knippen en te plakken waren. Waarom het niet werkte als ze het voorbeeld naar hun eigen code kopieerden, bleek regelmatig een raadsel voor hen. Als ik ze dan vroeg wat ze hadden willen programmeren, wezen ze vaak naar het voorbeeld en maakten duidelijk dat hetzelfde resultaat ook gewenst was in hun eigen programma, maar dan met een kleine aanpassing. Op zulke momenten bleek duidelijk dat de achterliggende concepten nog niet duidelijk waren. Daarom heb ik op die momenten de concepten zelf uitgelegd. Als leerlingen deze uitleg vervolgens konden reproduceren, is dat niet genoteerd alsof ze het begrepen, maar alleen dat ze het kenden. Latere lessen zijn er weer vragen gesteld over desbetreffende concepten. Dan bleek echter vaak dat ze opnieuw uitgelegd moesten worden.

5.3.3. Overige obstakels

Hoewel de leerlingen iedere les begonnen met een Daily Standup, was de samenwerking tussen de tweetallen die een team vormden niet optimaal. Het knelpunt lag er vooral in dat de geboekte resultaten samengevoegd moesten worden. De code van het ene tweetal leek in veel gevallen bijzonder weinig op die van het andere tweetal. Daardoor was het lastig om deze twee programma's samen te voegen tot één programma.

Verder gaven de leerlingen zelf aan dat ze het lastig vonden om een samenwerking vorm te geven met drie tweetallen. Ze gaven aan dat ze liever zouden samenwerken in groepjes van vier.

6. Conclusie en Discussie

In het uitgevoerde onderzoek heb ik een raamwerk ontworpen voor een lessenserie aan de hand van de Scrum-methode waarbij de conceptuele vorming van individuele leerlingen waargenomen moest kunnen worden. Uit de resultaten concludeer ik voorzichtig dat het ontwikkelde raamwerk daaraan kan voldoen. Mijn voorzichtigheid vindt zijn oorzaak erin dat het huidige onderzoek erg beperkt was door verschillende factoren. Hieronder beantwoord ik eerst de hoofdvraag van het onderzoek. Daarna ga ik verder in op de beperkingen die ik in dit onderzoek tegengekomen ben. Tot slot doe ik nog enige aanbevelingen voor vervolgonderzoek.

6.1. Beantwoording van de onderzoeksvraag

Om uiteindelijk de hoofdvraag te kunnen beantwoorden, rest mij alleen nog om een antwoord op deelvraag 3 te formuleren. Deelvraag 1 heb ik beantwoord door kenmerken voor het raamwerk te beschrijven. Deelvraag 2 heb ik beantwoord door het materiaal voor een raamwerk te ontwikkelen en daarbij ook een ontwerpopdracht te schrijven. Deelvraag 3 gaat erover of het ontwikkelde materiaal wel geschikt is voor het beoogde doel: conceptuele vorming van individuele leerlingen waarnemen tijdens het uitvoeren van een authentieke ontwerpopdracht.

Om die vraag te beantwoorden heb ik twee aspecten van het ontwikkelde materiaal onderzocht. Ten eerste heb ik bekeken of de momenten waarop conceptualisatie geobserveerd is daarvoor geschikt zijn. Ten tweede heb ik bepaald of de gebruikte hulpmiddelen daarbij voldoende ondersteuning boden. Beide aspecten beschrijf ik hieronder.

6.1.1. Momenten (deelvraag 3a)

Ik heb als docent bij leerlingen tijdens de werklessen inderdaad conceptuele vorming waargenomen, hoewel ik niet heb waargenomen dat leerlingen een bepaald concept beheersten op een niveau boven “kennen”. Er is onvoldoende bewijs om te bepalen of uit te sluiten dat dit komt door het ontwikkelde raamwerk. Ik vind het aannemelijk dat de oorzaak daarin niet ligt, maar in andere zaken die ik verderop beschrijf in paragraaf 6.2.

Tijdens de lessen had ik ook voldoende tijd om naast het beantwoorden van vragen ook zelf leerlingen te bevragen om zo conceptualisatie waar te nemen. Dit gold nog niet in de eerste sprint, omdat de leerlingen toen nog erg moesten wennen aan de manier van werken, maar vanaf de tweede sprint hadden leerlingen daar weinig begeleiding meer bij nodig.

Als docent had ik de indruk dat de werklessen wel geschikt waren om individuele conceptuele kennis waar te nemen, maar dat deze helaas niet opgedaan werd door de leerlingen. Daarom concludeer ik dat de momenten in het beschreven Scrum-proces geschikt zijn om conceptuele kennis waar te nemen bij individuele leerlingen tijdens het uitvoeren van een ontwerpopdracht.

6.1.2. Hulpmiddelen (deelvraag 3b)

In dit onderzoek heb ik verschillende hulpmiddelen gebruikt. Sommige daarvan heb ik zelf ontwikkeld, andere waren al beschikbaar. Ik begin met een terugblik op de zelf ontwikkelde hulpmiddelen: de Scrum-beschrijving, het observatiehulpmiddel en de opdrachtbeschrijving. Daarna evalueer ik het gebruikte online uitlegmateriaal.

6.1.2.1. Scrum-uitleg

De Scrum-uitleg is door leerlingen weinig gebruikt en is vermoedelijk ook niet geschikt als handleiding om Scrum in de praktijk toe te passen. Om leerlingen hiermee aan het werk te zetten is het wellicht geschikter om gebruik te maken van formulieren die de leerlingen op vaste momenten moeten invullen en inleveren bij hun docent. Deze conclusie wordt ook gestaafd door kenmerk K7, waarin

wordt aangemoedigd om formulieren te gebruiken zodat leerlingen zonder een ontwerpproces uit hun hoofd te leren, een ontwerpprobleem toch kunnen aanpakken zoals een professional dat zou doen. Hoewel ik dit kenmerk gevonden had, heb ik dit niet zo concreet toegepast in mijn onderzoek, omdat ik vermoedde dat het gebruik van een Scrumboard op Trello voldoende structuur en overzicht zou bieden aan de leerlingen om de ontwerpopdracht gestructureerd aan te kunnen pakken. Dit bleek echter niet het geval te zijn.

6.1.2.2. Observatiehulpmiddel

Het observatiehulpmiddel bood mij inderdaad overzicht op de concepten die de leerlingen beheersten en op welk niveau ze dat deden. Omdat ik in dit onderzoek zelf ook als docent fungeerde, kon ik niet goed bepalen of de vragen op de achterkant van het hulpmiddel van toegevoegde waarde waren. Deze had ik namelijk zelf bedacht en heb ik tijdens het onderzoek nauwelijks geraadpleegd. Ik verwacht dat andere docenten dit meer zouden waarderen.

6.1.2.3. Opdrachtbeschrijving

De opdrachtbeschrijving bleek een motiverende context te scheppen voor de leerlingen. Daarbij was de verleiding wel erg groot om veel tijd te besteden aan het bouwen van een robot en dus minder tijd met programmeren bezig te zijn. Om leerlingen productiever te laten worden in het leren van concepten, moet er nog kritisch naar de opdrachtbeschrijving gekeken worden. Wellicht kan die aangepast worden zodat het voor de leerlingen duidelijk is dat het opdoen van conceptuele kennis het hoofddoel van de opdracht is.

6.1.2.4. Online uitlegmateriaal

Het online uitlegmateriaal bleek te weinig aan te sluiten bij de context van de opdracht. De uitleg van de concepten gebeurde in een opdrachtprompt-scenario, terwijl ze toegepast moesten worden in een heel andere context: een robot. Verder bestaat het materiaal voornamelijk uit statische tekst, aangevuld met een enkele afbeelding. Dat wordt door Bennedsen en Caspersen afgeraden. Ze stellen namelijk dat het belangrijk is dat leerlingen leren om een programmeertaak op een systematische manier aan te pakken en dat statisch leermateriaal daar niet geschikt voor is, omdat er een dynamisch proces gepresenteerd moet worden in plaats van alleen een perfect resultaat (Bennedsen & Caspersen, 2008). Zij adviseren gebruik te maken van bijvoorbeeld video-opnamen van experts die een soortgelijke taak uitvoeren en daarbij hardop denken.

6.1.3. Antwoord op de hoofdvraag

De hoofdvraag van dit onderzoek luidt:

Hoe kan Scrum aangepast worden tot raamwerk voor een lessenserie om conceptuele vorming van individuele leerlingen waar te kunnen nemen tijdens het uitvoeren van een authentieke ontwerpopdracht voor informatica in de bovenbouw van het vwo?

Als antwoord op deze vraag heb ik materiaal ontwikkeld en getest waaruit blijkt dat het daarmee mogelijk is om conceptuele vorming waar te nemen bij individuele leerlingen. Daarbij is het mij wel opgevallen dat leerlingen niet zoveel conceptuele vorming opdoen als ik had gehoopt. Dit wijkt ik grotendeels aan factoren buiten het ontwikkelde materiaal. Hierop ga ik hieronder verder in. De potentie van het ontwikkelde raamwerk schat ik hoog in, maar daarvoor is het wel nodig dat het doorontwikkeld wordt. Aanbevelingen daarvoor doe ik in de laatste paragraaf.

6.2. Beperkingen

Het werken met Scrum bleek te leiden tot een goede werksfeer, waarbij iedere les veel leerlingen hun werk weer met plezier oppakten. Naar de precieze mate van effectiviteit van deze methode moet nog

meer onderzoek gedaan worden. De testgroep van dit onderzoek was namelijk erg klein en de aard van dit onderzoek was voornamelijk verkennend.

Ook het gebruikte online uitlegmateriaal vormde een beperking waar ik tijdens dit onderzoek tegenaan liep. Zoals gezegd, is dat voornamelijk opgebouwd uit voorbeelden van hoe sensoren en motoren gebruikt kunnen worden. Er wordt echter geen aandacht besteed aan de toepassing van concepten in deze context. Het is volgens kenmerk K2 echter wel een voorwaarde om dat expliciet te doen, zodat leerlingen de concepten gaan begrijpen en kunnen toepassen. Dat heb ik als docent gedaan door concepten mondeling te verduidelijken. Dat bleek erg veel tijd te kosten. Hier is dus duidelijk nog winst te behalen. Bovendien is het beter als leerlingen het zelf nog eens op kunnen zoeken, omdat leerlingen meestal niet veel onthouden als het alleen verbaal uitgelegd is.

Een andere mogelijke beperking die voor dit onderzoek gold was de beperkte tijd waarin de werkvorm toegepast is. Kolodner et al. (2003) beschrijft hoe het hen noodzakelijk leek om een volledige module van enkele weken als oefening uit te voeren. Het doel daarvan was tweeledig. Ten eerste wilden ze de klassencultuur dusdanig veranderen dat er sprake was van onderlinge afhankelijkheid. Vervolgens wilden ze leerlingen ervan doordringen dat je van je fouten kunt leren. Deze instelling is in het bijzonder bij programmeren erg belangrijk. Zelfs expert-programmeurs moeten ervan uitgaan dat ze ongeveer de helft van de tijd bezig zijn met code herschrijven (Bennedsen & Caspersen, 2008).

Bennedsen en Caspersen vermelden in hetzelfde onderzoek dat het daarom vreemd is om alleen perfecte uitwerkingen van opdrachten te laten zien in het lesmateriaal. Ook dit kan een tekort zijn in het lesmateriaal dat eraan bijgedragen heeft dan leerlingen minder conceptuele kennis opgedaan hebben dan gehoopt.

Als laatste beperking noem ik de mate waarin de Trello houvast bood aan leerlingen om Scrum op de juiste manier in te zetten. Om van deze WebApp optimaal gebruik te maken, moesten leerlingen de geschreven Scrum-uitleg erbij houden. Dat is echter een stap die leerlingen makkelijk overslaan. Daardoor werd Trello in veel gevallen suboptimaal gebruikt, wat de leerlingen minder overzicht gaf op het ontwerpproces waarin ze zich bevonden. Wellicht dat papieren formulieren hierbij een uitkomst zouden zijn. Als leerlingen daarin gestuurd worden om de juiste eisen te vertalen naar concrete taken, kunnen ze die daarna in Trello zetten. Het ingevulde formulier kan voor de docent interessante informatie bieden om leerlingen te begeleiden bij het opdoen van conceptuele kennis.

6.3. Aanbevelingen

In een onderzoek dat in dezelfde richting gedaan wordt, is het goed om met bovenstaande beperkingen rekening te houden. Naar aanleiding daarvan heb ik een aantal gedachten die ik graag met u deel. Eerst zal ik ingaan op de manier waarop Scrum gebruikt is, vervolgens op het observeren van conceptualisatie, daarna op de opdrachtbeschrijving en ten slotte zal ik aandacht geven aan het soort uitlegmateriaal dat waarschijnlijk beter zou werken.

6.3.1. Toepassing van Scrum

Om de leerlingen meer houvast te geven bij het werken volgens Scrum, zou het gebruik van stencils/formulieren goed van pas kunnen komen. Dat wordt ook gedaan bij LBD (Kolodner et al., 2003). Dat bleek een hulpmiddel te zijn dat leerlingen erg waarderen. Er waren in hun onderzoek namelijk leerlingen die ook een jaar later nog wel eens langs kwamen om te vragen om een stencil zodat ze het ook bij andere vakken konden gebruiken. Na afronding van het onderzoek heb ik een voorbeeld van een formulier om een Sprint te starten gemaakt (zie bijlage E) dat voortaan bij deze lessenserie gebruikt kan worden.

In dit onderzoek is gewerkt met groepjes van 5 à 6 leerlingen. Dit werd door leerlingen ervaren als te groot. Omdat er ook bij DBS in groepjes van vier leerlingen gewerkt wordt, is het goed mogelijk dat het reduceren van de groepsgrootte naar vier leerlingen zou resulteren in een beter werkproces.

6.3.2. Observatie van conceptualisatie

Ik had de indruk dat het observeren van conceptualisatie goed ging. Het lijkt me echter goed om de validiteit van de observaties te toetsen. Omdat dit onderzoek als doel had om te ontdekken of de Scrum-methode zich er überhaupt voor leent om ondertussen conceptuele vorming te meten, viel dit buiten het kader van deze studie.

6.3.3. Opdrachtbeschrijving

Om nog meer invulling te geven aan kenmerk K2 (maak concepten expliciet), zou in de opdrachtbeschrijving explicieter aandacht gegeven kunnen worden aan het toepassen van concepten. Daardoor zouden leerlingen meer bewust bezig gaan met het aanleren van concepten. Als leerlingen dan ook een overzicht van de te leren concepten krijgen waarop ze ook hun eigen voortgang kunnen bijhouden, zou dat mijns inziens het leerproces veel inzichtelijker maken. Het nadeel van deze verandering is echter dat de authenticiteit van de opdracht er duidelijk minder van wordt, aangezien die dan duidelijk didactisch van aard is.

Om te voorkomen dat leerlingen veel bezig zijn met bouwen van de robot en weinig met het programmeren ervan, kan het ook goed zijn om het scenario aan te passen of een ander scenario te kiezen waarvoor minder aanpassingen aan de robot gedaan hoeven te worden. Als er geen hefmechanisme nodig is voor de opdracht, zou dat waarschijnlijk al veel uitmaken. Als er ook een bouwstructuur zou zijn voor een hefmechanisme, zou dat ook veel tijd kunnen schelen.

6.3.4. Uitlegmateriaal

Het uitlegmateriaal dat leerlingen konden gebruiken bestond uit twee onderdelen. Het ene deel was bedoeld om de taal Small Basic te leren kennen. Het andere om te leren hoe je Small Basic kunt gebruiken om een Mindstorms EV3-robot te programmeren. Daardoor was het in de EV3-uitleg niet duidelijk welke programmeerconcepten er gebruikt werden. De programmeerconcepten moesten afgeleid worden van het uitlegmateriaal dat niet over robots ging.

Het lijkt mij een grote verbetering als er uitlegmateriaal gemaakt zou worden waarbij er met twee zaken rekening gehouden wordt. Ten eerste zouden in dat uitlegmateriaal de programmeerconcepten uitgelegd moeten worden aan de hand van voorbeelden in een robot-scenario met Mindstorms EV3. Volgens Van Merriënboer en Paas (1990) is het namelijk vooral effectief om leerlingen gebruik te laten maken van uitgewerkte voorbeelden die lijken op de oplossing van het probleem waaraan ze zelf werken. Met “uitgewerkte voorbeelden” bedoelen de auteurs “problemen die opgelost worden waarbij iedere stap expliciet uitgewerkt wordt”. Zij stellen ook dat het leerproces versneld kan worden door leerlingen vervolgens voorbeeldprogramma's te geven om te interpreteren en aan te passen.

Ten tweede beveel ik aan om het uitlegmateriaal geschikt te maken om naar te verwijzen. Als een leerling een vraag heeft, moet de docent in de meeste gevallen kunnen zeggen “kijk nog eens goed naar onderdeel X in de uitleg”. Dat was niet goed mogelijk met het gebruikte uitlegmateriaal in dit onderzoek, waardoor zelfs het beantwoorden van eenvoudige vragen over programmeerconcepten de docent veel tijd kostte. Leerlingen zouden dus zelfstandiger moeten kunnen ontdekken wat de principes van de programmeeromgeving en -taal zijn, zodat ze zelf oplossingen kunnen bedenken voor de problemen waaraan ze werken. Vermoedelijk is het daarom beter als het uitlegmateriaal encyclopedisch van aard zou zijn. Daarmee bedoel ik dat het voornamelijk begrippen en principes

beschrijft en daarbij steeds uitgewerkte voorbeelden geeft. Het expliciet maken van concepten, wat volgens kenmerk K2 belangrijk is, kan zo ook gerealiseerd worden in het uitlegmateriaal.

Referenties

- Barrows, H. S. (1985). *How to design a problem-based curriculum for the preclinical years* (Vol. 8). Springer Pub Co.
- Bennedsen, J., & Caspersen, M. E. (2008). Exposing the Programming Process. In J. Bennedsen, M. E. Caspersen, & M. Kölling (Eds.), *Reflections on the Teaching of Programming* (Vol. 4821, pp. 6–16). https://doi.org/10.1007/978-3-540-77934-6_2
- Doppelt, Y., Mehalik, M. M., Schunn, C. D., Silk, E., & Krynski, D. (2008). Engagement and achievements : A case study of design-based learning in a science context. *Journal of Technology Education, 19*(2), 22–39. <https://doi.org/10.1.1.145.1976>
- Examenprogramma informatica havo/vwo*. (2016, September 27). Retrieved from https://www.examenblad.nl/examenstof/informatica-havo-en-vwo-3/2019/f=/examenprogramma_Informatica_havo-vwo.pdf
- Fortus, D., Dershimer, R. C., Krajcik, J., Marx, R. W., & Mamlok-Naaman, R. (2004). Design-based science and student learning. *Journal of Research in Science Teaching, 41*(10), 1081–1110. <https://doi.org/10.1002/tea.20040>
- Koldenhof, E., Jeurig, J., & Ruth, S. (2011). *Rendement van objectgeoriënteerd programmeeronderwijs*. 83.
- Kolodner, J. L. (1992). *An Introduction to Case-Based Reasoning* *. <https://doi.org/10.1007/BF00155578>
- Kolodner, J. L., Camp, P. J., Crismond, D., Fasse, B., Gray, J., Holbrook, J., ... Ryan, M. (2003). Problem-Based Learning meets Case-Based Reasoning in the middle-school science classroom: putting Learning by Design(TM) into practice. *Journal of the Learning Sciences, 12*(4), 495–547. https://doi.org/10.1207/S15327809JLS1204_2
- Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (Moti). (2013). Learning computer science concepts with Scratch. *Computer Science Education, 23*(3), 239–264. <https://doi.org/10.1080/08993408.2013.832022>
- Ministerie van OCW. (2017). Taxonomie van Bloom | Stimulerend signaleren | Informatiepunt Onderwijs & Talentontwikkeling (SLO). Retrieved December 28, 2017, from <https://talentstimuleren.nl/thema/stimulerend-signaleren/rijke-leeractiviteiten/bloom>
- Nuutila, E., Törmä, S., Kinnunen, P., & Malmi, L. (2008). Learning Programming with the PBL Method — Experiences on PBL Cases and Tutoring. In J. Bennedsen, M. E. Caspersen, & M. Kölling (Eds.), *Reflections on the Teaching of Programming* (Vol. 4821, pp. 47–67). https://doi.org/10.1007/978-3-540-77934-6_5
- Pieters, M., Marsman, P., & Taminiau, A. (2015, February 19). *Vernieuwing examenprogramma's bètavakken havo/vwo - Op weg naar de invoering*. Retrieved from <http://downloads.slo.nl/Documenten/SLO-BrochureBetavakken-Op-weg-naar-de-invoering.pdf>
- Schmidt, V. (2018a, February 8). Karakterisering van de vernieuwing. Retrieved January 4, 2019, from Karakterisering van de vernieuwing website: <http://handreikingschoolexamen.slo.nl:80/informatica/algemene-informatie/karakterisering-van-de-vernieuwing>

- Schmidt, V. (2018b, February 15). Subdomein D1, D2: Ontwikkelen, inspecteren en aanpassen. Retrieved January 4, 2019, from <http://handreikingschoolexamen.slo.nl:80/informatica/het-examenprogramma/domein-d/subdomein-d1-d2>
- Schwaber, K., & Beedle, M. (2002). *Agile Software Development with Scrum*.
- Schwill, A. (1994). Fundamental ideas of computer science. *Bulletin -- European Association for Theoretical Computer Science*, 53, 274.
- Tritrakan, K., Kidrakarn, P., & Asanok, M. (2016). The use of engineering design concept for computer programming course: A model of blended learning environment. *Educational Research and Reviews*, 11(18), 1757–1765. <https://doi.org/10.5897/ERR2016.2948>
- Van Merriënboer, J. J. G., & Paas, F. G. W. C. (1990). Automation and schema acquisition in learning elementary computer programming: Implications for the design of practice. *Computers in Human Behavior*, 6(3), 273–289. [https://doi.org/10.1016/0747-5632\(90\)90023-A](https://doi.org/10.1016/0747-5632(90)90023-A)
- Winslow, L. E. (1996). Programming Pedagogy - A Psychological Overview. *ACM SIGCSE Bulletin*, 28(3), 17–22. <https://doi.org/10.1145/234867.234872>

Bijlagen

A: Werken met Scrum in de les

Inleiding

Om een projectopdracht tot een goed einde te brengen, is het noodzakelijk om te werken volgens een ontwikkelmethode. Het doel van een ontwikkelmethode is namelijk om een project beheersbaar te maken, zodat investering van tijd en geld resulteert in redelijke kwaliteit.

In hoofdzaak zijn de meeste softwareontwikkelmethoden in te delen in twee groepen: *watervalmethoden* en *iteratieve methoden*. De **watervalmethode** is een lineair proces; de stappen “ontwikkeling”, “testen”, “acceptatie” en “productie” worden achtereenvolgens uitgevoerd. Als een stap is gezet, kan daar moeilijk op teruggekomen worden, zoals een waterval altijd naar beneden stroomt. **Iteratieve methoden** delen de ontwikkeling van een systeem op in fasen waarin steeds een tussenproduct wordt geleverd dat werkt.

Iteratieve methoden zijn ontstaan als reactie op watervalmethoden. Ze zijn erop gericht om ontwikkeling flexibeler te laten verlopen en om zo snel mogelijk iets te maken dat werkt. Een variant hierop zijn de **agile softwareontwikkelmethoden**. In plaats van een planning gebruiken deze methoden feedback van de klant om te bepalen wat er moet gebeuren in iedere fase.

Scrum is een bekende en veelgebruikte agile softwareontwikkelmethode. Voor deze lesmodule zullen we een aangepaste versie van Scrum gebruiken. Daarom zal hieronder eerst worden uitgelegd hoe Scrum werkt. Vervolgens zal worden uitgelegd wat wij anders gaan doen. Ten slotte wordt er een overzicht gegeven van de Scrum-variant die we zullen gebruiken.

Inhoud

Inleiding	2
Inhoud	3
Scrum.....	4
Rollen.....	4
Artefacten.....	4
Bijeenkomsten.....	5
Toepassing van Scrum in de les.....	7
Rollen.....	7
Artefacten.....	7
Bijeenkomsten.....	7
Overzicht van de Scrum-variant	8
Eerste les	8
Planles.....	8
Werkles.....	8
Demoles.....	8

Scrum

Scrum is een ontwikkelmethode waarin zelforganiserende teams incrementeel een product ontwikkelen. Hiervoor zijn verschillende regels gemaakt voor rollen, bijeenkomsten en artefacten (benodigde onderdelen). Er wordt gebruik gemaakt van Sprints. Dat zijn perioden met een vaste lengte, meestal 2 weken of 30 dagen. Het team probeert iedere Sprint een werkende (en goed geteste) product-uitbreiding te maken.

Voor de eerste Sprint moet het project voorbereid worden door de rollen in het team te verdelen en een eerste idee van het product op te stellen. Onder de volgende kopjes staat achtereenvolgens uitgelegd welke rollen er zijn, welke artefacten Scrum kent en ten slotte welke bijeenkomsten er gehouden worden.

Rollen

Het team bestaat uit leden met allemaal verschillende kwaliteiten. De een kan beter programmeren, de ander is beter in testen en een derde is domeinexpert (kent de context waarin het product gebruikt gaat worden). Het team is qua organisatie onafhankelijk van externen. In overleg met de Product Owner wordt bepaald wat er bereikt moet worden. Hoe het bereikt moet worden is aan het hele team om te beslissen. Er zijn twee bijzondere rollen in Scrum.

Product Owner

De Product Owner is verantwoordelijk voor hoe waardevol het is wat het team maakt. Daarom moet hij een duidelijke visie hebben op wat er met het product gedaan moet kunnen worden. Hij is de baas over de eisen die aan het product gesteld worden (Product Backlog). Hij beslist uiteindelijk hoe eisen uitgewerkt moeten worden. Hij beslist of het project doorgaat, of een product geleverd wordt aan klanten en houdt de klant tevreden en op de hoogte.

Scrum Master

De Scrum Master is ondersteunend voor het Scrum-proces en heeft geen autoriteit. Hij zorgt voor een goede werkomgeving en -sfeer. Hij helpt belemmeringen uit de weg en zorgt ervoor dat het team gefocust blijft op waar ze mee bezig moeten zijn. Dat doet hij door de Scrum-artefacten zichtbaar te houden en door goede ontwikkeltechnieken aan te moedigen.

Artefacten

Scrum maakt gebruik van een aantal artefacten (ondersteunende onderdelen) waarmee wordt aangegeven wat er gebeuren moet. Ze worden hieronder een voor een besproken.

Product Backlog

De Product Backlog is de lijst waarop alle gewenste functionaliteit van het te ontwikkelen product gezet kan worden. Iedere belanghebbende mag dat doen. De Product Owner zal de prioriteit van ieder Product Backlog Item (PBI) bepalen en veranderen zodra hij dat nodig acht.

Ieder Product Backlog Item (PBI) gaat over *wát* het product moet kunnen, niet *hóe* het gerealiseerd moet worden. Op een PBI staat een *definition of done* die in samenspraak tussen het team en de Product Owner opgesteld zijn om technische tekorten te voorkomen. Er kunnen ook Item-specifieke acceptatiecriteria op staan die specificeren wat de klant verwacht. Het team bepaald hoeveel moeite een PBI ongeveer zal kosten. Dit wordt uitgedrukt in een tijdseenheid (uren/dagen) of Story Points (een vast deel aan werk).

PBI's worden vaak geschreven in de vorm van een *User Story*; dat is een beschrijving van een taak die een gebruiker uitvoert met het te ontwikkelen product.

Sprint Backlog

Dit is een selectie van PBI's die door het team en Product Owner is gemaakt. Op basis van deze PBI's worden Sprint Tasks (ST's) opgesteld. Deze beschrijven *hoe* een PBI bereikt gaat worden. De worden in een kolom "te doen" geplaatst, naast PBI waar ze bij horen. Daarnaast staan nog twee kolommen: "gestart" en "gedaan". Als het goed is staan alle ST's aan het einde van een Sprint in de laatste kolom.

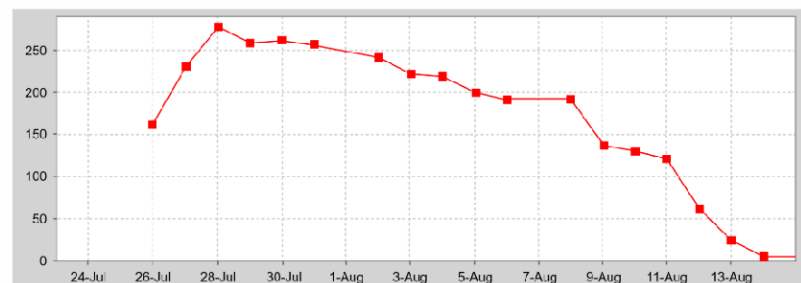
Van iedere ST wordt ingeschat hoeveel tijd het gaat kosten. Deze inschatting wordt iedere keer dat eraan gewerkt is, geüpdatet. Een Sprint Task omvat meerdere van de volgende activiteiten: eisen specificeren, analyseren, ontwerpen, implementeren, toepassen en testen.

Committed Backlog Items	Tasks Not Started	Tasks In Progress	Tasks Completed

de
ST's
het
vaak

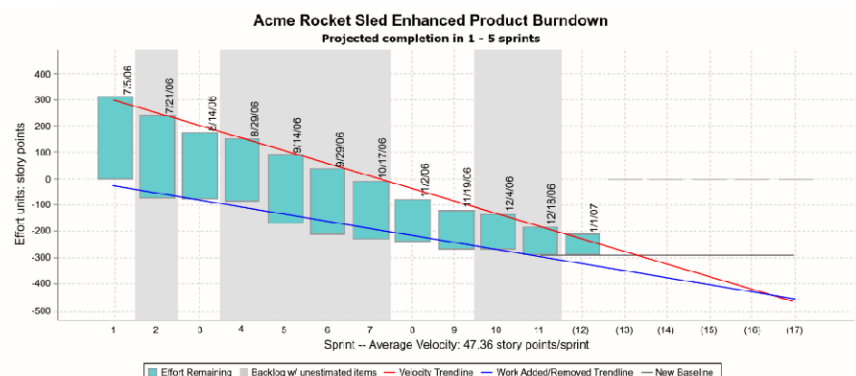
Sprint Burndown Chart

In een Sprint Burndown Chart wordt weergegeven hoeveel tijd een sprint nog gaat kosten, aan de hand van de inschatting van het team. Dit diagram geeft dus een indicatie voor hoe lang de sprint nog ongeveer gaat duren.



Product/Release Burndown Chart

De Release Burndown Chart geeft weer hoeveel tijd of Story Points er nog op de Product Backlog staan. Hierin wordt mogelijk onderscheid gemaakt tussen Story Points die erbij gekomen zijn sinds de laatste Sprint en die er al op stonden.



Bijeenkomsten

Om het Scrum-proces soepel te laten verlopen, zijn er op vaste momenten bepaalde bijeenkomsten (meetings). De Scrum Master faciliteert deze bijeenkomsten. Ze worden hier in chronologische volgorde behandeld.

Sprint Planning Meeting

Aan het begin van iedere Sprint wordt er een Sprint Planning Meeting gehouden. Tijdens deze bijeenkomst wordt bepaald welke PBI's het team deze Sprint zal proberen te realiseren. Het team houdt de haalbaarheid in de gaten, de Product Owner bepaalt de prioriteit van de PBI's. De gekozen PBI's worden in de Sprint Backlog geplaatst. Vervolgens worden de initiële ST's opgesteld in de "te doen"-kolom. Een Sprint Planning Meeting duurt 4 tot 8 uur voor respectievelijk 14 tot 30 dagen.

Daily Scrum/Daily Standup

Iedere dag wordt begonnen met 15 minuten rapportage van de voortgang. Ieder teamlid geeft antwoord op de volgende drie vragen:

1. Wat heb je de vorige keer gedaan?
2. Wat ga je vandaag doen?
3. Waar loop je tegenaan?

De eerste twee vragen zijn bedoeld om uitleg te geven hoe je een ST hebt aangepakt. Alleen vertellen met welke ST je bezig bent is geen informatie, omdat dat ook te zien is op de Sprint Backlog.

De Scrum Master houdt alleen in de tijd in de gaten. Als bepaalde zaken meer tijd nodig hebben, kunnen die na de Daily Scrum worden besproken met een deel van het team.

Wat besproken is, wordt bijgewerkt op de Sprint Backlog. Eventueel wordt ook de Sprint Burndown Chart bijgewerkt en de problemen op een problemen-lijst gezet. Vooral voor problemen die niet (op dat moment) door het team op te lossen zijn is dat handig.

Sprint Review Meeting

Tijdens de Sprint Review Meeting wordt het resultaat van de Sprint gepresenteerd. Dit wordt gedaan middels een demonstratie. De Product Owner zal op basis hiervan aangeven welke PBI's hij voldoende uitgevoerd vindt. De onafgeronde PBI's worden teruggeplaatst op de Product Backlog.

De Scrum Master zorgt ervoor dat feedback op het nieuwe tussenproduct wordt omgezet in nieuwe PBI's en/of wordt gebruikt om bestaande PBI's aan te passen. De Product Owner moet vervolgens de aangepaste Product Backlog opnieuw sorteren op prioriteit.

Sprint Retrospective Meeting

De Sprint Retrospective Meeting wordt door het team gehouden na iedere Sprint. Het doel is om te reflecteren op het ontwikkelproces. Het is belangrijk dat iedereen open durft te zijn, zodat er daadwerkelijk verbeterpunten ontdekt kunnen worden voor het proces.

Backlog Refinement Meeting

Voordat een nieuwe Sprint Planning Meeting gehouden kan worden, is het meestal nodig om de PBI's op de Product Backlog te verfijnen. Vaak zijn ze namelijk te groot/log en te onduidelijk. Zulke PBI's worden tijdens een Backlog Refinement Meeting opgesplitst en verduidelijkt met het oog op wat technisch mogelijk is en wat wenselijk is. De Product Owner en (een deel van) het team zorgen er op deze manier voor dat in ieder geval de bovenste PBI's duidelijk en haalbaar zijn. Vervolgens wordt met het hele team een inschatting gemaakt van de benodigde tijd per PBI.

Toepassing van Scrum in de les

In deze paragraaf wordt beschreven welke veranderingen er gemaakt zijn om Scrum toe te kunnen passen in een lessenserie.

Rollen

Je docent laat jullie een competenties-enquête invullen. Op basis hiervan zal er een aantal Scrum Masters aangewezen worden. Die kiezen vervolgens een team op basis van de enquêtes (waarbij de namen natuurlijk niet zichtbaar zijn).

Je docent vervult de rol van Product Owner. Hij is degene die beslissingen maakt aangaande functionaliteit van het product dat jullie maken.

Artefacten

Aan het begin van het project krijg je een lijst van *User Story's* als Product Backlog. Daarbij krijg je ook een eerste Sprint Backlog, zodat je eerst kan ontdekken hoe Scrum werkt voordat je zelf eisen opstelt. Sommige eisen zijn bedoeld om uit te zoeken hoe je een probleem het beste aan kan pakken, andere om de gekozen oplossing toe te passen en nog weer andere om te testen of de oplossing echt werkt.

Bij iedere ST wordt een indicatie van hoeveel tijd het zal kosten. Hierbij wordt als tijdseenheid kwartieren in plaats van uren gebruikt, omdat een lesuur geen uur duurt, en je ook geen uur bezig wilt zijn met huiswerk. Iedere les heb je tijd voor 2 kwartier en een Daily Scrum (à 10 min.) en thuis kan je ook 1-2 kwartier besteden per les.

Het maken van Burndown Charts is optioneel, zowel voor Sprints als voor het product. Het is een handig hulpmiddel voor jezelf om te zien of je als groep op schema loopt.

Bijeenkomsten

Voor de eerste Sprint is maar een korte Sprint Planning Meeting nodig, omdat de Sprint Tasks al zijn opgesteld. Er moeten nog tweetallen gevormd worden, die samen ST's gaan uitvoeren. Iedere volgende Sprint Planning Meeting zal zo'n 10 minuten duren. Als de ST's zijn toegewezen aan tweetallen, kan de eerste Daily Scrum plaatsvinden: ieder tweetal kiest een ST waar ze mee gaan beginnen en vertelt kort hoe ze het gaan aanpakken.

Iedere volgende les wordt begonnen met een Daily Scrum waarin besproken wordt wat er is gedaan sinds de vorige keer, hoe ieder tweetal verder gaat en welke problemen er zijn. Als er problemen zijn waar je samen niet uitkomt, vraag je je docent om raad en/of hulp.

Een Sprint duurt 4-6 lessen. Tijdens deze lessen is het handig om een Backlog Refinement Meeting te houden, waarin alvast wordt gekeken naar hoe groot en duidelijk de volgende eisen op de Product Backlog zijn. Aan het einde van iedere Sprint wordt er een les besteed aan Sprint Review en Sprint Retrospective Meetings. In deze les geef je een demonstratie aan je docent en bespreek je als groepje hoe het proces verbeterd kan worden. Als er tijd over is, kan je vast beginnen met de Sprint Planning Meeting (PBI's kiezen, ST's opstellen).

Overzicht van de Scrum-variant

Een Sprint duurt 4-6 lessen, inclusief start- en demoles.

Eerste les

De eerste les van de lessenserie. Dit is een planles waarin ook groepjes gevormd worden.

TIJDSDUUR	ACTIVITEIT	BESCHRIJVING
5 MIN.	Enquête invullen	Competentie-enquête invullen
5 MIN.	Groepjes vormen	Docent kiest Scrum Masters, zij kiezen groepsleden.
5 MIN.	Sprint Planning	ST's verdelen (ze zijn al opgesteld)
10 MIN.	Daily Scrum	Aanpak bespreken
15 MIN.	ST's uitvoeren	

Planles

Deze les is het begin van een nieuwe Sprint.

TIJDSDUUR	ACTIVITEIT	BESCHRIJVING
15 MIN.	Sprint Planning	ST's opstellen en verdelen
10 MIN.	Daily Scrum	Aanpak bespreken
15 MIN.	ST's uitvoeren	

Werkles

TIJDSDUUR	ACTIVITEIT	BESCHRIJVING
10 MIN.	Daily Scrum	Voortgang, aanpak en problemen bespreken
15 MIN.	ST's uitvoeren	
15 MIN.	ST's uitvoeren	

Demoles

TIJDSDUUR	ACTIVITEIT	BESCHRIJVING
25 MIN.	Sprint Review	Demonstratiemoment
15 MIN.	Retrospective	Groepsproces bespreken
REST	Sprint Planning	ST's verdelen (ze zijn al opgesteld)

B: Competenties-enquête

Naam:

lnr

naar achteren dubbelvouwen op deze lijn:

Competenties-enquête

Zet de volgende stellingen op volgorde van hoe erg je het ermee eens bent. Doe dit door de cijfers 1 t/m 5 ieder één keer in te vullen. 1 betekent: meest mee eens, 5 betekent: minst mee eens.

<input type="text"/>	Als ik iets doe, doe ik het goed
<input type="text"/>	Ik vind overleggen erg belangrijk
<input type="text"/>	Ik kan goed beslissingen nemen
<input type="text"/>	Ik kan goed "out-of-the-box" denken
<input type="text"/>	Ik houd overzicht bij complexe problemen

Naam:

lnr

naar achteren dubbelvouwen op deze lijn:

Competenties-enquête

Zet de volgende stellingen op volgorde van hoe erg je het ermee eens bent. Doe dit door de cijfers 1 t/m 5 ieder één keer in te vullen. 1 betekent: meest mee eens, 5 betekent: minst mee eens.

<input type="text"/>	Als ik iets doe, doe ik het goed
<input type="text"/>	Ik vind overleggen erg belangrijk
<input type="text"/>	Ik kan goed beslissingen nemen
<input type="text"/>	Ik kan goed "out-of-the-box" denken
<input type="text"/>	Ik houd overzicht bij complexe problemen

Robots in de haven

Casusbeschrijving

In de haven krioelt het van de heftrucks. Ze halen containers van vrachtwagens en treinen en brengen ze naar de kade. Grote kranen laden en lossen de schepen. Van de kade nemen heftrucks weer containers mee naar vrachtwagens en treinen. Dag en nacht gaat het zo door. Constant zijn er mensen aan het werk, hoewel er ook al veel door robots gedaan wordt.

Het havengebied wordt uitgebreid en het heftruck-bedrijf wil bij de nieuwe havens gebruik gaan maken van robots. Deze robots moeten dezelfde taken kunnen uitvoeren als de oorspronkelijke heftrucks. Het hoeven niet per se heftrucks te zijn, als het op een andere manier sneller/veiliger kan, is dat natuurlijk beter!

In dit document worden staan de eisen waaraan deze robots moeten voldoen. Het kan natuurlijk zo zijn dat er nog nieuwe inzichten komen tijdens de ontwikkeling hiervan.

Product Backlog

Containers verplaatsen

De robot moet een container op kunnen pakken, er een eindje mee rijden en hem vervolgens op een andere locatie neerzetten.

Acceptatiecriteria

- Er mogen geen botsingen gemaakt worden met obstakels die er staan.
- De robot mag nooit in het water komen.
- Er mag nooit een container vallen.

Waarschuwingsgeluid bij achteruitrijden

Als de robot achteruit rijdt, moet er een geluidssignaal gegeven worden.

Containers drie hoog stapelen

De robot moet containers op kunnen stapelen tot een maximale hoogte van 3 op elkaar. Hij moet ook van gestapelde containers de bovenste kunnen pakken om te verplaatsen.

Container zoeken op kleur (bestemming)

Iedere container heeft een bestemming. Die staat op een chip aan de bovenkant van iedere container. Voor deze opdracht is het gebruik van chips te ingewikkeld, en wordt er daarom gebruik gemaakt van de kleur van een container.

Op basis van de kleur van de container moet deze dus naar een bepaalde plek gebracht worden. Deze plekken zijn aan het begin van de haven en hebben ieder ook een kleur.

Containers sorteren op kleur (bestemming)

Het kan voorkomen dat er even geen vrachtwagen of trein gereed staat om een container van een bepaalde kleur mee te nemen. Dan moeten eerst alle nog lege plekken gevuld worden. Als er alleen nog containers staan waarvoor er geen verder vervoer klaarstaat, moeten die toch naar voren gebracht worden, zodat ze klaar staan als er een vrachtwagen of trein binnenkomt. Daar moeten ze dan gesorteerd staan op kleur in stapels van drie hoog. Zodra er voor een van deze containers vervoer beschikbaar komt, hebben deze voorrang op de containers die nog in de haven staan.

