



# Praktijkopdracht B1

“Vertaalopdracht”

## Over dit document

---

Blok: B  
Opleiding: Software Developer / Breda  
Week: 1 t/m 2

### Versiebeheer

Versie	Datum	Auteur	Aanpassingen
0.1	29-6-2020	Tim Lutt	Eerste opzet document
1.0	05-01-2021	Tim Lutt	Praktijkopdracht klaar voor review
2.0	16-07-2021	Wicher Hulzebosch	Start met 'Vertaalopdracht' (nieuwe B1-opdracht)
2.1	02-02	Bart Roos	Klaarzetten nieuwe blok
2.2	30-01	Eva van Os	Leesbaarheid aangepast

## Inhoudsopgave

---

I. Over deze praktijkopdracht.....	3
II. Studiewijzer .....	3
III. Afronding en beoordeling .....	3
1. Deelopdracht: Vertalen .....	4
2. Deelopdracht: Zoek de verschillen .....	10
Bijlage A: Vertalingen Python ↔ C#.....	17
IV. Voor de docent .....	19
V. Kwaliteit en verbeteracties .....	19

# I. Over deze praktijkopdracht

## Het doel van de opdracht

Je gaat in deze opdracht werken aan je eerste C# applicatie, waarbij we de nadruk leggen op de verschillen en overeenkomsten met wat je al kent: Python.

Aan het einde van deze twee weken heb je een goed beeld van C#, ken je wat eenvoudige commando's en kun je een eenvoudig Pythonprogramma omzetten naar C#.

## Samenhang met modules

In Blok A heb je veel met Python gewerkt. We zullen deze kennis gebruiken om met C# te beginnen.

## Materiaal- en middelenlijst

Materiaal	Versie	Toelichting
GitHub account	n.v.t.	Nodig om een 'fork' van de bestaande website te maken. <a href="https://github.com">https://github.com</a>
<b>Een browser, zoals:</b>		
(Optioneel) Mozilla Firefox	77 (of hoger) 64-bit	Je collega-ontwikkelaars hebben de site gemaakt en getest voor Mozilla Firefox.
(Optioneel) Google Chrome	80 (of hoger) 64-bits	De site zal ook moeten werken op Google Chrome, aangezien meer dan 60% <sup>1</sup> van de webbezoekers Chrome gebruiken
Microsoft Visual Studio	20	

# II. Studiewijzer

Week	Deelopdracht	Afronding
1	Deelopdrachten 1 en 2	Inleveren drie onderdelen, zie hfdst. III
2		

# III. Afronding en beoordeling

	Werkboek	Bijlage A	Eindopdracht 2.6
<b>Voldaan</b>	Vragen in het werkboek ingevuld	Volledig ingevuld	Ingeleverd en werkt
<b>Deels voldaan</b>	Eén van de drie zaken niet in orde		
<b>Niet voldaan</b>	Meer dan een van de drie niet in orde		

## Leeruitkomsten

13.08	Je kunt zelfstandig je code <u>debuggen</u> ; je leest foutmeldingen nauwkeurig en je gaat systematisch te werk om een fout op te sporen. Waar nodig schakel je hulp in.
CSD.01	Je kunt werken met Visual Studio (project maken, bestaand project openen, code delen, enzovoort).
CSD.02	Je kunt het basisconcept 'variabelen toepassen in C#.
CSD.03	Je kunt het basisconcept 'keuzes' toepassen in C#.
CSD.04	Je kunt het basisconcept 'herhalingen toepassen in C#.

# 1. Deelopdracht: Vertalen

---



Je hebt een stageplek gevonden bij het *Het meest leerzame pretpark voor het hele gezin*: DeveloperLand! Je gaat als programmeur aan de slag op de Developmentafdeling. Samen met andere collega's schrijf je software voor de attracties, administratie, HR: iedereen die software nodig heeft komt uiteindelijk bij jouw afdeling terecht.

Op je eerste dag meld je je bij de receptie en de receptionist belt je manager. Terwijl je van de automatenkoffie geniet komt de DevOps Manager de receptie binnen stormen.

"Ah, daar ben je! Euh, ja, loop maar even mee."

En weg is hij al. Je pakt snel je spullen en rent achter hem aan.

"We hebben een grote storing in de Turing Machine, een van onze attracties, dus ik heb weinig tijd om je wegwijs te maken

vandaag. Hier is jouw bureau. Normaal zitten er meer mensen op deze afdeling, maar iedereen is op het moment bij de Turing Machine. Euh... Ja... Euh... Jij had Python gehad op je opleiding toch? Dan weet ik wel een klusje."

Hij rommelt wat op een laptop en tovert een aantal Python projecten tevoorschijn.

"Hier, dit zijn alle applicaties die we binnen DeveloperLand gebruiken. Allemaal oude meuk, geschreven in de tijd dat we hier nog maar met z'n 12'en werkten. Het werkt allemaal nog, maar... Naja, ideaal is anders... De accountant zit al jaren te zeuren dat we onze administratie eigenlijk niet meer met deze scripts mogen doen. Hij begint er iedere keer over en blijkbaar kan iedereen de code van die scripts gewoon aanpassen en zo frauderen. Hij wil heel graag dat we overstappen op echte applicaties... Executables noemde hij het geloof ik. Er heeft al wel 's iemand een beginnetje gemaakt, maar die kon geen Python. Misschien kom jij er wel uit.

## Code lezen

Je hebt de opdracht gekregen om een Pythonprogramma om te zetten naar een C#-programma, maar je kunt nog helemaal geen C#! Omdat je toch even niets anders kunt doen zolang je manager nog weg is, besluit je om toch vast een blik te werpen op de programma's die je manager vertaald wilde hebben.

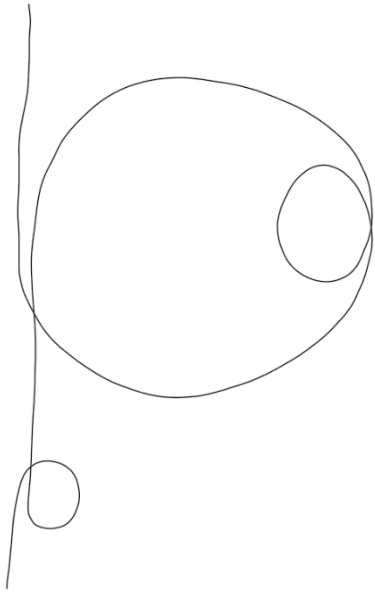
**Stap 1:** Clone de repo [https://github.com/curio-lesmateriaal/B1\\_Vertaalopdracht\\_DO1\\_PyReg](https://github.com/curio-lesmateriaal/B1_Vertaalopdracht_DO1_PyReg)

**Stap 2:** Open het project in Visual Studio (open PyReg/PyReg.sln) > Check of regelnummering aan staat!

**Stap 3:** Voordat we iets gaan vertalen, willen we eerst weten wat dit programma precies doet. Helaas is de inwerkdokumentatie niet veel meer dan een papertje waarop staat waar elk programma voor wordt gebruikt. Bekijk hoofdstuk 3 van het Personeelshandboek (zie itslearning).

**Stap 4:** Voer het programma uit, en test het (**grondig**), zodat je weet wat het doet, hoe kassiers/kassières het programma gebruiken, etc.

Je gaat nu de code van PyReg doornemen. Een Senior-developer heeft de hoofdstructuur van het programma voor je geschetst, vlak voordat hij ook naar de storing werd geroepen. In het eerste blok heb je al wat codeer structuren geleerd. De vragen hieronder gaan over één zo'n structuur.



*Figuur 1: Een bekende structuur, herken jij hem?*

**Vraag 1:** Welke structuur zie je in figuur 1? Hoe wordt deze structuur genoemd?

Loops

---

---

---

**Vraag 2:** Kijk naar het programma in VS. Op welke regel in de code begint de grote cirkel? Hoe vaak of tot wanneer wordt de code in dit blok uitgevoerd?

De loop start op regel 12 en blijft gaan totdat de keuze 9 is  
ingevuld in het hoofdmenu

---

**Vraag 3:** Op welke regel in de code begint de cirkel binnenin de grote cirkel? Hoe vaak, of tot wanneer, wordt deze code uitgevoerd?

De loop start op regel 27 en gaat door totdat de keuze 9 of Z is ingevuld

---

---

**Vraag 4:** Op welke regel in de code begint de cirkel onderaan? Hoe vaak, of tot wanneer, wordt deze code uitgevoerd?

Start op regel 86 en gaat door totdat de variabele 'inKassa' gelijk is aan 'bedragInKassaBegin' +  
'dagTotaal' - 'dagTotaalTerug'

---

---

Om het voor jezelf makkelijker te maken, wil je nu comments in de code plaatsen. Een comment vertelt in “menselijke” taal wat een stukje code moet doen. Soms gaat een comment over één regel, soms over een heel stuk code: Wanneer je bijvoorbeeld 20 regels nodig hebt om een menu te tonen, ga je niet bij iedere regel een comment zetten: Je kunt dan gewoon bovenaan zetten “Toon menu”. Aan de andere kant zijn er regels code die zo onduidelijk zijn, of waarin zoveel op één regel gebeurt, dat het nodig is om één comment (soms van méérdere regels!) aan één regel code te wijden.

**Stap 5:** Hieronder zie je een aantal comments staan, die in de code horen. Zet de comments op de juiste plek in de code.

```
# Verwerk retourbedrag in dagtotaal
# Gebruiker wil het programma afsluiten, vraag om bedrag in kassa
# Laat de gebruiker een keuze maken
# Herhaal tot de gebruiker aangeeft dat het programma afgesloten moet worden
# Toon retourbedrag
# Toon bon-menu
# Voeg één volwasseneticket toe aan de bon
# Voeg een kinderwagen/bolderkar toe aan de bon
# Start programma, stel de nodige variabelen in
# Kassa is correct afgesloten, toon dagtotalen
# Voeg een parkeerkaart toe aan de bon
# Gebruiker wil de dagtotalen zien
# Voeg een Developerland-kaart toe aan de bon
# Vraag gebruiker om keuze te maken
# Gebruiker wil een nieuwe bon maken, stel variabelen voor bon in en toon menu
# Voeg een familiepas toe aan de bon
# Gebruiker wil de bon annuleren; reset bon variabelen en keer terug naar het hoofdmenu
# Voeg één kinderkaartje toe aan de bon
# Vraag om bedrag en reden
# Bon is klaar, toon totaal en reken af
# Gebruiker wil bedrag retourneren
# Herhaal zolang het opgegeven bedrag in kassa niet overeenkomt met het berekende bedrag in kassa
# Toon het hoofdmenu
```

Net nadat je klaar bent komt één van je collega's de afdeling opgevlogen.

"Snel, weet jij waar de... Huh? Wie ben jij?"

Je stelt jezelf voor en verteld dat je bezig bent met het vertalen van alle Python scripts naar C#.

"Ow, daar was Diederik ook al aan begonnen, onze vorige stagair... Geen idee hoe ver hij is gekomen, maar kijk..."

Je collega opent Github en opent een project waarin een van je voorgangers al is begonnen met het vertalen van de PyReg-app.

**Stap 6:** Clone [https://github.com/curio-lesmateriaal/B1\\_Vertaalopdracht\\_DO1\\_C-Reg](https://github.com/curio-lesmateriaal/B1_Vertaalopdracht_DO1_C-Reg)

**Stap 7:** Open het project in Visual Studio door het bestand CSharpReg/CSharpReg.sln te openen

Bovenaan het programma zie je het volgende staan:

```
using System;

namespace CSharpReg
{
    class Program
    {
        static void Main(string[] args)
        {
```

Dit mag je (voor nu) vergeten. Voor nu mag je er even vanuit gaan dat de code 'begint' op regel hier direct onder. Alles wat je daar aan code neerzet, wordt uitgevoerd wanneer je het programma start.

De eerste regel (r9) die je dan ziet, is deze:

```
Console.WriteLine("==== C#Reg =====");
```

**Vraag 5:** Zoek in PyReg een regel code op die hier héél erg op lijkt. Wat denk je dat deze regel code in C# doet?

Print de tekst tussen de aanhalingstekens

---

---

---

**Vraag 6:** Wat zou 'Console.WriteLine' in C# betekenen, als je het naar Python 'vertaalt'?

```
p r i n t ( )
```

---

---

---

Dan kom je deze code tegen:

```
Console.WriteLine("Bedrag in kassa?");  
decimal bedragInKassaBegin = decimal.Parse(Console.ReadLine());
```

**Vraag 7:** Wat doet de tweede regel (r13) dan, denk je?

Zet de input om naar een kommagetal ipv een string

**Vraag 8:** Welke regel(s) in Python komen hiermee overeen?

```
i n p u t ( )
```

Je ziet hier de Python- en C#-code onder elkaar. In de tabel staat de code, maar dan opgehakt in kleine stukjes.

```
bedragInKassaBegin = float(input("Bedrag in kassa? "))
```

```
Console.WriteLine("Bedrag in kassa?");  
decimal bedragInKassaBegin = decimal.Parse(Console.ReadLine());
```

**Stap 8:** Verbind in de tabel hieronder alle C#-commando's met het bijbehorende Python-commando.

Beschrijving	Pyton	C#
Maak een nieuwe variabele	bedragInKassaBagin	Console.ReadLine()
Zet om naar datatype	float(..)	decimal bedragInKassaBegin
Toon vraag aan gebruiker en vraag om input	input("Bedrag in kassa? ")	Console.WriteLine("Bedrag in kassa?");
		decimal.Parse(..)



**Vraag 9:** Waarom staan er bij C# meer commando's dan bij Python denk je?

Omdat C# een typesafe taal is, je moet dus alle types ook omzetten als je het wilt opslaan

---

---

**Stap 9:** In bijlage A vind je een 'Woordenboek Python C#': Een beschrijving van wat je wilt doen, met daarbij de bijbehorende Python en C# code. Vul bijlage A aan met wat je tot nu toe geleerd hebt.

## 2. Deelopdracht: Zoek de verschillen

---

### 2.1 Comments

Je hebt nu geleerd dat *print* in Python *Console.WriteLine* in C# betekent, een beetje zoals *oui* in het Frans, *ja* is. Maar er zijn meer verschillen.

**Stap 1:** Voordat je verdergaat met het programma, wil je duidelijk de structuur aangeven in C#. Plaats **één** van de comments uit deelopdracht 1 in je C#-code.

**Vraag 1:** Wat gebeurt er wanneer je een python comment in C# code plaatst?

Het geeft een error, de comments in C# zijn // ipv een #

---

---

---

**Vraag 2:** Wat betekenen rode kringeltjes onder je code?

Dat die tekst een error is/geeft

---

---

---

**Vraag 3:** Blijkbaar werken comments in C# anders dan in Python, want deze comments zorgen voor heel veel fouten. Je moet dus eerst uitzoeken hoe je in C# een comment plaatst. Welke zoekterm(en) zou je gebruiken om uit te vinden hoe je in C# een comment plaatst?

"C# comment characters"

---

---

---

**Stap 2:** Google nu hoe je in C# comments plaatst. **LET OP:** In C# gebruik je soms *Documentation Comments*; deze zijn anders dan "normale" comments. Let erop dat je de juiste zoekresultaten bekijkt om te weten hoe je een normale comment plaatst.

**Vraag 4:** Hoe plaats je in C# een comment van één regel?

// <tekst>

---

---

---

een documentatie comment is `"/>`

**Stap 3:** Vul nu bijlage A aan met de nieuwe kennis die je hebt opgedaan.

## 2.2 Scoping

**Vraag 1:** In de C#-code zie je heel vaak een accolade voorbijkomen: { en }. Rond welke structuur code staan deze? Waarvoor dienen de accolades?

ze staan rond blokken code dat een groep vormt, het word gebruikt na bvb een if statement  
ipv bij python tabs te gebruiken gebruik je tab indents

**Vraag 2:** In Python gebruik je geen accolades. Maar hoe wordt dit wel gedaan in Python?

tab indents

**Vraag 3:** Selecteer alle C#-code en druk op <SHIFT> + <TAB> totdat alle code helemaal tegen de linkerrand staat. In Python zou je programma nu niet meer goed werken. Test het programma in C#; wat gebeurt er?

het werkt nog gewoon want C# gebruikt accolades ipv indentation

**Vraag 4:** Op welke regel begint de kleine cirkel binnen de grote cirkel? Was het makkelijker op te zoeken met, of zonder tabs?

Tabs maken dus voor je C# code niets uit, maar voor de leesbaarheid is het belangrijk dat je je code goed uitlijnt. Gelukkig helpt C# je met een hoop van dit soort zaken.

**Vraag 5:** Ga nu in Visual Studio naar Edit > Advanced en kies voor "Format Document". Wat gebeurt er?

alles word weer netjes neergezet met indents

**Stap 1:** Vul nu bijlage A aan met de nieuwe kennis die je hebt opgedaan.

## 2.3 Variabelen

**Vraag 1:** Hoe wordt in Python de variabele dagTotaal aangemaakt?

dagTotaal = 0

---

---

---

**Vraag 2:** Met welke code wordt dit in C# gedaan?

decimal dagTotaal = 0;

---

---

---

Python is een zogenaamde *dynamically typed* programmeertaal, omdat je het datatype niet hoeft op te geven. Dat betekent dat de code voor het aanmaken van een variabele met een stukje tekst erin, precies hetzelfde is als het aanmaken van een variabele met een integer: alleen de waarde die je erin opslaat is anders.

In C# is het echter noodzakelijk dat jij opgeeft welk datatype een variabele moet hebben. Dit heeft voor- én nadelen: in Python kun je een variabele waarin je net een integer hebt opgeslagen, later hergebruiken voor het opslaan van een stukje tekst; in C# zou je dan echter een foutmelding krijgen. C# noemen we daarom dan ook wel een *static typed* programmeertaal.

**Vraag 3:** Welke voordelen kun je bedenken van een *static typed* programmeertaal?

het is type-safe waardoor je niet perongelijk een string bij een int opteld bvb

---

---

---

**Vraag 4:** Waarom zou je een foutmelding willen krijgen wanneer je een stukje tekst opslaat in een variabele die eigenlijk gereserveerd is voor een integer?

omdat een int alleen voor volledige getallen is

---

---

---

**Stap 1:** Vul nu bijlage A aan met de nieuwe kennis die je hebt opgedaan.

## 2.4 If, loops

**Vraag 1:** Hoe schrijf je in Python een if-statement om te controleren of een variabele gelijk is aan 3?

`if ( <varNaam> == 3) { <code > }`

---

---

---

**Vraag 2:** Zoek een if-statement op in C#Reg. Hoe ziet dat eruit? Wat zijn de verschillen?

het gebruikt accolades ipv een dubbelepunt en tab indentation

---

de else if statements zijn volledig uitgescreven ipv elif

---

---

**Vraag 3:** Wat is de code om in Python een 'else' en 'else if' blok te maken?

`elif ()`

---

---

---

**Vraag 4:** Zoek nu een else en else if-statement op in C#Reg. Wat zijn de verschillen met Python?

`else if () { }`

---

het verschil is dat het is uitgetyped

---

---

**Vraag 5:** In C#Reg zie je twee keer de volgende comment staan:

```
// TODO: If-statement corrigeren
```

Bekijk de if-statements onder deze regel. Je ziet hier kringeltjes om het if-statement staan. Dit betekent dat er een foutje in de code zit. Wat is de foutmelding die je op deze regels krijgt?

het checkt of keuze gelijk is aan een int getal, maar keuze is een string variable dus dat kan niet

---

---

**Vraag 6:** Wat denk je dat deze foutmelding betekent?

het getal moet tussen aanhalingstekens

---

---

---

**Vraag 7:** Vergelijk deze twee if-statements met andere if-statements in het programma. Zie je een verschil?

ja, de getallen staan niet tussen aanhalingstekens, bij de rest wel

---

---

---

**Stap 1:** Los de foutmelding op door de if-statements aan te passen.

**Stap 2:** Vul nu bijlage A aan met de nieuwe kennis die je hebt opgedaan.

**Vraag 8:** Zoek nu één of meer loops op in PyReg en vergelijk die met de loops in C#Reg. Wat zijn de verschillen?

alles in de C# loops zit in een accolade { }

---

---

---

## 2.5 Casting

Zoals we eerder al gezien hebben is C# *statically typed*.

**Vraag 1:** Leg in je eigen woorden uit wat *statically typed* betekent. Als je het niet meer weet, probeer het dan op internet op te zoeken.

het is een soort van type-safety om ervoor te zorgen dat je niet een string + een int doet bvb

---

---

---

**Vraag 2:** Welke datatypes ken je allemaal?

int, boolean, string, char, float, double, decimal, array, object, long, short, tinyint

---

---

---

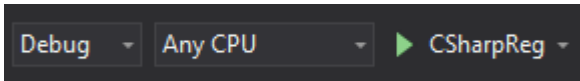
**Stap 1:** Zoek van ieder datatype uit hoe je een variabel aanmaakt. Vul het onderstaande schema aan door de naam van het datatype in te vullen en een voorbeeld van hoe je zo'n variabele aanmaakt. Je mag de naam en de waarde zelf bedenken.

Datatype	Naam	C# code
Heel getal	Int	Int dezeInt = 3;
Tekst	string	string hello = "hello world";
Komma getal	decimal	decimal kommaGetal = 21.3;
Waar / niet waar	bool	bool isBoolean = true;
Één letter	char	char singleCharacter = 'K';
Lijst van items	<type>[]	string[] names = ["Koen", "Daan", "Myron"]

## 2.6 Eindopdracht

**Stap 1:** Zoals je ziet zijn er twee onderdelen in C#Reg nog niet af (als je ze nog niet gevonden hebt, zoek dan op de comment "TODO: Afmaken"). Bekijk wat er op deze plekken in Python gebeurt en probeer met de kennis die je tot nu toe van C# hebt opgedaan de blokken af te maken.

**Stap 2:** Als het goed is werkt het programma nu helemaal. Zoals je manager al zei, wil hij graag met executables werken. Gelukkig kan Visual Studio dat eenvoudig voor ons doen. Bovenin Visual Studio zie je Debug staan, met een pijltje erachter. Klik op het pijltje en kies voor Release.



**Stap 3:** Ga in het menu naar Build en klik op "Build Solution".

**Stap 4:** Open nu je projectmap, ga dan naar de map 'bin', 'Release', 'netcoreapp3.1'

**Vraag 1:** Je ziet hier nu een aantal bestanden, waaronder CSharpReg.exe. Dubbelklik op CSharpReg.exe. Wat gebeurt er?

er opent een console window waar het programma op runt

---

---

---

**Gefeliciteerd!** Als dit gelukt is, heb je je eerste stand-alone app gebouwd! Je kunt dit programma op andere computers uitvoeren, simpelweg door alle bestanden uit deze map te kopiëren.

**Stap 5:** Zet in Visual Studio nu 'Release' weer terug naar 'Debug'.

**Stap 6:** Je hoeft nu je project alleen nog maar in te leveren, maar let op: Visual Studio maakt allerlei bestanden en mappen aan, die allemaal onderdeel zijn van het project. Ga in Windows verkenners naar je projectmap. Als het goed is zie je in de map waarin je de git-repo hebt gecloned de map CSharpReg, met daarin de volgende bestanden:

```
C# CSharpReg.csproj
  CSharpReg.sln
  Program.cs
```

**Stap 7:** Wanneer je gevraagd wordt om je hele project in te leveren, moet je altijd deze hele map zippen en inleveren. Selecteer alle bestanden en mappen in deze map, en zip ze door er met rechtermuisknop op te klikken en dan te kiezen voor 7Zip > Toevoegen aan Archief.

**Stap 8:** Upload het .zip-bestand op itslearning.



## Bijlage A: Vertalingen Python ↔ C#

Omschrijving	Python	C#
<i>If - statement</i>	<i>If a &lt; b :</i>	<i>If (a &lt; b)</i>
<i>Scoping</i>	<i>Met tabs</i>	<i>Tussen accolades: { en }</i>
<i>Comments</i>	<i># &lt;tekst&gt;</i>	<i>// &lt;tekst&gt;</i>
<i>Einde van iedere regel</i>	<i>niks</i>	<i>puntcomma ( ; )</i>
<i>Tekst op het scherm weergeven</i>	<i>print()</i>	<i>Console.WriteLine()</i>
<i>Input van de gebruiker vragen</i>	<i>input()</i>	<i>Console.ReadLine()</i>
<i>variable definiëren</i>	<i>&lt;varnaam&gt;</i>	<i>&lt;type&gt; &lt;varnaam&gt; ( string HelloWorld )</i>
<i>Clear console screen</i>	<i>os.system("cls")</i>	<i>Console.Clear()</i>
<i>If / Else If / Else</i>	<i>if (): / elif (): / else:</i>	<i>if () {} / else if () {} / else {}</i>

Omschrijving	Python	C#

## IV. Voor de docent

---

Bespreken nadat studenten dit hebben uitgevoerd:

- Plaatsen van Comments (3.1 stap 4)
- Invullen van woordenboek (laatste stap van 3.1)

## V. Kwaliteit en verbeteracties

---

curio