

Software Development Project I (ac 21-22)

Table of Contents

1. Voorwoord	1
2. Deelname aan het project	2
3. Planning gedurende het semester - zie welkomstekst in cursus	2
4. Contactmomenten	2
5. Puntenverdeling	3
5.1. Product (eindresultaat)	4
5.2. Productontwikkeling	4
5.3. Proces	5
6. Eindproducten	6
7. Opdracht	8
8. Tussentijdse presentatie en eindpresentatie	9
9. Een aantal extra's	9
9.1. Projectopgave Zatre	9
9.2. Overzicht criteria met beschrijving levels	13
9.3. Logboek	14
9.4. Opvolgingsformulier	14

1. Voorwoord

Beste student

Welkom in semester 2!

Tijdens dit semester werk je in team aan een opdracht waarin heel wat vaardigheden van een informaticus aan bod komen. Je leert door middel van een **vakoverschrijdende opdracht** in team werken maar ook zelfstandig informatie opzoeken. Op die manier verwerf je kennis en zet je die ineens om in de praktijk.

In deze brochure vertellen we je alles over dit opleidingsonderdeel Software Development Project I. We proberen klaarheid te scheppen in de chaos die vaak – ook in het werkveld – gepaard gaat met het opstarten van een project.

Op het einde van het project beoordelen wij niet enkel het eindproduct. We houden ook rekening met je individuele inbreng, je presentatie, je werkhouding, je rol en bijdrage in de groep en je vorderingen tijdens het verloop van het project. Voor dit opleidingsonderdeel voorzien we 4 studiepunten.

Projectwerking sluit goed aan bij wat de Amerikaanse pedagoog John Dewey ooit zei: ‘een school moet geen voorbereiding zijn op het leven van later, ze moet het leven zelf zijn’. Door te werken aan dit project kan je je voorbereiden op je professionele leven van later (lees: binnenkort): competent individueel en als teamplayer handelen in een voortdurend veranderende beroepspraktijk.

Ik wens je, namens alle lectoren die instaan voor de begeleiding en coaching, alvast veel succes!

Liesbeth Lewyllie Coördinator Software Development Project I

2. Deelname aan het project

In deze brochure vind je de globale projectopdracht en uitleg over de projectwerking.

Dit project is vakoverschrijdend. De leerstof gezien in de opleidingsonderdelen **Databases**, **Software Analysis** en **Object-oriented Software Development I** moeten hier toegepast worden. Ook hetgeen je dit semester leert in **Object-oriented Software Development II** moet onmiddellijk in praktijk omgezet worden.

Een projectgroep bestaat uit vier/vijf studenten. Iedereen in de projectgroep werkt mee aan elk onderdeel (analyse, ontwerpen, testen, programmeren, databank).

Deelnemen aan het project betekent dat je:

- wekelijks ongeveer 10u werkt aan dit project (week 2 - week 10)
- aanwezig bent op de verplichte contactmomenten

3. Planning gedurende het semester - zie welkomstekst in cursus

Wee k	TODO
2	online infosessie bijwonen + werkomgeving opzetten (uitleg documenten + instructiefilmpjes) + starten met use case 1
3 - 10	wekelijkse opvolging via het opvolgingsformulier en het logboek
11	indienen eindresultaat van het project + eindpresentatie

4. Contactmomenten

De beoordeling van het opleidingsonderdeel gebeurt op basis van **permanente evaluatie**. Dit betekent dat je dus voortdurend geëvalueerd zal worden tijdens de contactmomenten. Elke afwezigheid moet je dan ook wettigen. Indien je onmogelijk een bepaalde afspraak kan nakomen, verwittig dan de begeleidende lector én je groepsgenoten. De afwezigheid moet officieel gemeld worden (ten laatste op de dag van de afspraak zelf) via iBamaFlex, binnen de 48u na de melding

moet de correcte wettiging opgeladen worden.

Studenten die wel aanwezig zijn op alle afspraken, maar omwille van dwingende redenen afwezig zijn op de eindpresentatie, krijgen een AFW (afwezig) voor de evaluatie van de eerste zitting. Deze studenten worden verwezen naar de tweede zitting en halen tijdens de feedback de opgave voor de tweede zitting op.

Wat gebeurt er heel concreet bij **ongewettigde afwezigheden**?

De punten worden negatief beïnvloed door elke ongewettigde afwezigheid:

- je progressie voor die periode kan niet beoordeeld worden, hierdoor daalt je eindcijfer.
- punten groepswork dalen (pas op: dit cijfer neem je mee naar 2de zit, zie puntenverdeling)

Wat gebeurt er als je aanwezig bent maar **onvoldoende betrokken** bent bij het project?

Elke student moet voldoende bijdragen aan het project. **Een gebrek aan betrokkenheid kan ook resulteren in een vermelding afwezig voor dit opleidingsonderdeel.**

Bij een gebrek aan betrokkenheid krijg je als student een uitnodiging tot een persoonlijk gesprek met je begeleidende lector. De andere groepsleden kunnen ook uitgenodigd worden op dit gesprek. Indien je afwezig bent op dit gesprek of het leidt niet tot een terug correcte betrokkenheid in het project, krijg je die vermelding afwezig. Elk teamlid wordt hier via mail van op de hoogte gebracht. Om de mails te versturen, gebruiken de lectoren het standaard communicatiekanaal, nl. Chamilo.

Deze informatie kan je nalezen in het DOER:

- Artikel 4: Deelname aan onderwijs- en evaluatieactiviteiten (paragraaf 1,2 en 3)
- Artikel 6: Gevolgen bij afwezigheid bij evaluatie

Dit document kan je terug vinden op volgende locatie:

https://hogent.be/sites/hogent/assets/File/DOW/Bijlage%204_DOER%20DIT_2021-20.pdf

Algemeen Onderwijs- en examenreglement:

https://www.hogent.be/sites/hogent/assets/File/DOW/Onderwijs-%20en%20examenreglement_2021-2022_departementen%20en%20GO5.pdf

5. Puntenverdeling

De uiteindelijke verdeling van het totaalcijfer:



Hierbij wordt rekening gehouden met volgende criteria:

5.1. Product (eindresultaat)

Criterion 1: Volledigheid product qua vorm

1. einddossier met
 - inhoudstafel
 - genummerde blz.
 - per UC analyse, ontwerp, databank
 - reflectie ivm groepswork van elk groepsid
2. Eclipse project met code en VP UML-diagrammen
3. gegenereerde javadoc API van het domein

Criterion 2: Volledigheid product qua functionaliteit

Alle functionaliteiten zijn beschreven in de use cases.

Criterion 3: Volledigheid product qua niet functionele requirements

- user interface (console / grafisch)
- meertalige applicatie

5.2. Productontwikkeling

Criterion 1: Analyseren

Producten zijn:

- activity diagram
- domeinmodel
- SSD, bijhorende OC's

conform de regels toegelicht in Software Analysis en conform de opgave

Criterium 2: Ontwerpen

Opstellen van het DCD:

- DCD sluit aan op SSD's
- DCD bevat wat beschreven werd in OC

conform de regels toegelicht in OOSDI en conform de opgestelde analyse

Criterium 3: Testen

Elke "pure" domeinklasse, d.w.z. NIET DomeinController en repositoryklassen, wordt via JUNIT testen voldoende getest.

Criterium 4: Programmeren

Kwaliteitsvol implementeren betekent:

- respect voor OO
- respect voor de 3 lagen
- correct documenteren doorheen volledige proces
- robuuste applicatie
- in de code vind je de juiste methodes vanop het DCD terug

conform de regels toegelicht in OOSDI, OOSDII én conform het gemaakte ontwerp

Criterium 5: Databank

Eigen opgestelde databank voldoet aan:

- conceptueel model ERD
- overeenkomstig relationeel model
- overeenkomstige implementatie in relationele databank

Beheren betekent:

- lezen van data
- schrijven van data

conform de regels toegelicht in Databases

5.3. Proces

Criterium 1: Registreren

Logboek en opvolgingsformulier aanvullen.

Criterion 2: Plannen

Wekelijkse planning opstellen, werken volgens die planning en controleren of je op schema zit.

Criterion 3: Teamwork

Goed samenwerken en dit ook stimuleren.

Criterion 4: Voorbereiding wekelijks contactmoment

Vorbereiding betekent:

- Voorleggen en op de hoogte zijn van het werk van de voorbije week
- Vragen opsommen en meebrengen

Criterion 5: Omgaan met feedback / adviezen volgen

Dit gaat om zowel inhoudelijke feedback als procesmatige feedback.

Criterion 6: Stiptheid

Wees elke afspraak (online of op school) stipt op tijd.

Per criterium worden 5 levels beschreven (zie extra's). Twee maal (halverwege het semester (week 6) en op het eind (week 10)) zal je voor jezelf moeten bepalen op welk level je je bevindt per criterium en dat ook wat duiden. Zelfkennis is een belangrijke schakel in het succesvol werken in team.

Opgelet: je hoeft niet op elk criteria level 4 te halen om te kunnen slagen. De beschrijving van deze levels dient vooral om beter zicht te krijgen op jouw kunnen. Anderzijds, als je overal op level 0 zit, kan je wel onmogelijk slagen.

Wat gebeurt er als je in juni niet slaagt voor dit opleidingsonderdeel?

Tijdens de feedback kom je je **gepersonaliseerde taak** ophalen. Deze taak werk je dan **individueel** uit. Tijdens de tweede examenkans kom je dan **mondeling** je taak bespreken.

Voor het onderdeel groepswork kan je geen nieuwe taak krijgen. **Dit cijfer (/30) neem je mee uit de eerste zittijd.** Het is dus van cruciaal belang dat je je inzet voor het groepswork tijdens het semester! Als je een afwezig hebt op dit onderdeel (bv. te veel ongewettigd afwezig, onvoldoende betrokkenheid) neem je dit ook mee naar de 2de zittijd.

6. Eindproducten

Op het einde van de projectwerking geef je een **verzorgd dossier** af met daarin:

- Voorblad (zie Chamilo);
- ERD, relationeel model, script met DDL-instructies (per use case indien van toepassing);

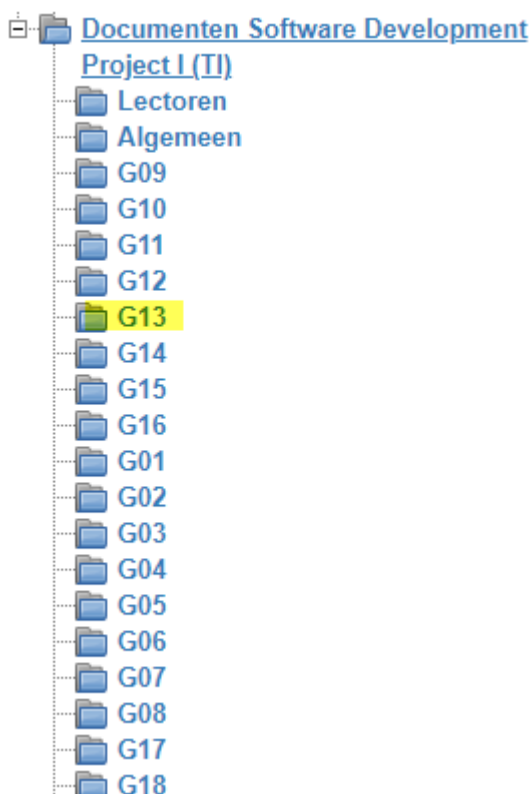
- Analyse en ontwerp (domeinmodel, activiteitendiagram, SSD, OC en DCD) per use case (of scenario indien van toepassing);
- 1 A4 blad tekst per groepslid met je ervaringen ivm groepswerk: wat heb je geleerd, wat waren de moeilijkheden, wat zijn jouw sterke punten, ...

Via de opdrachtenmodule van Chamilo dien je je project in:

Maak **1 zip-bestand** met daarin:

- Verzorgd dossier
- Eclipse project met de javacode
- De gegenereerde Java documentatie van de code (Javadoc)
- De volledige analyse en ontwerp (vpp-bestand van Visual Paradigm)
- Script met DDL-instructies van de MySQL database
- Exportbestand van de MySQL database (alle data).

Dit afgewerkte dossier laad je op via de opdrachtenmodule in Chamilo voor maandag 9 mei 23u59



Tijdens het semester bouw je ook een **digitaal dossier** op. Elke groep krijgt zijn eigen documentenfolder op Chamilo.

Deze folder is enkel toegankelijk voor de groepsleden én alle begeleidende lectoren.

Per week wordt hier het **logboek** geplaatst van de voorbije week. Ook het **opvolgingsformulier** krijgt hier een plaats.

Verder kunnen alle documenten waar je als groep samen aan werkt, hier gestockeerd worden (bv werkdossier, planningsdocumenten, ...).

Bewaar niets enkel lokaal op jouw pc. Zorg ervoor dat alles op elk moment online toegankelijk is!

De recentste code staat op github, de recentste analyse + ontwerp vinden we op VP online, andere documenten deel je via de Chamilo-folder of via google drive voor alle teamleden en de begeleidende lector.

7. Opdracht

1. Maak per use case (bevat alle vereiste functionaliteit):
 - a. Het **ontwerp van de database**. Doorloop hiervoor de stappen gezien tijdens de lessen Databases: opstellen van een ERD, daarna het relationeel model en tot slot het script met de DDL-instructies om de databank te implementeren in MySQL.
 - b. De **analyse** zoals aangeleerd in de lessen Software Analysis:
 - domeinmodel
 - alle zinvolle systeem sequentiediagrammen met bijhorende OC's
 - activiteitendiagrammen
 - c. Het **ontwerp** zoals aangeleerd in de lessen OOSDI : DCD
 - d. Een **console-applicatie** die ons toelaat om via deze interface de functionaliteit van de domein- en persistentielaag te testen. De implementatie van een drielagenproject leerde je in de lessen OOSD I (en OOSD II). Op een bepaald moment zal je mogen overschakelen naar een grafische user interface.
2. Ontwikkel een **grafische user interface**. Je bent vrij de GUI naar eigen inzicht vorm te geven. Wees creatief. Zorg ervoor dat de gevraagde functionaliteit gerealiseerd wordt. Begin niet aan de GUI voordat je begeleidende lector daarvoor **toestemming** geeft. Let op: ook **de console-applicatie blijft werken!**
3. De applicatie moet **meertalig** zijn. Gebruik hiervoor resourcebundles. Hoe deze precies werken, moet je zelf opzoeken! Je hoeft niet alle teksten te vertalen, maar technisch moet het eenvoudig te vertalen zijn!
4. Per use case wordt een **werkdossier** aangevuld met het gerealiseerde werk:
 - Het ERD, RM, script met DDL-instructies
 - De analyse (domeinmodel, activiteitendiagrammen en SSD's en OC's)
 - Het ontwerp (DCD)

Dit werkdossier vormt een belangrijke basis voor je **einddossier**. Houd het dus heel nauwkeurig bij.

5. Per groep wordt er een **logboek** bijgehouden. Dit logboek bevat een chronologisch overzicht

van de taken die de verschillende groepsleden op zich nemen. Telkens een groepslid aan het project werkt noteert die de details van het gedane werk (omschrijving en tijdsregistratie) meteen ook in het logboek. Elke week wordt een kopie van het logboek in het digitaal dossier geplaatst.

6. Wekelijks vul je het **opvolgingsformulier** in (wat is er gebeurd de voorbije week en wat zullen we doen de komende week, groepsafspraken, ...)

8. Tussentijdse presentatie en eindpresentatie

Op **28 maart** geef je een tussentijdse presentatie voor je projectbegeleider én de andere groepen.

Doel:

- voorbereiding op de eindpresentatie
- stand van zaken van de andere groepen

Belangrijke mijlpaal:

- Indienen voorlopig werkdoossier (Dit dossier moet in orde zijn als dossier, d.w.z. voorblad, inhoudsopgave, paginanummering, ... Verzorg het! Vergeet ook de persoonlijke teksten over groepswerking niet).
- Tonen van een stand van zaken (bespreek ook groepsaanpak, manier van werken)

Op **12 mei** volgt de eindpresentatie. Deze presentatie wordt bijgewoond door je begeleidende lector.

Elke student krijgt specifieke technische vragen (op basis van ingediend project, dossier en observaties tijdens het semester).

Het al dan niet goed beantwoorden van deze vragen beïnvloedt **héél sterk** je persoonlijk cijfer voor het project (deel informatica 70%).

9. Een aantal extra's

9.1. Projectopgave Zatre

afgewerkt is (d.w.z. alle steentjes liggen op het spelbord).

Tijdens een beurt leg je 2 (of 3) steentjes op het bord. Het kan zijn dat je door het leggen van je steentjes een lijntje maakt van 10, 11 of 12. Dan zet je een streepje in de juiste kolom, in het eerste lege vakje van die kolom. Maak je tijdens dezelfde beurt 2 lijntjes van bv 10 dan zet je twee streepjes in dat eerste lege vakje van die kolom. Komt jouw steentje op een grijs vakje te liggen, dan mag je een kruisje zetten in de kolom om de score te verdubbelen, terug in het eerste lege vakje van die kolom. Opgelet: een grijs vakje mag je enkel opvullen als je een 10, 11 of 12 kan maken.

- **Puntentelling**

Als een horizontale lijn op het scoreblad volledig opgevuld is, kan je in de laatste kolom je punten - voor die lijn - opschrijven. Elk streepje in het vakje van 10 levert 1 punt op, 2 punten voor de streepjes in het vakje van 11 en 4 punten voor elk streepje in het vakje van de twaalf. Als je dus zowel een 10, 11 als 12 hebt kunnen leggen krijg je ook nog de bonuspunten erbij. Staat er een kruisje bij het verdubbelen dan doe je nog eens de som van die punten maal 2.

De eindscore van de speler is de som van alle punten per lijn.

De speler met de hoogste eindscore is de winnaar.

Hieronder zie je een voorbeeld van een ingevuld scoreblad.

ZATRE®[illegible]

De opgave van het spel werd uitgeschreven in een aantal use cases. De use cases zijn het contract met je klant, hierin staat beschreven hoe het precies allemaal moet werken. Begin dus niet in het wilde weg te programmeren. Volg exact wat er beschreven staat in de use case. Doorloop alle fasen van een iteratie (analyse, ontwerp, implementatie, testen). Als de ene iteratie afgewerkt is kan je met de volgende starten.

- ### 1. UC1 - Registreer speler

2. UC2 - Selecteren speler
3. UC3 - Start Zatre
4. UC4 - Speel beurt

9.2. Overzicht criteria met beschrijving levels

Categorieën	Criteria	Level 0	Level 1	Level 2	Level 3	Level 4
Product	Volledigheid product qua vorm	Het product is onvolledig : Er ontbreken essentiële onderdelen.	Het product is volledig met alle onderdelen, maar sommige onderdelen zijn onvolledig .		Het product is volledig en conform de eisen.	Het product is volledig en conform de eisen op een professionele wijze .
	Volledigheid product qua functionaliteit	Het product bevat minder dan de helft van de gevraagde functionaliteiten.	Het product bevat minstens de helft van de gevraagde functionaliteiten.	Het product bevat alle gevraagde functionaliteiten.	Het product bevat alle gevraagde functionaliteiten.	Het product bevat alle gevraagde functionaliteiten.
		De opgeleverde applicatie is niet stabiel , vormt geen geheel , moet vaak herstart worden.	De opgeleverde applicatie is niet stabiel maar vormt één geheel .	De opgeleverde applicatie is bijna volledig stabiel , maar af en toe ontbreekt een nodige validatie.	De opgeleverde applicatie is volledig stabiel . Alle gevraagde validaties zijn aanwezig.	De opgeleverde applicatie is volledig stabiel . Alle gevraagde validaties zijn aanwezig. Extra validaties zijn toegevoegd.
	Volledigheid product qua niet functionele requirements	De student kan de vooropgestelde niet functionele requirements niet uitwerken.	De student kan de vooropgestelde niet functionele requirements voor een beperkt deel van de functionaliteiten uitwerken.	De student kan de vooropgestelde niet functionele requirements uitwerken, sturing nodig.	De student kan de vooropgestelde niet functionele requirements autonoom uitwerken.	De student kan de vooropgestelde niet functionele requirements autonoom uitwerken (bewust en met nodige toelichting).

Categorieën	Criteria	Level 0	Level 1	Level 2	Level 3	Level 4
Productontwikkeling	Analyseren	De student kan de producten uit de analysefase niet opstellen en toelichten .	De student kan de producten uit de analysefase opstellen en toelichten , maar weinig kwaliteitsvol. Er is frequent intense bijsturing nodig.	De student kan de producten uit de analysefase kwaliteitsvol opstellen en toelichten . Er is soms bijsturing nodig (ganse periode).	De student kan de producten uit de analysefase kwaliteitsvol opstellen en toelichten . De bijsturing kan geleidelijk aan verminderen.	De student kan de producten uit de analysefase kwaliteitsvol opstellen en toelichten . Er is amper bijsturing nodig.
	Ontwerpen	De student kan de producten uit de ontwerpfase niet opstellen en toelichten .	De student kan de producten uit de ontwerpfase opstellen en toelichten , maar weinig kwaliteitsvol. Er is frequent intense bijsturing nodig.	De student kan de producten uit de ontwerpfase kwaliteitsvol opstellen en toelichten . Er is soms bijsturing nodig (ganse periode).	De student kan de producten uit de ontwerpfase kwaliteitsvol opstellen en toelichten . De bijsturing kan geleidelijk aan verminderen.	De student kan de producten uit de ontwerpfase kwaliteitsvol opstellen en toelichten . Er is amper bijsturing nodig.
	Testen	De student kan geen correcte unittesten schrijven voor de verschillende methodes uit de domeinklassen.	De student kan beperkt unittesten schrijven voor de verschillende methodes uit de domeinklassen.	De student kan unittesten schrijven voor de verschillende methodes uit de domeinklassen maar de testen zijn niet volledig .	De student kan unittesten schrijven voor de verschillende methodes uit de domeinklassen en de testen zijn volledig .	De student kan unittesten schrijven voor de verschillende methodes uit de domeinklassen, de testen zijn volledig . De student doet dit volledig op een TDD wijze en moet hiertoe niet verplicht worden.
	Programmeren	De student kan functionaliteiten niet programmeren .	De student kan functionaliteiten weinig kwaliteitsvol programmeren . Er is frequent intense bijsturing nodig.	De student kan functionaliteiten kwaliteitsvol programmeren . Er is soms bijsturing nodig (ganse periode).	De student kan functionaliteiten kwaliteitsvol programmeren . De bijsturing kan geleidelijk aan verminderen en de student vraagt gericht hulp waar nodig.	De student kan functionaliteiten kwaliteitsvol programmeren . Er is amper bijsturing nodig.
	Databank	De student kan de databank niet opstellen conform de eisen.	De student kan de databank opstellen conform de eisen. Er is frequent intense bijsturing nodig.	De student kan de databank opstellen conform de eisen. Er is soms bijsturing nodig (ganse periode).	De student kan de databank opstellen conform de eisen met beperkte bijsturing.	De student kan de databank opstellen conform de eisen. Er is amper bijsturing nodig.
		De student kan de eigen opgestelde databank niet beheren vanuit de applicatie.	De student kan de opgestelde databank beheren vanuit de applicatie. Er is frequent intense bijsturing nodig.	De student kan de opgestelde databank beheren vanuit de applicatie. Er is soms bijsturing nodig (ganse periode).	De student kan de opgestelde databank beheren vanuit de applicatie en ook toelichten (niet enkel nabootsen van voorbeeldapplicatie). De bijsturing kan geleidelijk aan verminderen.	De student kan de opgestelde databank beheren vanuit de applicatie en ook toelichten (niet enkel nabootsen van voorbeeldapplicatie). Er is amper bijsturing nodig.

Categorieën	Criteria	Level 0	Level 1	Level 2	Level 3	Level 4
P R O C E S	Registreren	De student registreert <i>niet of onvoldoende</i> (aantal keer + correct).	De student registreert op expliciete vraag maar <i>niet altijd correct of niet consequent</i> . Er is frequent intense bijsturing nodig.	De student registreert <i>consequent</i> . Er is soms <i>inhoudelijke</i> bijsturing nodig (ganse periode).	De student registreert <i>spontaan en inhoudelijk meestal goed</i> . De inhoudelijke bijsturing kan geleidelijk aan verminderen.	De student registreert van bij de start <i>spontaan en inhoudelijk correct</i> . Er is <i>amper</i> bijsturing nodig.
	Plannen	De student plant niet: neemt <i>ad hoc</i> taken op, <i>volgt afspraken niet</i> en controleert <i>niet</i> of hij/zij op schema zit.	De student plant zelden: kijkt <i>dag per dag</i> naar de taken, volgt afspraken op als hij/zij <i>erop gewezen</i> wordt, <i>controleert zelden</i> de volledige planning.	De student plant <i>zelfstandig</i> op korte termijn, <i>respecteert</i> de gemaakte <i>weekplanning</i> en controleert af en toe of het hij/zij nog op schema zit.	De student plant <i>zelfstandig</i> op korte termijn, <i>rekening houdend met vorige werkervaringen</i> , <i>respecteert</i> weekplanning, <i>controleert</i> planning en <i>stuurt bij</i> waar nodig.	De student plant zelfstandig op korte termijn, rekening houdend met vorige werkervaringen, <i>respecteert</i> weekplanning, <i>controleert</i> planning en <i>stuurt bij</i> waar nodig. <i>Geen bijsturing nodig, van bij de start op hoog niveau!</i>
	Teamwork	De student werkt <i>niet samen</i> in groep en werkt voor zichzelf binnen de groepsoopdracht.	De student werkt <i>soms</i> samen, maar houdt zich vooral bezig met eigen taken en opdrachten.	De student werkt <i>samen</i> in groep maar stelt <i>af en toe</i> het <i>eigenbelang</i> boven het groepsbelang.	De student werkt <i>samen</i> in groep en <i>motiveert anderen</i> . De student stelt het groepsbelang meestal boven het eigenbelang.	De student werkt <i>samen</i> in groep en <i>stimuleert</i> samenwerken. De student stelt het groepsbelang prioritair.
	Vorbereiding wekelijks gesprek	De student heeft <i>niets</i> voorbereid.	De student heeft <i>gedeeltelijke</i> of <i>onsamenhangende voorbereiding</i> gemaakt.	De student heeft een <i>relevante voorbereiding</i> gemaakt.	De student heeft een <i>grondige voorbereiding</i> inclusief vragen gemaakt.	De student heeft een <i>grondige voorbereiding</i> gemaakt in een ruimere context waardoor hij <i>anticipeert</i> op mogelijke antwoorden.
	Omgaan met feedback / Adviezen opvolgen	De student houdt <i>geen of amper rekening met feedback</i> van groepsleden en lector. De student reageert onverschillig of aangevallen.	De student houdt <i>rekening met feedback</i> van groepsleden en lector <i>na meermaals aandringen</i> . De student reageert soms onverschillig of aangevallen.	De student houdt <i>meestal</i> rekening met feedback van groepsleden en lector.	De student <i>houdt rekening</i> met feedback van groepsleden en lector. <i>Vraagt actief en spontaan feedback</i> .	De student houdt <i>rekening met feedback</i> van groepsleden en lector. <i>Vraagt proactief</i> spontaan feedback en geeft zelf suggesties.
	Stiptheid	Is altijd te laat op sessies.	Is sporadisch op tijd op sessies.	Is meestal op tijd op sessies.	Is altijd op tijd op sessies	

9.3. Logboek

Noteer alle uren die je besteedt aan dit project. Je krijgt geen punten voor je uren, maar op die manier krijgen we een zicht op de tijdsbesteding in een project. Welk luik vraagt meest tijd?

ZATRE - groep XX										
Datum	Omschrijving taak	Student	Tijd in min.	DB	OA	OO	OP	?	commit/revision	#
			→							
			→							
			→							
			→							
			→							
			→							
			→							
			→							
			→							
				Totalen in uren						
				Globaal	DB	OA	OO	OP	Andere	
st1				0	0	0	0	0	0	
st2				0	0	0	0	0	0	
st3				0	0	0	0	0	0	
st4				0	0	0	0	0	0	
Groep				0	0	0	0	0	0	
Gemiddeld per persoon				0	0	0	0	0	0	

9.4. Opvolgingsformulier

Elke week vullen we het opvolgingsformulier aan.

- **DONE:** Wat heb je de voorbije week gedaan?
- **TODO:** Wat ben je van plan te doen de komende week?
- **Hoe verloopt het groepswork?:** Noteer hier alle probleempjes, vragen, sterke punten, ...

Deel 1 - In te vullen door de studenten

	DONE - Wat werd voorbije week gerealiseerd?
Algemeen	
Student1	
<u>Student2</u>	
<u>Student3</u>	
Student4	

	TODO - Wat plannen we volgende week te realiseren?
Algemeen	
Student1	
<u>Student2</u>	
<u>Student3</u>	
Student4	

Hoe verloopt het groepswerk?