



SLAGEN VOOR HET OSM PROJECT

Voldoen aan de beoordelingscriteria

Michel Koolwaaij (OOSE)

Tineke Jacobs (OOSE & OSM)

Bram Knippenberg (OSM)

4 november 2022

INHOUDSOPGAVE

1	COMPETENTIES.....	4
2	BEROEPSPRODUCTEN.....	5
2.1	Plan van aanpak.....	5
2.2	Onderzoeksverslag(en).....	5
2.3	SRS.....	5
2.4	SDD.....	6
2.5	Code.....	6
2.6	Testrapport.....	6
2.7	Individueel projectverslag.....	6
2.8	Algemeen.....	7
3	PROCES EN TOOLING.....	8
3.1	Scrumceremonies.....	8
3.2	Professionaliteit en zelfstandigheid.....	8
3.3	Gebruik van de tooling.....	8
3.4	Samenwerking en hulpvaardigheid.....	9
3.5	Verdiepen in de beroepstaak.....	9
4	DEELTENTAMENS EN -CIJFERS.....	10
4.1	IND-PVB.....	10
4.2	Tussentijds groep.....	10
4.3	Tussentijds individueel.....	10
4.4	Groepsproducten- EIND.....	11
4.5	Individuele bijdrage- EIND.....	11
4.5.1	Individuele eindverslag.....	12
	KNOCK OUTS.....	14
4.6	Onvoldoende functioneren gedurende het project.....	14
4.7	Belangrijke randzaken.....	14

INLEIDING

Inmiddels ben je gestart met het OSM-project en heb je alles wat in het document “Starten met het OSM-project” staat op orde.

Nu wil je weten wat je moet doen om je studiepunten te krijgen. Voor dit project zijn dat er 15. Net als voor alle andere vakken staan de competenties en de bijbehorende beoordelingscriteria voor het OSM-project in de OWE-beschrijving en die vind je in het OER (Onderwijs- en ExamenReglement). Simpel gesteld krijg je studiepunten als je alle competenties aantoont volgens de bijbehorende beoordelingscriteria. In dit document geven we antwoord op veelgestelde vragen als “wat betekenen die competenties?”, “hoe toon ik die competenties aan?” en “hoe beoordelen docenten dat?”.

Het document start met een opsomming van de competenties. Daarna volgen de beroepsproducten die je gaat opleveren en welke competenties je hiermee aantoont. Als laatste leggen we uit hoe de beoordelingsmomenten in elkaar steken.

Dit document is met zorg samengesteld. De auteurs staan open voor opmerkingen, correcties en aanvullingen op de tekst. We wensen de studenten veel plezier in het uitvoeren van het project. De hoofdlijn van deze handleiding is ook terug te vinden in de sheets van de aftrap, maar de details staan in dit document, én in ‘Starten met het OSM-project’, lees beide documenten dus zorgvuldig.

Namens de OOSE en OSM teams,

Tineke Jacobs & Michel Koolwaaij & Bram Knippenberg

Neem voor het OSM-Project contact op met Bram Knippenberg

1 COMPETENTIES

In het OSM-project (studiejaar 2022-2023) moet je voldoen aan tien competenties. Dit zijn:

OSM P-1

Je past Scrum als ontwikkelmethodiek toe in het project en beargumenteert deze achteraf op bruikbaarheid, gegeven de specifieke eigenschappen van het project.

OSM P-2

Je maakt een analyse van de eisen en wensen voor de software van een systeem, en schrijft hierover een analyserapport met op functionele en niet-functionele eisen, rekening houdend met technische beperkingen en mogelijkheden van het systeem.

OSM P-3

Je voert een kwalitatief en kwantitatief onderzoek op een systeem uit en levert hierover een onderzoeksrapport op.

OSM P-4

Je maakt een ontwerp voor de software van een systeem, rekening houdend met functionele en niet-functionele eisen en beoordeelt de consistentie tussen ontwerp en gestelde functionele en niet-functionele eisen.

OSM P-5

Je ontwikkelt de software voor een systeem op professionele wijze, en gebruikt daarbij op adequate wijze relevante ontwikkeltools.

OSM P-6

Je implementeert in multidisciplinair teamverband een ontwerp van een gedistribueerd, algoritmisch complex systeem, en evalueert het ontwerp en de realisatie daarvan.

OSM P-7

Je beoordeelt tussentijds de kwaliteit van de software en stuurt zo nodig bij door gebruik van design patterns en refactoring.

OSM P-8

Je test de werking van het systeem op gestructureerde wijze.

OSM P-9

Je draagt bij aan het goed functioneren van het ontwikkelteam en ondersteunt de leden van het team bij hun taakuitoefening.

OSM P-10

Je verdiept zich zelfstandig verder in de beroepstaak.

2 BEROEPSPRODUCTEN

Professioneel software ontwikkelen, use cases realiseren, betekent meer dan alleen code kloppen. Dat betekent dat code niet het enige beroepsproduct is dat je in het OSM-project oplevert. Hieronder vind je een overzicht van de beroepsproducten die je gaat opleveren tijdens het project. Zoals je geleerd hebt tijdens de courses van Software en Modelling: deze beroepsproducten zijn niet onafhankelijk van elkaar.

2.1 Plan van aanpak

Je schrijft zo snel mogelijk na bekendmaking van de groepen en opdrachten het plan van aanpak (PvA). Hiermee krijg je inzicht in het Waarom Wat en Hoe van het project. Daarmee hebben projectteam, opdrachtgevers en begeleiders houvast gedurende het project. Je schrijft het PvA aan de hand van de actuele Toelichting op het Plan van Aanpak. In de skills-lessen heb je daar les over gehad.

Je levert het Plan van Aanpak in onderwijsweek 1 in. Daarna word je uitgenodigd voor een assessment. Tijdens dit assessment word je individueel beoordeeld aan de hand van de beoordelingscriteria. Het plan van aanpak zelf wordt beoordeeld tijdens de tussentijdse beoordeling en telt mee voor dat cijfer.

Competenties: OSM P-1.

2.2 Onderzoeksverslag(en)

Tijdens het uitvoeren van het project zul je diverse onderzoeken uitvoeren. Deze komen ten dele voort uit de opdracht, maar mogelijk ook omdat je keuzes moeten maken en nieuwe technieken en tooling moeten leren kennen. Beschrijf je onderzoek(en) dusdanig in je onderzoeksverslag(en), zodat het helder wordt hoe je tot je conclusie(s) komt. Het moet duidelijk zijn waarom het onderzoek wordt uitgevoerd en wat de concrete bijdrage is aan het project.

Competenties: OSM P-3.

2.3 SRS

Bij OSM-Modelling heb je geleerd je requirements te specificeren in een Software Requirements Specification. Het SRS is onderdeel van de IEEE software levenscyclus, bekend als IEEE 29148. Op basis van het SRS moet een ontwikkelaar een systeem kunnen ontwikkelen. Een nieuw teamlid moet op basis van het SRS een compleet beeld krijgen van wat het systeem doet en kan. De onderdelen van het SRS worden beoordeeld zoals bij OSM-Modelling.

Gebruik voor het SRS het sjabloon zoals aangeboden op Confluence.

Competenties: OSM P-2

2.4 SDD

Bij OSM-Modelling heb je geleerd om je ontwerp te beschrijven in een SDD. Een Software Design Description is onderdeel van de IEEE software levenscyclus, bekend als IEEE 1016. Op basis van het SDD moet een ontwikkelaar begrijpen hoe de software in elkaar steekt en waarom. Een nieuw teamlid moet aan de hand van het SDD kunnen zien hoe requirements uit het SRS zijn geïmplementeerd en met welke reden. Een SDD bevat daarom ontwerpbeslissingen waarin je verwijst naar patronen en principes zoals geleerd tijdens de Modelling- en Software-courses. Je mag hier dus eigen keuzes maken. De onderdelen van het SDD worden beoordeeld zoals in de courses.

Gebruik voor het SDD het sjabloon zoals aangeboden op Confluence.

Competenties: OSM P-4.

2.5 Code

Je schrijft nette goed onderhoudbare code, voorzien van documentatie (doxygen) en maakt hierbij gebruik van de aangereikte tooling voor code management, continuous integration en codekwaliteit. In de course fase heb je bij OSM-Software geleerd om nette en duidelijke code te schrijven die goed testbaar is en bij OSM-Modelling om verantwoordelijkheden goed te scheiden. De tooling zul je zelf onder de knie moeten krijgen. Je wordt beoordeeld op de kwaliteit van je code, de mate en kwaliteit van documentatie, en het gebruik van de tooling.

Competenties: OSM P-6, OSM P-7

2.6 Testrapport

Als goede software-ontwikkelaar wil je natuurlijk kwaliteit opleveren. De software betrouwbaar zijn en de resultaten/uitkomsten valide. Niet alleen wil je van elke regel code de zekerheid dat deze goed werkt, je wil ook dat al je requirements door je software worden afgedekt.

Om dit te behalen zet je meerdere test-technieken in: unit-test, integratie-tests, testplannen (bijvoorbeeld op use case niveau) en diverse middelen voor validatie van de resultaten uit software.

Dat alles maak je inzichtelijk door het maken van een testrapport waarin je beschrijft welke requirements je hebt gerealiseerd, hoe je dat hebt getest en wat het resultaat van die test is.

Competenties: OSM P-8

2.7 Individueel projectverslag

In het individuele projectverslag neem je de docent/beoordelaar mee in jouw project. Dit document gebruik je om jouw bijdrage aan het project zichtbaar te maken.

Tijdens het OSM-project leer je veel over software bouwen met een team voor een opdrachtgever. Je past theorie uit de courses toe, leert nieuwe technieken, bouwt ervaring op in het omgaan met een opdrachtgever en werkt met professionele tooling. Bewustwording hiervan staat centraal in de opleiding en deze is niet te halen uit de overige beroepsproducten. In je individueel verslag laat je daarom zien wat je geleerd hebt, je reflecteert op je eigen ontwikkeling, het groepsproces en de kwaliteit van de geleverde producten. Kijk voor meer informatie hierover op Onderwijs Online in het document “Alle info over het schrijven van je projectverslag”.

Competenties: OSM P-1, OSM P-10.

2.8 Algemeen

Voor alle beroepsproducten geldt: je moet ze op tijd op de juiste plaats inleveren én ze voldoen aan de AIM-controlekaart. Voldoe je hier niet aan dan is je werk ‘niet ontvankelijk’ en kunnen we het niet beoordelen.

3 PROCES EN TOOLING

Binnen het OSM-project gebruiken we Scrum, tooling en werken we aan een grote opdracht.

Dit zijn zaken die niet direct leiden tot een beroepsproduct, maar van grote invloed zijn op kwaliteit, samenwerking en productiviteit. Als softwareontwikkelaar vinden we het vaak leuker om te programmeren dan om te overleggen of te documenteren.

Hoe effectiever je overlegt, hoe beter je gebruik maakt van de tooling en hoe eerder je documenteert, hoe meer tijd je overhoudt om te programmeren en oplossingen te bedenken. Daarom beoordelen wij ook de volgende punten:

3.1 Scrumceremonies

Hiertoe behoren daily standups, sprint review, planningpoker en retrospectives. Voor de Daily Stand Up en de Retropective zijn een checklists beschikbaar (zie bijlage C en D). Gebruik die. Maak passende notulen/actiepuntenlijsten en nodig af en toe de begeleiders uit voor feedback.

Competentie: OSM P-1

3.2 Professionaliteit en zelfstandigheid

Communiceer netjes en proactief met je opdrachtgever en begeleiders. Nodig ze uit, maak een agenda, betrek ze en informeer ze. Jullie zijn het ontwikkelteam!

Competentie: OSM P-1

3.3 Gebruik van de tooling

Jullie gaan verplicht gebruik maken van Jira, Confluence, Bitbucket, Jenkins en SonarQube. Deze tools gebruik je ter ondersteuning van het werkproces. Hoe beter je deze gebruikt, hoe meer voordeel je ervan hebt. Hoe meer voordeel je ervan hebt, hoe hoger je beoordeling.

Gebruik de tools niet alleen omdat ze er zijn, maar voor de meerwaarde die deze bieden bij de projectuitvoering. Voorbeelden:

- Taken staan beschreven in Jira.
- De status van taken wordt bijgewerkt.
- (Tussen) resultaten van taken zijn duidelijk vastgelegd in Jira.
 - Zo kan een ander de informatie ook vinden.
 - Bijvoorbeeld in de vorm van een code-commit of een confluence-diff.
- Code is geschreven in feature-branches en zijn naar Bitbucket ge-pushed.
 - Maar niet voordat de feature branch bouwt zonder problemen.
- Pull requests worden ingezet voor code-reviews.
- Code komt pas development te staan als deze getest is (unit-test).

- Urenschattingen en daadwerkelijk bestede uren worden zo correct mogelijk ingevoerd.
 - o Alleen dan is het mogelijk om beter te leren schatten en krijg je meer controle over je proces.

Competenties: OSM P-5

3.4 Samenwerking en hulpvaardigheid

Een goede samenwerking zorgt voor een betere sfeer, meer motivatie en een beter eindresultaat. Hoe stel jij je op in de groep? Tenminste twee keer per projectperiode houden jullie een IPV waarin je van je groepsgenoten persoonlijke feedback ontvangt, vaak wordt de bespreking van de IPV gekoppeld aan een retrospectieve. Je bereid je voor met de vragenlijst iSAS, waarin je GEIN-feedback noteert. In de bespreking in de groep staat steeds de vraag centraal: hoe kunnen we door het verbeteren van onze samenwerking de beste beroepsproducten opleveren. Steeds maak je concrete verbeterafspraken. Zo help je elkaar het beste eruit te halen. Vergeet niet om met elkaar te spreken over je (vorderingen op) je leerdoelen. In Semester 2 maak je actief gebruik van het Persoonlijk AnalyseFormulier. Stimuleer je teamgenoten het beste in zichzelf naar boven te halen.

Competenties: OSM P-9

3.5 Verdiepen in de beroepstaak

Als software-ontwikkelaar bouw je (vrijwel altijd) software voor anderen. Vaak is dat ook in een jou nog onbekend domein. Je zult met de klant in overleg moeten (vaak expert in zijn domein). Ook is het mogelijk dat je nieuwe ontwikkeltechnieken moet verkennen.

Hier heb je waarschijnlijk je gehele carrière als softwareontwikkelaar mee te maken: verdiepen in de beroepstaak.

Tip: Hou opgedane kennis niet alleen voor jezelf, maar deel deze ook met anderen. Schrijf bijvoorbeeld een bruikbaar stukje op een confluence-pagina, zodat ook andere er hun voordeel mee kunnen doen in het project.

Competenties: OSM P-10

4 DEELTENTAMENS EN -CIJFERS

In de OWE-beschrijving die je vindt op han.nl/insite> Informatica, Media en Communicatie> HBO-ICT> Rechten en Plichten> naar OS/OER HBO-ICT vind je de formele toetskaders. Hieronder geven we een praktische vertaling. *Op iSAS kun je de beoordelingscriteria per toet bekijken.*

4.1 IND-PVB

Deze toets is een zogenaamde 'vinktoets': je krijgt er geen cijfer voor, maar je haalt hem (V) of je haalt hem niet (NV). In deze toets worden jouw evenredige bijdrage aan, en jouw individuele begrip van het Plan van Aanpak getoetst. Dit gebeurt in week 1, 2 of 3. Beoordelaars zijn de procesbegeleider en de skills-docent. De vorm van deze toets is een assessment waarbij de hele groep aanwezig is. Hoe dit in zijn werk gaat vind je in bijlage A.

4.2 Tussentijds groep

Dit cijfer telt mee voor 10% van je eindcijfer. We beoordelen het plan van aanpak, de onderzoeks-verslag(en), alle producten die jullie hebben opgeleverd en het verloop van sprints tot zover. Dit gebeurt in week 4 of 5. Beoordelaars zijn de procesbegeleider en de assessor.

4.3 Tussentijds individueel

Dit telt mee voor 10% van je eindcijfer. We beoordelen je bijdrage tot nu toe en je individuele verslag dat je tot nu toe hebt gemaakt. Dit gebeurt in week 4 of 5. Beoordelaars zijn de procesbegeleider en de skills-docent. Bij het schrijven van het verslag geldt het volgende m.b.t het nakijkmodel dat je hebt gevonden in het document 'Alle info over het schrijven van je projectverslag'.

1. Je schrijft een inleiding (punt 1 nakijkmodel)
2. Bij deelproducten (punt 2 nakijkmodel) ga je tussentijds in op:
 - Het plan van aanpak
 - Het SRS
 - Het SDD
 - Een stuk code naar keuze
 - Unit-tests

Hierbij hoeven deze deelproducten niet af te zijn: je kunt uitstekend een kwaliteitsoordeel geven over een product dat nog niet af is. Mocht een product nog niet bestaan dan kan je daar uiteraard geen oordeel over hebben.

3. Op het eindproduct (punt 3 nakijkmodel) ga je in op het product zoals dat tot nu toe is opgeleverd (tweede sprint-review).
4. Je schrijft een eerste oordeel over de projectmethode (punt 4 nakijkmodel)
5. Je schrijft een eerste reflectie op je roluitvoering, en plaatst dit in de kader van de samenwerking met je team (punt 5 nakijkmodel)
6. Voor de competenties (punt 6 nakijkmodel) ga je in op:
 - Twee competenties naar keuze uit OSM P-2, 4, 6 en 'eigen keuzecompetentie'
 - o Voor de 'eigen keuzecompetentie' mag je kiezen uit OSMP-3, OSMP-7, OSMP-8.
 - Ten overvloede: competenties OSMP-5 mag je dus niet kiezen.
7. Je laat zien hoe ver je bent met je leerdoelen (punt 7 nakijkmodel)
 - Je moet tenminste één situatiebeschrijving geven en/of één "nulmeting" die inzichtelijk maakt dat je leerdoel relevant is.
8. De conclusie (punt 8 nakijkmodel) gaat tussentijds over de gang van zaken in het project tot zover en hoe *jij* de rest van het project gaat aanpakken. Dit zowel voor de zaken binnen het project in het algemeen (groepswork) als voor jouw persoonlijk (competenties en leerdoelen). Betrek daar waar relevant het online werken in je conclusie.
9. Bijlage: Factsheet tot zover, met daarin per competentie (*alle* competenties) je 3-5 belangrijkste facts met directe links naar jouw bijdrage (dus naar de specifieke jirataak, commit, of diff op confluence). Zie voor een opzet Confluence.

Belangrijk:

- Voor het gehele verslag geldt dat je je verhaal onderbouwd met concrete links naar relevante documenten, taken, commits, etc.
- Voor het aantonen van competenties moet je concreet verwijzen naar eigen bijdrage. Denk hierbij bijvoorbeeld aan een confluence-diff (verschil tussen twee documentversies; zie onder history van de pagina), een specifieke code-commit (of set van commits in een pull-request).

4.4 Groepsproducten- EIND

Dit telt voor 40% mee voor je eindcijfer. Beoordeling gebeurt in week 9, 10 of 11 op basis van je ingeleverde werk in ISAS. Beoordelaars zijn de procesbegeleider en de assessor.

Dit deelcijfer moet een 5,5 of hoger zijn en is als volgt opgebouwd.

- SRS: 16%
- SDD: 22%
- Code: 22%
- Proces: 40%

Als dit deelcijfer lager is dan een 5,5 dan ben je gezakt voor het project.

4.5 Individuele bijdrage- EIND

Dit cijfer telt voor 40% mee aan het eindcijfer. Beoordeling gebeurt in week 9, 10 of 11 op basis van je ingeleverde werk in ISAS. Beoordelaars zijn de procesbegeleider en de skills-docent.

De basis voor dit cijfer is het cijfer voor de groepsproducten. Om een voldoende te halen voor je individuele bijdrage zijn twee factoren essentieel:

- Het deelcijfer voor groepsproducten is voldoende
- Je toont al je competenties aan.

Inmiddels heb je per competentie gelezen op welke manier en met welk product je deze aantoont.

Hierbij hanteren de beoordelaars vier verschillende methoden om dit te beoordelen:

1. Je individuele verslag met je factsheet.
2. De toolstack.
 - Als je de tooling goed gebruikt dan zien wij precies welke bijdrage ieder heeft geleverd.
3. Onze waarneming.
4. Een (eventueel) assessment.

Als dit voldoende wordt bevonden en je deelcijfer voor groepsproducten is voldoende, dan heb je een voldoende voor je individuele bijdrage. Daarbij geldt als *knock out* dat je op alle vlakken een serieuze poging moet hebben gewaagd.

Een betere of mindere bijdrage dan gemiddeld kan leiden tot een hoger of lager cijfer voor je individuele bijdrage dan voor je groepsproducten.

4.5.1 Individuele eindverslag

In week 8 lever je je eindverslag in. In principe is dit *een verder aangevulde versie* van het tussentijdse verslag.

Bij het schrijven van het verslag hou je rekening met de eisen uit "Alle info over het schrijven van je projectverslag ". Let wel op de verschillen tussen de aanwijzingen voor *tussentijds* en *eind*.

1. De inleiding (punt 1 nakijkmodel) verbeter je als de TT feedback daar aanleiding voor gaf
2. Bij deelproducten (punt 2 nakijkmodel) ga je in op:
 - Het SRS
 - Het SDD
 - Code in het algemeen
 - Testplan + rapport
 - Unit-tests
 - Het adviesrapport
3. Onder eindproduct (punt 3 nakijkmodel) wordt het eindproduct bedoeld zoals opgenomen in het PvA.
4. Je schrijft definitief oordeel over de projectmethode (punt 4 nakijkmodel), waarbij je ook ingaat op de tweede helft van het project.
5. Je schrijft een definitieve reflectie op je roluitvoering, in de kader van de samenwerking met je team. Hierin neem je de TT gekregen feedback, en de tweede helft van het project mee. (punt 5 nakijkmodel)
6. Voor de competenties (punt 6 nakijkmodel) geldt dat je ingaat op de volgende competenties:
 - Competenties OSM P-2, 4, 6 en 'eigen keuzecompetentie'
 - o Voor de 'eigen keuzecompetentie' mag je kiezen uit OSMP-3, 7 en 8.

7. Je laat zien waar je nu staat ten aanzien van de door jou geformuleerde leerdoelen (punt 7 nakijkmodel).
8. Je grijpt in de conclusie (punt 8 nakijkmodel) terug op de opdracht: in hoeverre hebben jullie je doel gehaald en wat waren daarin de belangrijkste factoren, je gaat in op je belangrijkste uitdaging, en je geeft je belangrijkste persoonlijke ontwikkelpunt(en), en je laat zien waar je staat ten opzichte van de competenties van OSM-P.
9. **Bijlage:** Factsheet met daarin per competentie (voor alle competenties) je belangrijkste facts (3-5) met directe links naar jouw bijdrage (dus naar de specifieke jirataak, commit, of diff op confluence), en een korte toelichting daarop.

Wellicht ten overvloede: Als het verslag niet voldoet aan alle gestelde eisen is je 'tentamen' niet succesvol verlopen en krijg je een (individuele) onvoldoende. Je moet dan het hele project opnieuw doen. Laat het daar niet op aan komen.

KNOCK OUTS

4.6 Onvoldoende functioneren gedurende het project

In zeer uitzonderlijke gevallen komt het voor dat iemand al gedurende het project het team moet verlaten.

Dit kan het geval zijn als iemand zich na herhaaldelijke opbouwende feedback door medestudenten (initieel) en docenten (in tweede instantie) niet aan de herhaaldelijk gemaakte afspraken houdt, en het team hiervan zoveel hinder ondervindt dat ze geen goede voortgang kunnen maken.

Uiteraard mag dit geen willekeur zijn, en moet een zorgvuldige procedure worden gevolgd. Die kun je lezen in het boekje 'Projectonderwijs ICA' dat je op OnderwijsOnline bij het OSM-project kunt vinden.

Voorwaardelijk voor een goede procedure is dat feedback en afspraken worden vastgelegd. Dat eerste kan door middel van de IPV-PB-beoordelingsenquête in iSAS. Uiteraard moet deze enquête persoonlijk worden besproken (GEIN-feedback!). Wanneer je bijdragen door je teamleden als onvoldoende worden beschouwd (criteria daarbij zijn: niet 40 uur per week werken, niet samenwerken, geen verantwoordelijkheid nemen voor de invulling van je rol en/of de begeleiders niet in staat stellen je te begeleiden), maak je met elkaar concrete afspraken over de manier waarop jij je in de volgende sprint gaat verbeteren.

Het is geen gek idee om dit gesprek te houden gelijktijdig met, of rondom de retrospective. Let er in elk geval op dat een van je docenten tenminste bij de eerste bij de bespreking van de IPV aanwezig is.

Zorg dus dat je hier tijdig afspraken over maakt.

4.7 Belangrijke randzaken

Je werkt tijdens het project in de toolstack. Dat zorgt ervoor dat we je werk kunnen op ieder moment kunnen inzien, waardoor onze begeleiding makkelijker wordt. Andersom geformuleerd: als je ons het begeleiden onmogelijk maakt moeten we je een onvoldoende geven. Hoe kun je dat voorkomen?

- Je gebruikt je HAN-emailadres in de toolstack. Bijdragen die via andere mailadressen komen worden niet meegeteld.
- Je vervangt de anonieme avatar in de toolstack door een voor jou unieke. Zo zien docenten makkelijker wie het werk heeft gemaakt
- Alle teksten die je schrijft voldoen aan de eisen zoals gesteld in de AIM-controlekaart.

- Teksten die je in Confluence schrijft schrijf je daar direct. Het is dus niet de bedoeling dat je eerst in Word iets opschrijft en dan in één klap uploadt naar Confluence. We zien zo of jullie goed samenwerken aan de producten.
- Aan het einde van de dag staat er geen code meer lokaal. Mocht je ziek worden kunnen je team ermee verder.
- Je zorgt dat al je werk (óók je individuele verslag) voortdurend direct toegankelijk is voor alle docenten die tijdens het OSM-semester begeleiden. Begeleiding én beoordeling vinden plaats aan de hand van het werk dat in de toolstack staat. Mocht een docentenwissel nodig zijn kan die vlot plaatsvinden. Het makkelijkste los je dit op door geen restricties te gebruiken. Wil je tóch dat je medestudenten niet meelesen? In confluence is de er de mogelijkheid om een Pagina van restricties te voorzien. Geef daar **Groep Docenten én jezelf** toegang. Let op: Als je jezelf geen toegang geeft kun je niet mee bij je eigen werk. Wij kunnen die niet voor je oplossen.
- Je levert je werk op tijd in op iSAS. Dit is voor archiveringsdoeleinden, maar er geldt: niet ingeleverd op iSAS, is geen cijfer:
 - o Vóór het assessment op het Plan van Aanpak (IND PVB), trek je een export van het PVA van Confluence, en dit Word-document load je up in iSAS. Is dit document er niet, dan kunnen we geen Vink voor je invoeren
 - o Tussentijds lever je uitsluitend je individuele werk in op iSAS bij (IND-TT). Je trekt een export van je verslag, en dit Word-document load je up in iSAS. Wellicht ten overvloede: het groepswerk hoeft tussentijds NIET op iSAS te worden ingeleverd.
 - o Aan het EIND Je trek je een export van je projectverslag, en dit Word-document load je up in iSAS bij IND-EIND). Daarnaast load ieder groepslid ook een dump up van alle gegevens uit de toolstack, bij GROEP-EIND. Je ontvangt van de trekkers van het OSM-project tijdig een link naar die dump.
- Je stelt je begeleidbaar op. Dat betekent je voor je teamgenoten en docenten bereikbaar en op afgesproken momenten aanwezig bent, en dat je verkregen feedback serieus neemt.
- Werken aan het verslag doe je binnen de 40 uur die je per week werkt aan het project. Een vaste dag inroosteren voor werken aan het verslag is niet toegestaan. Zo benut je de tijd suboptimaal: het schrijven moet dan per se op die ene dag gebeuren, terwijl er dagen zullen zijn waarin je je tijd veel nuttiger kunt besteden dan door een verslag te schrijven. Zo kan het ook voorkomen dat je een inzicht hebt waarvan je denkt: als dat dat meteen opschrijf dan heb ik het maar mooi 'in the pockt'. In het algemeen geldt dat je in het begin van het project minder 'verslagtijd' nodig zult hebben dan valk voor de inlevermomenten. Houd daar in de planning rekening mee.

BIJLAGE A. GANG VAN ZAKEN TOETS IND PVB

UITGANGSPUNTEN BIJ DE TOETSING

1. De toets maakt gebruik van het plan van aanpak dat door de projectteams voor de semesterprojecten wordt geschreven
2. De aard van de toets is een 'verrijkend assessment'. Daarmee bedoelen we dat de toets, naast een beoordelingsmoment, nadrukkelijk bedoeld is om studenten inzicht te geven in de kwaliteit van hun plan van aanpak, en op basis daarvan –als nodig- het plan van aanpak te verbeteren.
3. Er wordt getoetst op basis van het geldende document voor het schrijven van het plan van aanpak (op het moment van schrijven 'toelichting op het plan van aanpak v3.1')
4. Het plan van aanpak zelf wordt NIET beoordeeld (dat 'gaat mee' in de tussentijdse beoordeling van de groep)
5. In de toets wordt getoetst of:
 - o De individuele student de begrippen die nodig zijn om te komen tot een volledig en consistent plan van aanpak begrijpt, en kan plaatsen in het kader van het betreffende project.
 - o De student voldoende bijdrage heeft geleverd aan het tot stand komen van het plan van aanpak.
6. Dus: het plan van aanpak kan inhoudelijk slecht, of zelfs onvoldoende zijn, maar als de individuele student in zijn mondelinge toets in staat is de juiste verbeterpunten aan te geven kan hij slagen voor deze toets.

EERSTE KANS

Werkwijze

1. Het docententeam van het semesterproject geeft de projectgroepen een deadline voor het inleveren van het definitieve plan van aanpak.
2. Het plan van aanpak wordt voor de deadline klaargezet op Confluence.
3. Binnen 2 weken na de deadline nodigt de professional skills-docent de studenten uit voor het mondeling assessment, samen met de procesbegeleider.
4. Tijdens het assessment wordt iedere individuele student over minimaal twee willekeurige onderwerpen van het plan van aanpak ondervraagd.
5. Het plan van aanpak mag daarbij worden meegenomen naar het assessment, de toelichting op het plan van aanpak niet.
6. Studenten die de eerste kans missen, nemen deel aan de herkansing.

Cesuur

Een student krijgt een voldoende voor de toets als hij: voldoende heeft meegewerkt bij het maken van het plan van aanpak, als hij de onderliggende begrippen die nodig zijn voor het schrijven van het plan van aanpak voldoende begrijpt, en als hij samenhang/verbanden tussen de enkele begrippen aan de hand van het voorliggende plan van aanpak kan toelichten.

TWEEDE KANS

1. Wanneer het mondeling assessment onvoldoende is beoordeeld, nodigt de professional skills-docent de student uit voor een herkansing. Deze vindt plaats uiterlijk in week 6 van het project.
2. Voor deze toets neemt de student zijn eigen plan van aanpak mee naar de toets. De toets bestaat eruit dat hij dit plan van aanpak door middel van schriftelijke feedback moet verbeteren tot een consistent verhaal.

DERDE KANS

1. Wanneer een student ook de herkansing niet heeft gehaald levert hij –met het projectverslag aan het einde van het project- een kritische evaluatie van het plan van aanpak in. Daarin gaat hij in op het verloop van het project, de rol die het plan van aanpak daarin heeft gespeeld, en de rol die het plan van aanpak erin had kunnen of moeten spelen.

BIJLAGE B. FEEDBACKKAART DAILY STAND UP

Scores: Blanco = niet waargenomen; - = niet aanwezig; + = aanwezig; ++ = sterk aanwezig

	Score	Toelichting, geef concrete voorbeelden, waarnemingen (geen interpretaties zonder waarnemingen!)
Standup begint op tijd		
JIRA-scrumbord is zichtbaar voor alle teamleden		
JIRA-scrumbord wordt tijdens de standup actief gebruikt		
JIRA-scrumbord is na afloop up-to-date		
Standup verloopt volgens een logische opbouw (gisteren, vandaag, blocks)		
Er wordt een eensluidend en d.m.v. tooling (burndownchart) onderbouwd antwoord gegeven of de sprint gehaald gaat worden		
Standup duurt niet langer dan 15 minuten		
Alle deelnemers aan de standup waken ervoor dat sprekers voldoende concreet zijn. Er wordt gesproken over: <ul style="list-style-type: none"> - Concrete taken (issuenummers) - Concrete statussen (to do-in progress- (to review)– done) 		
De Standup wordt duidelijk geleid door de scrummaster		
Scrummaster zorgt dat bespreek/discussiepunten die ontstaan tijdens de standup direct erna worden ingepland		
Scrummaster vraagt of taken moeten worden toegevoegd aan sprintbacklog		
Scrummaster vraagt of teamleden vorige dag iets nieuws hebben geleerd dat relevant is voor anderen		

BIJLAGE C. HOE VOER JE EEN RETROSPECTIVE UIT BIJ OSM?

Wanneer een retrospective?

Scrum schrijft voor dat de sprint retrospective wordt gehouden nadat de sprintreview is geweest. Op school draaien we dit om: voordat je je product in de sprintreview toont aan de opdrachtgever houden jullie een sprint retrospective en plan je de volgende sprint. Dit heeft als voordelen:

- Je kunt professioneel uitleggen waarom het product is geworden wat het is geworden omdat je een analyse 'bij kunt leveren'. Anders gezegd: "je komt verder dan: sorry sprintdoelen niet helemaal gehaald". Dit geldt andersom ook: als je het wel gehaald hebt moet je kunnen uitleggen of dat een gelukje was (te weinig hooi op de vork), of resultaat van een beheerst proces.
- Je hebt een idee van wat er in de volgende sprint nodig is en mogelijk is (qua tijd) en je kunt dus niet alleen een voorstel doen, maar ook verzoeken van de opdrachtgever sneller op waarde schatten.

Uiteraard bespreek je de sprintreview altijd na, en bepaal je of het gesprek aanleiding geeft om toch nog een keer naar de sprintplanning te kijken. Als nodig voer je dus nog een herplanning uit.

Centrale vragen in de retrospective

De retrospective centreert zich op een drietal kernvragen:

- Wat ging er goed?
- Wat ging er fout?
- Wat doen we de volgende keer weer zo?

Dit veronderstelt op een antwoord op de volgende vragen:

- Wat is "wat"? Waar moeten we naar kijken?
- Wat is "goed" of "fout"? Wat is een goede norm?

Een sprint retrospective helpt je maatregelen te formuleren ter verbetering van de product- en proceskwaliteit, eerder genomen maatregelen op hun effectiviteit te evalueren en de maatregelen te verbeteren of vervangen door andere (betere) maatregelen.

Vorbereiding retrospective

Om een goede retrospective te doen moet die worden voorbereid. Daarvoor moeten de volgende metriecken in kaart worden gebracht. Deze gegevens moeten worden uitgesplit naar de aard van de taken, in de volgende categorieën:

- Uitzoeken
- Planning & projectmanagement
- Analyse & Ontwerp
- Implementatie & testen
- Overige

Hieronder vind je de tabel die je kunt gebruiken om de retrospective voor te bereiden

Aanvullend aan deze tabel geldt:

- Per type taak zijn de plannings- en uitvoeringstijden in een grafiek uitgezet tegen de ingeschatte complexiteit van het planningpokeren.
- Van alle meetgegevens moet de ontwikkeling in de tijd in grafieken of tabellen beschikbaar zijn. Je moet dus in een grafiek zichtbaar maken dat het inderdaad beter gaat (of niet). De grafieken 'vullen' zich dus, gedurende het project.

Hoe ga je te werk:

- In de eerste sprint retrospective wordt de basiskwaliteit van het team vastgelegd door een eerste meting van de gegeven metrieken (zie hiervoor de tabel. In de volgende sprints dienen die metrieken verbeterd te worden.
- Van de sprintplanning en -uitvoering wordt expliciet de huidige kwaliteit beoordeeld, vergeleken met de vorige sprint en wordt de verbeterdoelstelling (indien van toepassing) vastgelegd.
- De maatregelen uit de vorige sprint retrospective worden expliciet geëvalueerd op hun effectiviteit
- Per verbeterdoelstelling worden de reeds bestaande maatregelen eventueel aangescherpt dan wel (nieuwe) maatregelen genomen.
- Uiteraard wordt er van iedere retrospective een verslag gemaakt, met actiepunten

	Onderdeel	Metric	Toelichting	Te verzamelen door
Proces	Werktijd	beschikbare werktijd	40 uur per week minus planmatig niet werkbaar: verlof, vakantie, feestdagen	In S1 geen vrije dagen dus uitgangspunt is 40 uur p/p Bijvoorbeeld $5 \times 2 \times 40 = 400$ uur of $6 \times 2 \times 40 = 480$ uur In S2 per sprint corrigeren voor het aantal werkdagen in de voorliggende sprint
		% benut als werktijd	Som van het aantal uren besteed aan Jira-taken gedeeld door beschikbare werktijd.	Jira->Tempo->Reports->Project Report en dan juiste start- en einddatum selecteren. Bijvoorbeeld $176 / 400 = 44\%$
	Sprintplanning	Aantal taken toegevoegd na start sprint		Project->Report-> Created vs. Resolved Issues Report. Days Previously terugtellen naar eerste dag van de sprint. Neem het aantal taken op dat bij Created staat op de dagen na de start van de sprint.
		Aantal taken gewijzigd na start sprint	De taak moet inhoudelijk sterk gewijzigd zijn en/of de tijdschatting moet voor de taakuitvoering met meer dan 25% gewijzigd zijn	Jira->Tempo->Reports->Project Report. Group By Issue, Kies voor een List View en selecteer de kolommen Worked Σ en Estimated. Exporteren naar Excel, je krijgt een rapport waarin taken meerdere keren voorkomen, tel de Billed Hours hiervoor bij elkaar op en vergelijk deze met de Original Estimate.
	Planningpoker	% correct ingeschat	De taakduur is correct ingeschat als de uitvoering niet meer dan 25% afwijkt van de oorspronkelijke ingeschatte tijd en deze afwijking niet heeft geleid tot uitlopen van, of inlopen op de planning	Als je in uren geschat hebt is dit hetzelfde als bovenstaande metric. Als je de hoofdtaken in punten gepokerd hebt, beschrijf dan hier: <ol style="list-style-type: none"> 1. de som van het aantal geschatte punten 2. de som van het aantal gerealiseerde punten 3. de hoeveelheid geschatte uren

	Onderdeel	Metric	Toelichting	Te verzamelen door
				4. de hoeveelheid gemaakte uren 5. de verhouding tussen 1 en 3 6. de verhouding tussen 2 en 4
		% te lang ingeschat	Is hetzelfde als "Aantal taken gewijzigd na start sprint"	nvt
		% te kort ingeschat	Is hetzelfde als "Aantal taken gewijzigd na start sprint"	nvt
Product	Sprintdoelen	Sprintdoel gehaald	Sprintdoel gehaald of niet (ja/nee). Je kunt een sprintdoel halen zonder dat alle geplande features gerealiseerd zijn, neem deze nuance/informatie wel op.	Ja/Nee
		Aantal afgesproken user stories of use cases	Het aantal features dat bij de planning is meegenomen.	Het aantal hoofdtaken/issues/use cases. Project->Report->Sprint Report. Tel het aantal taken van het type Task/Issue/Use Case/User Story.
		Aantal gehaalde user stories of use cases	Het aantal features die bij de sprintreview werkend gedemonstreerd zouden kunnen worden	Zou kunnen blijken uit een testrapport en/of sprint review verslag waarin de PO aangeeft dat de feature akkoord is.
		Aantal user stories of use cases gecontroleerd op, en voldoende aan de kwaliteitscriteria	Dit zijn de kwaliteitscriteria zoals opgenomen in het PvA, eventueel bijgewerkt op basis van ervaringen in het project	Bespreek met je team hoe je dit zou kunnen meten. Je kunt iedereen individueel een schatting laten doen en deze opschrijven en uitmiddelen en besluiten of je voldoende "hard" kunt maken dat de features terecht in "Done" staan.
	Jira-taken	Aantal geplande taken	Het aantal taken dat bij de start van de sprint in de sprintbacklog zijn opgenomen	Project->Report-> Created vs. Resolved Issues Report. Days Previously terugtellen naar eerste dag van de sprint. Neem het aantal taken op dat bij Created staat op de dag van de start van de sprint.
		Aantal uitgevoerde taken	Het aantal taken uit de sprintbacklog die bij de sprintreview op "resolved" (of "Done") staan	Project->Report-> Created vs. Resolved Issues Report. Days Previously terugtellen naar eerste dag van de sprint. Neem de som van het aantal taken op dat bij Resolved staat vanaf de

	Onderdeel	Metric	Toelichting	Te verzamelen door
				dag van de start van de sprint. Project->Report->Pie Chart Report->Assignee. Bekijk de werkverdeling.
		Aantal taken gecontroleerd op, en voldoende aan de kwaliteitscriteria	Dit zijn de kwaliteitscriteria zoals opgenomen in het PvA, eventueel bijgewerkt op basis van ervaringen in het project	Bespreek met je team hoe je dit zou kunnen meten. Je kunt iedereen individueel een schatting laten doen en deze opschrijven en uitmiddelen en besluiten of je voldoende "hard" kunt maken dat de taken terecht in "Done" staan.
	Code	Aantal commits	Het aantal commits in de branch die heeft geleid tot het product dat gedemonstreerd is bij de sprintreview (development of eventueel master branch)	BitBucket->Project->Graphs. Selecteer de juiste branch en noteer het aantal commits uit de tab Activity. BitBucket->Project->Graphs. Selecteer de juiste branch en noteer het aantal commits per developer uit de tab Contributors. Als teamleden er twee keer in staan dan zijn er meerdere git-accounts gebruikt, geen probleem, registreer dan alle e-mailadressen bij de betreffende gebruiker, dan worden de commits bij elkaar opgeteld.
		Aantal commits gecontroleerd op, en voldoende de kwaliteitscriteria	Dit zijn de kwaliteitscriteria zoals opgenomen in het PvA, eventueel bijgewerkt op basis van ervaringen in het project	Bespreek met je team hoe je dit zou kunnen meten. Je kunt iedereen individueel een schatting laten doen en deze opschrijven en uitmiddelen en besluiten of je voldoende "hard" kunt maken dat de programmeertaken terecht in "Done" staan.
		% geslaagde unittesten en % code-coverage	Gebruik SonarQube of een alternatieve reportage	SonarQube Dashboard. Als de test-coverage en tests niet zichtbaar zijn check dan of de xml-bestanden van de rapporten (LCOV of Cobertura formaat) worden gemaakt en worden uitgelezen door SQ (eventuele fouten zijn te zien in de Jenkins build log)
		Metrics statische codecheckers	Gebruik SonarQube of een alternatieve reportage	SonarQube Dashboard. Noteer het aantal bugs, violations, technical debt en

	Onderdeel	Metric	Toelichting	Te verzamelen door
				duplications.

**OPEN UP
NEW **HAN_**UNIVERSITY
OF APPLIED SCIENCES
HORIZONS.**