

Handleiding Proftaak S6 – Ordina JTech Coin

Versie: januari 2018
Samenstellers: Edwin Derks, Kevin Donkers

1. De casus	2
1. Voorkennis.....	2
2. Context	3
3. Functionele beschrijvingen van de systemen	3
1. De Blockchain	3
2. Transacties	3
3. De portemonnee.....	4
4. Monitoring.....	4
5. Mining	4
6. De infrastructuur	4
7. Shop	4
4. Niet-functionele requirements van de systemen	4
2. Organisatie	5
1. Studiebelasting	5
2. Weekindeling	5
3. Verantwoordelijkheden projectteam en omgeving	6
4. De werkmethode	7
5. Voorwaarden vorm eindproducten.....	7
3. Beoordeling	9
1. Proftaakbeoordeling door tutor	12
2. Proftaakbeoordeling door bedrijfsexpert.....	12
3. Vakinhoudelijke beoordeling: SOP6	12
4. Vakinhoudelijke beoordeling: DPI6	13
5. Vakinhoudelijke beoordeling: JEA6	14
6. Vakinhoudelijke beoordeling: OND6.....	15
7. Vakinhoudelijke beoordeling: DBO6	15

We beschrijven hier de organisatorische en inhoudelijke aspecten van de proftaak ESD6: Enterprise Software Development van semester 6 van het S-profiel.

Gedurende dit semester werk je in de proftaak aan de casus "Ordina JTech Coin". In deze casus kom je in aanraking met softwarearchitectuur, rich internet applications, enterprise applicaties, enterprise architectuur, schaalbare databases & softwarebeheer en -kwaliteit.

In meerdere iteraties wordt een omvangrijk Enterprise Software systeem opgebouwd dat bestaat uit meerdere samenwerkende software systemen in Java EE 8. Je wordt begeleid door een tutor die als product owner nieuwe requirements aanlevert.

In deze proftaak werk je aan de ontwikkeling van alle competenties voor het Software-profiel op niveau 3. Een voorbeeld van zo'n competentie die in deze proftaak aan bod komt is:

Een requirementsanalyse uitvoeren voor een softwaresysteem met verschillende belanghebbenden in een context van bestaande systemen. Bouwen en beschikbaar stellen van een softwaresysteem dat aansluit bij bestaande systemen, volgens de ontworpen architectuur met gebruik van bestaande frameworks.

Binnen de proftaak willen we betekenis geven aan de complexiteit die inherent is aan het ontwikkelen van software, zoals het gebruik van innovatieve technieken, rekening houden en anticiperen op evoluerende requirements, het vrijgeven van software op basis van continuous deployment, efficiënt communiceren over architectuurontwerpen, het realiseren van berichtuitwisseling tussen servers en kwaliteitsborging door automatische testen in verschillende testomgevingen.

Hierbij is de casus geformuleerd vanuit de vraag van de klant en is het de verantwoordelijkheid van de studenten om uit de geleerde technieken een beargumenteerde keuze te maken en deze op een professioneel niveau toe te passen.

Verderop worden eerst de doelstellingen, de productbeschrijving en de vereiste voorkennis van de proftaak beschreven. Vervolgens de organisatie van de proftaak, met taakverdeling en betrokken begeleiders. Als laatste staat een hoog niveau omschrijving van de beoordeling van de proftaak.

1. De casus

1. Voorkennis

Om de proftaak succesvol te kunnen uitvoeren moet je kunnen analyseren, ontwerpen en programmeren zoals in de competenties op niveau 2 vastgelegd. Daarnaast moet je de onderwerpen die parallel aan dit project in de verschillende deelvakken aan de orde komen, voldoende snel oppakken.

De cursus JEA6 moet gevolgd worden om mee te kunnen doen in de proftaak. Een student die de leerstof van JEA6 aan het einde van de eerste 10 weken in onvoldoende mate beheerst wordt uitgesloten voor de proftaak.

2. Context

De afdeling JTech binnen Ordina is op zoek naar een betaalmiddel waarbij medewerkers beloond worden voor het bezoeken van evenementen en het bijhouden van hun kennis. Het idee is om een eigen cryptovaluta te ontwikkelen waarbij middels een blockchain munten worden uitgegeven aan medewerkers.

Elke medewerkers zal zijn eigen **portemonnee** aanmaken waarna hij kan beginnen met het verzamelen van munten. Deze munten worden uitgegeven in een **centrale blockchain** en het moet ten alle tijde mogelijk zijn om de historie van een bepaalde block in de chain terug te zien. Er dient een **administratief account** te komen die de mogelijkheid heeft munten te verkrijgen. Deze munten kan hij vervolgens **overmaken** naar medewerkers die bijvoorbeeld een evenement bezoeken of een andere prestatie leveren. Om inzicht te krijgen in de grootte van de chain dient er een **monitoringsysteem** te zijn waarin het administratieve account kan inloggen en de huidige totale uitgave kan inzien. Daarnaast dient er een mogelijkheid te zijn om voor medewerkers munten te "**minen**", dit zal gebeuren middels het beantwoorden van vragen uit een database, die naar grootte van de chain telkens moeilijker en meer worden.

3. Functionele beschrijvingen van de systemen

1. De Blockchain

Het basisconcept van blockchain is vrij eenvoudig: een gedistribueerde database die een voortdurend groeiende lijst met geordende records bijhoudt. Elke individuele block heeft een eigen identificatie waarde en een waarde van de vorige block zodat we ook daadwerkelijk weten welke positie deze block in de chain heeft.

Het is niet mogelijk om met de gegevens van een block te zien wie de eigenaar van dit block is. Dit is namelijk geheel anoniem en niet te herleiden. Belangrijk is dus dat er nagedacht wordt hoe dit te beveiligen zodat het niet mogelijk is om blocks van iemand anders te stelen.

2. Transacties

Transacties bestaan uit twee componenten: inputs en outputs. Uitgangen specificeren waar de munten naartoe worden verzonden en invoeren geven een bewijs dat de munten die daadwerkelijk worden verzonden in de eerste plaats bestaan en eigendom zijn van de "afzender".

Om een munt te versturen dien je eerst zijn openbare sleutel te ontvangen. Er dient hierbij gebruik gemaakt te worden van de Public-key cryptografie methodiek. De openbare sleutel kan worden afgeleid van de geheime sleutel, maar de geheime sleutel kan niet worden afgeleid van de openbare sleutel. De openbare sleutel (zoals de naam al aangeeft) kan veilig worden gedeeld met iedereen.

Het is belangrijk dat de inhoud van de transactie niet kan worden gewijzigd nadat deze is ondertekend. Omdat de transacties openbaar zijn, kan iedereen toegang krijgen tot de transacties, zelfs voordat ze zijn opgenomen in de blockchain.

3. De portemonnee

Het doel van de portemonnee is om een meer abstracte interface voor de eindgebruiker te creëren.

Hierbij moet de gebruiker de volgende acties kunnen uitvoeren:

- Maak een nieuwe portemonnee (= privésleutel in dit geval)
- Bekijk het saldo van zijn portemonnee
- Stuur munten naar andere adressen

4. Monitoring

Om de historie van een munt in de blockchain te zien, dient er een interface te komen die middels een waarde de gehele historie aan transacties inzichtelijk maakt. Daarnaast zou je informatie kunnen aanvragen over een specifiek block waarbij je weet vanaf wanneer dit block bestaat.

5. Mining

Voor Ordina dient het minen als onderdeel om ontwikkelaars middels Gamification hun kennis te trainen. Er dienen vragen beantwoord te worden middels een interface waarna er munten kunnen worden verdiend. Deze vragen worden naar mate de grote van de blockchain steeds meer en lastiger. Hiervoor dient een aparte database te worden benaderd die helemaal los staat van de verdere blockchain.

6. De infrastructuur

Er dient rekening gehouden te worden met een groot aantal gebruikers. Het is daarom essentieel dat er niet dubbel dezelfde munten worden uitgegeven tijdens het minen. Ook moeten de saldo's ten alle tijden kloppen van de gebruikers. Houdt daarom rekening met een poolingsmechanisme om te voorkomen dat munten meerdere malen worden verkocht/verkregen.

7. Shop

In de shop kunnen middels coins items besteld worden waarna dit van het saldo van de medewerker zal worden afgeschreven. De shop zal hierbij een eigen wallet hebben geconfigureerd waar deze coins naartoe zullen worden geschreven.

4. Niet-functionele requirements van de systemen

Het is de ontwikkelaars om kritisch aan de slag te gaan met de niet-functionele requirements:

- De systemen moeten onafhankelijk van elkaar zijn, dus ze beschikken niet over een gedeeld bestandssysteem of een gedeelde database.

- Alle systemen moeten goed beveiligd zijn tegen misbruik. De voorzorgen tegen misbruik moeten expliciet in de documentatie worden vermeld.
- Het uiteindelijke systeem bestaat uit deelsystemen die volgens te ontwikkelen protocollen met elkaar communiceren. Deze protocollen zullen, net zoals de systemen, aan verandering onderhevig zijn tijdens het ontwikkelen.
- Het is van belang via goed configuratiebeheer vast te leggen wat de afhankelijkheden tussen de verschillende protocol versies en de verschillende systeemversies zijn.

2. Organisatie

Commented [DK1]: Graag aanvullen door Frank op basis van huidige situatie

1. Studielast

De proftaak heeft een pittige casus waar veel nieuwe technologieën aan bod komen. Je zal veel tijd nodig hebben om alles te realiseren. Houd hier rekening mee bij de planning van je werk.

Opgelet: enkel contacturen met de tutor staan in je rooster. Het is de verantwoordelijkheid van de groep om **extra tijd te reserveren** om zelfstandig te werken.

2. Weekindeling

In de eerste periode wordt er **een dagdeel ingeroosterd** om te werken aan de proftaak. Het is de verantwoordelijkheid van de groep om **een extra dag te reserveren** om zelfstandig te werken.

Bij elk blok van ingeroosterde projecturen zal **een deel van de tijd de tutor aanwezig** zijn. Als projectgroep maak je hierover afspraken. Omdat gezamenlijk een product gemaakt wordt, waarbij ieder individueel een deeltaak voor zijn rekening neemt is het essentieel de samenwerking goed op elkaar af te stemmen.

In de tweede periode moet er meer tijd besteed worden aan de proftaak, door hieraan 3 'volle' dagen te werken.

De proftaak wordt in lesweek 2 opgestart tijdens een bijeenkomst met de tutor.

In onderstaande figuur wordt de samenhang met de andere vakken schematisch weergegeven. PiE staat voor Partner in Education, waarbij er gastcolleges geregeld worden.

	Wk 1	Wk 2	Wk 3	Wk 4	Wk 5	Wk 6	Wk 7	Wk 8	Wk 9	Wk 10	Wk 11	Wk 12	Wk 13	Wk 14	Wk 15	Wk 16	Wk 17	Wk 18	Wk 19	Wk 20
vrije dagen																				
ESD6		Opstart			Expert	Sprint 2		Expert	Sprint 3		Expert	Sprint 4		Expert	Sprint 5		Expert			
SOP6					OTAP															
DBO6										DBO6	WebServices vanuit de database									
DPI6																				
JEA6		PIE	PIE					PIE						PIE						
Back-end		Backend						Front-end		Integratie				Capita selecta						
PES6		OND	SOL6	OND	SOL6	OND	SOL6	OND	SOL6	PIE		OND	SOL6	OND	OND	OND	OND	OND	OND	OND
		AFS6		AFS6		AFS6		AFS6				AFS6	OVH6	AFS6	OVH6	AFS6	OVH6	AFS6	OVH6	AFS6
												PIE/RA	RAP6	RAP6						

3. Verantwoordelijkheden projectteam en omgeving

Het projectteam bestaat uit (bij start) van ongeveer 5 en maximum 6 studenten. Dit wordt ten laatste in week 1 via je studenten-email bekend gemaakt, en gepubliceerd in Canvas.

Ieder binnen de groep is verantwoordelijk voor het **noteren van de feedback** die door de verschillende belanghebbenden gegeven wordt, waarbij deze notities voor het verlaten van het overleg gelezen moeten kunnen worden.

Voor de uitvoering van de proftaak **bewaakt de tutor het groepsproces** en neemt de verschillende soorten feedback samen in **een beoordeling van de groepsproducten**.

Er is regelmatig **een aan de school externe bedrijfsbegeleider** aanwezig die vanuit zijn specifieke expertise vragen kan beantwoorden, feedback kan geven op de gemaakte keuzes en advies kan formuleren om in de komende weken iets anders te doen. De aanwezigheid is afhankelijk van agenda's, maar wordt om de 3 weken ingepland.

Er zullen driewekelijks momenten gepland worden waarbij de voortgang zal worden gepresenteerd aan de **bedrijfsexperts** van Ordina. Hierbij zullen tevens mogelijke nieuwe requirements worden bepaald voor volgende sprints.

Samenvattend zijn de volgende personen bij de proftaak betrokken:

Wie	Aanwezigheid	Rol
Tutor	wekelijks	Procesbegeleider Product Owner: bewaakt de functionaliteit van het systeem en geeft hierover namens de klant zijn visie. Backlog manager Bepaalt eindpunt (zie beoordeling)
Groepslid	dagelijks	Software ontwikkelaar Peer- en self reviewer Presentator Ev. afgevaardigde van de groep
Extern groepslid	dagelijks	Integratie afgevaardigde Monitoring afgevaardigde Peer assessor
Vakdocent	3 x op afspraak	Consultant (niet-functioneel) als technisch adviseur. Deelpuntbepaler (zie beoordeling)
Externe expert	3 wekelijks	Consultant Deelpuntbepaler (zie beoordeling)

4. De werkmethode

De proftaak zal volgens de **Scrum-methode** worden uitgevoerd, zoals gezien in semester 4, waarbij er **sprints van 3 weken** gepland worden.

Er zijn 5 sprints:

Wk 1	Wk 2	Wk 3	Wk 4	Wk 5	Wk 6	Wk 7	Wk 8	Wk 9	Wk 10	Wk 11	Wk 12	Wk 13	Wk 14	Wk 15	Wk 16	Wk 17	Wk 18	Wk 19	Wk 20
								vr	ma	vr			do, vr	di	ma				
Next	Opstart																		
	Voorbereiding																		
	Sprint 1			PIE						Sprint 2			PIE						Uitloop/ Reparatie
							Sprint 3			PIE									
										Sprint 4			PIE						
												Sprint 5			PIE				

De exacte data van de oplevermomenten en de beschrijving van de deliverables zijn te vinden in **Canvas**.

Wekelijks zal er een projectvergadering gehouden worden met de tutor.

De proftaak moet door de studenten in meerdere incrementen worden opgedeeld. De projectgroep maakt zelf een detailplanning om het pad naar de deliverables in kaart te brengen en te borgen.

Na elk oplevermoment wordt er een bijeenkomst houden met de producteigenaar, de groepsleden en (eventueel) vakdocenten en de externe bedrijfsexpert, waarin er gereflecteerd wordt, vooruitgekeken en een demo gegeven wordt van de werkende systemen.

De tutor moet zicht hebben op de voortgang van de proftaak, waarbij elke oplevering een overzicht bevat van gerealiseerde, uitgestelde en openstaande use cases. Deze wordt als een leesbare PDF bij elke sprintoplevering toegevoegd.

De tutor treedt op als beheerder tijdens de GO-/NO GO-beslismomenten in het proces. Wijzigen aan de features die wel of niet uitgewerkt worden, dienen bijgehouden te worden, zodat de reden achter proceswijzigingen toegelicht kan worden.

5. Voorwaarden vorm eindproducten

In overleg met de tutor wordt de wijze van de oplevering bepaald. Zo is er de mogelijkheid om dit in te dienen in Canvas, waarbij de feedback genoteerd wordt in Feedpulse (een plugin van Canvas).

Commented [DK2]: Graag aanvullen door Frank op basis van huidige leertraject en wensen op basis van Java / Angular / Spring.

De proftaakgroep is in beperkte mate vrij om **de technologieën** te kiezen:

- **Java EE 8** moet in de meerderheid van de deelproducten gebruikt worden.
- Exact 1 oplossing moet draaien in een ander type Enterprise Application Server, eventueel met **een andere software stack**, zodat jullie een oplossing laten zien voor de integratie moeilijkheid.
- Voor de front-end mag je kiezen uit **de programmeertalen die gezien zijn in JEA6**.
- De databaselaag en de database moet gerealiseerd worden binnen een Persistence Framework. Zo moet object relation mapping mogelijk zijn, en correct gebruikt, zoals gezien in JEA6.
- De database moet kunnen werken met de technieken zoals stored procedures, zoals gezien in **DBO6**.

Jullie mogen dus één van de applicaties (naar jullie eigen keuze) realiseren in C#. De overige applicaties moeten in Java EE ontwikkeld worden. De reden dat we vasthouden aan Java EE voor de proftaak is dat we de technieken die worden aangeleerd in de vakken zoveel mogelijk terug willen laten komen in de proftaak. Niet alleen conceptueel, maar ook technisch-inhoudelijk. Om jullie te laten ervaren hoe het is om andere enterprise applicaties met Java EE-applicaties te laten samenwerken mogen jullie één applicatie in bv. C# of Java Spring ontwikkelen, maar niet meer dan dat.

De eindproducten zijn te onderscheiden in onderstaande categorieën en per product vermelden we ook de nodige documentatie.

Wat	Verantwoordelijkheden	Documentatie
Draaiend systeem dagelijks	Portal website zodat er geen url's getypt moeten worden <ul style="list-style-type: none"> • Systemen kunnen gegevens versturen en ontvangen van andere systemen • Moet altijd ter beschikking staan van de belanghebbenden • Moet on demand aangepast kunnen worden volgens een gecontroleerd en kwalitief goed releaseproces 	Infrastructuur <ul style="list-style-type: none"> - Infrastructuur diagram met betrokken servers en services - Oplijsting IP's/url's & (test)username en wachtwoorden Functionaliteiten <p>Screenshots van belangrijkste, werkende functionaliteiten en uitleg indien nodig</p> OTAP documentatie
Resultaten van monitoring dagelijks	Geeft overzicht van beschikbaarheid van systemen	Monitoring <ul style="list-style-type: none"> - Gebruikte tooling, connectiegegevens + authenticatie en ev. gebruikte configuratie - Historiek: per systeem up-time overheen semester en aantal werkende end points

Wat	Verantwoordelijkheden	Documentatie
Verslag van planning en werkopvolging 3 wekelijks	Planningstool Overzicht taakverdeling, gepresteerde uren, prioriteiten en afgewerkte functionaliteiten	Planning met per sprint welke userstories uitgewerkt zouden worden, ontwikkelaars, aantal gewerkte uren en status op het einde van de sprint Notulen van stand-up meetings
Opgeleverde software 3 wekelijks	Versiebeheer Broncode, configuratiebestanden, systeemopzet, ... Moet altijd ter beschikking staan van de belanghebbenden. ZIP Archief zonder versiebeheergeschiedenis met broncode.	Ontwerp Overzicht ontwerp en argumentatie De ontwerpschema's moeten in werkformaat (bv. Visual Paradigm) en publicatieformaat PNG afgeleverd worden.
Kennisdeling systeem op afspraak met tutor, maar minstens 2x in semester	Presentaties slides, & video/geluidsopnames van presentaties	Feedback Verzameling van gewijzigde requirements of ontwerpkeuzes, verkregen feedback van expert/vakdocenten. Soms is het logischer om de feedback te verweven met de andere hoofdstukken. Duidt gekregen feedback helder aan.

3. Beoordeling

De proftaak levert op verschillende tijdstippen een aantal beroepsproducten op, welke door verschillende personen geëvalueerd worden.

Hierbij toont een student zijn of haar competentie op het gebied van enterprise software-ontwikkeling:

De student kan

- vanuit een complexe casus in een team en overheen teams samenwerken om een enterprise architectuur te ontwerpen, realiseren en beheren
- een requirementanalyse opstellen van nodige componenten
 - rekening houdend met verschillende belanghebbenden en kwaliteitseigenschappen
 - keuzes van componentverantwoordelijkheden beargumenteren
- nieuwe software componenten ontwerpen
 - waarbij de productdocumentatie de belangrijkste beslissingen weergeeft en beargumenteert.
 - er onderzoek gedaan is naar de koppelingsmogelijkheden tussen verschillende applicaties, en verslaglegging van de argumentatie
 - de uitgewerkte koppelingsmogelijkheden volgens afspraken gedocumenteerd werden

- software componenten realiseren op een Enterprise niveau
 - waarbij er gekeken wordt naar de demonstratie van de werking
 - de kwaliteit van de broncode.
 - functionele volledigheid (en de complexiteit ervan)
 - gewaarborgd niveau van uitvoerings-, uitbreidbaarheid, schaalbaarheid en robuustheid
 - testen van de software koppelingen
- de software componenten operationeel aanbieden waarbij
 - die door verschillende stakeholders gebruikt kunnen worden
 - wijzigingen gecontroleerd doorgevoerd kunnen worden
 - de beschikbaarheid gemeten kan worden

De tutor bepaalt het eindpunt. Hij heeft verschillende handvaten om het eindpunt van de proftaak te geven. Hiervoor worden de feedback van meerdere personen in beschouwing genomen.

De vakdocenten geven feedback op basis van **vakspecifieke rubrics** en **een beoordeling op basis van feedbackverwerking**.

De kwaliteit van de producten wordt bepaald aan de hand van **evaluatie door medestudenten, externe begeleiders en automatisch te berekenen indicatoren op basis van systeemmonitoring**.

De professionaliteit van het samenwerkingsproces wordt afgeleid op basis van **self en peerreviews**, waarbij de tutor werkpunten kan vaststellen.

Het is in het belang van de groep om dus zelf eigenaar te zijn van de feedback, en hier bij de vakdocenten gericht naar te vragen. Feedback over inschattingfouten of beperkingen is een mogelijkheid om te verbeteren. Het totale feedback rapport kan beheerd worden binnen Canvas, waarbij de groep de rol van notulist heeft, en de betrokken vakdocenten de feedback van anderen kunnen lezen en hier verder op in kunnen gaan. Het professioneel omgaan met feedback staat centraal.

Het eindproduct is een verzameling van deelproducten die je per oplevermoment gemaakt hebt en wordt beoordeeld door de vakdocenten, andere studenten, experts uit het bedrijfsleven en de tutor.

Om deze beoordelingen te kunnen uitvoeren hebben jullie een live omgeving draaiende waarop de software-componenten staan en een rapportagedocument dat de beslissingen en de argumentaties bevat.

Elke vakdocent kijkt op basis van de vakleerdoelen naar het product en kan daarover vragen stellen.

Je wordt in de groep individueel beoordeeld, waarbij gelet wordt op de samenwerking binnen de groep, je argumentatie en presentatie (zowel mondeling als schriftelijk) en de taken die je op jou genomen hebt en wat het resultaat daarvan is.

Mocht de bijdrage van de student aan het beroepsproduct positief of negatief afwijken ten opzichte van de groepsgenoten, dan kan de beoordeling van de student naar boven of beneden bijgesteld worden.

Doordat in de proftaak met een grote groep studenten wordt samengewerkt binnen een strakke tijdsplanning is het essentieel dat afspraken nagekomen worden. Projectdeelnemers die bij herhaling afspraken niet nakomen kunnen door de groep in samenspraak met de tutor **van deelname worden uitgesloten**. Hieraan zal altijd een officiële waarschuwing in de vorm van **een gele kaart** voorafgaan.

Samenvattend zijn de volgende personen betrokken bij de evaluatie:

Wie	Frequentie	Feedback op
Tutor	wekelijks	Proces Als product owner op functionele requirements
	3 evaluatie momenten	Eindpunt bepaler op basis van peer- en self reviews, evaluatie van vakrubrics door vakdocent(en) en evaluatie van externe bedrijfsexpert. Kwaliteit van presentaties Onderbouwing van gemaakte keuzes
Groepslid	3 wekelijks	Peer- en self reviewer Kwaliteit van presentaties
Extern groepslid	3 wekelijks	Kwaliteit van aangeboden interfaces Performantie van aangeboden interfaces Doeltreffendheid van communicatie Kwaliteit van presentaties
Vakdocent: DBO, SOP, DPI en JEA	3 evaluatie momenten	Evaluatie van vakrubrics Onderbouwing van gemaakte keuzes
Externe expert	3 evaluatie momenten	Onderbouwing van gemaakte keuzes Kwaliteit van presentaties

1. Proftaakbeoordeling door tutor

De proftaakgroep is in staat om in een complexe groepsomgeving op een professionele wijze software ontwikkelen. Hierbij zet de proftaakgroep maakt de groep goede afspraken, zet zich in conform de gemaakte afspraken en werken de studenten binnen de groep op een professionele manier samen.

Voldoende	Goed	Excellent
De groep werk aantoonbaar professioneel maar sommige zaken kunnen nog beter worden ingericht, taken worden structureel opgepakt maar worden af en toe nog niet geheel correct uitgevoerd en samenwerking verloopt op correcte wijze, waarbij er af en toe nog communicatieproblemen optreden.	De groep werkt aantoonbaar professioneel, alle taken worden structureel opgepakt en op correcte wijze uitgevoerd en afgerond en de samenwerking laat geen noemenswaardige problemen zien.	Het Goed niveau is gehaald, waarbij door de groep aangedragen non-triviale onderbouwing

2. Proftaakbeoordeling door bedrijfsexpert

Afhankelijk van de beschikbaarheid is er 5 keer een externe bedrijfsexpert beschikbaar waar de studenten de volgende feedback van verzamelen:

Professioneel gebruik maken van de expertise van de expert	Afgevaardigden van de groep zijn in staat om analyse, ontwerp of realisatie keuzes te presenteren en te beargumenteren
	Bij problemen in de realisatie kan de groep een professioneel incident rapport opstellen, waardoor de expert gericht kan helpen in het oplossen van het probleem.
Waarborgen van kwaliteit van ontwerp en realisatie	Op 2 momenten wordt het resultaat van de groep voorgesteld aan de externe expert, waarbij gericht naar het resultaat gekeken wordt en inhoudelijke vragen beantwoord worden. Hierbij is het de verantwoordelijkheid van de groep om de feedback te noteren en verbetervoorstellen in te plannen.

3. Vakinhoudelijke beoordeling: SOP6

De student is in staat om een software-ontwikkelstraat in te richten en te gebruiken.

Ontwikkelstraat

De student kan op elk moment een nieuwe release van een applicatie uitrollen op een gecontroleerde manier waarbij de kwaliteit van de source code steeds blijft voldoen aan vooraf gestelde relevante criteria.

Onvoldoende	Voldoende	Goed	Excellent
Er is geen ontwikkelstraat en/of componenten ontbreken	De ontwikkelstraat wordt gebruikt en bevat alle benodigde componenten. Sommige stappen zijn nog niet geautomatiseerd maar de progressie van het proces vindt plaats volgens een gedefinieerd proces	De ontwikkelstraat wordt correct gebruikt, bevat alle benodigde componenten en een release vindt volledig geautomatiseerd plaats.	Voorgaande plus non-triviale eigen uitbreiding

Documentatie en evaluatie softwarewijzigingsbeheer

De student is in staat om een softwarewijzigingsproces op te stellen en te documenteren in de vorm van een procesbeschrijving inclusief werkinstructies.

Onvoldoende	Voldoende	Goed	Excellent
Er is geen procesbeschrijving en/of relevante rollen zijn niet onderkend en/of beslismomenten zijn niet aangegeven en/of de werkinstructies ontbreken	Er is een procesbeschrijving met swimlanes die gebruikt wordt in het wijzigingsproces. De relevante rollen zijn onderkend en de beslismomenten zijn duidelijk aangegeven. De werkinstructies zijn gedocumenteerd.	De procesbeschrijving en werkinstructies worden exact gevolgd tijdens het gebruik van de ontwikkelstraat	Voorgaande plus non-triviale eigen uitbreiding

4. Vakinhoudelijke beoordeling: DPI6

De student kan een applicatie interface standaarden met meerdere stakeholders opstellen, en ervoor zorgen dat de opgeleverde applicatie hieraan voldoet.

Hierbij zorgt hij voor een optimale samenwerking tussen de applicaties door het asynchroon uitwisselen van messages goed (en goed onderhoudbaar) te structureren. Hierbij pas je beargumenteerd "Enterprise Integration Patterns" toe in een technologie naar keuze, b.v. Java Messaging Service (JMS).

Ontwerpen van geïntegreerde systemen

De student kan nieuwe software componenten ontwerpen waarbij er onderzoek gedaan is naar de koppelingsmogelijkheden tussen verschillende applicaties, met verslaglegging van de argumentatie.

Onvoldoende	Voldoende	Goed	Excellent
Er is geen integratie ontwerpbeschrijving en/of relevante integratie scenario's zijn niet onderkend en/of de integratie instructies ontbreken	Er is een integratie ontwerp beschrijving waarmee de student de scenario's en ontwerpbeslissingen kan toelichten.	Er zijn op verschillende manieren rekening gehouden met standaard integratie patronen, waarbij de belangrijkste risico's vermeden worden	Het ontwerp laat toekomstgericht en technisch volledige ontwerpkeuzes zien voor systeem integratie.

Realiseren van geïntegreerde software componenten op een Enterprise niveau

De student kan een werkend, geïntegreerd software systeem demonstreren, en kwaliteitsvolle broncode opleveren met een gewaarborgd niveau van uitvoerings-, uitbreidbaarheid en schaalbaarheid.

Onvoldoende	Voldoende	Goed	Excellent
Er ontbreken koppelingen en/of de belanghebbenden hebben veel last gehad van onstabiele of onbetrouwbare interfaces en/of er kan niet aangetoond worden dat er integratie patronen geïmplementeerd werden adv een werkend systeem en/of ontwerpdocumentatie en/of code niet aantonen dat er integratie patronen geïmplementeerd werden.	Er werden beargumenteerd verschillende integratiepatronen opgenomen in het ontwerp en geïmplementeerd, waarbij de minimale werking van het systeem gewaarborgd wordt.	Het Voldoende niveau is gehaald, waarbij de studenten tevens toekomstgerichte verbetervoorstellen ontworpen hebben op het vlak van integratie patronen en/of een technisch uitdagend integratiepatroon werd uitgewerkt.	De uitwerking bevat verschillende integratie uitwerkingen, waarbij de betrouwbaarheid van de interfaces uitmuntend is.

5. Vakinhoudelijke beoordeling: JEA6

De student kan een Enterprise architectuur met meerdere samenwerkende systemen opzetten, waarbij de proftaakgroep rekening houdt met verschillende stakeholders en (niet-) functionele requirements. Je moet gebruik maken van de JEA specifieke technologie voorkeuren.

Ontwerpen van Enterprise applicaties

De student kan iteratief op voorhand nadenken over de te ontwikkelen omgeving volgens de methodieken die binnen JEA aangeleerd worden.

Zo wordt er gericht gebruik gemaakt van enterprise faciliteiten die een applicatie server aanbiedt en is er voldoende aandacht voor ontkoppeling van front-end/back end, van business logica en database, als tussen applicaties onderling.

Verdere thema's zijn het beargumenteren van ontsluiting aan het internet, beveiliging, betrouwbaarheid, schaalbaarheid, distributie, omgang met grote hoeveelheden data en integratie met andere systemen.

Realiseren van robuuste software componenten op een Enterprise niveau

De student kan verschillende applicaties ontwikkelen die met elkaar in real-time gegevens uitwisselen, waarbij er de gebruiker een responsief systeem heeft. Hierbij kan de proftaakgroep aantonen dat er een gewaarborgd niveau van uitvoerings-, uitbreidbaarheid, schaalbaarheid en robuustheid is; bijvoorbeeld aan de hand van ontwerpdocumentatie en/of broncode en/of demonstraties en/of geautomatiseerde testen.

Er wordt gericht gebruik gemaakt van enterprise faciliteiten die een applicatie server aanbiedt. Communicatie met de database moet gebeuren aan de hand van een Persistence Framework. Zo moet object relation mapping mogelijk zijn, en correct gebruikt.

Onvoldoende	Voldoende	Goed	Excellent
Er is onvoldoende gebruik gemaakt van frameworkondersteunende middelen om loose-coupling, beveiliging en schaalbaarheid te realiseren, waarbij er ofwel fouten in software zijn ontstaan, ofwel de broncode onvoldoende onderhoudbaar is.	Er is op onderbouwde en correcte wijze gebruik gemaakt van verschillende technieken en frameworks in de context van Enterprise Softwaresystemen	Er is voldaan aan de criteria voor beoordeling <i>voldoende</i> en men kan aantonen dat de opgeleverde systemen voldoen aan alle niet-functionele requirements	Er is voldaan aan de criteria voor de beoordeling <i>goed</i> en er zijn aantoonbaar niet triviale zaken zoals uitbreidingen, niet-triviaal onderzoek e.d. aanwezig

6. Vakinhoudelijke beoordeling: OND6

Aan de hand van 8 workshops ga je in kleine groepjes aan de slag met het opzetten en uitvoeren van een onderzoek in het kader van je proftaak. Je vakdocent geeft je bijkomend binnen de proftaak een beoordeling op onderstaand leerdoel.

De student kan binnen een opdracht onderzoek uitvoeren en documenteren, ter ondersteuning van het realiseren van de nodige componenten, waarbij hij uitspraken kan doen gebaseerd op feiten en niet op aannames, rekening houdend met verschillende belanghebbenden en kwaliteitseigenschappen.

Voldoende	Goed	Excellent
De documentatie bevat objectieve argumentatie voor een zelf gekozen onderzoek welke zijn resultaten bijgedragen hebben aan de kwaliteit van de applicatie en/of systeem.	Het uitgevoerde onderzoek is aantoonbaar ordelijk uitgevoerd, waarbij er een beter beeld binnen het project ontstaan is voor de werkzaamheden	Er is voldaan aan de criteria voor de beoordeling <i>goed</i> en er zijn aantoonbaar niet triviale zaken zoals uitbreidingen, niet-triviaal onderzoek e.d. aanwezig

7. Vakinhoudelijke beoordeling: DBO6

De database kan op enterprise niveau hoeveelheden aan gegevens verwerken, waarbij de technieken zoals gezien in DBO gericht toegepast worden. In de uitwerking willen we een gepast gebruik van stored procedures zien, query optimalisatie en materialised views, gebruik makend van bestaande frameworks (tools en technieken).