



SOGETI

M

PTSE6

Enterprise Software Development

Masters of Code

Fontys Hogescholen

Versie informatie

Versie	Datum	Bijzonderheden	Auteur
0.1	12-1-2015	Initiële versie	JvG
1.0	26-1-2015	Bijgewerkt a.h.v. review Sogeti	JvG
1.1	30-1-15	Bijgewerkt a.h.v. review Sogeti	JvG
1.2	3-2-15	Bijgewerkt a.h.v. review FHICT	JvG

Verzendlijst

Titel	
Java Core Team	Sogeti Nederland BV
Frank Coenen	Fontys Hogeschool voor ICT

Copyright Sogeti Nederland B.V. © te Vianen

Niets uit deze uitgave mag vereenvoudigd en/of openbaar worden gemaakt (voor willekeurig welke doeleinden) door middel van druk, fotokopie, microfilm, geluidsband, elektronisch of op welke andere wijze dan ook zonder voorafgaande schriftelijke toestemming van <Company>. Dit rapport is enkel en alleen bedoeld voor intern gebruik voor hierboven genoemd(e) bedrijf/bedrijven.

Inhoudsopgave

1	Inleiding	1
2	Proftaak	2
2.1	Productbeschrijving	2
2.1.1	Spelregels	2
2.1.2	Deelsystemen	3
2.1.3	Algemene richtlijnen	3
3	Organisatie	4
3.1	Studiebelasting	4
3.2	Weekindeling	4
3.3	Werkwijze	4
3.4	Planning	6
4	Beroepsproducten en beoordeling	7
A.	Maven archetype voor opdrachten	9
B.	Workspace management module	11
C.	Service module	12
D.	HTML client voor deelnemers	13
E.	HTML admin console voor wedstrijdleiding	14
F.	HTML client voor publiek	15
G.	JavaFX client voor deelnemers	16

1 Inleiding

Deze studiewijzer beschrijft de essentiële organisatorische en inhoudelijke aspecten van het project PTSE6: Enterprise Software Development van semester 6, S-profiel. Gedurende periode P11 en P12 werk je aan de casus "Masters of Code" uitgeschreven door Sogeti Nederland BV, die door middel van de Scrum-methodiek wordt uitgevoerd.

In meerdere iteraties, wordt een omvangrijk systeem opgebouwd bestaande uit meerdere samenwerkende software systemen. De Java EE technieken die je geleerd hebt in de cursus JEA6 moet je hierbij toepassen in groter verband.

In hoofdstuk 2 worden de doelstellingen, de productbeschrijving en de vereiste voorkennis van het project beschreven. In hoofdstuk 3 de organisatie van het project. De beoordeling en planning van het project worden weergegeven in respectievelijk hoofdstuk 4 en 5. In hoofdstuk 6 worden tenslotte de op te leveren producten beschreven.

2 Proftaak

2.1 Productbeschrijving

[Masters of Java](#) is een roemruchte “funprogging contest” in Java SE, toegankelijk voor iedere Java ontwikkelaar. De wedstrijd wordt jaarlijks georganiseerd door de Nederlandse Java User Group ([NLJUG](#)) in samenwerking met partners zoals [Sogeti](#), en in de nabije toekomst.... jullie! In de wedstrijd worden API kennis en programmeervaardigheid getest en dingen de deelnemers in teamverband naar diverse prijzen.

Het huidige software systeem waarin de deelnemers hun opdrachten maken heet *Masters of Code (MoC)*. Het bestaat momenteel uit een Java SE client met bijbehorende Java SE server. Het systeem is tamelijk beperkend voor bedenkers van opdrachten en kan niet naar een grote aantallen deelnemers schalen. Ook blijkt het lastig het systeem aan nieuwe wensen aan te passen omdat het onvoldoende modulaair is opgebouwd. De wens bestaat daarom het systeem compleet opnieuw te bouwen als modulaire JEE 7 web applicatie die goed horizontaal schaalbaar en makkelijk aanpasbaar is, en makers van opdrachten bovendien meer vrijheid geeft. Je opgeleverde systeem zal als open source project doorontwikkeld worden en voor toekomstige Masters of Java wedstrijden gebruikt worden.

2.1.1 Spelregels

Hoe zit een jaarlijkse wedstrijd in elkaar? Een wedstrijd bestaat uit verschillende rondes. Per ronde wordt eenzelfde opdracht door ieder team afzonderlijk gemaakt. De tijd waarbinnen een team met een correcte oplossing komt bepaald het aantal behaalde punten in die ronde .

Een opdracht voor Masters of Java wordt aangeleverd in één archief zoals een gezippt Maven-project. Deze bevat onder andere:

- Een omschrijving van de opdracht
- Informatie over de auteur en zijn/haar organisatie;
- Eén of meerdere *read-only* broncode-bestanden. Deze worden in de opdracht gebruikt en zijn door de deelnemers alleen te lezen.
- Eén of meerdere bewerkbare broncode-bestanden. Elk team kan hun eigen versie hiervan lezen en schrijven. Elk team start met dezelfde initiële versies van de bewerkbare bestanden van een opdracht. Het bewerken van deze bestanden vormt de kern van de opdracht.
- Van de onderstaande typen geautomatiseerde testscripts tenminste één van de 1^e categorie, maar bij voorkeur ook tenminste één van de 2^e:
 1. *Systeem gestuurd testscript*: dit betreft een testscript wat alleen bij indiening van de opdracht, automatisch wordt aangeroepen voor finale validatie. Indien de test faalt wordt de ingediende opdracht niet geaccepteerd. Dit zonder gedetailleerde terugkoppeling over het falen.
 2. *Deelnemer gestuurd testscript*: dit betreft een testscript wat door de deelnemers tijdens het maken van de opdracht zo vaak als zij willen kan worden aangeroepen. De betreffende deelnemers krijgen feedback over de uitkomst van de test ('pass'/'fail'). Bij het falen van een test wordt ook terugkoppeling gegeven over de oorzaak van het falen. Elke deelnemer gestuurd test van een opdracht is afzonderlijk aanroepbaar door de deelnemers tijdens het maken van die opdracht. Deelnemers kunnen tijdens een opdracht de naam en omschrijving inzien van elke deelnemer gestuurd test.

Het is overigens mogelijk dat een de maker van een opdracht een test zowel als systeem gestuurd als deelnemer gestuurd aanmerkt. De broncode van een testscript mag nooit door de deelnemers in te zien zijn.

- Eén of meerdere hints voor de deelnemers. Hints kunnen na een vooraf ingestelde tijdsduur (gemeten vanaf de start van de ronde) automatisch geopenbaard worden aan alle teams.

De maximale tijd per ronde wordt door de wedstrijdleiding vooraf aan de wedstrijd ingesteld. Het team welke de opdracht van een ronde het eerst volbrengt, verdient hierin de meeste punten. Het aantal hiermee verdiende punten is evenredig aan het aantal resterende seconden bij indiening in die ronde vermenigvuldigt met de moeilijkheidsgraad van de opdracht. De moeilijkheidsgraad zorgt ervoor dat met

lastige opdrachten waarvoor veel tijd nodig is ongeveer evenveel punten verdient kunnen worden als met opdrachten die sneller gemaakt kunnen worden. De wedstrijd wordt uiteindelijk gewonnen door het team met de hoogste totaalscore.

2.1.2 Deelsystemen

Het beoogde product kent in de logische onderdelen weergegeven in Tabel 1.

Bijlage	Naam	Pagina
A	Maven archetype voor opdrachten	9
B	Workspace management module	11
C	Services module	12
D	HTML client voor deelnemers	13
E	HTML admin console voor wedstrijdleiding	14
F	HTML client voor publiek	15
G	JavaFX client voor deelnemers	16

Tabel 1 logische onderdelen systeem

2.1.3 Algemene richtlijnen

De onderhoudbaarheid en uitbreidbaarheid van het systeem dient optimaal gewaarborgd te zijn. Waar mogelijk gebruik je bekende ontwerp patronen en documenteer je dit zowel in code als ontwerpdocumenten. In je ontwerpdocumenten focus je op het 'wat' in plaats van het 'hoe': alleen de globale structuur van het systeem wordt hier beschreven. In je code schrijf je documentatie (javadoc) op package-, class- en public method niveau. Deze moet dusdanig zijn dat een nieuwe programmeur snel begrijpt wat genoemde zaken doen en betekenen.

Afhankelijkheden buiten de JEE 7/JRE 8 standaard bibliotheken dienen geminimaliseerd te worden. Dit heeft verschillende redenen. Ten eerste voorzien we langdurig gebruik en opensource doorontwikkeling van het systeem. Niet alle bibliotheken worden lange tijd onderhouden en ondersteund. Ook gaan overgangen naar nieuwe versies in externe bibliotheken niet altijd even soepel. Dit kan op den duur doorontwikkeling van het systeem bemoeilijken. Ten tweede dienen bibliotheken als open source, gratis en vrijelijk bruikbaar te zijn. Masters of Code wordt dit namelijk zelf ook.

Je mag er van uitgaan dat gebruikers van het systeem één van de volgende browsers heeft in de meest recente productieversie. Momenteel zijn dit:

- Google Chrome v39
- Mozilla FireFox v32
- Microsoft Internet Explorer v11
- Apple Safari v8

Dit betekent dat je welhaast onbeperkt van standaard HTML 5 technieken gebruik kan maken. Het is niet de bedoeling browser-specifieke APIs direct te gebruiken. Binnen de JEE 7/JRE 8 bibliotheken gebruik je liefst de nieuwste technieken wanneer verschillende alternatieven voorhanden zijn.

We voorzien in de toekomst andere clients dan thans gepland, bijvoorbeeld voor smartphones en tablets. In je ontwerp en implementatie dien je hier rekening mee te houden. Zorg ervoor dat de HTML client voor deelnemers voor deelnemers zo 'dom' mogelijk is, en zoveel mogelijk bedrijfslogica en functionaliteit in herbruikbare software componenten en services terecht komt. Tot slot dient het systeem volledig internationaliseerbaar te zijn. Momenteel voorzien we dat het systeem door Engels- en Nederlandstalige deelnemers zal worden gebruikt. De voertaal in code en documentatie is het Engels.

Tijdens wedstrijden dient het Masters of Code systeem in haar geheel zonder internettoegang goed te functioneren. Dit om het opzoeken van informatie op het internet door deelnemers te bemoeilijken.

3 Organisatie

3.1 Studielast

Aan de proftaak zijn 6 EC's toegekend. Met een doorlooptijd van negen weken in periode 11 en vijf weken in periode 12 gaat dit om een inspanning van ruim 6 en 16 uur per week per student. Omdat het om een pittige opdracht gaat, waar veel nieuwe technologieën aan bod komen, zal deze tijd ook echt nodig zijn. Houd hier rekening mee bij de planning van je werk.

3.2 Weekindeling

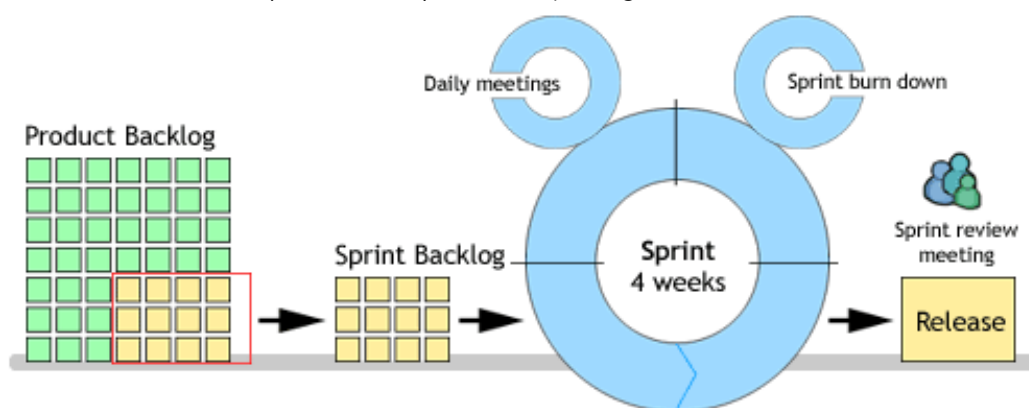
De proftaak wordt in lesweek 1 opgestart tijdens een bijeenkomst met de docentbegeleider van Fontys en Product Owner van Sogeti.

In de eerste periode wordt er een dagdeel gewerkt worden aan de proftaak. Hierdoor resteert voldoende tijd om de benodigde technieken in de cursus JEA6 aan bod te laten komen. In de tweede periode moet er meer tijd besteed worden aan de proftaak, door hieraan twee 'volle' dagen te werken. De afronding van de proftaak is in week 5 van periode 12, door het opleveren van de proftaak aan de docentbegeleider en producteigenaar. Hierna wordt de cursus JEA6 weer opgepakt om nieuwe technieken te bestuderen.

Bij elk blok van ingeroosterde projecturen zullen een deel van de tijd de docentbegeleider en producteigenaar aanwezig zijn. Elke projectgroep maakt hierover afspraken met de betrokkenen. Buiten de ingeroosterde lesuren moet nog een aanzienlijke tijd op school of thuis verder gewerkt worden. 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.

3.3 Werkwijze

Het project moet volgens de Scrum-methodiek worden uitgevoerd. Scrum is een methodiek voor het uitvoeren en managen van Agile software ontwikkel projecten. Bij Scrum wordt het project uitgevoerd in meerdere kort iteraties (2 a 4 weken), ook wel Sprints genoemd.



Figuur 1 SCRUM methodiek

Aan het begin van de sprint wordt er Sprint Backlog samengesteld, waarin alle User Stories voor de huidige Sprint zijn opgenomen opgesplitst in taken door het team. Iedere Sprint wordt afgerond met een demonstratie en aansluitend wordt de volgende Sprint gepland. Meer informatie over Scrum kun je vinden bij het cursusmateriaal van semester 4 waar de Scrum-methodiek uitvoerig is behandeld. Voor de studenten die niet dit onderwijs over Scrum hebben gevolgd, wordt er een gastcollege gegeven over het praktisch toepassen van deze methodiek.

Het projectteam bestaat uit 5 a 6 studenten, waarvan één van de studenten de Scrum Master rol op zich neemt. Voor de uitvoering van het project bewaakt de begeleidende docent het groepsproces en

beoordeeld beroepsproducten. De Product Owner van Sogeti bewaakt de functionaliteit van het systeem en geeft hierover namens de klant zijn visie. Ook kent de Product Owner in samenspraak met het team prioriteiten toe aan items in het backlog. De Product Owner treedt in dit project ook op als technisch adviseur. Waar nodig voor specifieke expertisegebieden zijn vanuit Sogeti ook andere adviseurs beschikbaar. Ook docenten van parallel uitgevoerde vakken kunnen een rol spelen als adviseur.

Wekelijks zal er een projectvergadering gehouden worden met de docentbegeleider en Product Owner. Hierbij wordt de voortgang van het project besproken. Zij zullen ook kijken of het Scrum-proces goed verloopt, en waar nodig bijsturen. Zorg ervoor dat het scrum board altijd bijgewerkt is. Aangezien de tijd beperkt is, is het belangrijk om vooraf een agenda op te stellen, en tijdens de vergadering de notulen op te maken.

Tijdens het "Sprinten" beginnen de studenten de projectdag met een scrum, stand-up meeting. Het doel van deze bijeenkomst is te zorgen dat iedereen zo effectief mogelijk bezig is. Iedereen beantwoordt de volgende 3 vragen:

- Wat heb ik gedaan sinds de vorige meeting?
- Wat ga ik doen tot de volgende meeting?
- Welke issues heb ik en welke hulp heb ik daar bij nodig?

Daarbij moet het Scrum-board natuurlijk bijgewerkt worden. Wanneer de Product Owner hierbij niet fysiek aanwezig kan zijn, zal deze proberen via video conference deel te nemen.

De studenten moet voor elke sprint een Sprint planning bijeenkomst houden met de Product Owner, waarin de volgende Sprint wordt besproken en een demo gegeven wordt van de afgelopen Sprint.

Het projectteam moet gebruik maken van een Agile project management tool VersionOne. De docentbegeleider moet natuurlijk toegang krijgen tot deze tool, om inzicht te krijgen in de voortgang van het project.

Doordat in het project 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 begeleidende docent van deelname worden uitgesloten. Hieraan zal altijd een officiële waarschuwing in de vorm van een gele kaart vooraf gaan

3.4 Planning

Het project moet door de studenten in 2 of meer incrementen worden opgedeeld. Voor het project zijn de onderstaande deliverables door de Product Owner gedefinieerd. De projectgroep maakt zelf een detailplanning om het pad naar deze deliverables in kaart te brengen en te borgen.

Deliverable	Weeknr
Requirement: Backlog, Features, User Stories	3
Software Ontwerp (m.b.v. Visual Paradigm)	
Software Architectuur	
Configuratiebeheer (SVN of GitHub)	1-16
Testen	
Prototype (standalone) systemen A, B en C	11
Prototype (integratie) systemen A, B, en C	
Prototype (standalone) systemen E, F en G	
Prototype van het eindproduct	15

De producteigenaar heeft de onderstaande kwaliteitseisen gedefinieerd:

- Het product moet in Java SE 8 en Java EE 7 geschreven zijn
- Visual Paradigm moet gebruikt worden voor de ontwerpen
- SVN òf GitHub moet gebruikt worden voor configuratiebeheer
- Maak gebruik van een Agile project management tool, VersionOne
- Gebruik een MockUp-tool om de ui-schermen ontwikkelen, Balsamiq is een handig tool.
- De intergratie-technieken : hiervan moeten minstens JMS en Restfull Web Services gebruikt worden. De overige integratietechnieken zijn optioneel.
- Alle deliverables behalve binaire code staan in het versiebeheersysteem

Op de site van [RUPopMaat](#) staan een aantal handige templates, voor o.a. de software architectuur.

In het onderstaande schema wordt een voorbeeld planning van de eerste weken van de proftaak gegeven.

Wk	Activiteit	Product
1	Gastcollege scrum, Groep formatie, Lezen opdracht,	
2	User stories en Backlog op basis van casus beschrijven	Product backlog
3	Sprint PlanningX	
4	SprintenX	
5	Sprinten	
6	Demo + Sprintplanning + Sprinten	Broncode, Ontwerp, testrapportage

4 Beroepsproducten en beoordeling

Het project is onderverdeeld in een aantal producten. Elk product wordt door de docent-begeleider beoordeeld met een cijfer van tenminste 10 en ten hoogste 100. Eén van de beoordelingen wordt aangevuld met kwalitatieve feedback. Dit oordeel wordt gegeven door de begeleidende docent op basis van kwantiteit en kwaliteit van de bijdrage van elk individueel project lid. Mocht de bijdrage van de student aan het product positief of negatief afwijken ten opzichten van de groepsgenoten, dan kan de beoordeling van de student naar boven of beneden bijgesteld worden.

Hieronder volgt een overzicht van de producten van de proftaak en wat de criteria zijn bij de beoordeling hiervan.

1) Enterprise Applicatie Integratie (EAI)

Bij dit beroepsproduct gaat het om de uiteindelijk opgeleverde programma: de demonstratie en de kwaliteit van de broncode. Gekeken gaat worden naar de omvang, complexiteit en kwaliteit van het eindproduct.

De beoordelingscriteria voor de omvang van het eindproduct wordt in Tabel 2aangegeven. Hierbij wordt gebruik gemaakt van de [MoSCoW methode](#). Indien aan alle Must Have's voldaan is, levert dit een cijfermatige voldoende op voor de omvang.

	Must	Should	Could
A Maven archetype voor opdrachten	X		
B Workspace management module	X		
C Services module	X		
D HTML client voor deelnemers	X		
E HTML admin console voor wedstrijdleiding		X	
F HTML client voor publiek		X	
G JavaFX client voor deelnemers			X

Tabel 2 vereisten P12

Het criterium voor de kwaliteit is dat er een er een werkend systeem wordt opgeleverd met leesbare en gedocumenteerde code in overeenstemming met de opgestelde specificaties en het opgestelde ontwerp.

2) Enterprise Applicatie (EA)

Bij dit beroepsproduct gaat het om de tussenstand in periode 11 zoals aangegeven in Tabel 3. Gekeken gaat worden naar de omvang, complexiteit en kwaliteit van het eindproduct.

	Must	Should	Could
A Maven archetype voor opdrachten		X	
B Workspace management module	X		
C Services module	X		
D HTML client voor deelnemers		X	
E HTML admin console voor wedstrijdleiding			X
F HTML client voor publiek			X
G JavaFX client voor deelnemers			X

Tabel 3 vereisten P11

3) Productdocumentatie(PD)

Het product Productdocumentatie beschrijft het eindproduct. Het gaat hier tenminste om de deelp producten:

- a) Requirements: de functionele en niet functionele eisen van het systeem worden in de scrum-methode beschreven in Backlogs, Features en User Stories.
- b) Ontwerp: het ontwerp beschrijft hoe het systeem gemaakt moet worden, hierbij gaat het om het domeinmodel, software ontwerp, database ontwerp en de ontwerpen van de ui-schermen.
- c) Architectuur: moet een overzicht geven van de software architectuur van het te bouwen systeem, waarbij diverse aspecten van het systeem worden belicht. Het 4+1 model van Kruchten zou je kunnen gebruiken voor het beschrijven van de architectuur.
- d) Testen: beschrijving van de relevante testscenario's voor de deelsystemen en hun onderlinge samenhang, deze kunnen onderdeel zijn van de User Story beschrijvingen in Scrum of in een apart document beschreven worden.

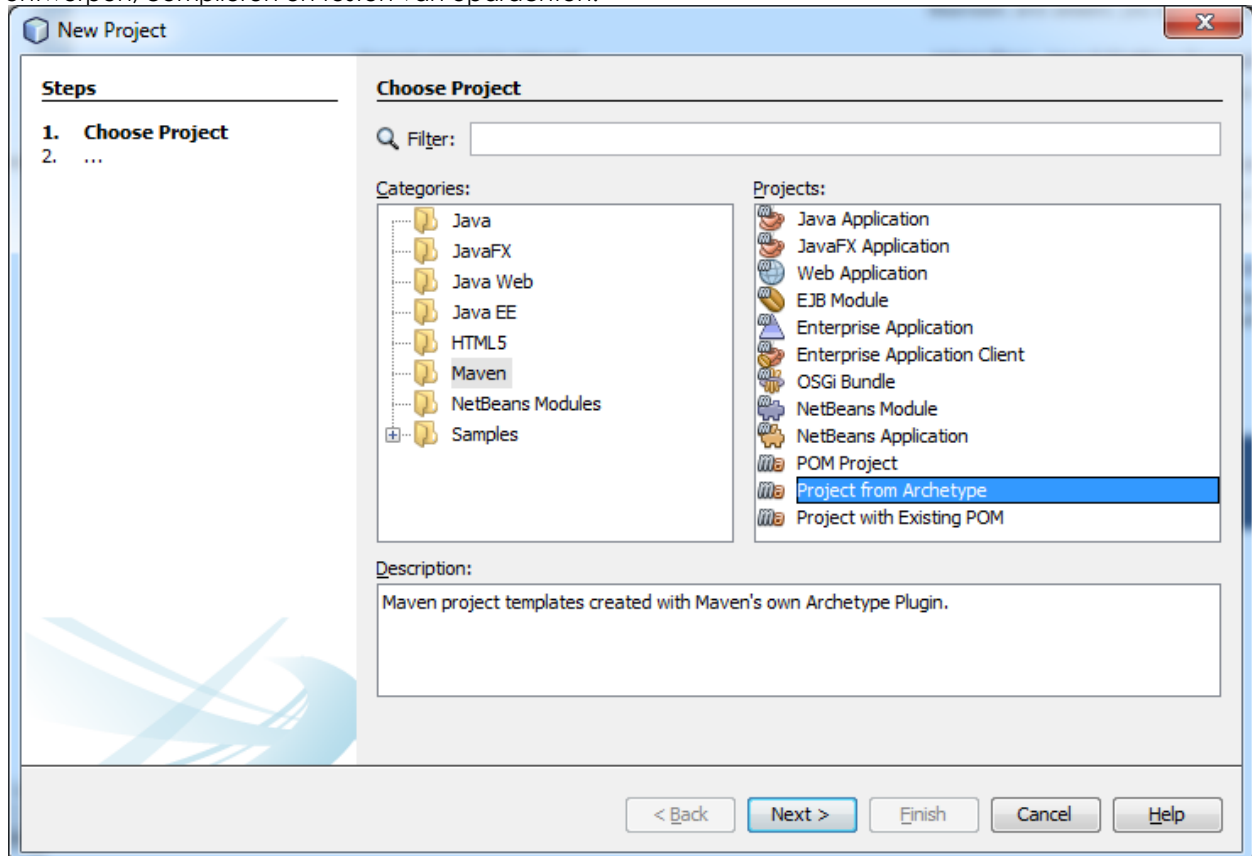
Er moet een bruikbaar ontwerp en architectuur gemaakt worden die in overeenstemming met de instructies van de cursus GSO in het 3 de semester.

Op je cijferlijst komt slechts één cijfer voor dit vak: een gewogen gemiddelde van de drie producten: $0.2 \cdot EA + 0.6 \cdot EI + 0.2 \cdot PD$.

Elk team zal zes, met elkaar communicerende, subsystemen gaan realiseren. Voor elk subsysteem is een team van twee studenten verantwoordelijk. Daarnaast krijgt ieder hierbij individuele taken.

A. Maven archetype voor opdrachten

Een Maven archetype is een blauwdruk voor een type project, zoals 'Masters of Java opdracht'. Archetypen worden veel gebruikt in Java-ontwikkeling. Als je bijvoorbeeld een nieuw Maven Web project genereert in je IDE, wordt onder de motorkap een archetype gebruikt om de projectstructuur te genereren. Ook kun je een specifiek archetype opgeven om een project op te baseren, zie Figuur 2. Aangezien Maven nagenoeg onbeperkte flexibiliteit kent in het definiëren van dependencies, testen en het 'packagen' van artefacten ligt het voor de hand een Maven archetype te ontwikkelen voor het ontwerpen, compileren en testen van opdrachten.



Figuur 2 Kiezen van een archetype als basis voor een project in Netbeans

De functie van het Maven-archetype voor opdrachten is het genereren van een eenduidige en gestandaardiseerde projectstructuur voor opdrachten. Een gegenereerd project moet met een standaard IDE die Maven ondersteunt met minimale configuratie te maken zijn door de bedenker van een opdracht.

Het archetype dient een maven-project structuur voor de opdrachtontwikkelaar te genereren waarin de ontwikkelaar een opdracht kan definiëren. Ook dient het maven-project een archief te kunnen genereren van zo'n project wat na uitpakken op het bestandssysteem te compileren en/of testen is gegeven dat geen van eerdergenoemde read-only of bewerkbare broncode bestanden fouten bevatten. De maker van een opdracht dient de volgende attributen op uniforme wijze te kunnen instellen. Deze gegevens moeten bij elk project op dezelfde manier eenvoudig automatisch uitgelezen kunnen worden:

- Over de bedenker van de opdracht:
 - Naam;
 - Organisatie;
 - web link;
 - URI binnen het project naar een logo van de organisatie;
- Over de opdracht:
 - Naam;
 - Omschrijving voor de deelnemers (HTML5);
 - Omschrijving voor het publiek (HTML5);

- Eén of meerdere read-only broncode bestanden;
- Eén of meerdere bewerkbare broncode bestanden;
- Eén of meerdere hints;
- Per deelnemer gestuurde test: een naam en omschrijving van wat er getest wordt, alsook het maximaal aantal seconden die een test in beslag mag nemen. Dit laatste is bedoeld om tests die vanwege programmeerfouten van deelnemers tot oneindige loops leiden af te breken.
- Zowel deelnemer- als systeem gestuurde test-classes moeten eenvoudig machinaal geïdentificeerd kunnen worden.

Voorts dient een opdracht-project een artefact op te leveren wat uitgepakt in folder precies de Maven project structuur geeft van het opdracht project. Informatie over het maken van een Maven archetype vind je op het web, bijvoorbeeld [hier](#) en [hier](#).

B. Workspace management module

Een 'workspace' in de IDE-wereld is een folder van Java projecten. In een wedstrijd heeft elk team haar eigen workspace nodig met daarin de opdracht-projecten van de diverse opdrachten. De hoofdfunctie van de workspace management module is het beheer van deze workspaces. De module dient de volgende functionaliteit via haar API beschikbaar te stellen:

- Het aanmaken of verwijderen van een workspace voor een team in een wedstrijd;
- Het inlezen van een opdracht-artefact en het beschikbaar stellen van de meta-data van een opdracht via een API.
- Het bijwerken van de bewerkbare broncode bestanden van een opdracht;
- Het compileren van een opdracht-project met terugkoppeling omtrent eventuele fouten in de bewerkbare broncode bestanden. Compilatie van een opdracht-project mag niet de compilatie of testuitvoering van een opdracht-project van een andere workspace hinderen.
- Het uitvoeren van één of meerdere testen in de opdracht met terugkoppeling omtrent eventuele testfouten voor gebruiker-gestuurde tests. Uitvoeren van één of meerdere testen binnen een workspace mag de compilatie of testuitvoering van een andere workspace niet hinderen/beïnvloeden.

Deze module zal zowel in een JEE container als in een Java SE client bruikbaar moeten zijn. Daarom kunnen binnen deze module geen technieken worden gebruikt die van een container afhankelijk zijn zoals dependency injection. Alleen de eigenaars van de workspace (het team), het Masters of Code systeem en de wedstrijdleiding hebben toegang tot die workspace.

C. Services module

De services module wordt het centraal zenuwstelsel van Masters of Code en houdt immer de regie over wedstrijden, ronden, winnaars en verliezers, beveiliging, en alle overige bedrijfslogica. Het starten, stoppen en pauzeren van wedstrijden en ronden, afgeven van hints en het bijhouden van de toestanden van wedstrijden, ronden en opdrachten waaronder puntentelling hoort tot de verantwoordelijkheden van deze module. Ook het bijwerken van bewerkbare broncode bestanden, compileren, testen en indienen wordt vanuit deze module aangestuurd. Hierbij worden andere module(s) dusdanig gebruikt dat het systeem horizontaal (via meerdere servers) naar honderden teams kan schalen met standaard JEE technieken. Ook stelt de module een API beschikbaar waarmee componenten zich betrouwbaar kunnen abonneren op gebeurtenissen in een wedstrijd of ronde.

Tot slot biedt de module een API voor het beheer van gebruikers, wachtwoorden, rollen, teams en wedstrijden in de Masters of Code database en een JEE serverkoppeling die authenticatie en autorisatie met standaard containermechanismen mogelijk maakt. Ongeautoriseerde toegang tot een team- of gebruikersprofiel, wedstrijd, ronde of opdracht wordt ook binnen deze module uitgesloten.

D. HTML client voor deelnemers

Deze module vormt het gezicht van Masters of Code naar de deelnemers toe. Na inloggen krijgt een team voorafgaand aan de wedstrijd globale informatie hierover te zien zoals het aantal ronden, de totale verwachte duur van de wedstrijd en het aantal seconden tot start van de wedstrijd.

Zodra een ronde start krijgt het deelnemende team alle details hiervan te zien. Het team kan ondermeer de bewerkbare broncode bestanden wijzigen, compileren en testen, de read-only broncode bestanden inzien, de opdrachtschrijving lezen en de JavaDocs inzien van de onderhavige Java Platform Standard Edition. Teams kunnen in een soort van newsfeed zien welke acties andere teams ondernemen. Berichten over andere teams mogen echter geen details over de bewerkbare broncode bestanden geven! Andere informatie die voor het team altijd beschikbaar moet zijn is de resterende tijd in een ronde, de score die behaald zou worden indien een opdracht 'nu' zonder fouten wordt ingediend, en de scores van de andere teams.

Wanneer een hint wordt geopenbaard, wordt deze op het scherm van de deelnemers getoond. Eenmaal binnen een ronde geopenbaarde hints dienen binnen die ronde altijd inzichtelijk te zijn. Het mechanisme waarmee server hints asynchroon naar de HTML client worden verstuurd heeft een generiek ontwerp zodat het ook voor andere berichttypen kan worden hergebruikt. Vanwege het belang van hints voor de deelnemende teams moet de aflevering ervan aan de HTML client uiteindelijk altijd gebeuren, ook na kortstondige netwerkstoringen! Na een netwerkstoring moet de HTML client (eventueel na kleine gebruikersacties) weer de toestand kunnen tonen juist vóór de storing. Zodra de wedstrijd klaar is toont de client de scores van de teams van hoog naar laag op het scherm.

Deelnemers moeten zich individueel kunnen registreren bij het systeem. Het systeem dient hierbij het opgegeven e-mailadres te verifiëren met een activatie e-mail. Na deze stap kunnen gebruikers zelf over hun profiel en wachtwoord beschikken.

Tot slot dient de HTML client voor deelnemers nog een functie te bieden voor het registreren van een team voor deelname aan een wedstrijd. Een team bestaat dus maar voor één wedstrijd en wordt initieel gevormd door één persoon: de initiator. Ook mag een persoon aan een wedstrijd binnen ten hoogste één team deelnemen. De initiator kan andere personen uitnodigen door hun naam en e-mail op te geven aan het systeem, waarop het systeem een email stuurt aan die persoon met hierin een link om de uitnodiging te bevestigen dan wel af te slaan. Als die persoon op de link klikt terwijl zijn of haar e-mail adres nog onbekend is bij Masters of Code, wordt de persoon eerst doorverwezen naar de registratiepagina. Eenmaal lid van een team, mag alleen het lid zelf, de initiator van het team of de wedstrijdleiding het genoemde lid verwijderen uit het team. Deze registratiefunctie is *tijdens* wedstrijden niet beschikbaar omdat het Masters of Code systeem dan mogelijk niet verbonden is met internet. Uiteraard mag men zich alleen registreren voor toekomstige wedstrijden. Ook de wedstrijdleiding moet voorafgaand aan een wedstrijd deelnemers en teams kunnen registreren en teams kunnen samenstellen of wijzigingen zonder dat daar internet bij vereist is voor e-mail verificatie.

Een mechanisme wat configureerbare audio-profielen van muziek en geluidseffecten tijdens of juist voor of na de wedstrijd biedt zou een leuk extraatje zijn.

E. HTML admin console voor wedstrijdleiding

Deze component maakt het de wedstrijdleiding mogelijk het systeem te configureren en monitoren.

Het configureren van het systeem bestaat uit:

- Algemene informatie over een wedstrijd: deze wordt getoond in de (start-)schermen van de HTML clients voor deelnemers en publiek.
- Het registreren van deelnemers, teams en het kunnen samenstellen van teams.
- Het selecteren van de teams die zich hebben opgegeven voor deelname aan een wedstrijd;
- Het selecteren en ordenen van opdrachten binnen een wedstrijd;
- Het instellen van de moeilijkheidsgraad van een opdracht en overschrijven van de meta-data van een opdracht;
- Het configureren van de tijdsduur van een ronde, en de momenten binnen een ronde waarop hints worden afgegeven;
- Het afgeven van een hint tijdens een ronde door de wedstrijdleiding in plaats van geplande openbaarmaking van die hint via een eerder ingestelde timer;

Hiernaast dient via de admin console de wedstrijd gestart of gestopt te kunnen worden. Wordt een wedstrijd vroegtijdig gestopt, dan gelden de tot dan toe door teams behaalde punten als eindscore. Ook individuele ronden kunnen worden gestart, gestopt, gepauzeerd of hervat (na pauzeren) via de admin console. Wordt een ronde vroegtijdig gestopt, dan gelden de tot dan toe behaalde punten in die ronde als score voor die ronde. Bij het pauzeren van een ronde wordt de klok die de resterende tijd voor die ronde bijhoudt stilgezet tot hervatting van die ronde. Deze laatste functie is bijvoorbeeld nodig als tijdens de wedstrijd meer tijd nodig is dan voorzien. Bijvoorbeeld omdat een opdracht moeilijk blijkt. De HTML client voor deelnemers blijft tijdens het pauzeren van de wedstrijd bruikbaar. Ook moet het mogelijk zijn binnen een ronde de resterende hoeveelheid tijd met een opgegeven hoeveelheid te vermeerderen. Als laatste moet een ronde bevroren kunnen worden. Dit heeft hetzelfde effect als het pauzeren van een ronde, met dien verstande dat de HTML client voor deelnemers geblokkeerd is totdat de ronde hervat wordt.

Het compileren en testen van opdrachten kan een aanzienlijke systeemplas veroorzaken die bij grote aantallen deelnemers over meerdere machines gedistribueerd moet kunnen worden. Om deze last inzichtelijk te maken dienen zaken zoals de gemiddelde en piek-belasting van een server inzichtelijk gemaakt te kunnen worden. De console moet goed beveiligd worden tegen niet geautoriseerde toegang: alleen de wedstrijdleiding mag de console openen en achterliggende functies aanroepen.

F. HTML client voor publiek

Evenals de HTML client voor deelnemers kent de HTML client voor publiek drie hoofdtoestanden: die van voor, tijdens en na een wedstrijd.

De HTML client voor publiek heeft als doel 'het publiek' van de ontwikkelingen tijdens en juist voor of na de wedstrijd op de hoogte te houden. Belangrijke informatie omvat:

- Informatie over teams en hun leden;
- De naam van de wedstrijd;
- Het (resterende) tijd per ronde;
- Actuele scores;
- Een 'newsfeed' van acties van individuele teams. Deze feed mag echter geen informatie bevatten die een deelnemend team zou kunnen helpen bij de opdracht. Alleen details over opdrachten uit voorafgaande ronden zijn inzichtelijk voor het publiek.

G. JavaFX client voor deelnemers

De JavaFX client voor deelnemers is een server-resource besparend alternatief voor de HTML client voor deelnemers. Door compilatie en deelnemer gestuurde tests van opdrachten lokaal op de machine van de deelnemer te doen, kunnen de servers worden ontlast. Hiervoor zou de Workspace management module herbruikt moeten worden binnen. Ook de HTML client voor deelnemers kan eenvoudig in JavaFX [herbruikt kunnen worden](#). Door slim gebruik te maken van bestaande modules en [communicatie tussen JavaFX en JavaScript](#) zou de implementatie van de JavaFX client voor deelnemers met minimale inspanning mogelijk moeten zijn.