Hertentamen Software Engineering

Woensdag 25-8 9-12 uur.

Dit tentamen gaat uit van een doorlopend voorbeeld, beschreven op aparte bladzijden. Lees dat voorbeeld helemaal door. Lees ook alle vragen en onderdelen door. Er is echter geringe afhankelijkheid tussen vragen en onderdelen. Probeer daarom alle onderdelen te beantwoorden, ook als je vorige onderdelen niet kon. Geef antwoorden volledig, maar bondig.
LET OP: Eerst Centrale Voorbeeld op bladzijden 3 en 4 doornemen!!
Het eindcijfer is het behaalde aantal punten +1.

1 Hergebruik (3 pt.)

a) Waarom is het toepassen van software hergebruik (waarvan objectgeörienteerd hergebruik een voorbeeld van is) een voorwaarde voor het samenwerken van bedrijven –zoals BIT en ECCOO dat doen– bij het produceren van Software? Tip: Vergelijk samenwerking d.m.v. software hergebruik met samenwerken waarbij je samen aan een software module werkt. (0,4 pt.)
b) Leg uit welke drie(!) fundamentele mechanismen in objectgeörienteerd programmeren met name uitnodigen tot hergebruik. Welke daarvan zijn het meest geschikt voor toepassing in een dergelijke samenwerking tussen bedrijven en waarom? (0,8 pt.)
c) Noem op z'n minst 4 voordelen van hergebruik, en hoe die in dit voorbeeld met name kunnen worden ingevuld. (0,8 pt.)
d) Leg kort uit wat een Pattern is, en hoe een pattern in het framework van BIT zou kunnen worden toegepast. Geef een voorbeeld van een mogelijke toepassing van een pattern in het framework, geef ook waarom je het toepast. (0,6 pt.)
e) In hoeverre zou voor het voorbeeld Generatorgebaseerd hergebruik mogelijk zijn? Hoe stel je je dat voor? (0,4 pt.)

2 Requirement management & Processen (3 pt.)

a) Figuur 2 geeft grofweg aan hoe verandering processen lopen in het voorbeeld. Beschrijf hoe elk van de organisaties (BIT en ECCOO) met change requests om moeten gaan om er voor te zorgen dat change requests de huidige implementatie processen niet doorkruisen. Geef evt. een figuur. (0,4 pt.)
b) Wat gebeurt er volgens Sommerville allemaal met een binnenkomend verzoek voor een nieuwe of te veranderen requirement? Welke stappen worden daarbij genomen? (0,6 pt.)
c) Hoe passen die stappen in je antwoord van vraag 2a? Leg eventueel (niet verplicht) uit aan de hand van een voorbeeld. (0,4 pt.)
d) Requirements management is het proces waarbij veranderende requirements gemanaged worden. Geef aan welke hulpmiddelen en informatie daarbij gebruikt wordt, en hoe die ingezet kunnen worden om b.v. de impact van een change request in te schatten. (0,6 pt.)
e) Sommerville beschrijft in hoofdstuk 3 een aantal fundamentele proces modellen of proces paradigma's. Waar lijkt het proces van Figuur 2 het meest op? (0,2 pt.)
f) Wat is een belangrijk probleem dat in dit proces model en ook in het voorbeeld op kan treden? Geef ook aan waarom dat probleem hier op kan treden. (0,4 pt.)
g) Geef aan hoe je kan voorkomen dat het onder 2f genoemde probleem optreedt. (0,4 pt.)

3 Configuratie Management (1,6 pt.)

a) ECOO gebruikt het framework van BIT om applicaties te ontwikkelen. Leg uit hoe ECCOO in de problemen kan komen als BIT met een upgrade van het framework komt. (0,4 pt.)
b) Beschrijf wat configuratie management is. Doe dat door in ieder geval 3 belangrijke termen uit de wereld van configuratie management te noemen en kort te beschrijven. Je mag de engelse termen gebruiken. (0,6 pt.)
c) Leg uit hoe als BIT en ECCOO er een *gezamenlijke* configuratie management op na houden het probleem van ECCOO bij upgrades kan worden verminderd. (0,2 pt.)
d) In hoeverre is het nuttig en nodig, dat bij een dergelijke configuratie management bij upgrades ook echt (tekstuele) code van BIT naar ECCOO vloeit? Geef een toelichting. (0,2 pt.)
e) Indien geen code van BIT naar ECCOO vloeit, hoe kan dan toch de configuratie management van ECCOO in overeenstemming blijven met die van BIT? (0.2 pt.)

4 Testen (1,4 pt.)

a) Leg uit wat drivers en stubs zijn en welke je wanneer kunt gebruiken bij het testen. (0,4 pt.)
b) Leg uit voor zowel het BIT als voor het ECCOO of Stubs en Drivers van nut voor ze zijn bij het testen van software. Zo ja, hoe dan. Tip: Bedenk en schrijf op wat ECCOO zou moeten testen. (0,6 pt.)
c) Leg uit waarom bij objectgeoriënteerde systemen integratie testen moeilijker is dan bij procedurele talen. Geef tenminste 2 redenen. (0,4 pt.)

Centrale voorbeeld.

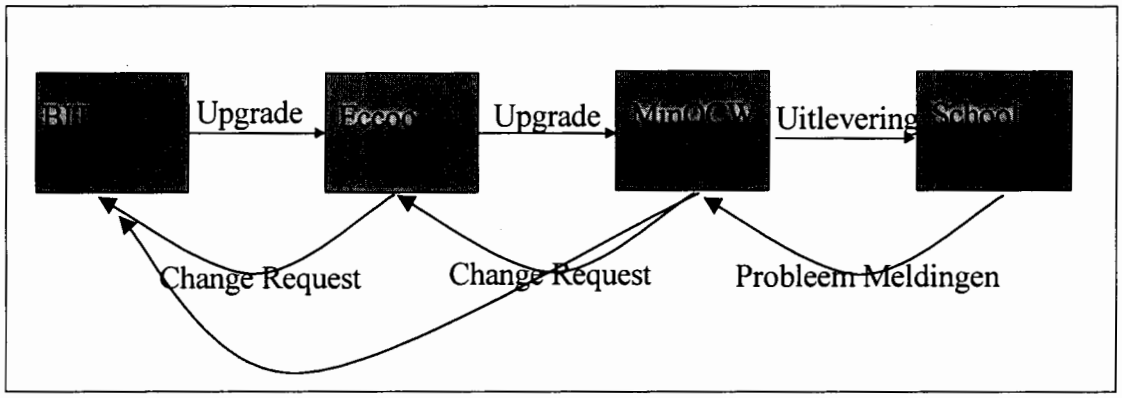
Dit voorbeeld betreft elektronische formulier applicaties zoals die door scholen gebruikt kunnen worden voor het tellen van hun leerlingen en voor het (digitaal) doorgeven van die tellingen naar het ministerie van Onderwijs Cultuur en Wetenschap (afgekort MinOCW). Op basis van die tellingen krijgen scholen subsidie. Figuur 1 toont een voorbeeld de user interface van één(!) vraag in zo'n elektronische formulier applicatie. Een vraag ziet er uit als een matrix met verschillende dimensies (b.v. achterstand niveau, sekse) en per dimensie een aantal eenheden b.v. J (=jongen) M (=meisje). Per veld wordt het aantal leerlingen ingevuld. Bij het invullen van individuele velden van de matrix worden er checks gedaan, b.v. het aantal 4 jarigen met achterstandscategorie 1.00 kan niet groter zijn dan het totaal aantal 4 jarigen op een school. Ook bij het springen van vraag naar vraag kunnen er checks worden uitgevoerd, b.v. dat het totaal aantal leerlingen in de tabel niet het vooraf opgegeven totaal aantal leerlingen van de school overstijgt. De gegevens uit een matrix worden opgeslagen in een database tabel.

1 Aantal leerlingen naar leeftijd, achterstandscategorie en geslacht

	Achterstansdkategorie/geslacht												Totaal
	1.00		1.25		1.50		1.40		1.70		1.90		
Leeftijd	J	M	J	M	J	M	J	M	J	M	J	M	
4 jarigen													
5 jarigen													
6 jarigen													
7 jarigen													
8 jarigen													
9 jarigen													
10 jarigen													
11 jarigen													
12 jarigen													
13 jarigen													
14 en ouder													
Totaal													

Figuur 1 Voorbeeld van de visualisatie van één vraag in een elektronisch formulier

Per jaar worden er tenminste twee van die applicaties gebouwd voor twee verschillende schoolsoorten. En van jaar tot jaar vertonen applicaties voor zelfde schoolsoorten flinke veranderingen. Dergelijke applicaties verschillen van elkaar: Het aantal vragen, precieze inhoud van de vragen, per vraag b.v. hoeveel dimensies in de tabel, welke eenheden per dimensie, maar ook: de checks die worden uitgevoerd per ingevuld veld en de checks die uitgevoerd worden bij het springen van vraag naar vraag.



Figuur 2 Structuur en Verandering processen

In dit voorbeeld werken er vier organisaties samen, zie Figuur 2. BIT (Bosch Information Technologies), ECCOO (Expertisecentrum Computer Ondersteund Onderwijs), het ministerie (MinOCW) en een (niet met name genoemde) school. MinOCW is de opdrachtgever van zowel BIT als ECCOO. BIT is de leverancier van een objectgeoriënteerd framework voor het bouwen van digitale formulieren, met name ten behoeve van het tellen van leerlingen. ECCOO is de leverancier van specifieke applicaties gebouwd met behulp dit framework, iedere applicatie is één elektronisch formulier met diverse vragen. Applicaties worden in eerste instantie geleverd aan het ministerie, die ze zelf met behulp van uitgebreide acceptatie tests test en weer uitlevert aan scholen. De school is de echte gebruiker van de applicatie.