Tentamen Software Engineering

3 mei 2000

In dit tentamen staat de case "FlexPay" centraal. Vragen over modelleertechnieken hebben betrekking op deze case.

Een grote onderneming in het zuiden des lands wil de productiviteit van haar werknemers verhogen door invoering van het principe loon-naar-werken. Hoewel de ondernemingsraad en de vakbonden hiermee nog akkoord moeten gaan, wil de directie alvast van start gaan met de ontwikkeling van de hiervoor benodigde automatisering. Het idee is om de administratie bij te houden met een geautomatiseerd systeem, genaamd FlexPay. Een gesprek tussen de directie en de automatiseringsafdeling van de onderneming resulteerde in de onderstaande probleemstelling.

> Iedere werknemer maakt jaarlijks met zijn directe leidinggevende afspraken over de prestaties (targets) die hij of zij het komende jaar gaat leveren. Het salaris dat de werknemer gaat verdienen is voor een gedeelte afhankelijk van het al dan niet behalen van deze targets. Het salaris is derhalve opgebouwd uit een vast en een flexibel gedeelte. De verhouding vast-flexibel wordt per individuele werknemer bepaald, omdat de prestaties van werknemers in sommige gevallen afhankelijk zijn van externe factoren die buiten hun controle vallen.
>
> Omdat de af te spreken prestaties nog niet vastliggen, gaat men vooralsnog uit van de volgende twee varianten:
> - het halen van milestones in een project;
> - het halen van sales targets.
>
> In de toekomst zullen evenwel nog meer soorten targets worden toegevoegd. Het halen van een target is overigens niet binair (wel/niet): de leidinggevende bepaalt de mate waarin één of meerdere targets zijn behaald.
>
> Het FlexPay systeem moet naast het bestaande financiële systeem gaan functioneren. De koppeling met dit systeem wordt in eerste instantie zo eenvoudig mogelijk gehouden. Na ieder functioneringsgesprek wordt het nieuwe salaris bepaald en ingevoerd in het FlexPay systeem. FlexPay stuurt vervolgens de juiste gegevens (personeelnummer, salaris, ingangsdatum) naar het financiële systeem. In de toekomst wil men werknemers bovendien de mogelijkheid bieden om een gedeelte van de behaalde targets om te zetten in aandelenopties.
>
> Om achteraf problemen te voorkomen worden alle gemaakte afspraken in het FlexPay systeem vastgelegd. Dit zijn privacy-gevoelige gegevens. Het FlexPay systeem moet dan ook aan hoge veiligheidseisen voldoen.

Merk op dat deze probleemstelling een aantal onduidelijkheden en ambiguïteiten bevat. Bij het beantwoorden van de vragen ben je vrij om onvolledigheden zelf in/aan te vullen. Vergeet niet gemaakte aannames kort toe te lichten.

Vraag 1: Requirements Engineering

a) Wat wordt verstaan onder de begrippen "universe of discourse (UoD)" en "conceptueel model van UoD"?
b) Identificeer de gebruikersgroepen in de bovenstaande FlexPay-case.
c) Schrijf de requirements op in telegramstijl en deel ze naar eigen inzicht in volgens de MOSCOW (Must, Should, Could and Won't Have) methode.
d) Werk twee "Must Haves" uit d.m.v. use cases (in natuurlijke taal) ".

z.o.z.

Vraag 2: Analyse

a) Binnen de automatiseringsafdeling die het FlexPay systeem moet gaan ontwikkelen is een verschil van mening ontstaan over de te gebruiken modelleertechnieken. De ene groep is een groot voorstander van object-oriented (OO) technieken, terwijl de andere groep zweert bij structured-analysis technieken. Men komt er niet uit en besluit een deskundige in te huren. Dat ben jij. Maak een afweging en licht die in maximaal 250 woorden toe.
b) Analyseer de FlexPay-case m.b.v. een class diagram (met gebruikmaking van de specialisatie/generalisatie (is-a), aggregatie (part-of) en/of associatie (is-related-with) relatie). Het gaat in deze vraag om een conceptueel model met de nadruk op concepten en hun onderlinge relaties. Class attributen en methoden hoeven niet in het diagram te worden opgenomen. Verklaar de gebruikte notatie.
c) Analyseer dezelfde case met een data flow diagram (DFD). Er wordt hier een uitwerking verwacht die bestaat uit tenminste twee niveaus: een level 1 context diagram en een level 2+ DFD met een aantal data stores en processen. Verklaar de gebruikte notatie.

Vraag 3: Software Architectuur

a) Wat maakt software architectuur anders dan high-level design?
b) Wat wordt verstaan onder een architectural/design pattern? Waar dienen ze voor?

Vraag 4: Design

a) Twee belangrijke design criteria zijn: cohesion and coupling.
   a1. Wat precies wordt met deze twee begrippen bedoeld?
   a2. Waarom zijn ze belangrijk?
   a3. Geef tenminste 3 vormen van cohesion en 3 vormen van coupling.
b) Laat m.b.v. een sequence diagram (SD) zien hoe in de FlexPay-case de resultaten van een functioneringsgesprek (dus de mate waarin de targets zijn gehaald) in het systeem worden ingevoerd en doorwerken naar het financiële systeem. Verklaar de gebruikte notatie.

Vraag 5: Testen

a) Wat wordt verstaan onder black box (functional) en white box (structural) testing?
b) Beschrijf tenminste 2 black box testtechnieken.

Puntentelling

Met dit tentamen zijn 90 punten te verdienen. Het eindcijfer wordt als volgt bepaald:

Eindcijfer = (#punten + 10) / 10	zonder huiswerkopdracht bonus
Eindcijfer = min((#punten + 15) / 10 , 10)	met huiswerkopdracht bonus

Het gewicht van de vragen is als volgt verdeeld:

1a: 4 b: 3 c: 3 d: 10
2a: 10 b: 10 c: 10
3a: 5 b: 5
4a: 6 (a1: 2, a2: 2, a3: 2) b:14
5a: 4 b: 6