

Tentamen Software Engineering

10 januari 2000

De case: "de automatisering van video-winkelketen Look Sharp (LS)" staat centraal in dit tentamen. Vragen over modelleertechnieken hebben betrekking op deze case.

Een video-verhuurbedrijf kan niet goed functioneren zonder een geoliede administratie. De directie van Look Sharp ziet automatisering van de belangrijke bedrijfsprocessen als een absolute noodzaak. Uit een interview met de directie kwam de onderstaande probleemstelling naar voren.

Look Sharp (LS) verhuurt en verkoopt video's aan klanten via één van haar vele winkels. Een klant moet lid zijn van LS om een video te kunnen huren. Deze eis geldt niet voor het kopen van een video.

Leden kunnen een video reserveren als een exemplaar tijdelijk niet beschikbaar is. Als een gereserveerd exemplaar wordt teruggebracht, dan wordt het bewuste lid ingelicht, en het exemplaar blijft dan voor een duur van drie dagen voor het lid beschikbaar. Na die drie dagen vervalt de claim.

In een winkel is een beperkte hoeveelheid video's beschikbaar voor de verkoop. Een klant kan echter een order plaatsen die dan wordt doorgestuurd naar het hoofdkantoor. Het hoofdkantoor beheert de catalogus van video's en stelt de prijs van een verkoopvideo vast.

Iedere winkel verhuurt een deelverzameling van de bovengenoemde catalogus. In tegenstelling tot een verkoopvideo, wordt de prijs van een verhuurvideo bepaald door de winkel zelf, dit om de lokale concurrentie het hoofd te kunnen bieden.

In de administratie van een winkel wordt de lokale catalogus bijgehouden, waarin o.a. verhuur- en reserveringsgegevens zijn opgenomen.

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

Vraag 1: Requirements Engineering

- Wat is het doel van het requirements specification document? Bespreek tenminste 4 eisen waaraan dit document moet voldoen.
- Identificeer de gebruikersgroepen in de bovenstaande LS-case.
- Schrijf de requirements op in telegramstijl en deel ze naar ~~eigen inzicht~~ in volgens de MOSCOW (Must, Should, Could and Won't Have) methode.
- Werk een gedeelte van de user requirements uit d.m.v. use cases (in natuurlijke taal) van twee "Must Haves" geïdentificeerd in de vorige vraag.

Z.O.Z.

Vraag 2: Object Oriented Analysis and Design (OOAD)

- a) Beschrijf het verband tussen OOA en OOD.
- b) Analyseer de LS-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) Modelleer het reserveren en het eventueel ophalen van een video m.b.v. een state transition diagram. Er hoeft geen rekening te worden gehouden met problemen zoals het niet lid zijn van LS of gehuurde video's die over tijd zijn. Verklaar de gebruikte notatie.

Vraag 3: Software Architectuur

- a) Geef een definitie van software architectuur.
- b) Noem tenminste 3 voordelen van het ontwikkelen van een software architectuur.
- c) Beschrijf het verband tussen systeem structuren, kwaliteit attributen en architecturale view modellen.

Vraag 4: Design

- a) Beschrijf tenminste 3 principes waaraan een "goed" design moet voldoen.
- b) Laat m.b.v een Data Flow Diagram (DFD) zien hoe in de LS-case video's worden verkocht, verhuurd en gereserveerd. 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 5: Software Maintenance

- a) Leg uit waarom in de praktijk software altijd zal evolueren.
- b) Wat voor soorten maintenance kunnen we onderscheiden? Geef van iedere soort een korte omschrijving.

Puntentelling

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

$$\text{Eindcijfer} = (\# \text{punten} + 10) / 10$$

$$\text{Eindcijfer} = \min((\# \text{punten} + 15) / 10, 10)$$

zonder huiswerkopdracht bonus

met huiswerkopdracht bonus

Het gewicht van de vragen is als volgt verdeeld:

- 1a: 5 b: 2 c: 3 d: 10
- 2a: 5 b: 10 c: 10
- 3a: 4 b: 6 c: 5
- 4a: 6 b: 14
- 5a: 2 b: 8