

Tentamen Software Engineering

23 augustus 2002

In dit tentamen staat de case "Bos-, Duin- en Zee-Zicht (BDZZ)" centraal. Vragen over modelleertechnieken hebben betrekking op deze case.

Hotel "Bos-, Duin- en Zee-Zicht" (BDZZ) trekt ieder jaar meer gasten. Tot nu toe geschiedde de registratie en het afrekenen handmatig, maar dit leverde op den duur steeds meer problemen op. In sommige gevallen werden de telefoon-, minibar- en ontbijt-rekeningen niet eens in rekening gebracht wanneer gasten uitcheckten. Deze door het personeel uit het hoofd gevoerde administratie leidt dus onder meer tot een toename van het aantal onbetaalde rekeningen. De directie besluit dat dit niet langer zo kan en gaat over op een hotel-management-systeem. Dit systeem, genaamd BDZZ, is verantwoordelijk voor kernactiviteiten zoals reservering, check-in/out en catering (ontbijt en diner). In de zeer nabije toekomst moet het systeem ook worden uitgerust met zaken als room-service, faciliteiten voor conferenties, speciale arrangementen en reserveringen via het Internet. In eerste instantie echter moeten de kernactiviteiten zo snel mogelijk door het BDZZ systeem worden ondersteund.

De directie van het hotel geeft automatiseringsbedrijf Van Vliet & De Bruin opdracht het BDZZ systeem te ontwikkelen. Omdat in de toekomst veel nieuwe producten zullen worden aangeboden, is het van groot belang dat het systeem flexibel wordt opgezet. Ook van essentieel belang zijn hoge betrouwbaarheid, performance en veiligheid.

Let op

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 is het doel van het requirements specification document? Bespreek tenminste 4 eisen waaraan dit document aan moet voldoen.
- b) Identificeer de stakeholders in de bovenstaande BDZZ-case en geef een profiel (de karakteristieke eigenschappen van de stakeholders die relevant zijn in de context van de BDZZ-case).
- c) Een overduidelijke functionele eis voor het BDZZ-systeem is dat een gast een reservering kan maken voor overnachtingen. Werk deze "Must Have" uit d.m.v. een use case in de vorm van een opeenvolging van genummerde stappen en alternatieven voor bepaalde stappen.
- d) Stel niet-functionele eisen op m.b.t. de aspecten hoge betrouwbaarheid, performance en veiligheid. Van ieder aspect wordt tenminste één (gekwantificeerde) eis gevraagd.

Vraag 2: Analyse

- Analiseer de BDZZ-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. Geef korte omschrijvingen van de classes en hun onderlinge relaties. Verklaar de gebruikte notatie.
- Formaliseer de use case uit vraag 1.c m.b.v. één van de dynamische OO technieken (bijvoorbeeld, interaction (scenario) diagram of activity diagram). Verklaar de gebruikte notatie.
- Analiseer dezelfde BDZZ-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

- In de IEEE-1471 standaard wordt de volgende definitie van software architectuur gegeven:

Architecture is the fundamental organization of a system embodied in its components, their relationships to each other and to the environment and the principles guiding its design and evolution.

Verklaar deze definitie. Ga vooral in op de begrippen organisatie, componenten, relaties en principes.

- Beschrijf twee rollen van een software architect.
- Beschrijf het verband tussen systeem structuren, kwaliteit attributen en architecturale view modellen.
- Stel een (globale) software architectuur op voor het BDZZ-systeem. Maak aan de hand van deze architectuur aannemelijk dat het te bouwen BDZZ-systeem aan de niet-functionele eisen voldoet die zijn opgesteld in vraag 1.d.

Vraag 4: Software Design

- Wat wordt verstaan onder software design?
- Twee belangrijke design criteria zijn cohesion en coupling.
 - Wat wordt precies met deze twee begrippen bedoeld?
 - Waarom zijn ze belangrijk?
 - Geef tenminste 3 vormen van cohesion en 3 vormen van coupling.

Vraag 5: Software Life Cycle

- Welke software life cycle model zou jij toepassen in de BDZZ-case? Beargumenteer je keuze.

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: 3 b: 3 c: 8 d: 6
2a: 10 b: 5 c: 10
3a: 3 b: 3 c: 4 d: 15
4a: 2 b: 8
5a: 10