

Tentamen Software Engineering

Donderdag 14-4-2005 9-12 uur.

Geef je naam en studentnummer op elk vel.

Lees nauwkeurig. Beantwoordt alle vraag onderdelen, vergeet b.v. een motivatie niet indien daar om gevraagd wordt. Geef **beknopte(!)** antwoorden en schrijf leesbaar.

Neem voldoende ruimte voor je antwoord, b.v. een blaadje per vraagnummer.

Voorbeeld

Het volgende voorbeeld komt in diverse vragen terug.

Zgn. automotive systems zijn systemen die bij auto's en vrachtwagens gedrag automatiseren op (o.a.) het gebied van:

- Motor controle. Het gedrag van de motor wordt volledig software gestuurd op basis van de omstandigheden. Eén van de belangrijkste omstandigheden (de "user interface") is het gebruik van de gaspedaal.
- Veiligheids systemen. O.a., systemen die b.v. "pre-crash" airbags opblazen op het moment vlak voor een aanrijding, maar ook anti-blokkeringsystemen van remmen.
- Systemen tegen ongeautoriseerd gebruik. Alarm systemen, controle van sloten e.d.

Van deze systemen kan het gewenste gedrag nauwkeurig beschreven worden.

1 **Processen (1,6 pt.)**

- a) Een aantal activiteiten vinden plaats in elk software ontwikkelingsproces. Noem minimaal 5 van dergelijke activiteiten (0,5 pt.).
- b) Geef aan waarom een waterval methode een geschikte methode zou kunnen zijn om automotive systemen (zie Voorbeeld bovenaan) "van de grond af" te ontwikkelen. (0,5 pt.)
- c) Onder welke voorwaarden is het mogelijk een complex automotive systeem, iteratief te ontwikkelen? Belangrijke hint: Bedenk eerst wanneer het *niet* mogelijk is een dergelijk systeem iteratief te ontwikkelen, en draai die redenering vervolgens om. (0,2 pt.)
- d) Hoe zou je een project met dergelijke iteratieve ontwikkeling plannen (0,2 pt.)
- e) Als het kan een dergelijk ontwikkelingsproces iteratief op te zetten wat zijn dan de voordelen daarvan? Geef er minstens twee (0,2 pt.)

2 **Requirements (1,1 pt.)**

- a) Benoem een aantal non-functionele requirements die aan een systeem voor motor controle gesteld zouden kunnen worden. Noem er tenminste drie, noem ook het type van de non-functionele requirement. (0,3 pt.)
- b) Noem de non-functionele requirement die de kosten van het ontwikkelen van dergelijke systemen het sterkst zal beïnvloeden. Waarom? (0,2 pt.)
- c) Leg uit hoe je voor het voorbeeld van motor controle in automotive systemen de requirements zou willen opschrijven. Of m.a.w.: in welke vorm zou je die requirements expliciet willen maken? Motiveer je antwoord. (0,2 pt.)
- d) Bij motor controle systemen zal vaak op het laatst, tijdens het proefdraaien van een motor en het testen van een auto op een testbaan nog veel

verbeteringen/ veranderingen worden voorgesteld. Eerder gaven we aan dat de ontwikkeling van een dergelijk systeem vaak een zwaar, soms waterval-achtig proces is. Leg uit hoe je je requirement management zo moet opzetten dat dergelijke aanpassingen toch nog mogelijk worden gemaakt. (0,2 pt.)

- e) Het vinden van de juiste instelling van allerlei parameters van de software, ten behoeve van optimaal rijgedrag is vaak trial-and-error werk. Wat is het gevolg hiervan voor het totale ontwikkelproces? (0,2 pt.)

3 Architectuur (2,7)

- a) Leg uit wat de relevantie is van een expliciet architectuur ontwerp voor een groot ontwikkelteam (0,3 pt.)
- b) Sommerville legt uit dat een goede architectuur vooral van belang is voor het vervullen van non-functionele eisen. Geef tenminste vier types van dergelijke non-functionele requirements. (0,4 pt.)
- c) Sommerville beschrijft tevens een aantal "Standard architectural models" zowel structurele als controle modellen. Structurele modellen die hij geeft zijn:
- Repository model
 - Client-server model
 - Abstract machine model

Beschrijf voor een *tweetal* van deze standaard modellen voor en nadelen, noem telkens minstens twee voordelen en twee nadelen. Laat ook zien dat deze voor en nadelen eigenlijk allemaal te maken hebben met non-functionele eisen. (2 pt.)

4 Implementatie en Testen (0,8 pt.)

- a) Bij sommige ontwikkelmethoden (onder andere "Extreme Programming" afgekort XP) worden tests geschreven voordat de software geïmplementeerd wordt, en dient zelfs de test als een soort specificatie. Geef twee redenen waarom dat een goede aanpak zou kunnen zijn. (0,2 pt.) Geef ook een mogelijk nadeel van deze aanpak (0,1 pt.)
- b) Zou dit principe vooral toepasbaar zijn op unit tests of op integratie tests? Waarom? (0,2 pt.)
- c) Leg uit wat in de RUP methode de link is tussen use-cases en tests. (0,3 pt.)

5 Change en configuratie management (0,8 pt.)

- a) Leg uit hoe een iteratief proces een belangrijke rol kan spelen bij change management. Geef details hoe dat gaat. (0,2 pt.)
- b) Leg uit waarom in een groot software project het gebruik van een configuratie management systeem van groot belang is. Geef een drietal functionaliteiten (of groepen van functionaliteit) van een configuratie management systeem met hun nut/doel. (0,6 pt.)

6 **Grootschalig Hergebruik (1 pt.)**

Voor de eerdergenoemde “automotive systemen”, met name motor control, wordt nagedacht over grootschalig hergebruik, d.w.z., dat niet langer voor elke klant (nieuw auto-type) een dergelijk systeem opnieuw volledig wordt ontworpen en gebouwd, maar dat er zoveel mogelijk software hergebruik wordt toegepast. Er wordt gestreefd naar een zgn. product familie.

- a) Leg uit waarom een goede architectuur essentieel is voor een product familie. (0,2 pt.)
- b) (In hoeverre) moet de architectuur voor een dergelijk product familie rekening houden met de mogelijke variabiliteit tussen verschillende producten die met dezelfde product familie gemaakt moeten gaan worden? (0,2 pt.)

In het volgende is het van belang te begrijpen dat een product familie niet alleen bestaat uit een standaard architectuur, maar ook daaraan gekoppeld standaard, generieke componenten die in diverse individuele projecten gebruikt kunnen worden door middel van selectie (welk van de componenten wordt gebruikt) en van specialisatie van de geselecteerde componenten, b.v. door middel van parameters.

- c) Leg uit waarom er in een product familie twee gedeeltelijk ontkoppelde ontwikkel processen ontstaan. (0,2 pt.)
- d) Leg uit waarom change management moeilijker is voor een dergelijke product familie dan voor een “normaal” één-systeem proces. Leg dit uit door te vertellen wat er met een change request op een specifiek product moet gebeuren. (0,2 pt.)
- e) Leg uit waarom configuration management moeilijker is voor een dergelijke product familie dan voor een “normaal” één-systeem proces. Leg dit uit door te vertellen waarom het doorvoeren van een verandering op de product family architectuur en generieke componenten problematischer is dan het veranderen van software in een één systeem proces. (0,2 pt.)

7 **Kwaliteit (1 pt.)**

- a) Beschrijf de diverse fases van de kwaliteit management van een normaal project (één systeem ontwikkeling) (0,3 pt.)
- b) Hoe hangen deze fases samen met het software ontwikkel proces? (0,3 pt.)
- c) Leg uit waarom het gebruik van een document standaard voor b.v. een ontwerp, goed kan zijn voor de kwaliteit van dat ontwerp. Geef minstens twee redenen. (0,2 pt.)
- d) Geef een voorbeeld hoe het volgen van een processtandaard goed kan zijn voor de kwaliteit van het eindresultaat. (0,2 pt.)