

Software Engineering

3de BAC Informatica (Computer Science)
[Academic year 2010-2011]

© Prof. Serge Demeyer



Universiteit Antwerpen

Doel

Programmaboekje

- "Het doel bestaat erin om de student een brede basis te verschaffen in het bouwen van softwaresystemen die te complex zijn om door één persoon gerealiseerd te worden."
- Dus
 - brede basis
 - ➔ véél technieken
 - complexe systemen
 - ➔ schaalbare technieken

HOOFDSTUK 0 – Praktische Zaken



Doel

- Professionele Informaticus
- Plaats in het Curriculum
 - + Kerncompetenties
- Beoordelingscriteria
- Examen
- Voorbeeldvragen

Literatuur

Inhoudstafel

Professionele Informaticus

diverse domeinen

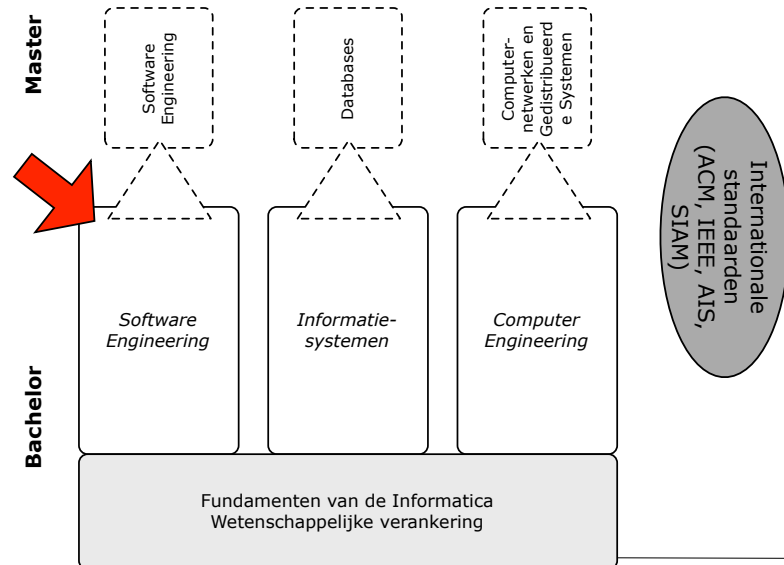
- traditionele "data processing"
(banken, verzekeringen)
- spits technologie
(multimedia, embedded systems)
- telecommunicatie
(netwerkbeheer, e-commerce)
- ...
- onderzoek

zij verwachten

- technische virtuozen
 - + diverse specialiteiten
- groepsspelers
 - + sociale vaardigheden



Bachelor Programma - overzicht



0.Praktische Zaken

5

Kerncompetenties Software Engineering

Bachelor = bekwame informaticus

Analyse en ontwerp (kleinschalige software projecten)	+
Implementatie (nieuwe softwaresystemen)	+
Onderhoud (bestaande softwaresystemen)	++
Databanken (implementatie en onderhoud van)	
Netwerk (beheer van)	
Vakbekwaamheid	++
Maatschappij	+++
Communicatievaardigheden	+

Academische Bachelor = wetenschappelijke vorming

Wiskundige basis	
Formeel denken (abstraherend vermogen)	+
Levenslang leren	++
Wetenschappelijk aanpak	+
Wetenschappelijke basis	
Autonoom en creatief	+

0.Praktische Zaken

6

Criteria - Selectie

Accuraatheid

- Een professionele software engineer werkt in groep en moet dus op een accurate manier kunnen communiceren met zijn collega's en eindgebruikers.
 - Juist gebruik van terminologie
 - Parate kennis definities

Toepasbare kennis

- Een professionele software engineer moet in staat zijn gekende technieken toe te passen in een variërende context.
 - "Know-how"
 - Oefeningen

SELECTIE

- Je *moet* dit kunnen demonstreren tijdens het examen om te slagen !



0.Praktische Zaken

7

Criteria - Diversificatie

Inzicht

- Een professionele software engineer moet technische keuzes kunnen verantwoorden
 - "Know when"
 - Afwegingen maken

DIVERSIFICATIE:

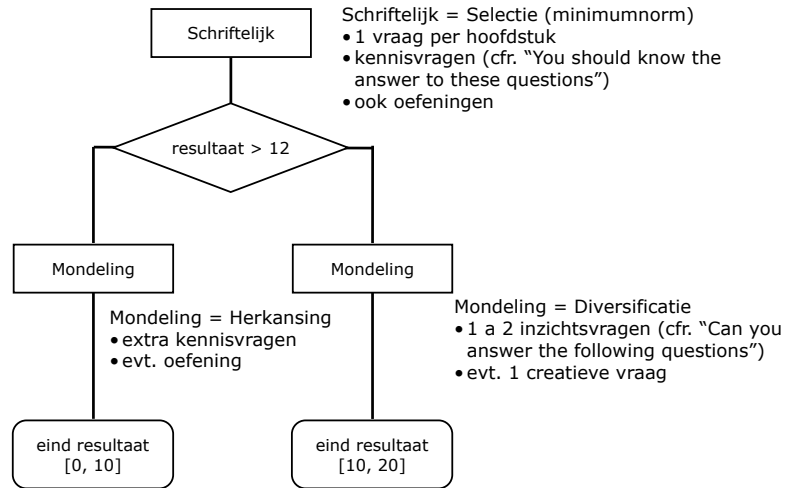
- De mate waarin je dit kunt demonstreren tijdens het examen laat je toe je te onderscheiden van je collega's.



0.Praktische Zaken

8

Examen



Voorbeeldvragen (schriftelijk)

Enkele voorbeeldvragen

- 1. Geef 2 redenen waarom het waterval model onrealistisch is.
- 2. "Het systeem is voor 93% correct" is een geldige uitspraak.
 - Ja / Neen
 - Waarom?
- 3. Bij het overschrijven van een methode in een subclasse ...
 - + (a) moet de preconditione gelijk blijven
 - + (b) mag de preconditione zwakker worden
 - + (c) mag de preconditione sterker worden
 - Waarom?

Modelantwoorden

- 1.
 - + gebruikers kunnen behoeften nooit volledig specificeren
 - + een werkende versie is veel te laat beschikbaar
- 2. Ja / Neen
 - + Correctheid is een absolute eigenschap.
- 3. (b)
 - + Een subclasse moet minstens hetzelfde contract vervullen.

Voorbeeldvragen (mondeling)

Een voorbeeld v. e. inzichtsvraag

- Leveren CRC-kaarten het best mogelijke ontwerp? Argumenteer?

- Modelantwoord
 - + er is geen eenduidig criterium om te meten wat het "beste" ontwerp is
 - + de techniek is heel vrij: elke stap kan verscheidene goede antwoorden bieden
 - + veel hangt af van de groepsdynamiek

Een voorbeeld v.e. creatieve vraag

- Je baas heeft op de radio gehoord van het "I Love You Virus" en vraagt een veiligheidsplan. Wat zul je allemaal in dat veiligheidsplan opnemen ?
- Modelantwoord
- De context is niet voldoende duidelijk en je moet zelf vragen stellen om die helder te krijgen.
 - + Wat voor soort systeem is het? Hoe hangt het aan het internet?
 - + Wat voor soorten risico's loopt het systeem? Hoeveel risico is je baas bereid te lopen? Hoeveel is hij bereid te investeren ?

Criteria (ii)

Levenslang leren

- Een professionele software engineer zal zijn leven lang de technische evoluties op de voet moeten volgen.
 - Vele referenties naar boeken, artikels, world-wide web
 - "Engels" als voertaal voor de transparanten

ACHTERGRONDINFORMATIE:

- Je wordt verondersteld zelf selectief met diverse bronnen om te gaan.



Literatuur

Aanbevolen is 1 van onderstaande boeken aandachtig door te nemen (INZICHT)

- [Ghez91a] Fundamentals of Software Engineering, C. Ghezzi, M. Jazayeri, D. Mandroli, Prentice Hall, 1991. ([Ghez02a] = Second edition)
 - Tijdloos door de nadruk op onderliggende principes, maar daardoor moeilijk.
- [Pres97a] Software Engineering — A Practitioner's Approach, R. Pressman, Mc-Graw Hill, Fourth Edn., 1997. ([Pres09a] = Ninth edition)
 - Zeer praktisch en zeer diep, maar anderzijds weinig selectief en volumineus (dus duur).
- [Somm96a] Software Engineering, I. Sommerville, Addison-Wesley, Fifth Edn., 1996. ([Somm06a] = Eighth edition)
 - Zeer populair, zeer breed en makkelijk leesbaar, maar mist af en toe wat diepgang.

Andere literatuur wordt per hoofdstuk vermeld, incl. referenties op het web.