Tentamen
"Software Engineering"
B.Sc. Informatica &
B.Sc. Informatica en Economie

Faculteit der Wiskunde en Natuurwetenschappen, Universiteit Leiden

Docenten: Drs. Werner Heijstek & Dr. Natallia Kokash

dinsdag 31 juli 2012
13:45 - 16:45

Dit tentamen bestaat uit **12** vragen verdeeld over **4** pagina's, inclusief dit voorblad. Er zijn **77** punten te verdienen met het beantwoorden van de vragen. Het aantal te verdienen punten is per vraag aangegeven.

Nota bene (student and invigilator):

- Je mag bij dit tentamen geen boeken of enig ander geschreven, geprint of digitaal materiaal gebruiken. (*No books or any other materials are allowed.*)
- **Dit blad met tentamenvragen moet weer worden ingeleverd.**
  (The exam questions are to be handed in / cannot be taken home.)

Nota bene (student):

- Schrijf je naam en studentnummer *op ieder antwoordvel*.
- Motiveer al je antwoorden maar doe dit *kort en bondig* (gebruik nooit meer dan 150 woorden per (deel-)vraag).
- Schrijf de antwoorden op met blauwe of zwarte pen (niet met potlood) en schrijf *leesbaar*.
- Je mag antwoorden formuleren in het Nederlands of in het Engels.

1a. *(3 pt.)* Noem drie belangrijke verschillen tussen het traditionele waterval softwareontwikkelproces en de meer recente 'agile' methoden.
1b. *(3 pt.)* Geef een voorbeeld van een situatie waarin toepassing van een methode als bijvoorbeeld SCRUM valt te verkiezen boven een meer watervalachtige aanpak.

2a. *(2 pt.)* Geef aan uit welke activiteiten (minimaal 5) de software development lifecycle bestaat.
2b. *(4 pt.)* Wat is globaal de verdeling van effort over deze activteiten tijdens de software development lifecycle?

3. *(3 pt.)* Leg uit hoe polymorphisme bijdraagt aan de uitbreidbaarheid van software.

4a. *(3 pt.)* Wat zijn de voor- en nadelen van het toepassen van 'prototyping'?
4b. *(3 pt.)* In welke situaties is het gebruik van 'prototyping' vooral nutting?

5. Aanschouw de volgende hypothese:

> *"Het gebruik van agile ontwikkelprocessen leidt vaker tot hogere maintenance kosten van software projecten dan waterval processen."*

5a. *(3 pt.)* Stel een empirische onderzoeksstrategie voor om bewijs te vergaren dat bovenstaande hypothese ondersteunt of tegenspreekt.
5b. *(3 pt.)* Geef aan hoe u de onderhoudskosten van een project gaat beoordelen.

6.*(3 pt.)* Leg uit hoe softwaremetrieken gebruikt kunnen worden om de onderhoudbaarheid van een software systeem te beoordelen.

7. *(2 pt.)* Geef een voorbeeld van een niet-functionele systeemeis.

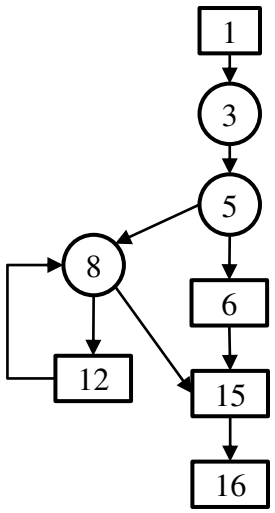
Denk aan vragen 8, 9, 10, 11 en 12 op de volgende pagina's!

8a. *(4 pt.)* Noem vier redenen om softwareontwikkeling offshore uit te besteden.
8b. *(2 pt.)* Wat wordt bedoeld met de 'sociaal-culturele afstand' die gemoeid is met global software development?
8c. *(2 pt.)* Wat kun je als projectleider doen om de ondervonden hinder van deze afstand te beperken?
8d. *(3 pt.)* Leg uit wat de relatie is tussen koppeling en cohesie en gedistribueerde softwareontwikkelteams.

9a. *(3 pt.)* Noem drie voordelen van het maken van documentatie tijdens software ontwikkeling.
9b. *(2 pt.)* Noem twee methoden om de kwaliteit van softwaredocumentatie te waarborgen.

10. *(3 pt.)* Motiveer in welke fase van het softwareontwikkelproces men zou moeten beginnen met testen.

11. *(4 pt.)* Wat is de cyclomatische complexiteit van de flow graph van het programma in Figuur 1?



Figuur 1: Flowchart van een programma behorende bij tentamenvraag 11

Denk aan vraag 12 op de volgende pagina!

12. U bent de software architect voor de software van een (eenvoudige) mobiele telefoon.

12a. *(5 pt.)* Stel minstens twee relevante requirements op. Formuleer beide requirements aan de hand van de S.M.A.R.T. methode.

Maak (aan de hand van de volgende vier deelvragen) met behulp van de UML, een ontwerp voor een (eenvoudige) mobiele telefoon die voldoet aan de volgende eisen

- de architectuur moet gelaagd zijn
- de diagrammen moeten begrijpelijk zijn
- gebruik design patterns, indien mogelijk

12b. *(6 pt.)* Ontwerp het use-case diagram en neem hierin tenminste drie niet-triviale use cases op.
12c. *(6 pt.)* Ontwerp tenminste één niet-triviaal klassendiagram
12d. *(6 pt.)* Ontwerp tenminste drie 'sequence diagrams' om de 'use cases' uit deelvraag b te realiseren.