Automatisering

Paul Sohier 0806122
Sebastiaan Polderman 0820738

31 oktober 2010

Inhoudsopgave

1 Week 1

1.1 Opdrachten bij hoofdstuk 1

1.1.1 Geef voorbeelden van computerprogramma's die voornamelijk ontwikkeld worden volgens het watervalmodel

Een pakket als MS Office zou volgens dit model ontwikkeld worden omdat het van te voren bepaalde eisen heeft aan de functionaliteit. Zodra het basis pakket ontwikkeld is zal er niet veel meer veranderd worden aan het uitendelijke doel van het programma.

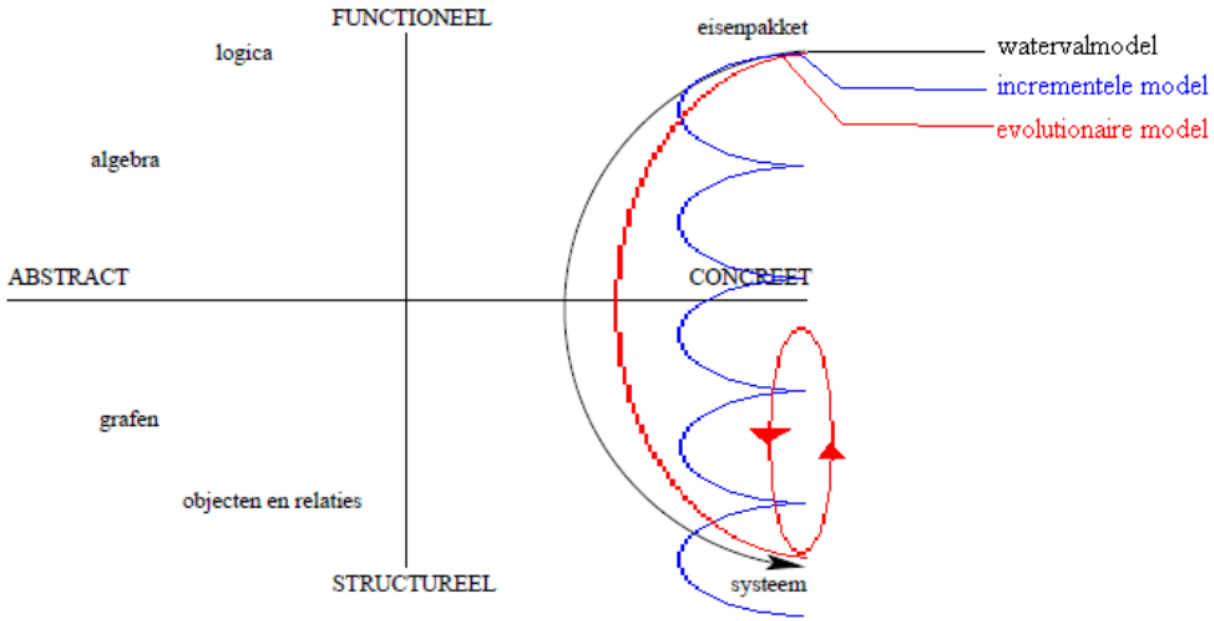
1.1.2 Geef voorbeelden van computerprogramma's die voornamelijk ontwikkeld worden met prototyping.

Een goed voorbeeld hierbij is een project op school. Hierbij wordt eigenlijk bijna direct begonnen met het werken aan het eindproduct, terwijl er nog niet echt bedacht is wat het eindproduct moet worden. Op diverse punten tijdens de ontwikkeling van het programma zal het eisenpakket worden aangepast naar de uiteiendelijke wensen.

1.1.3 Wat is het verschil tussen validatie en verificatie

Bij validatie (Valideren) wordt gekeken of het ontwerp voldoet aan het eisenpakket, terwijl er bij verificatie wordt gekeken of het process voldeed.

1.1.4 Hoe zou u de ontwikkeltrajacten van het waterval-, het incremenetele- en het evolutionare model aangeven in figuur 1.1?



1.1.5 Zal de evolutionare ontwikkelstrategie de grens tussen ontwikkeling en onderhoud laten verdwijnen?

Ja, zodra je onderhoud doet kan je tegelijkertijd ook nieuwe, door de klant gewenste, features kunnen toevoegen aan het al bestaande programma.

1.1.6 Zoek op internet voorbeelde van repositories.

Een voorbeeld hiervan zijn de door debian gebruikte repositories, waaruit alle Debian installaties software vandaan installeren.

1.1.7 Geef de voor en nadelen van open-source software.

Een groot voordeel is dat de sourcecode van de applicatie/programma vrijelijk te bekijken is. Hierdoor zie je dat bijvoorbeeld security problemen welke in deze source code aanwezig zijn sneller gevonden worden en hierdoor het algemener bekend is of software veilig is of niet.
Helaas is dit grote voordeel ook direct een nadeel. Wanneer een security probleem gevonden wordt in een applicatie/programma en dit wordt niet opgelost door de vendor kan dit makkelijk misbruikt worden door de vaak uitgebreide verspreiding van de software.
Om dit probleem in het geheel op te lossen zal je dus een combinatie moeten maken tussen veilig gebruik van software (Kijkend naar de geschiedenis van software) en ander soort software.

1.1.8 De eis "Alle uitvoer moet normaal binnen 10 seconden gegeven worden" is om één van de volgende redenen fout. Welke?

a Dubbelzinnig
b Niet concreet
c Tegenstrijdig

De eis is niet concreet genoeg, doordat alle uitvoer heel algemeen is. Een verbetering op deze eis zou zijn: "De klant moet binnen 10 seconden een bevestiging van zijn bestelling op het scherm zien."

1.1.9 Wat mankeert er aan de eis: "Het bestand moet een afsluitteken bevatten."?

De eis is onduidelijk, doordat het afsluitteken niet is vastgesteld in de eis.

1.1.10 Welke maatregelen kunnen positief of negatief werken op:

a De correctheid
b De beschikbaarheid
c De herstelbaarheid

a Regelmatige validatie en verificatie tijdens alle stappen van het project.
b Het systeem zo laten functioneren dat, mocht er een storing plaatsvinden in een bepaald gedeelte, dat de rest van het systeem nog naar behoren blijft functioneren.
c Optimalisatie van de programmatuur.

2 Week 2

2.1 Opdrachten bij hoofdstuk 3

2.1.1 De projectkosten worden begroot op 100000euro. De winst wordt gesteld op 15%.Het risico dat men met dit project denkt te lopen, is gebaseerd op de ervaring dat 20% van dit soort projecten mislukken. De BTW bedraagt 19%. Hoeveel is de aanbestedingsprijs?

$$(100 * 1.15 * 1.2) * 1.19 = 164.20 euro$$

2.1.2 Wat zijn de verschillen tussen kosten en investeringen?

Kosten zijn uitvagen die direct van de winst mogen worden afgetrokken. Inversteringen daarintegen moeten over meerdere jaren worden afgeschreven. Die jaarlijkse afschrijven mogen wel als kosten worden opgevoerd.

2.1.3 Noem 3 investeringscriteria

- Maximale gemiddelde boekhoudkundige rendabiliteit
- Minimale terugverdienperiode
- Concurrentievoordeel krijgen

2.1.4 Men kan een productiemiddel huren voor de prijs van 6100 e per maand. Indien men dit productiemiddel voor 120000 e aanschaft, moet men 100 e per maand onderhoud betalen. De restwaarde is nihil.

- Bepaal het omslagpunt.
- Bereken het omslagpunt indien de discontovoet 1% per maand is.
- Welke financiele overweging kan bij deze keuze een belangerijke rol spelen?

- Het omslagpunt:

$$\frac{120000}{(6100-100)} = 20$$

  Het omslagpunt ligt dus bij 20 maanden
- Het omslagpunt bij een discotovoet van 1% per maand

$$120000 * 1.01^{-20} = 98345.34$$

$$\frac{98345.34}{6100-100} = 16.39$$

  Het omslagpunt ligt dan bij 16.36 maanden.
- Welke overweging kan bij deze keuze een belangerijke rol spelen?
  De duurzaamheid of het gebruik van de inverstering. Hieruit bepaal je dan of het goedkoper is om te kopen of juist andersom.

3 Week 3

3.1 Opdrachten bij hoofdstuk 4

3.1.1 Noem minimaal 7 factoren die de individuele productiviteit beïnvloeden.

1 Kennisniveau
2 Sfeer
3 Taalbeperkingen
4 Afhankelijk van hulpmiddelen
5 Budget
6 Werkprocessen
7 Beloning

3.1.2 Wat is het bezwaar tegen de definitie van individuele productiviteit: “De omvang van de objectcode (gecompileerde programmatuur) in bytes per tijdseenheid”?

Iemand die minder efficient programeert levert volgens deze stelling beter werk af. Ook wordt er hiermee niet gekeken naar de complexiteit van de code.

3.1.3 Verklaar waarom het coderen in de uitwerkingsfase meestal minder dan 10% van de totale kosten van de levenscyclus bedraagt.

Door het gebruik van hulpmiddelen krimpt het coderingsaandeel.

3.1.4 Verklaar hoe uit de COCOMO methoden blijkt dat de projecttijd niet afhankelijk is van het aantal programmeurs.

De geschatte projecttijd T wordt berekend door het aantal mensmaanden tot de macht c (Compactheid), vermenigvuldigt met $2,5$. Op basis hiervan wordt het minimum aantal benodigde mensen geschat. Het model rekent niet de projecttijd uit op basis van het aantal teamleden.

3.1.5 Een administratief systeem van 9 netto functiepunten werd met een inspanning van 3 mensmaanden ontwikkeld. Is dit kenmerkend voor een administratief automatiseringsproject?

Volgens figuur 4.6 (Tabel 1) in de reader is de gemiddelde productiviteit voor administatieve systemen $15,2$. De parameters van vraag 5 geven een productiviteit van 3.

Tabel 1: Figuur 4.6

Toepassingsgebied	Productiveit P [nfp/mensmaand]		
	gemiddeld μ	standaarddeviatie σ	variatiecoëfficiënt ρ
systemen met microcode	$1,5$	$2,7$	$1,80$
ingebedde realtime systemen	$6,8$	$3,1$	$0,46$
vliegtuigsystemen	$6,4$	$3,3$	$0,52$
commando en controle systemen	$9,9$	$4,1$	$0,41$
procesbesturings systemen	$10,3$	$4,3$	$0,42$
besturingsprogrammas en utilities	$11,3$	$3,9$	$0,35$
wetenschappelijke systemen	$12,5$	$4,0$	$0,32$
netwerksystemen	$10,3$	$3,1$	$0,30$
administratieve systemen	$15,2$	$3,8$	$0,24$

3.1.6 Een computerprogramma heeft bij functiepuntanalyse voor de systeemkarakteristieken een totale correctiefactor $\bullet \sum_{i=1}^{14} c_i = 50$. Uit de specificatie blijkt dat er 3 verschillende invoer-, 7 uitvoer- en 5 vraagtypen nodig zijn. Daarnaast zijn er 2 externe gegevensverzamelingen en 2 interne gegevensverzamelingen nodig. Alle geruiktsfuncties hebben een gemiddelde complexiteit. Men zal het computerprogramma coderen in de taal JAva ($C_t = 53$). Er wordt geen gebruik gemaakt van hergebruik. Maak een schatting voor de broncode.

$$f = 3 * 4 + 4 * 5 + 5 * 4 + 2 * 10 + 2 * 7 = 12 + 20 + 20 + 20 + 14 = 86$$

$$NFP = (0,65 + 0,01 * 50) * 86 = 98,9$$

$$S = 98,9 * 53 = 5241,7$$

Het aantal regels broncode is dus ongeveer 5241

3.1.7 Verklaar waarom juist computertalen met een lage ct waarde (zie paragraaf 4.1.1) geschikt zijn voor prototyping.

Dit komt doordat de hoeveelheid code kleiner is en hierdoor dus sneller is aan te passen

3.1.8 Gegven een studentinformatiesysteem bestaande uit de volgende onderdelen:

soort	omschrijving	complexiteit
module	inschrijving student	gemakkelijk
module	uitschrijving student	gemakkelijk
module	rekening collegegeld sturen	gemakkelijk
module	machtiging betaling collegegeld verwerken	makkelijk
module	betaling collegegeld verwerken	gemiddeld
module	aanmaningen sturen	gemakkelijk
module	tentamencijfers verwerken	moeilijk
module	studieresultaten naar Informatie Beheer Groep	gemiddeld
scherm	inschrijven van een student	gemiddeld
scherm	uitschrijven van een student	gemiddeld
scherm	betaling student invoeren	gemiddeld
scherm	machtiging betaling invoeren	gemiddeld
scherm	tentamencijfers invoeren	moeilijk
scherm	vakkentabel invoeren	gemiddeld
scherm	vakkentabel wijzigen	moeilijk
formulier	studentgegevens	gemiddeld
formulier	jaarlijkse voortgangsrapportage	moeilijk
formulier	rapportage van tentamenresultaten	moeilijk
formulier	aanmaningen collegegeld	gemiddeld

- Maak met een objectpuntanalyse (OPA) een schatting voor de inspanning E als het hergebruik rond de 30% ligt en het ontwikkelteam een lage productiviteit heeft. Maak een schatting met COCOMO-81 van de projecttijd T, indien het project een 'organische' karakter heeft.
- Maak een schatting van de projecttijd T met COCOMO-II, indien er geen sprake is van tijdsdruk en het project de volgende karateristieken heeft:

Karakteristieken	b_i
ontwerpervaring	$0,05$
ontwerpvrijheid	$0,02$
ontwerprisico	$0,3$
ontwerpteam	$0,03$
organisatie	$0,01$

- OPA

$$NOP = 50 * (1 - 0.3) = 35$$

$$E = \frac{35}{7} = 5$$

- COCOMO-81

$$T = 2.5 * 5.2^{0.38} = 4.78$$

- COCOMO-II

$$T = (2.5 * 5.2^{0.33+0.2*0.14}) = 4.51$$

4 Week 4

4.1 Opdrachten bij hoofdstuk 5

4.1.1 Gegeven een netwerk met 7 activiteiten:

activiteit	omschrijving	afhankelijk van	t_b	t_m	t_w
A	Definitie		1	2	3
B	Hardware specificatie	A	3	4	5
C	Software specificatie	A	1	2	9
D	Hardware ontwikkeling	B	1	2	3
E	Software ontwikkeling	B C	4	5	12
F	Documenteren	D	3	4	5
G	Systeemintergratie	D E	3	3	3

a Bepaal met PERT het mijpalenplan en het kritieke pad
b Bepaal het Ganttscheme. Welke conclusies kan men daaruit trekken?
c Bepaal het 95%-betrouwbaarheidsinterval voor de werkelijke projecttijd.

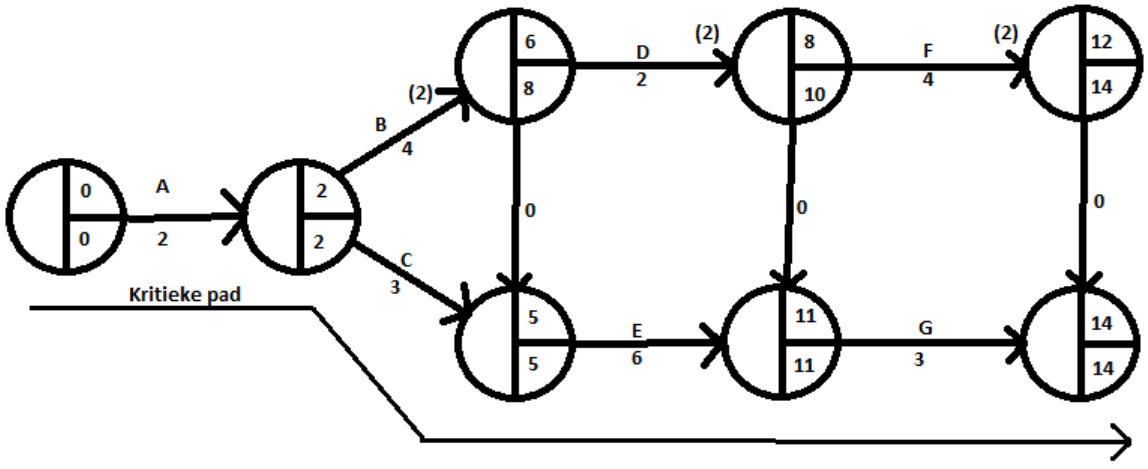
a Zie figuur 1.
b Zie tabel 2.
c

$$o^2 = 2 + 2 + 8 + 2 = 2 + 8 + 2 + 0 = 24$$

$$14 - (1,96 * \sqrt{24}) =< T =< 14 + (1,96 * \sqrt{24}))$$

$$4,40 =< T =< 26,60$$

Figuur 1: Antwoord bij Hoofdstuk 5 Vraag 1.a



Tabel 2: Antwoord bij Hoofdstuk 5 vraag 1.b

	0..1	1..2	2..3	3..4	4..5	5..6	6..7	7..8	8..9	9..10	10..11	11..12	12..13	13..14
A 2	—	—												
B 4			—	—	—	—	- - -	- - -						
C 3			—	—	—									
D 2							- - -	- - -						
E 6						—	—	—	—	—	—			
F 4									- - -	- - -	—	—	- - -	- - -
G 3												—	—	—

4.1.2 bepaal van de volgend activiteiten met PERT het mijlpalenplan en een schatting van het kritieke pad T_k

activiteit	t	afhankelijk van
A	2	K
B	3	H
C	4	
D	2	F G
E	1	D I J
F	3	B
G	3	C K
H	3	
I	2	A F
J	1	F
K	2	H

4.1.3 Van een collectief team is gegeven dat de verliesfactor per communicatiekaneel 5% is, bereken de maximale groepsgrootte

$$N = \frac{(1+i)}{(2*i)} = \frac{1+0,05}{2*0,05} = \frac{1,05}{0,10} = 10,5$$

4.1.4 Toon aan dat de volgende formule geldt voor het hiërarchieke team met N teamleden en maximaal 6 ondergeschikten per echelon:

$$\lfloor {}^{6}\log(5N+1) \rfloor \leq echelons \leq N$$

5 Week 5

5.1 Opdrachten bij hoofdstuk 7

5.1.1 Van een objectgeoriënteerd computerprogramma in Java, zijn van alle classes de metrieken NOM,CBO, RFC, WMC, DIT en NOC gemeten. Een tiental classes had een verhoogd risico.

a Maak een tabel met riskante waarden van de waarden van de metrieken.

Metrieken	Riskante waarde	reader bladzijde	reader paragraaf
NOM	Java gemiddeld: ≈ 8 C++ gemiddeld: ≈ 25 Kritiek: > 40	84	7.2.2
CBO	5	86	7.2.5
RFC	≥ 50	87	7.2.7
RFC/NOM	Java: ≥ 10 C++: ≥ 5	88	7.2.7
WMC	aanbevolen: ≥ 25 kritiek: > 75	83	7.2.1
DIT	5	84	7.2.3
NOC	Geen aanbevolen waarde, hoge NOC geeft slechte metriek	85	7.2.4

b Geef in de volgende tabel bij elke class aan welke metriek een kritieke waarde heeft.

class	NOM	CBO	RFC	RFC/NOM	WMC	DIT	NOC
1	54	8	536	9,9	175	1	0
2	7	6	168	24,0	71	4	0
3	33	4	240	7,2	105	2	0
4	54	8	381	6,7	117	2	2
5	62	6	378	6,1	163	2	0
6	63	7	235	3,7	156	2	0
7	81	10	285	3,5	161	2	0
8	42	5	127	3,0	69	3	0
9	20	17	325	16,2	139	4	4
10	46	5	186	4,0	238	1	3

Bij NOC is er geen aanbevolen kritieke waarde, maar er is wel hoe hoger deze waarde is hoe slechter de metriek is. Wij hebben bij een NOC van ≥ 3 aangenomen dat de NOC kritiek is. Dit baseren wij op dat de NOC is gebaseerd op het aantal kinderen een class heeft, en hoe meer kinderen, hoe groter de kans op fout. Bij ≥ 3 is de kans op fouten volgens onze mening zeer goed aanwezig.
Wanneer een waarde kritiek is voor een bepaald metriek, staat er in onderstaande tabel *kritiek*. Wanneer er niets staat is de waarde niet kritiek. Er is uitgegaan dat de classes zijn geschreven in de taal JAVA. Indien de waarde uitmaakt voor de kritieke waarde, en de kritieke waarde voor C++ anders is, staat dit tussen haakjes.
De waarde voor RFC is in alle gevallen kritiek. De waarde ligt flink boven de aangegeven kritieke waarde, welke tevens ook zeer hoog boven de kritieke waarde ligt. De kritieke waarde voor RFC ic ≥ 50, terwijl de laagste waarde voor RFC van de classen 127 is.
De waarde voor DIT ligt in alle gevallen onder de kritieke waarde van 5.

class	NOM	CBO	RFC	RFC/NOM	WMC	DIT	NOC
1	kritiek	kritiek	kritiek	(kritiek)	kritiek		
2		kritiek	kritiek	kritiek(kritiek)			
3			kritiek	(kritiek)	kritiek		
4	kritiek	kritiek	kritiek	(kritiek)	kritiek		
5	kritiek	kritiek	kritiek	(kritiek)	kritiek		
6	kritiek	kritiek	kritiek		kritiek		
7	kritiek	kritiek	kritiek		kritiek		
8	kritiek	kritiek	kritiek				
9		kritiek	kritiek	kritiek(kritiek)	kritiek		kritiek
10	kritiek	kritiek	kritiek		kritiek		kritiek

5.1.2 Bepaald WMC, DIT, NOC, CBO, RFC en LCOM van de volgende pseudo objecten broncode:

Class	trade
Variables	trade.id, counterparty, trade_value
Methods	evaluate_counterpart() get_trade_id(trade_id) position_update() {position_manager::report_trade()}
Class	bond_trade
Variables	bond_detials
Methods	get_bond_info()
Class	fx_trade
Variables	forex_detials
Methods	calculate_exchange_rates()
Class	equity_trade
Variables	company, stock_market, PE_ratio, earnings, week_hi_lo
Methods	Estimate_beta() get_stock_quotes() {quotron::quotes()}
Class	municipal_bond_trade
Variables	state_or_federal, over_the_counter
Methods	calculate_coupon_rate() {Tbil_server::rate()}
Class	corporate bond_trade
Variables	adr, sp_rating
Methods	calc_rating(sp_rating) {if adr then fx_trade::calculate_exchange_rates()}
Class	international_equity
Variables	exchange_rate, quitation
Methods	perform_anaylysis_roa() {fx_trade::calculate_exchange_rates()} get_quotron(quotation)
Class	domestic_equity
Variables	variables: attribute1
Methods	