

Automatisering

Paul Sohier 0806122
Sebastiaan Polderman 0820738

14 oktober 2010

Inhoudsopgave

1 **Week 1 2**

1.1 *Opdrachten bij hoofdstuk 1* 2

1.1.1 Opdracht 1 2

1.1.2 Opdracht 2 2

1.1.3 Opdracht 3 2

1.1.4 Opdracht 4 2

1.1.5 Opdracht 5 2

1.1.6 Opdracht 6 2

1.1.7 Opdracht 7 2

1.1.8 Opdracht 8 2

1.1.9 Opdracht 9 3

1.1.10 Opdracht 10 3

2 **Week 2 3**

2.1 *Hoofdstuk 3* 3

2.1.1 Opdracht 1 3

2.1.2 Opdracht 2 3

2.1.3 Opdracht 3 3

2.1.4 Opdracht 5 3

3 **Week 3 4**

3.1 *Hoofdstuk 4* 4

3.1.1 Opdracht 1 4

3.1.2 Opdracht 2 4

3.1.3 Opdracht 3 4

3.1.4 Opdracht 4 4

3.1.5 Opdracht 5 4

3.1.6 Opdracht 6 4

3.1.7 Opdracht 7 4

3.1.8 Opdracht 8 5

4 **Week 4 5**

4.1 *Hoofdstuk 5* 5

4.1.1 Opdracht 1 5

4.1.2 Opdracht 2 5

4.1.3 Opdracht 3 5

4.1.4 Opdracht 4 5

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 uiteindelijke 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 uiteindelijke 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 ontwikkeltrajecten van het waterval-, het incremenetele- en het evolutionaire model aangeven in figuur 1.1?

Nog doen.

- 1.1.5 Zal de evolutionaire 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

Nog doen?

2 Week 2

2.1 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?

$$(100 * 1.15 * 1.2) * 1.19 = 164.22 \text{euro}$$

2.1.2

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

2.1.3

- Maximale gemiddelde boekhoudkundige rendabiliteit
- Minimale terugverdienperiode
- Concurrentievoordeel krijgen

2.1.4

- Het omslagpunt:

$$\frac{120.00}{(6100 - 100)} = 20$$

Het omslagpunt ligt dus bij 20 maanden

- Het omslagpunt bij een discotovoet van 1% per maand

$$120 * 1.01^{-20} = 98345.35$$

$$\frac{98345.35}{6100 - 100} = 16.36$$

Het omslagpunt ligt dan bij 16.39 maanden.

- Welke overweging kan bij deze keuze een belangrijker 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 Hoofdstuk 4

3.1.1

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

3.1.2

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

Door het gebruik van hulpmiddelen krimpt het coderingsaandeel.

3.1.4

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

Volgens figuur 4.6 in de read is de gemiddelde productiviteit voor administratieve systemen 15, 2. De parameters van vraag 5 geven een productiviteit van 3.

3.1.6

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

$$NFP = (0.65 + 0.01 * 50) * 86 = 1.15$$

$$S = 1.15 * 53 = 60.95$$

NOTE: Nog narekenen?

Het aantal regels broncode is dus ongeveer 60.95

3.1.7

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

3.1.8

- OPA

$$NOP = 50 * (1 - 0.3) = 36.4$$

$$E = 36.4/7 = 5.2$$

- COCOMO-81

$$T = 2.5 * 5.2^{0.38} = 4.8$$

- COCOMO-II

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

4 Week 4

4.1 Hoofdstuk 5

4.1.1 Gegeven een netwerk met 7 activiteiten:

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

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

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

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

$$\lfloor \log_6 (5N + 1) \rfloor \leq \text{echelons} \leq N$$

5 Week 5

5.1 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.
- b Geef in de volgende tabel bij elke class aan welke metriek een kritieke waarde heeft.

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